Быстрый старт OpenSCADA

Роман Савоченко (rom_as@oscada.org)

Максим Лысенко (mlisenko@oscada.org)

28. ноя. 2011

Оглавление

Введение

Открытая система OpenSCADA является предельно модульной, гибкой и многофункциональной SCADA-системой. Как следствие этого первое знакомство с OpenSCADA может быть достаточно сложным по причине малых шансов совпадения предыдущего опыта пользователя или полного его отсутствия, с подходами работы в OpenSCADA. Однако, в значительной степени это только первое впечатление, поскольку вся мощь OpenSCADA оказывается на ладони у пользователя, от изобилия которой пользователь может растеряться, и ему могут понадобиться значительные усилия для отбора функций, нужных в решении его задачи.

По этой причине и для наглядного представления общей концепции работы в OpenSCADA создан данный документ. Документ в достаточно краткой и наглядной форме представляет путь от запуска OpenSCADA до создания элементов пользовательского интерфейса на реальных примерах. Кроме того, документ содержит раздел с рецептами по конфигурации, реализации и решению типовых задач пользователя.

Документ не содержит детального описания концепции и глубокого погружения в детали OpenSCADA, а предоставляет ссылки на документы OpenSCADA, содержащие такую информацию.

Описание документа ведётся синхронно с реализацией примеров на демонстрационной базе данных (БД), модели АГЛКС. Следовательно, пользователь должен получить дистрибутив OpenSCADA с этой БД, для наглядного изучения и опробования примеров.

1. Термины, определения и аббревиатуры

АРМ (аббревиатура) - Автоматизированное Рабочее Место. Обычно представляет из себя системный блок вычислительной системы, дисплей, манипулятор «мышь» иногда с клавиатурой, и другое периферийное оборудование, которое служит для визуального представления данных технологического процесса, а также выдачи управляющих воздействий на ТП.

Блокировка (термин) – условная граница технологического параметра, по преодолению которой выполняются предустановленные алгоритмом действия по предотвращению аварии. В некоторых режимах ТП (пуск) в соответствии с регламентом может быть необходимо отключение блокировки (деблокирование).

Деблокирование (термин) – процесс отключения блокировки на время работы ТП в режимах, для которых в регламенте предусмотрена данная операция. Внимание, деблокирование технологических параметров является строго отчётной операцией и должно производится оперативным персоналом в соответствующем порядке.

Квитация (термин) – процесс подтверждения факта того, что оперативный персонал обратил внимание на нарушение в работе ТП. Обычно этот процесс подразумевает принятие мер оператором для устранения нарушения и нажатие соответствующей кнопки для прекращения сигнализации.

ПЛК (аббревиатура) – Промышленный Логический Контроллер. Микропроцессорное электронное устройство, на которое через устройство сопряжения с объектом (УСО) собираются сигналы технологических параметров. ПЛК выполняет функцию непосредственного сбора данных, их обработку и выдачу управляющих воздействий посредством алгоритмов автоматического регулирования. Кроме того ПЛК предоставляет данные для визуализации ТП, а также получает данные ручного вмешательства оператора от системы верхнего уровня.

Сигнализация (термин) – процесс уведомления оперативного персонала о нарушении технологического процесса или работы оборудования автоматизации. Способы сигнализации могут быть различных типов воздействия на органы чувств человека с целью привлечения внимания. Часто предусматриваются следующие типы сигнализации:

- *Световая сигнализация* – обычно осуществляется посредством смены цвета графического объекта (миганием) для возникающих событий и установкой статичных аварийных цветов (красный и жёлтый) для сквитированных событий.
- *Звуковая* – осуществляется выдачей звукового сигнала в момент возникновения события. Тип звукового сигнала может быть как монотонным, так и синтезированным речевым сообщением с информацией о нарушении.

ТП (аббревиатура) - Технологический Процесс. Весь комплекс технологического оборудования производственного процесса.

УСО (аббревиатура) – Устройство Сопряжения с Объектом. Ряд устройств или модулей ПЛК, к которым непосредственно подключаются сигналы с датчиков ТП для последующего преобразования из аналогового в цифровой вид и обратно. Преобразование осуществляется с целю последующей обработки значений технологических параметров в ПЛК.

Уставка сигнализации (термин) - условная граница значения технологического параметра, преодоление которой считается нештатной ситуацией. Обычно предусматриваются следующие границы:

- *Верхняя и нижняя аварийные границы* – границы аварийных значений технологического параметра.
- *Верхняя и нижняя предупредительные границы* – границы предупреждения, регламентные границы, о выходе технологического параметра за рабочий диапазон.
- *Отказ* - признак выхода значения параметра за аппаратные границы технологического оборудования. Обычно характеризует отказ датчика, обрыв канала связи с датчиком или ПЛК.

SCADA (аббревиатура-ан.) - Supervisory Control And Data Acquisition (Диспетчерская система управления и сбора данных). Программное обеспечение, выполняющее комплекс задач по сбору данных ТП, их архивирование и представление, а также выдачу управляющих воздействий оператором в ручном режиме.

2. Установка и запуск

Установку дистрибутива OpenSCADA можно осуществить двумя способами. Первый и простой способ - это получить готовые пакеты для используемого дистрибутива ОС Linux. Второй - собрать систему OpenSCADA из исходных текстов. В целом процедура установки сильно зависит от используемого дистрибутива Linux и исчерпывающе её описать в данном руководстве не представляется возможным! Поэтому может понадобиться глубокое знакомство с механизмами установки ПО выбранного дистрибутива Linux в его документации.

В случае отсутствия у пользователя достаточно глубоких знаний и умений в выбранном дистрибутиве Linux, настоятельно рекомендуется выбирать дистрибутив Linux по критерию наличия для него пакетов системы OpenSCADA в репозиториях дистрибутива, что позволит и гарантирует простейшую и безпроблемную установку!

Если у пользователя вызывает затруднение установка не только OpenSCADA, но и дистрибутива Linux, то на первое время он может воспользоваться "живым" дистрибутивом Linux, с установленной и готовой для работы и изучения демонстрацией OpenSCADA. На данный момент доступны "живые" сборки на основе дистрибутива ALTLinux в виде CD и Flash-образов, на странице: http://oscada.org/ru/glavnaja/zagruzit.

Внимание! Динамическая модель компрессорной станции, на 6 газовых компрессоров, которая лежит в основе демонстрационной БД, требует значительных вычислительных ресурсов, а именно процессора с частотой более 1 ГГц. Данные ресурсы требуются именно для динамической модели и не являются общим показателем ресурсоёмкости программы в конечных задачах!

2.1. Установка OpenSCADA из готовых пакетов

Установка OpenSCADA из готовых пакетов, в свою очередь, может осуществляться двумя методами. Первый - простейший, когда пакеты OpenSCADA уже присутствуют в официальных или дополнительных репозиториях используемого дистрибутива ОС Linux, и установка их - вопрос запуска типичной программы управления пакетами дистрибутива с последующим выбором пакетов OpenSCADA. Второй подразумевает получение пакетов OpenSCADA и установку их вручную.

На данный момент пакеты системы OpenSCADA можно встретить в репозиториях таких дистрибутивов ОС Linux: ALTLinux и дистрибутивах, основанных на пакетной базе Fedora.

Проверить на наличие пакетов OpenSCADA в репозиториях используемого дистрибутива Linux, а также загрузить пакеты OpenSCADA для ручной установки можно на странице загрузки официального сайта OpenSCADA (http://oscada.org/ru/zagruzka).

Описание установки из репозитория выбранного дистрибутива Linux пропустим и отошлём читателя к документации соответствующего дистрибутива.

Для ручной установки пакетов OpenSCADA загрузим их из официального сайта или другого источника. Загрузить можно набор пакетов двух типов.

Первый тип представлен набором из девяти пакетов:

- **openscada** - пакет со всеми файлами, нужными для запуска OpenSCADA в полном объёме, включая все модули;
- **openscada-LibDB.Main** - основные библиотеки OpenSCADA для сбора данных и другого в БД SQLite;
- **openscada-LibDB.VCA** - библиотеки визуальных компонетов в БД SQLite;
- **openscada-Model.AGLKS** - БД и конфигурация модели "АГЛКС" (Демо: EN,RU,UK);
- **openscada-Model.Boiler** - БД и конфигурация модели "Котёл" (только Русский);
- **openscada-docEN** - документация на систему OpenSCADA - Английский язык;
- **openscada-docRU** - документация на систему OpenSCADA - Русский язык;
- **openscada-docUK** - документация на систему OpenSCADA - Украинский язык;

- **openscada-devel** - пакеты разработки, для отдельного создания модулей к системе OpenSCADA.

Второй тип представлен набором из порядка пятидесяти пакетов с выделением модулей OpenSCADA в отдельные пакеты:

- **openscada-core** - содержит ядро OpenSCADA, базовую конфигурацию и запускающие файлы;
- **openscada-DB.*** - модули подсистемы "БД";
- **openscada-DAQ.*** - модули подсистемы "Сбор данных";
- **openscada-Archive.*** - модули подсистемы "Архивы";
- **openscada-Transport.*** - модули подсистемы "Транспорты";
- **openscada-Protocol.*** - модули подсистемы "Транспортные протоколы";
- **openscada-UI.*** - модули подсистемы "Пользовательские интерфейсы";
- **openscada-Special.*** - модули подсистемы "Специальные";
- **openscada-LibDB.Main** - основные библиотеки OpenSCADA для сбора данных и другого в БД SQLite;
- **openscada-LibDB.VCA** - библиотеки визуальных компонетов в БД SQLite;
- **openscada-Model.AGLKS** - БД и конфигурация модели "АГЛКС" (Демо: EN,RU,UK);
- **openscada-Model.Boiler** - БД и конфигурация модели "Котёл" (только Русский);
- **openscada-docEN** - документация на систему OpenSCADA - Английский язык;
- **openscada-docRU** - документация на систему OpenSCADA - Русский язык;
- **openscada-docUK** - документация на систему OpenSCADA - Украинский язык;
- **openscada-devel** - пакеты разработки, для отдельного создания модулей к системе OpenSCADA;
- **openscada** - виртуальный пакет, содержащий зависимости, для установки типовой конфигурации OpenSCADA;
- **openscada-plc** - виртуальный пакет, содержащий зависимости, для установки типовой конфигурации OpenSCADA как ПЛК;
- **openscada-server** - виртуальный пакет, содержащий зависимости, для установки типовой конфигурации OpenSCADA как SCADA-сервер;
- **openscada-visStation** - виртуальный пакет, содержащий зависимости, для установки типовой конфигурации OpenSCADA как визуальная SCADA-станция.

Первый тип набора пакетов предназначен для простой-ручной установки, поскольку содержит только девять пакетов. Второй тип предназначен для помещения в репозиторий дистрибутива Linux и последующей установки их с помощью пакетного менеджера, осуществляющего автоматическое разрешение зависимостей. Второй тип набора пакетов позволяет установить только нужные компоненты OpenSCADA, тем самым оптимизируя рабочее окружение, чего не позволяет осуществить набор пакетов первого типа.

При установке из репозитория выбираем только пакет "openscada-Model.AGLKS". Всё остальное, в соответствии с зависимостями, будет выбрано и установлено автоматически.

Ручную установку RPM-пакетов первого типа можно осуществить командой, предварительно сменив рабочую директорию на директорию с пакетами:

```
# rpm -i openscada-LibDB.Main-0.7.1-alt1.noarch.rpm openscada-LibDB.VCA-0.7.1-
alt1.noarch.rpm openscada-Model.AGLKS-0.7.1-alt1.i586.rpm openscada-0.7.1-
alt1.i586.rpm
```

Ручную установку DEB-пакетов первого типа можно осуществить командой, предварительно сменив рабочую директорию на директорию с пакетами:

```
# dpkg -i openscada-libdb.main-0.7.1-1_all.deb openscada-libdb.vca-0.7.1-
1_all.deb openscada-model.aglks-0.7.1-1_all.deb openscada_0.7.1-1_i386.deb
```

В процессе выполнения могут возникнуть ошибки, связанные с неудовлетворёнными зависимостями. При ручной установке из пакетов удовлетворять их прийдётся в ручную, подобно установке пакетов OpenSCADA, или через менеджер пакетов дистрибутива Linux. Детальнее

ознакомиться с процессом установки ПО в RPM-пакетах можно по ссылке: http://skif.bas-net.by/bsuir/admin/node51.html.

2.2. Установка из исходных текстов

Если нет возможности получить готовые пакеты OpenSCADA для выбранного дистрибутива, то остаётся только вариант сборки OpenSCADA из исходных текстов. Процесс сборки OpenSCADA детально описан в руководстве по ссылке http://wiki.oscada.org/Doc/SborkaIzIsxodnikov. Однако нужно иметь в виду, что если Вам удалось собрать OpenSCADA из исходных текстов, то это руководство не для Вас и Вы, скорее всего, можете легко освоить базовую документацию OpenSCADA (http://wiki.oscada.org/Doc/OpisanieProgrammy).

Данный раздел приведён здесь только для полноты и целостности рассмотрения вопроса, поскольку требование квалификационного уровня пользователя для этого раздела значительно выше уровня документа в целом!

3. Первичная конфигурация и запуск

После корректной установки OpenSCADA с демонстрационной БД никакой предварительной настройки не требуется. Если нужно выполнить особую настройку, которая отличается от базовой, то воспользуйтесь документом описания программы OpenSCADA по ссылке: http://wiki.oscada.org/Doc/OpisanieProgrammy#h827-1 .

Внимание! Демонстрация OpenSCADA на основе демонстрационной БД это не то же, что обычно предоставляют коммерческие производители ПО с целью продемонстрировать возможности, но исключить или усложнить нормальную работу путём ограничения функций. Демонстрация OpenSCADA это полно-функциональная система, предоставляющая примеры реализации и настройки различных компонентов. На основе демонстрационной БД OpenSCADA можно легко создавать собственные проекты, используя предоставленные наработки.

Запустить на исполнение OpenSCADA с демонстрационной БД можно из меню окружения рабочего стола в разделе "Графика", пункт "Модель 'АГЛКС' на открытой системе визуального контроля и сбора данных" с характерной иконкой (рис.3.1).

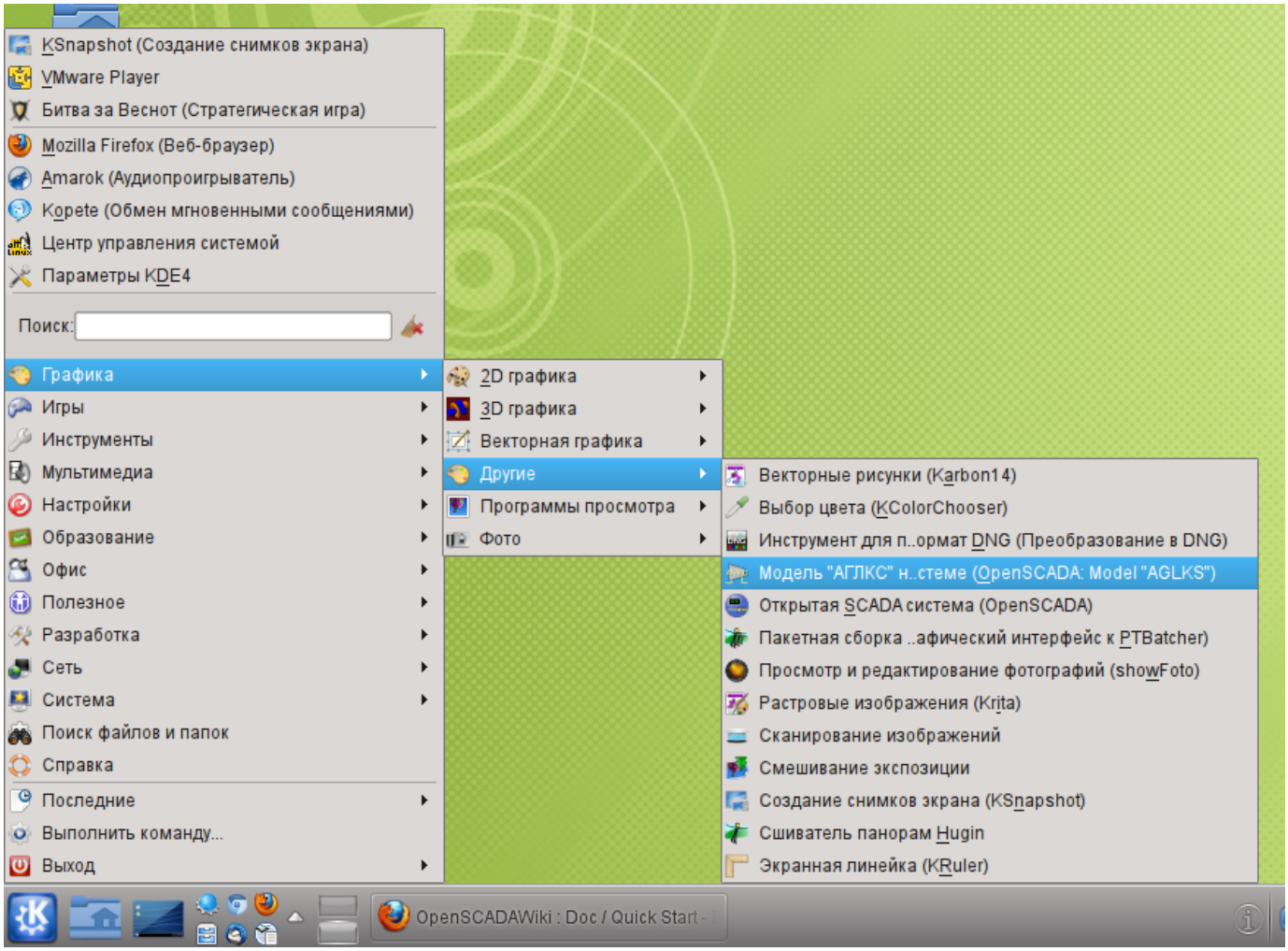

Рис. 3.1. Пункт меню окружения рабочего стола для запуска демонстрации OpenSCADA.

Запуск также можно осуществить из консоли командой:

```
# openscada_demo
```

Внимание! При запуске OpenSCADA из консоли, с помощью команды **"# openscada"**, осуществляется запуск без конфигурации, в результате которого запрашивается пользователь и пароль для входа. По умолчанию в системе OpenSCADA предусмотрен привилегированный пользователь "root" (пароль "openscada") и непривилегированный "user" (пароль "user"), которые не имеют никакого отношения к пользователям операционной системы. Запуск OpenSCADA таким образом имеет смысл только от суперпользователя ОС ("root") или в режиме демона.

После запуска получим окно графического конфигуратора системы OpenSCADA - QTCfg (рис.3.2) с открытой корневой страницей. Демонстрационная БД специально настроена так, что бы первым при запуске появлялось окно конфигуратора. В дальнейшем можно открыть окно разработки графических интерфейсов пользователя, а также запустить проект пользовательского интерфейса на исполнение.

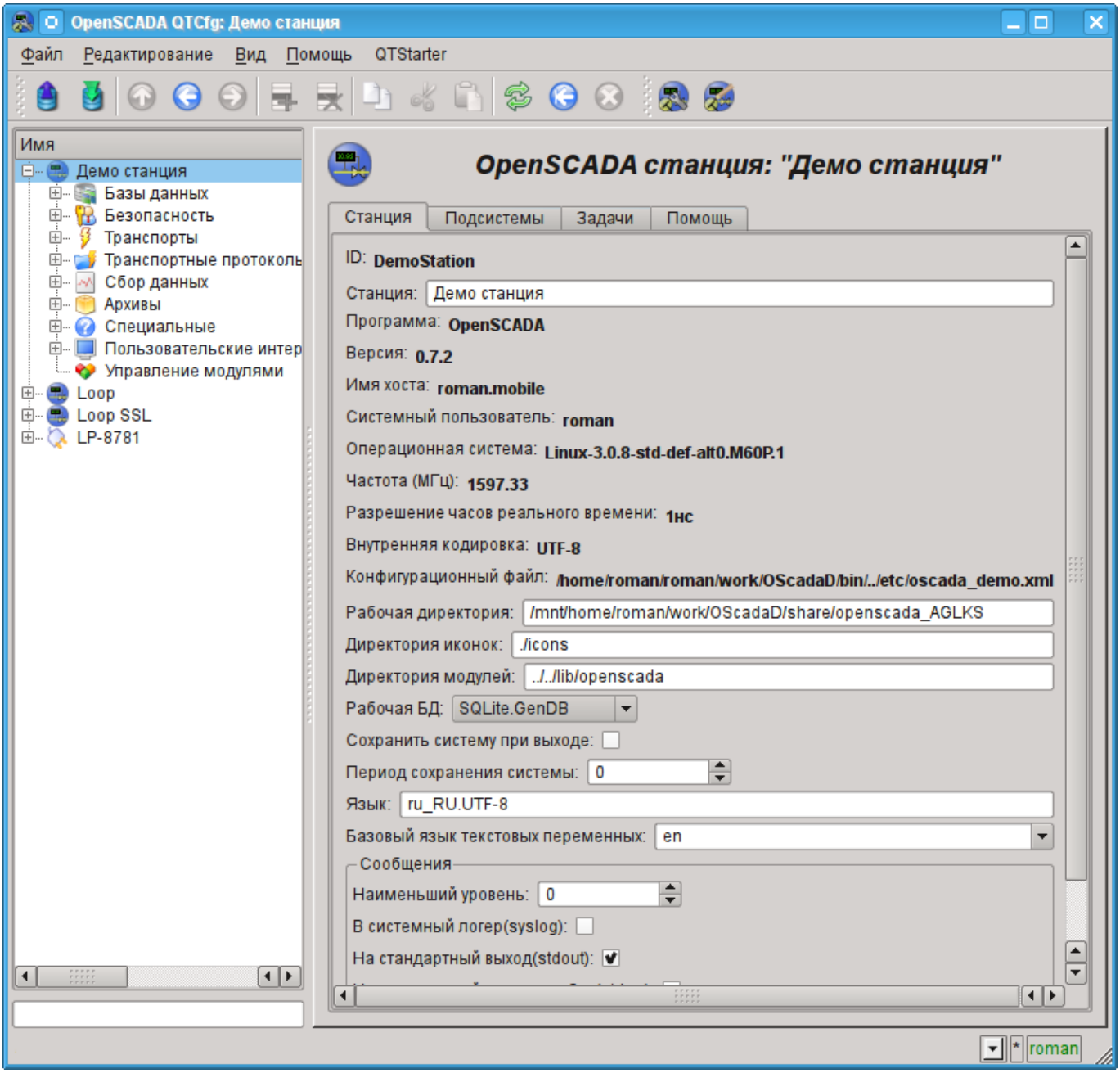

Рис. 3.2. OpenSCADA конфигуратор - QTCfg, корневая страница.

Конфигуратор OpenSCADA является основным и достаточным средством для конфигурации любого компонента системы. Как и многие компоненты OpenSCADA, конфигуратор реализован в виде модуля. Кроме конфигуратора QTCfg могут быть доступны другие конфигураторы, выполняющие те же функции, но реализованные на основе других технологиях. Например, таковыми являются Web-конфигураторы: WebCfg и WebCfgD.

Все действия в дальнейшем мы будем рассматривать только в конфигураторе QTCfg, хотя все их можно будет выполнить и в других конфигураторах.

Структуру интерфейса окна конфигуратора можно детально рассмотреть по ссылке http://wiki.oscada.org/Doc/QTCfg. Для нас же сейчас более важно рассмотреть все доступные интерфейсы OpenSCADA, поэтому нажмём предпоследнюю, с верху, иконку на панели инструментов. После нажатия на эту иконку откроется окно разработки пользовательского интерфейса (рис.3.3).

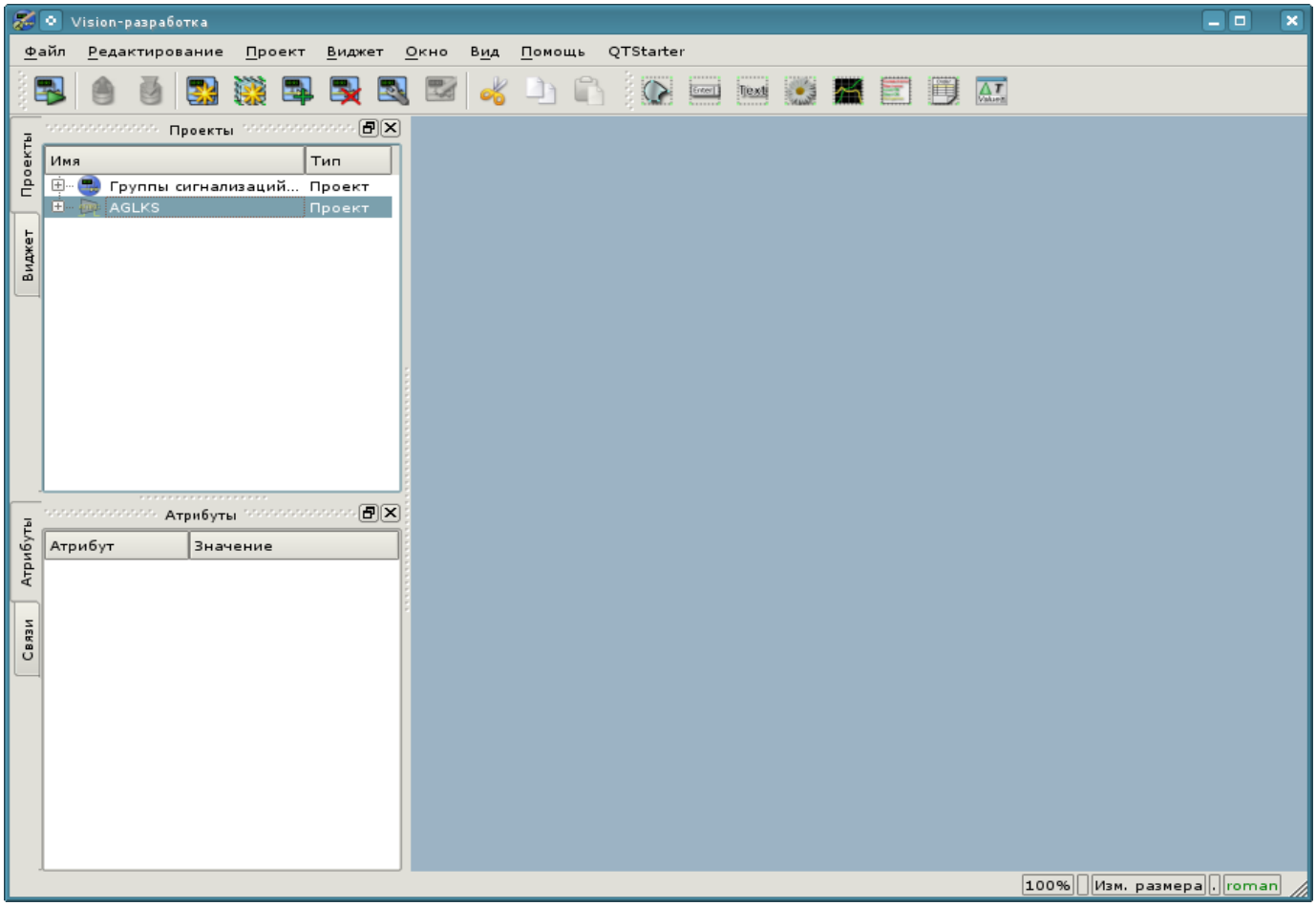

Рис. 3.3. Окно разработки пользовательского интерфейса.

Далее можем запустить проект "AGLKS" на исполнение. Для этого выбираем его в перечне проектов и запускаем путём нажатия на первую слева иконку на панели инструментов или в контекстном меню. В результате получим окно пользовательского интерфейса (рис.3.4).

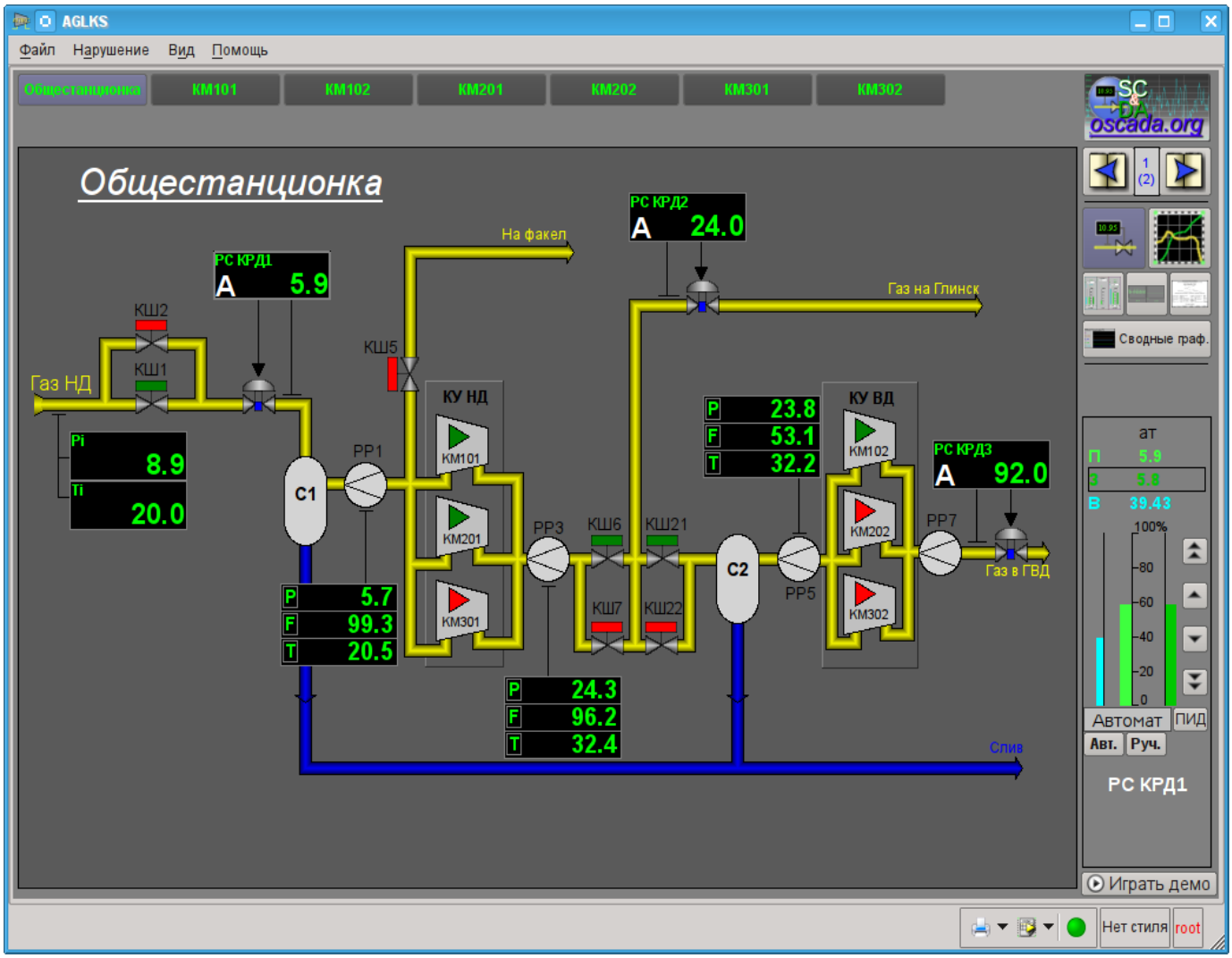

Рис. 3.4. Окно пользовательского интерфейса проекта "AGLKS".

Построение и исполнение пользовательских интерфейсов осуществляется модулем "Vision" подсистемы "Пользовательские интерфейсы". Кроме этого модуля могут быть доступны иные модули визуализации. Например, OpenSCADA предоставляет модуль WebVision, который позволяет исполнять проекты, ранее разработанные в интерфейсе модуля "Vision", посредством Web-технологий и стандартного Web-браузера. Все действия в дальнейшем мы будем рассматривать только в интерфейсе модуля "Vision".

Таким образом мы запустили демонстрацию OpenSCADA и познакомились с основным набором инструментов. В дальнейшем будем их использовать для конфигурации OpenSCADA, создания задач сбора данных, обвязки собранных данных, с целью их обработки и выдачи воздействий, а также для создания пользовательского интерфейса визуализации полученных данных и выдачи управляющих воздействий.

Закроем окно исполнения проекта "AGLKS" и окно разработки пользовательского интерфейса для подготовки к изучению следующих разделов.

Весь процесс конфигурации SCADA-системы для выполнения функций верхнего уровня можно условно разделить на два этапа:

- Конфигурация источников данных и создание базы данных (БД) параметров этих источников.
- Формирование визуального представления данных ТП путём создания интерфейса оператора в виде мнемосхем, групп графиков, групп контуров, документов и т.д.

4. Работа с источниками данных

Основной функцией любой SCADA-системы является работа с источниками данных, а именно опрос программируемых логических контроллеров (ПЛК) и простых модулей УСО. Детальнее ознакомиться с этим вопросом можно в документе "Сбор данных в OpenSCADA" по ссылке: http://wiki.oscada.org/Doc/DAQ .

Поддержка того или иного источника данных зависит от протокола или API, по которому источник свои данные предоставляет, и наличия для протокола/API модуля подсистемы "Сбор данных" в OpenSCADA. Общий перечень модулей подсистемы "Сбор данных" и документацию по ним можно получить по ссылке http://wiki.oscada.org/Doc#h79-4 в соответствующем разделе.

Полученные из источников данные в последствии архивируются, обрабатываются и используются для визуального представления оператору ТП.

4.1. Опрос данных аппарата ТП

В качестве примера рассмотрим и создадим опрос данных для аппарата воздушного холодильника. Демонстрационная БД содержит модель реального времени ТП компрессорной станции из шести компрессоров. Данные для двух аппаратов воздушных холодильников "AT101_1" и "AT101_2" компрессорной станции "KM101" доступны по протоколу ModBus/TCP на порту 10502.

Мы создадим контроллер опроса по протоколу ModBUS/TCP и получим эти данные, тем самым фактически реализовав задачу опроса реальных данных, поскольку от настоящего внешнего устройства наша конфигурация будет отличаться адресом этого устройства и адресами регистров ModBUS.

Для опроса данных по протоколу ModBUS/TCP в OpenSCADA присутствует модуль "ModBUS" подсистемы "Сбор данных". Для добавления нового контроллера откроем в конфигураторе страницу модуля "ModBUS" ("Демо станция"->"Сбор данных"->"Модуль"->"ModBUS") и в контекстном меню пункта "ModBUS" нажмём "Добавить" (рис.4.1.1).

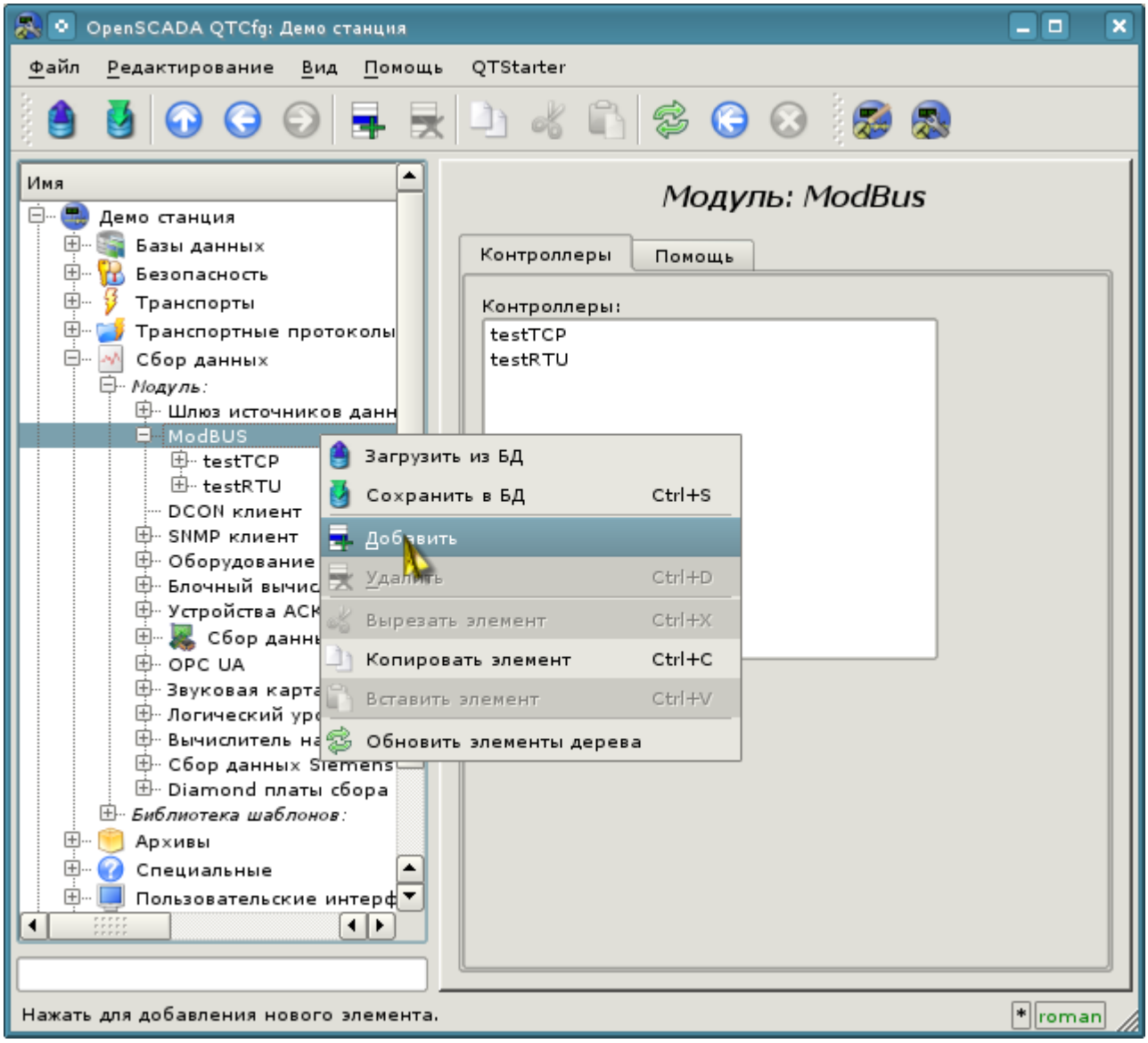

Рис. 4.1.1. Добавление контроллера в модуле "ModBUS" подсистемы "Сбор данных".

В результате появится окно диалога (рис.4.1.2) с предложением ввести идентификатор и имя нового контроллера. Идентификаторы любых объектов в OpenSCADA ограничены размером в 20 символов и их рекомендуется вводить символами английского алфавита и цифрами. Кроме этого, начинать идентификатор желательно с буквы. Это связано с тем, что идентификатор в последствии может использоваться в скриптах. Имена объектов OpenSCADA ограничены размером в 50 символов и могут вводиться любыми символами. Имена обычно используются для отображения. Если поле имени оставить пустым, то вместо него для отображения будет использоваться идентификатор. Введём идентификатор "KM101" и имя "KM 101".

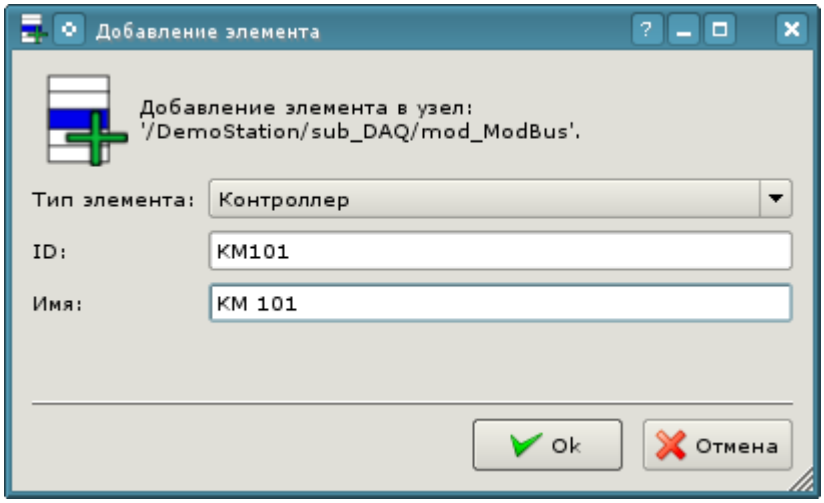

Рис. 4.1.2. Диалог для указания идентификатора и имени нового объекта.

После подтверждения у нас появится объект нового контроллера. Выберем его в конфигураторе и познакомимся с настройками (рис.4.1.3).

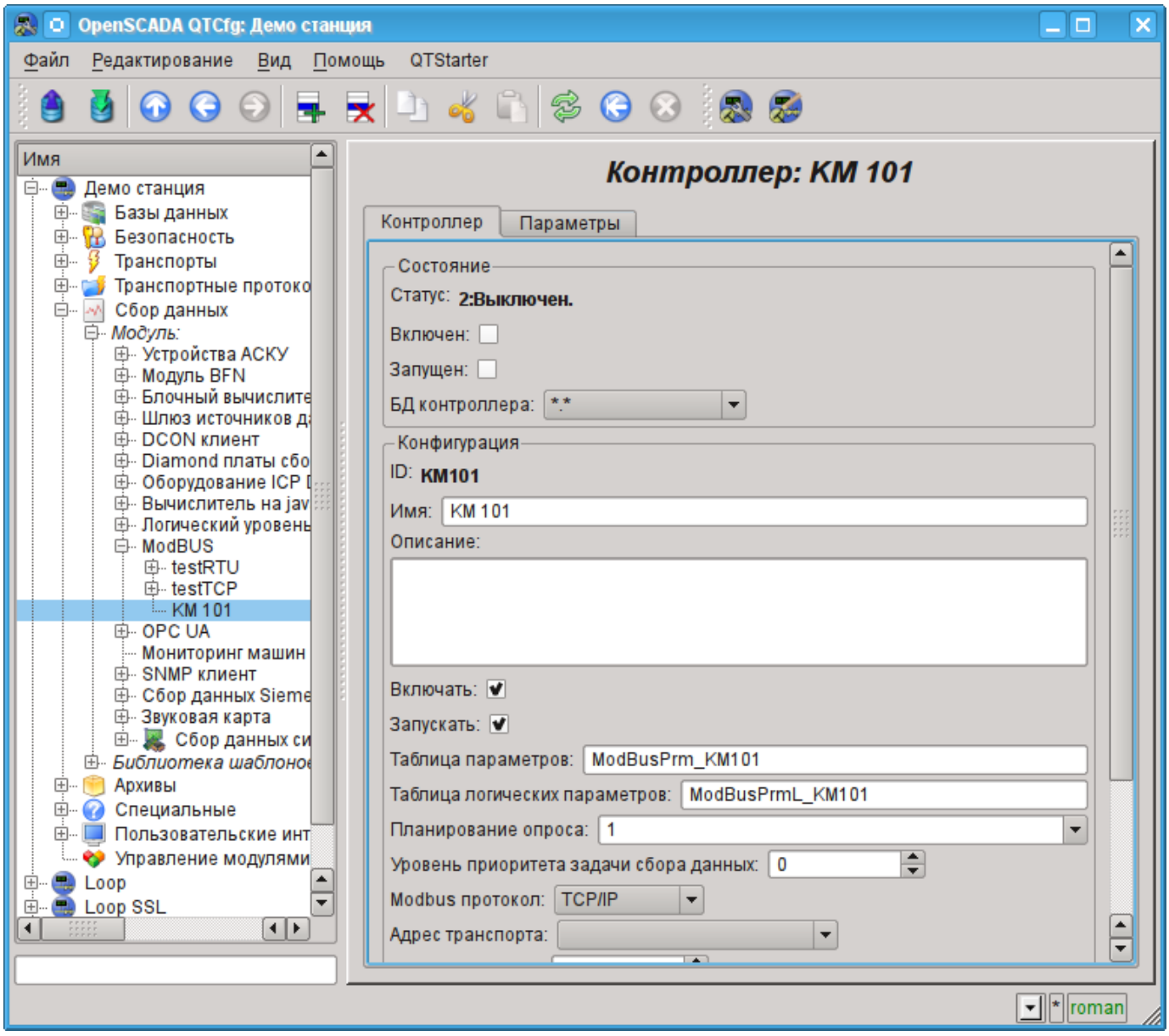

Рис. 4.1.3. Главная вкладка настройки объекта контроллера модуля ModBUS.

Настройки объекта контроллера, как правило, специфичны для разных типов источников данных и протоколов. Детально ознакомиться с настройками объекта контроллера модуля "ModBUS" можно по ссылке http://wiki.oscada.org/Doc/ModBus?v=16m6#h592-14. Мы же рассмотрим общие настройки объекта контроллера и ключевые настройки для модуля "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 контролера 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. Нас 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 мы можем поменять прямо здесь, а вот идентификатор невозможно изменять прямо, однако объект можно вырезать (Ctrl+X) и затем вставить (Ctrl+V), тем самым переименовав его.
- "Описание" может содержать развёрнутую характеристику и назначение объекта контроллера. В нашем случае значение этого поля не принципиально.
- "Включать" и "Запускать" указывают на то, в какое состояние переводить объект контроллера при запуске OpenSCADA. Установим оба поля.
- "Таблица параметров" - содержит имя таблицы БД, в которой будет храниться конфигурация 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. Оставим одну секунду.
- "Уровень приоритета задачи сбора данных" - указывает насколько приоритетна данная задача (от -1 до 99). Приоритеты выше нуля имеют смысл только при запуске OpenSCADA от привилегированного пользователя. Оставим это поле без изменений.
- "ModBUS протокол" - указывает на вариант протокола ModBUS. Нас интересует вариант "TCP/IP", поэтому оставим как есть.
- "Адрес транспорта" - указывает на исходящий транспорт подсистемы "Транспорты", который используется для соединения с контроллером. В случае с вариантом "TCP/IP" нам нужен транспорт в модуле "Sockets". На создании исходящего транспорта в "Sockets" детальнее остановимся ниже.
- "Узел назначения" - указывает узел ModBUS. В нашем случае это должно быть "1".
- "Объединять фрагменты данных" - включает объединение не смежных фрагментов регистров в один блок запроса, до 100 регистров, вместо генерации отдельных запросов. Позволяет уменьшить общее время опроса. Установим эту опцию.
- "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.
- "Время восстановления" - Указывает на время в секундах, через которое в случае отсутствия связи повторять попытку восстановить соединение.

Сохраним наши изменения в БД, нажав вторую слева иконку на панели инструментов.

Теперь таким же образом, как и объект контроллера, создадим исходящий транспорт в модуле "Sockets" ("Демо станция"->"Транспорты"->"Сокеты") посредством контекстного меню (рис.4.1.4). И назовём транспорт так же, как контроллер: "KM101" и имя "KM 101". Обратите внимание, что в поле "Тип элемента" диалога ввода идентификатора и имени (рис.4.1.2) нужно выбрать "Выходной транспорт".

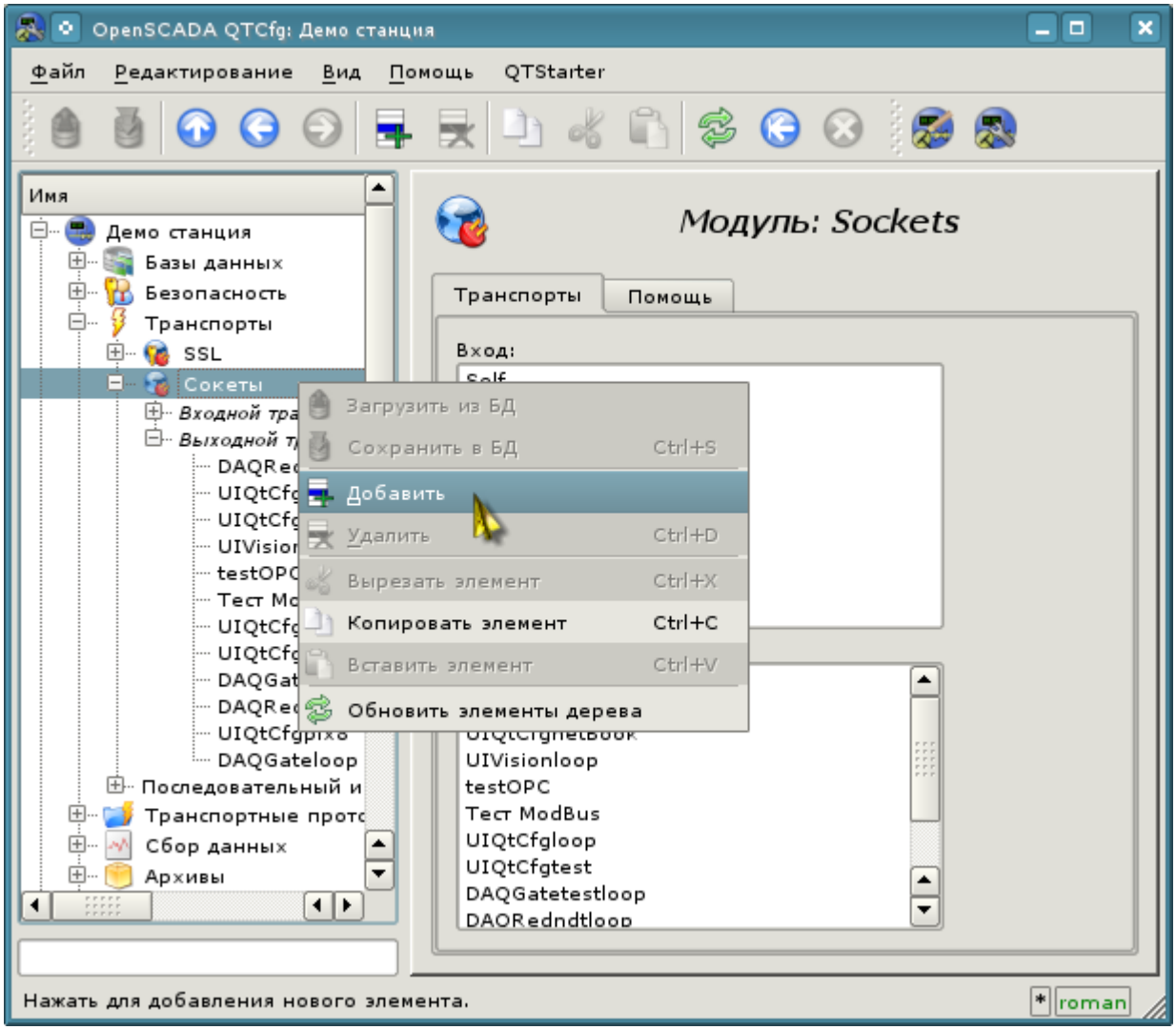

Рис. 4.1.4. Добавление исходящего транспорта в модуле "Sockets" подсистемы "Транспорты".

Страница конфигурации полученного исходящего транспорта приведена на рис.4.1.5. 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.

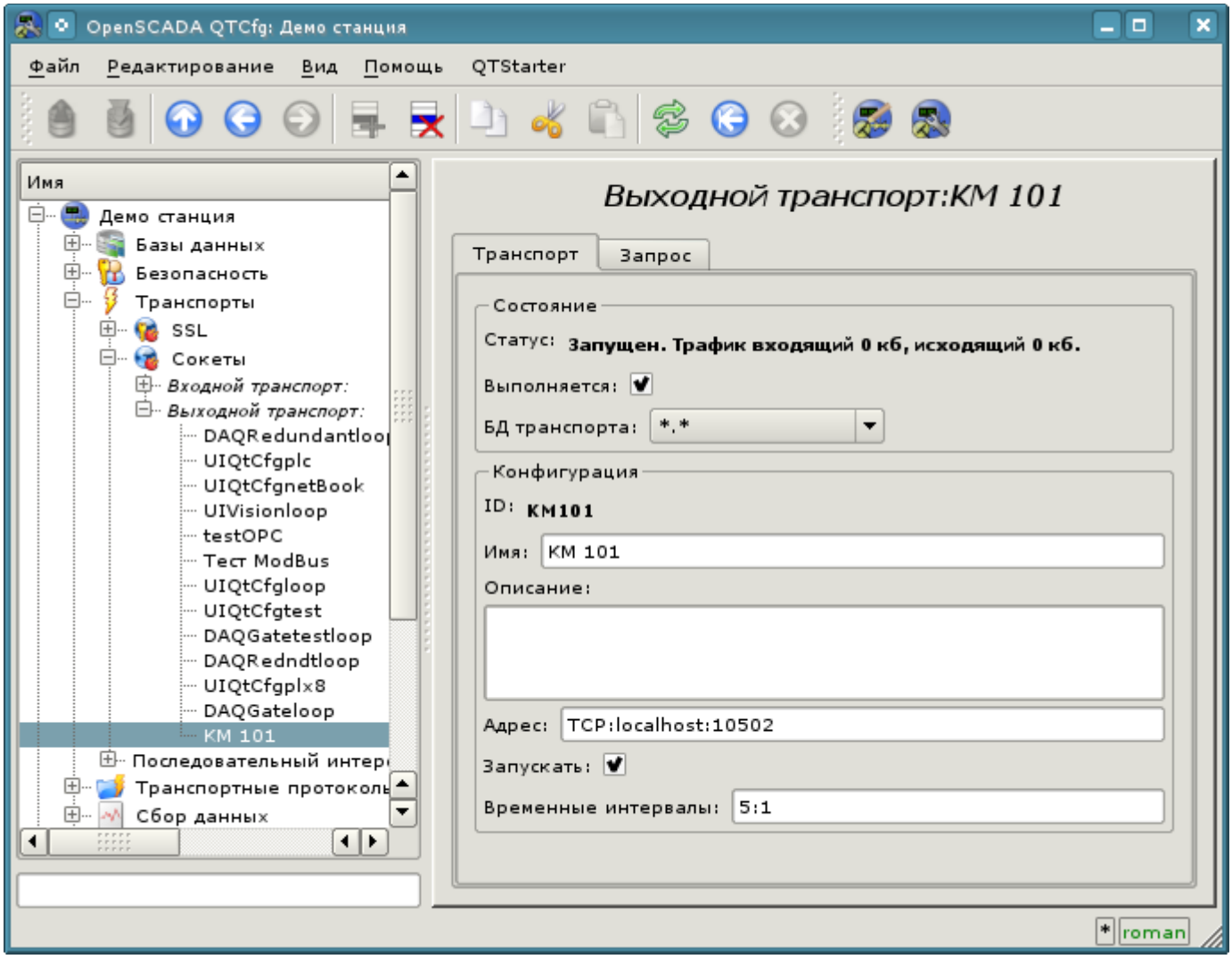

Рис. 4.1.5. Страница конфигурации исходящего транспорта модуля "Sockets" подсистемы "Транспорты".

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 запуска OpenSCADA. Установим поля.
- "Временные интервалы" указывают продолжительность ожидания ответа от удалённой станции. Ознакомиться с форматом записи можно из всплывающей подсказки. Оставим значение неизменным.

Сохраним объект транспорта и вернёмся к конфигурационному полю "Адрес транспорта" объекта контроллера, где выберем адрес "Sockets.KM101". На этом настройка объекта контроллера закончена. Следующим этапом является конфигурация и выбор тех данных, которые нужно опрашивать из контроллера. Эта настройка производится путём создания объекта "Параметр"

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

Для добавления нового объекта параметра откроем в конфигураторе страницу нашего объекта контроллера и в контекстном меню пункта "KM101" нажмём "Добавить". Объект параметра назовём: "AT101_1" и имя "AT 101_1".

Страница конфигурации полученного параметра приведена на рис.4.1.6. Эта страница содержит раздел состояния и оперативного управления. В поле "Тип" содержится идентификатор типа параметра, в нашем случае возможен только тип "Стандартный" (std). Параметр мы можем включить, установив флажок напротив соответствующего поля. Включенный параметр участвует в процессе обмена с контроллером.

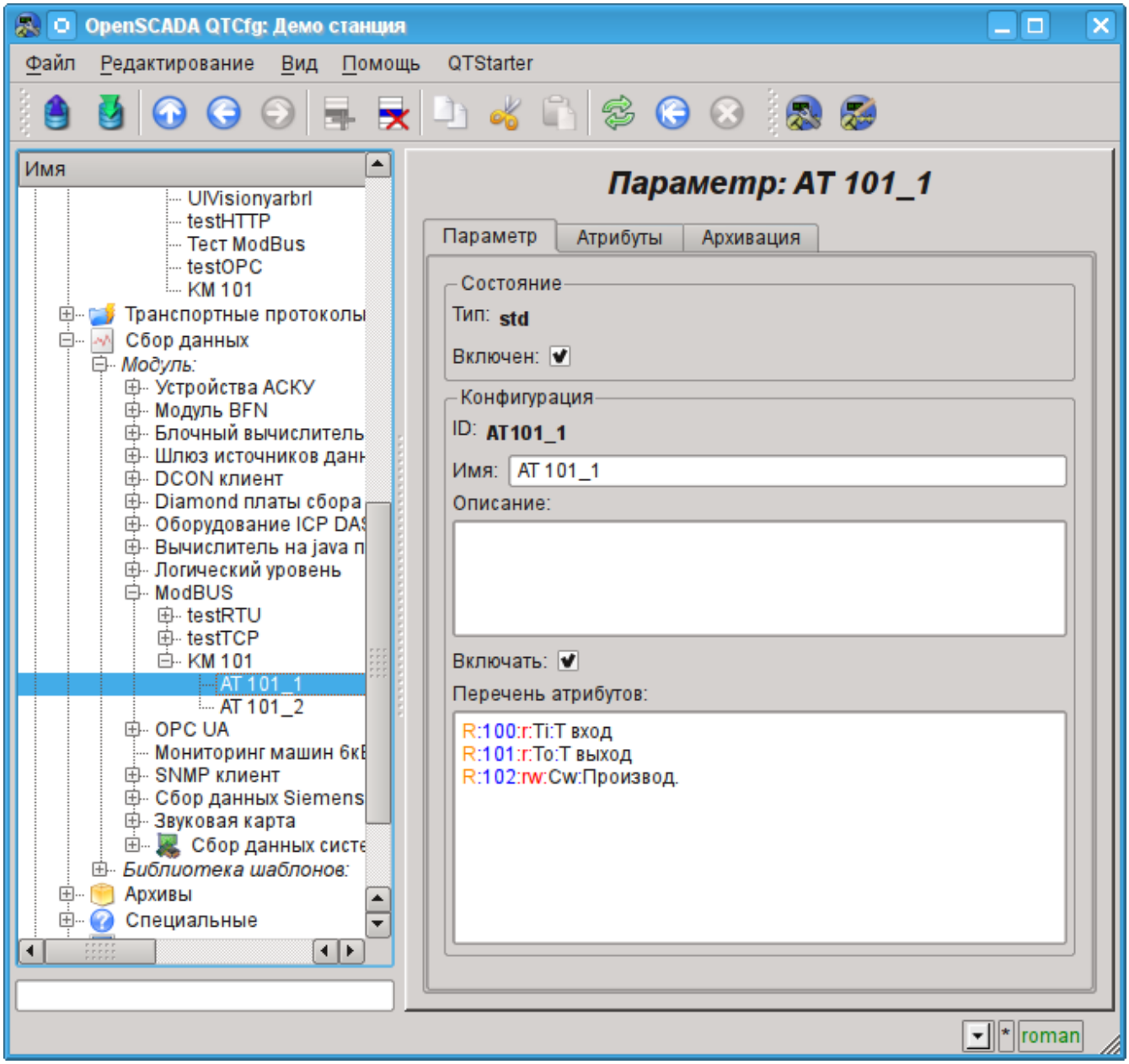

Рис. 4.1.6. Страница конфигурации параметра контроллера "ModBUS".

В разделе "Конфигурация" непосредственно содержится конфигурация объекта параметра:

- "Идентификатор" и "Имя" содержат названия, которые мы вводили при создании объекта.
- "Описание" может содержать развёрнутую характеристику и назначение объекта.
- "Включать" указывает на то, в какое состояние переводить объект при запуска OpenSCADA. Установим поле.

- "Перечень атрибутов" содержит конфигурацию атрибутов параметров в соотношении их с регистрами и битами ModBUS. Ознакомиться с форматом записи можно из всплывающей подсказки. 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: "AT101_2" с именем "AT 101_2". 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". Если мы ничего не пропустили, то обмен успешно запустится и в поле "Статус" мы получим подобное представленному на рис.4.1.7.

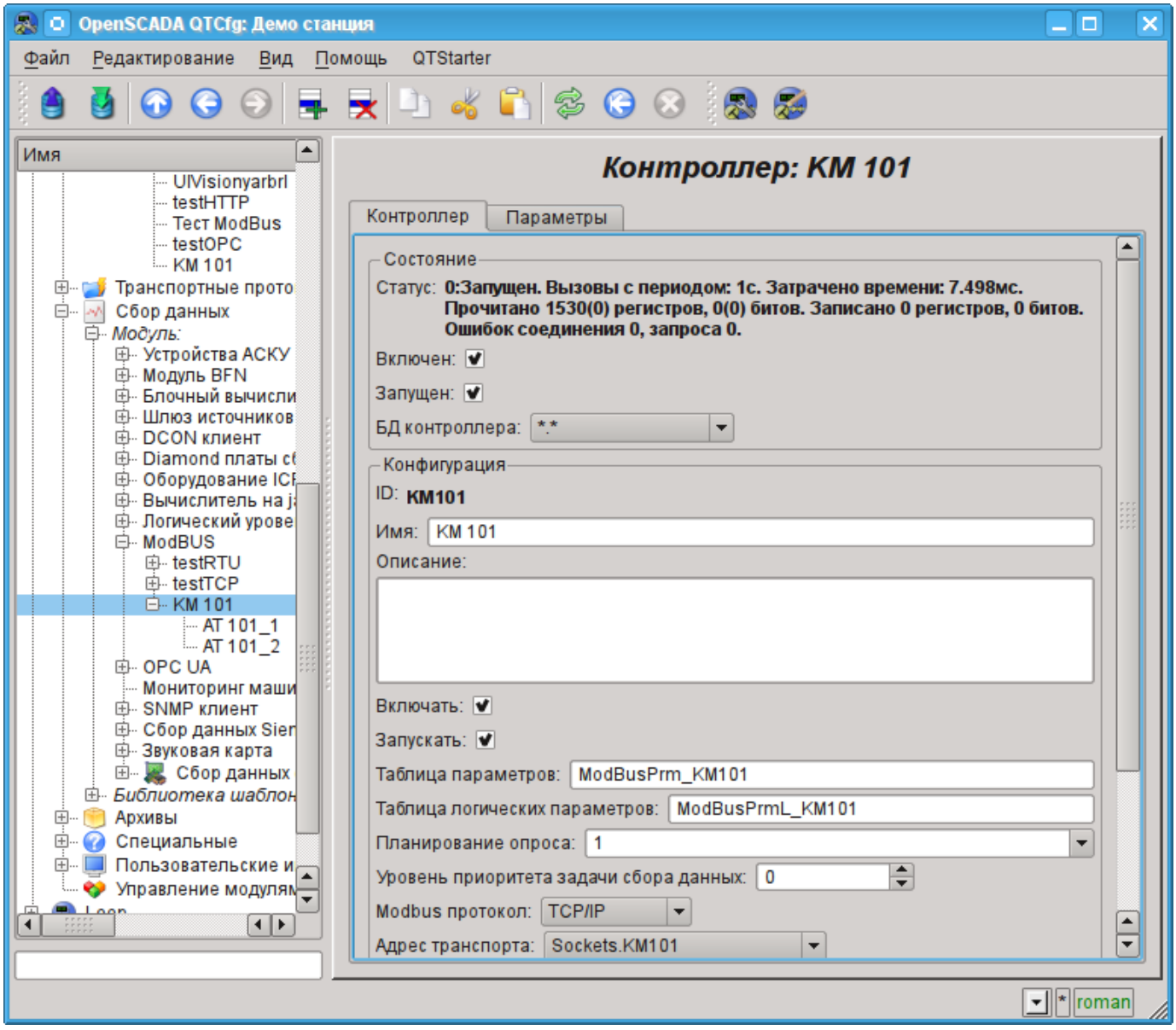

Рис. 4.1.7. Страница объекта контроллера при успешном обмене с физическим контроллером.

В случае успешного обмена с физическим контроллером мы получим описанные данные контроллера в инфраструктуре OpenSCADA. Увидеть эти данные можно на вкладке "Атрибуты" наших параметров AT101_1 (рис.4.1.8) и AT101_2. Поскольку опрос производится регулярно и с периодичностью в секунду, то мы можем наблюдать их изменение, нажимая кнопку "Обновить текущую страницу" на панели инструментов.

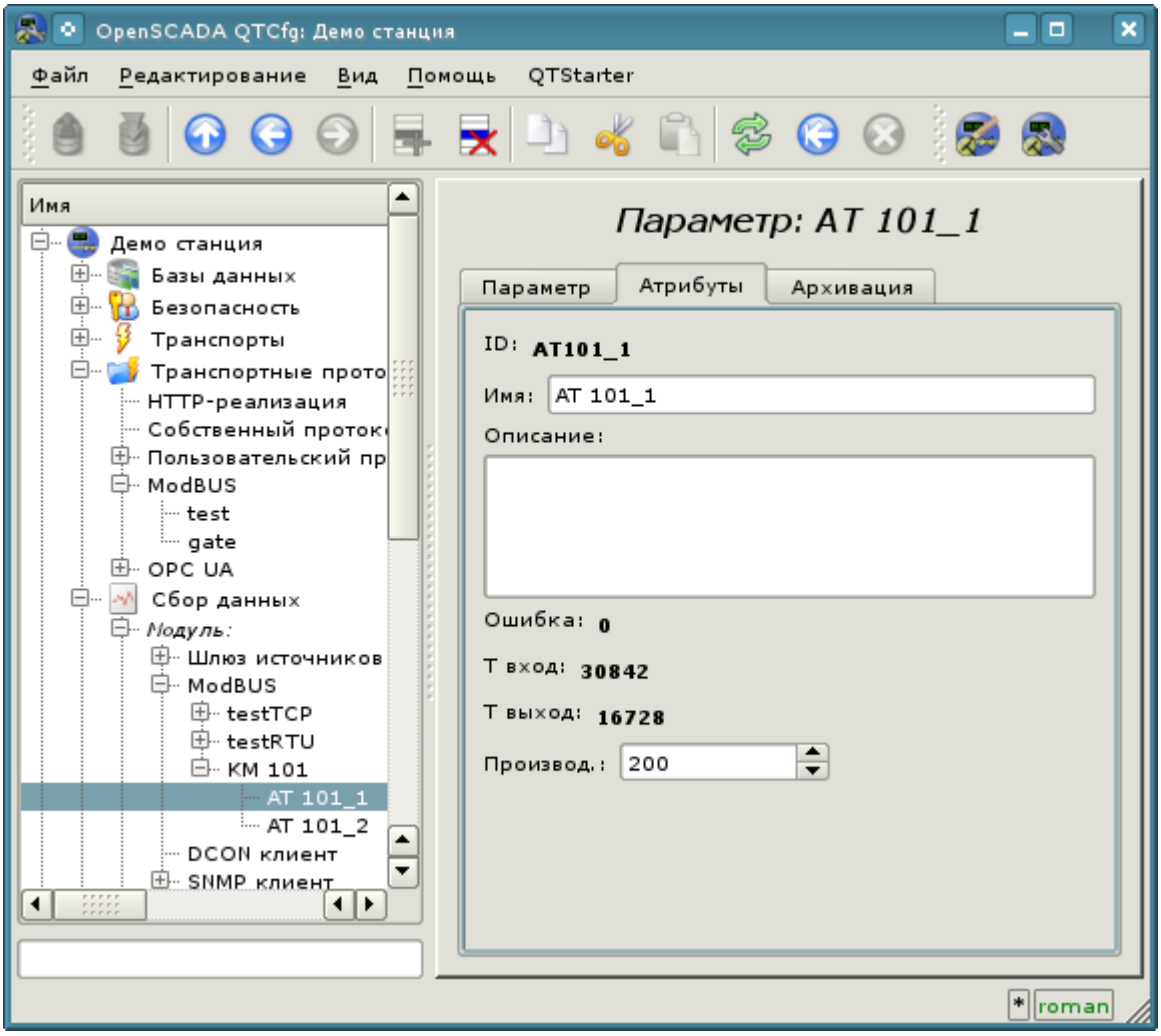

Рис. 4.1.8. Страница описанных атрибутов параметра AT101_1.

На этом конфигурация сбора данных считается законченной.

4.2. Обработка полученных данных ТП

Часто встречается ситуация, когда исходные данные, полученные из источника данных, являются "сырыми", т.е. неготовыми или неудобными для визуального представления, поэтому необходимо выполнить эту подготовку. В нашем примере мы получили данные, которые поступают в коде от шкалы внутри контроллера. Наша задача - выполнить вычисление реальных значений из полученных данных. Обработку данных в OpenSCADA можно делать, как непосредственно при визуализации, так и в подсистеме "Сбор данных". Однако, смешивание процесса визуализации и обработки исходных данных вносит путаницу в конфигурацию, а также делает полученные образы визуализации мало пригодными для повторного использования. По этой причине выполним подготовку данных в подсистеме "Сбор данных".

Вычисления в подсистеме "Сбор данных" выполняются посредством модуля логического уровня "LogicLev" и шаблонов параметров подсистемы "Сбор данных". Детальнее ознакомиться с концепцией "Логического уровня" можно по ссылке http://wiki.oscada.org/Doc/DAQ#h831-9 .

Для выполнения вычислений в модуле логического уровня нужно предварительно создать шаблон параметра подсистемы "Сбор данных". Для этого откроем страницу библиотеки шаблонов

"Базовые шаблоны" ("Демо станция"->"Сбор данных"->"Библиотека шаблонов"->"Базовые шаблоны") и посредством контекстного меню создадим объект шаблона "airCooler" с именем "Воздушный холодильник". Страница конфигурации полученного объекта изображена на рис.4.2.1. Эта страница содержит раздел состояния и оперативного управления. Мы можем сделать шаблон доступным, установив флажок напротив соответствующего поля. Доступные шаблоны могут подключаться к параметрам сбора данных, а параметры будут выполнять вычисления по этому шаблону. В поле "Использовано" указывается число объектов, которые используют данный шаблон для вычисления образа параметра. В разделе "Конфигурация" содержаться только уже знакомые нам поля конфигурации.

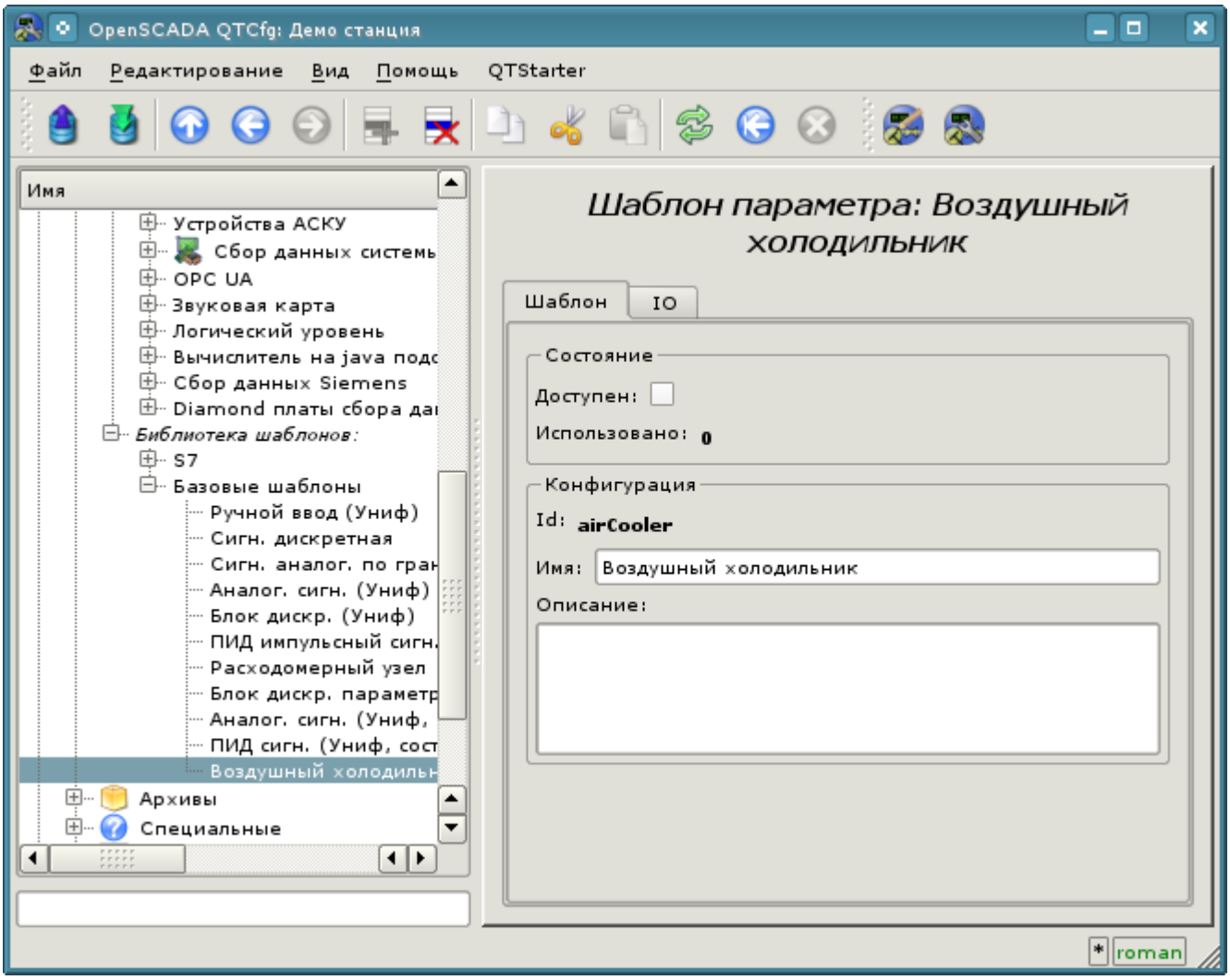

Рис. 4.2.1. Страница конфигурации объекта шаблона.

Основная конфигурация и формирование шаблона параметра сбора данных осуществляется во вкладке "IO" (рис.4.2.2) шаблона. Детальное описание процесса формирования шаблона можно получить по ссылке: http://wiki.oscada.org/Doc/OpisanieProgrammy#h827-6 .

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", а "Cw" в "Полный доступ" для формирования трёх атрибутов: два только на чтение и один на полный доступ у результирующего параметра сбора данных.

Язык программы установим в "JavaLikeCalc.JavaScript", а программу в:

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

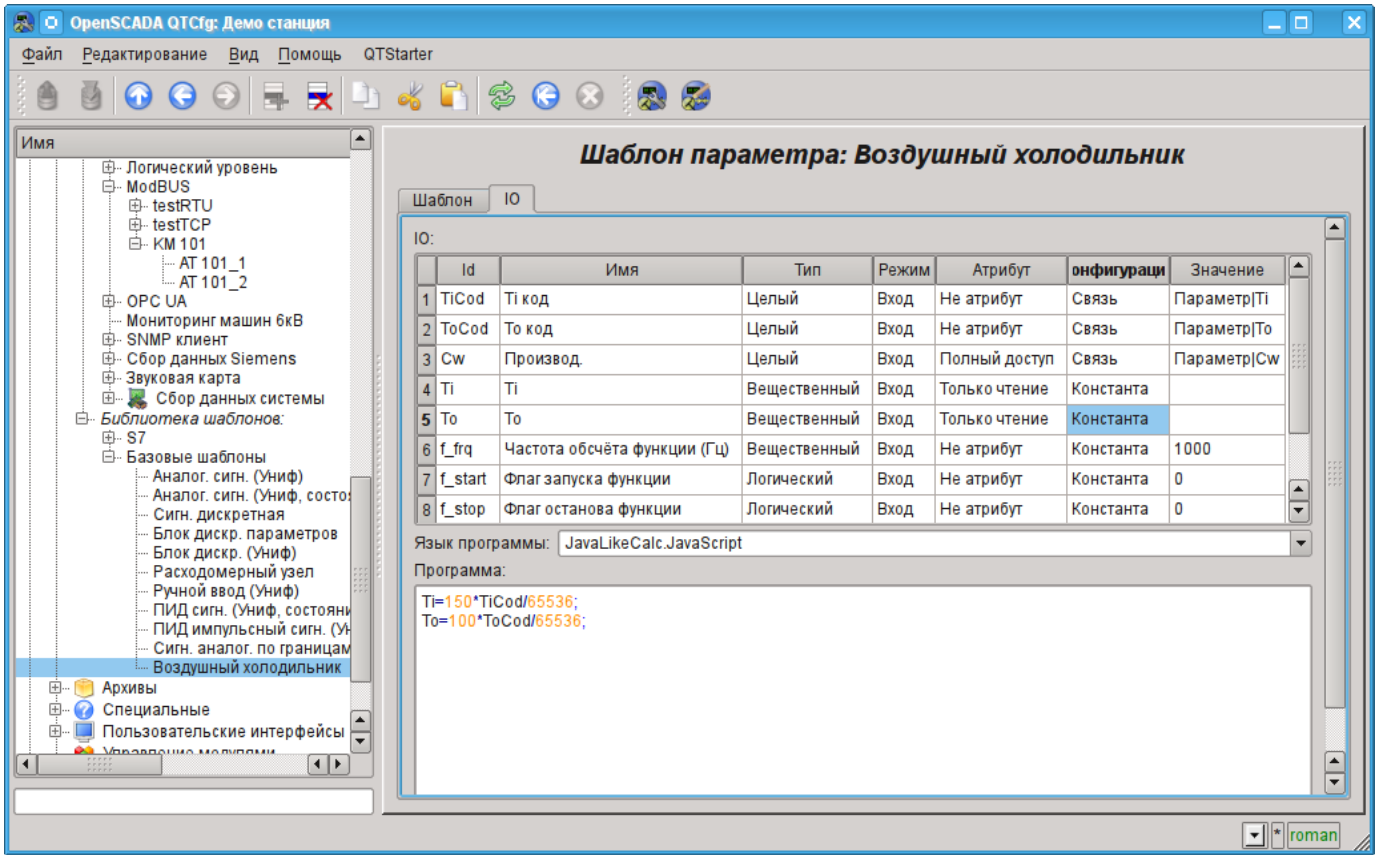

Рис. 4.2.2. Вкладка "IO" страницы конфигурации объекта шаблона.

Полученный шаблон сохраним и установим флаг доступности.

Теперь создадим объекты контроллера и параметров в модуле "LogicLev" подсистемы "Сбор данных". Контроллер и его параметры в модуле "LogicLev" создаются идентично ранее созданным в модуле "ModBUS" на странице: "Демо станция"->"Сбор данных"->"Модуль"->"Логический уровень". Объект контроллера и параметров назовём идентично объектам в модуле "ModBUS".

Объект контроллера модуля "LogicLev" (рис.4.2.3) не имеет специфических настроек и установленные по умолчанию можно не трогать.

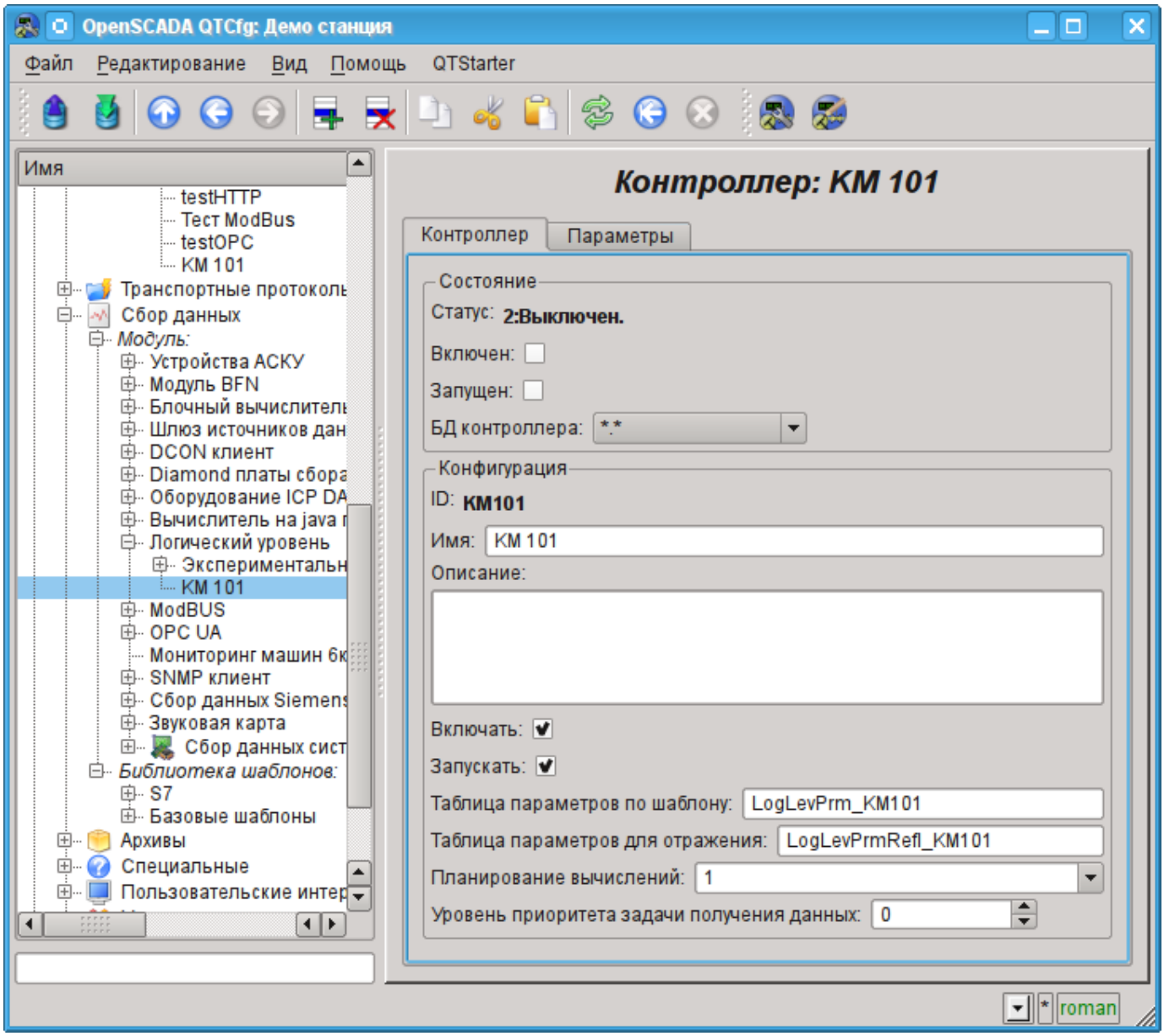

Рис. 4.2.3. Главная вкладка настройки объекта контроллера модуля LogicLev.

Объект параметра контроллера модуля "LogicLev" (рис.4.2.4) из специфических настроек имеет только "Режим", где нужно установить "Шаблон", а справа выбрать адрес созданного нами шаблона.

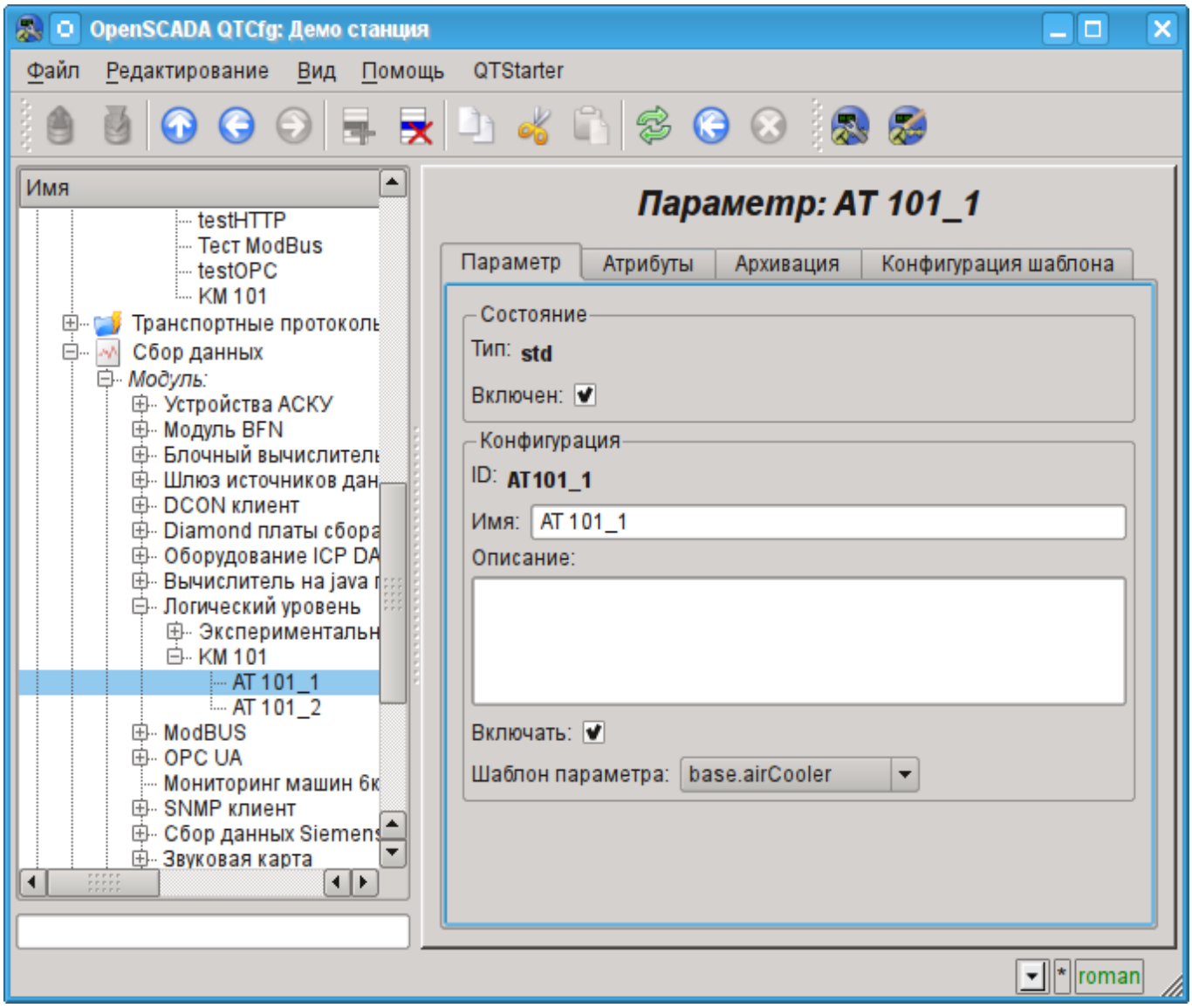

Рис. 4.2.4. Страница конфигурации параметра контроллера "LogicLev".

Кроме базовой конфигурации параметра нужно сконфигурировать подключенный шаблон (рис. 4.2.5). Вкладка конфигурации шаблона появляется в режиме параметра "Включен". Включить параметр можно, включив предварительно контроллер. Флаг "Показывать только атрибуты" позволяет устанавливать отдельно каждую связь (рис.4.2.6). Поскольку мы в шаблоне предусмотрительно описали формат связей в виде "Параметр|Ti", то все три связи мы можем установить просто указав адрес к параметру в контроллере "ModBus". Укажем адреса "ModBus.KM101.AT101_1" и "ModBus.KM101.AT101_2" в соответствующих параметрах.

Нужно отметить, что все поля ввода адресов объектов в OpenSCADA снабжены механизмом набора адреса. Данный механизм подразумевает поэлементный выбор, в процессе которого происходит движение в глубь. Например, при наборе адреса "ModBus.KM101.AT101_1" вначале мы будем иметь возможность выбора типа источника данных, в числе которых будет "ModBus". Выбрав пункт "ModBus" в перечень доступных элементов для выбора добавятся контроллеры модуля "ModBus", в числе которого будет "ModBus.KM101". Выбор элемента "ModBus.KM101" добавит в список параметры контроллера и т.д. до конечного элемента в соответствии с иерархией объектов (http://wiki.oscada.org/Doc/OpisanieProgrammy#h827-6). Для возможность возврата на уровни выше в список выбора вставляются элементы всех вышестоящих уровней от текущего значения адреса.

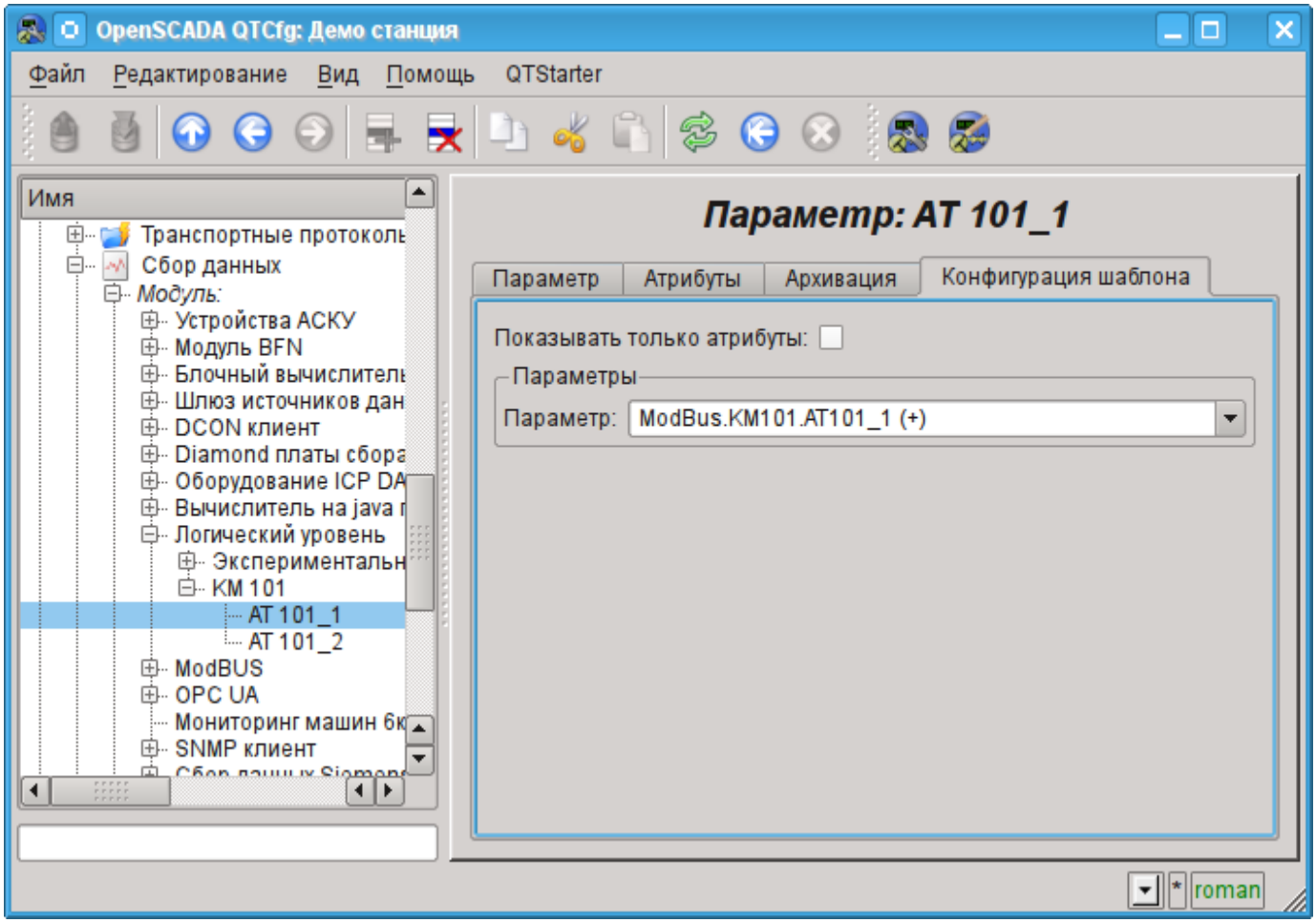

Рис. 4.2.5. Вкладка "Конфигурация шаблона" страницы параметра контроллера "LogicLev".

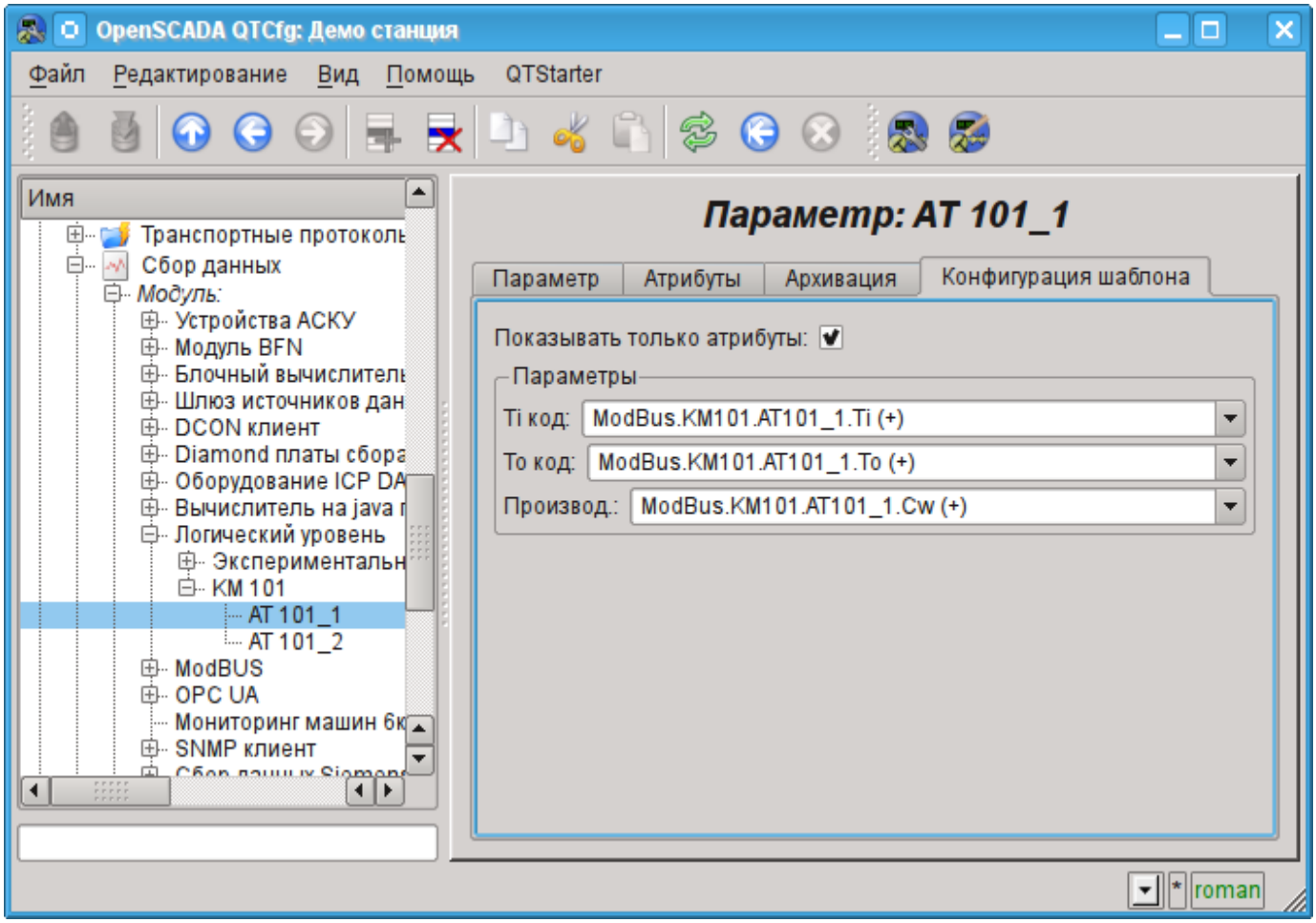

Рис. 4.2.6. Вкладка "Конфигурация шаблона" страницы параметра контроллера "LogicLev" с разворотом связей.

Сохраним созданные объекты контроллера и параметров. После этого запустим контроллер на исполнение, установив флаг контроллера "Запущен" в разделе "Состояние". Если мы ничего не пропустили, то вычисление успешно запустится и в поле "Статус" мы получим подобное представленному на рис.4.2.7.

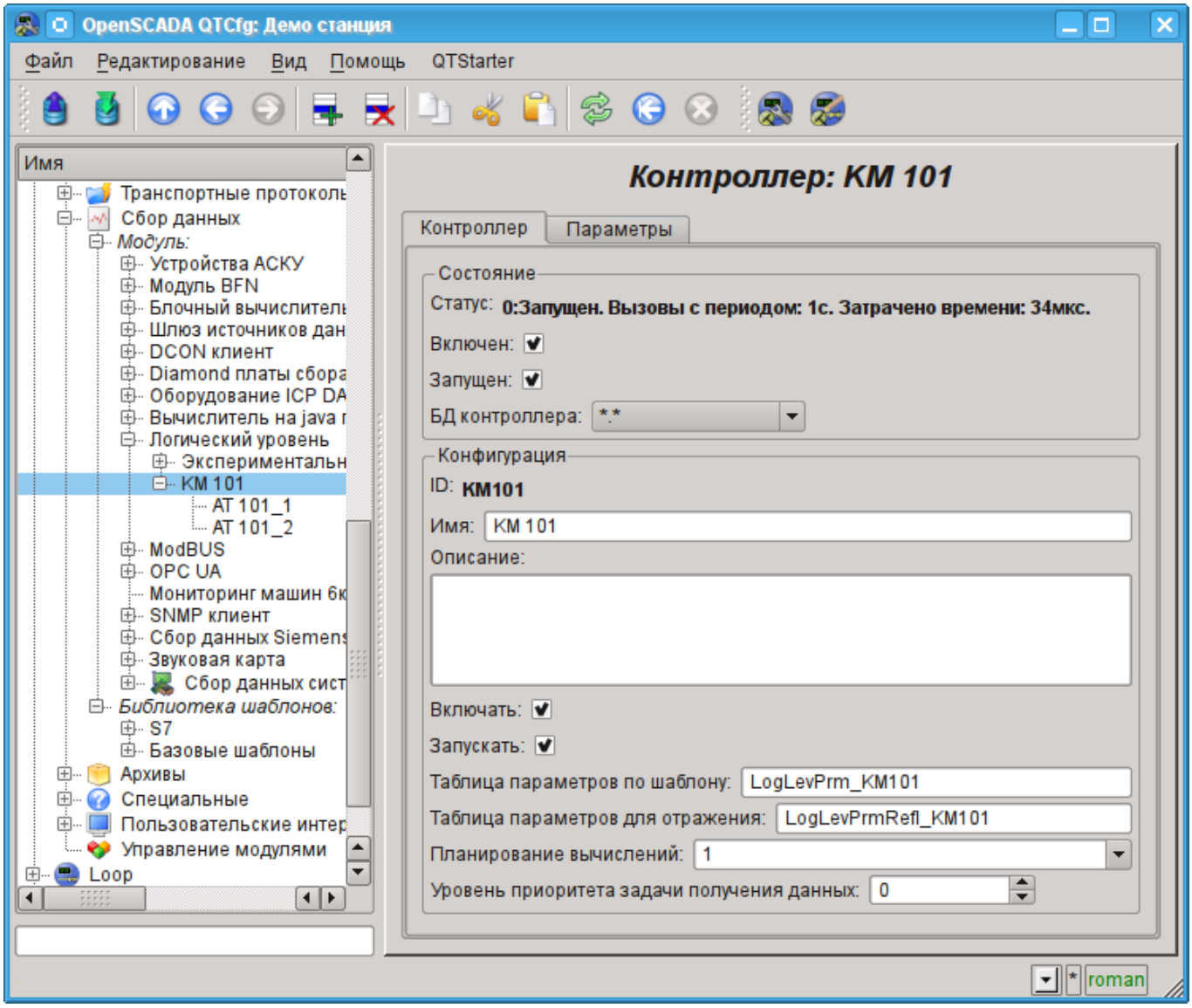

Рис. 4.2.7. Страница объекта контроллера при успешном вычислении контроллера в модуле "LogicLev".

В случае успешной обработки кода шаблона в параметрах мы получим обработанные данные в инфраструктуре OpenSCADA. Увидеть эти данные можно на вкладке "Атрибуты" наших параметров AT101_1 (рис.4.2.8) и AT101_2.

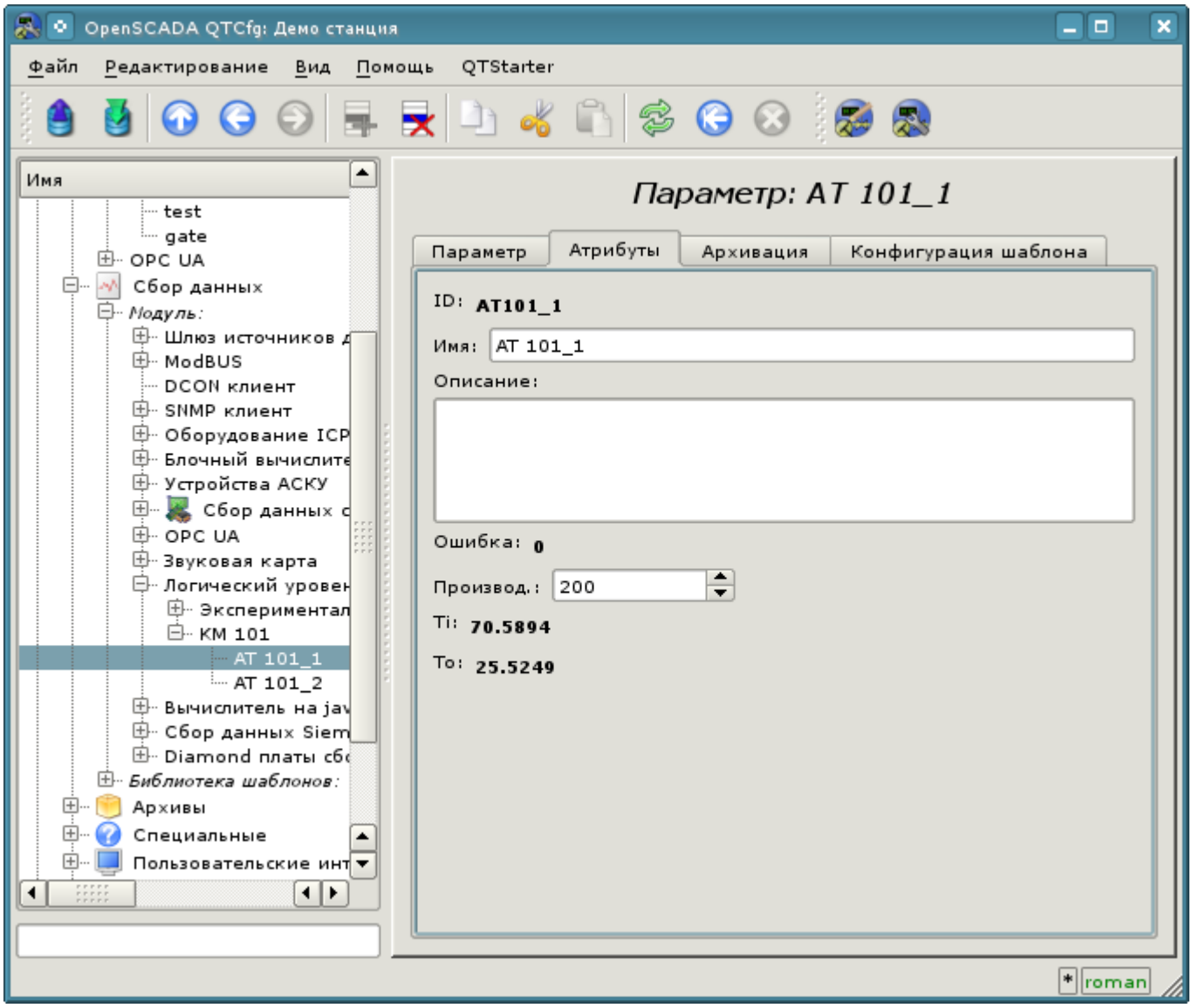

Рис. 4.2.8. Страница атрибутов параметра AT101_1 модуля "LogicLev".

На этом конфигурация обработки данных считается законченной.

4.3. Включение архивирования данных ТП

Для многих задач требуется ведение истории параметров ТП. Для включения архивирования атрибутов "Ti" и "To" параметров AT101_1 и AT101_2 в ранее созданном контроллере модуля "LogicLev" достаточно на вкладке "Архивация" страницы параметров выбрать, какие атрибуты архивировать и на каких архиваторах (рис.4.3.1). Выберем архивацию атрибутов "Ti" и "To" в архиваторе "FSArch.1s".

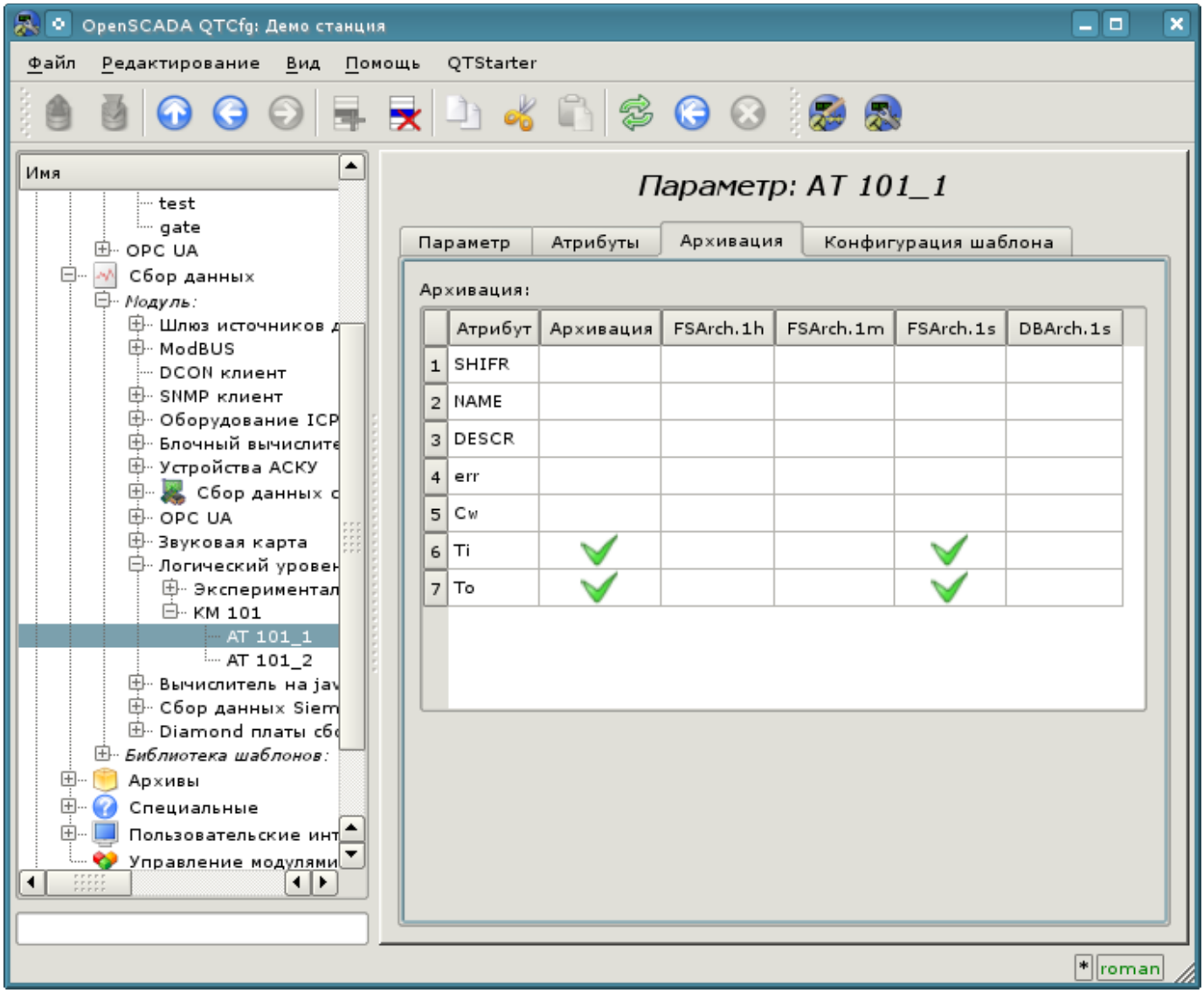

Рис. 4.3.1. Вкладка "Архивация" параметра AT101_1 модуля "LogicLev".

В результате этой операции будут автоматически созданы объекты архивов для выбранных атрибутов. Например, объект архива для атрибута "Ti" параметра AT101_1 представлен на рис.4.3.2.

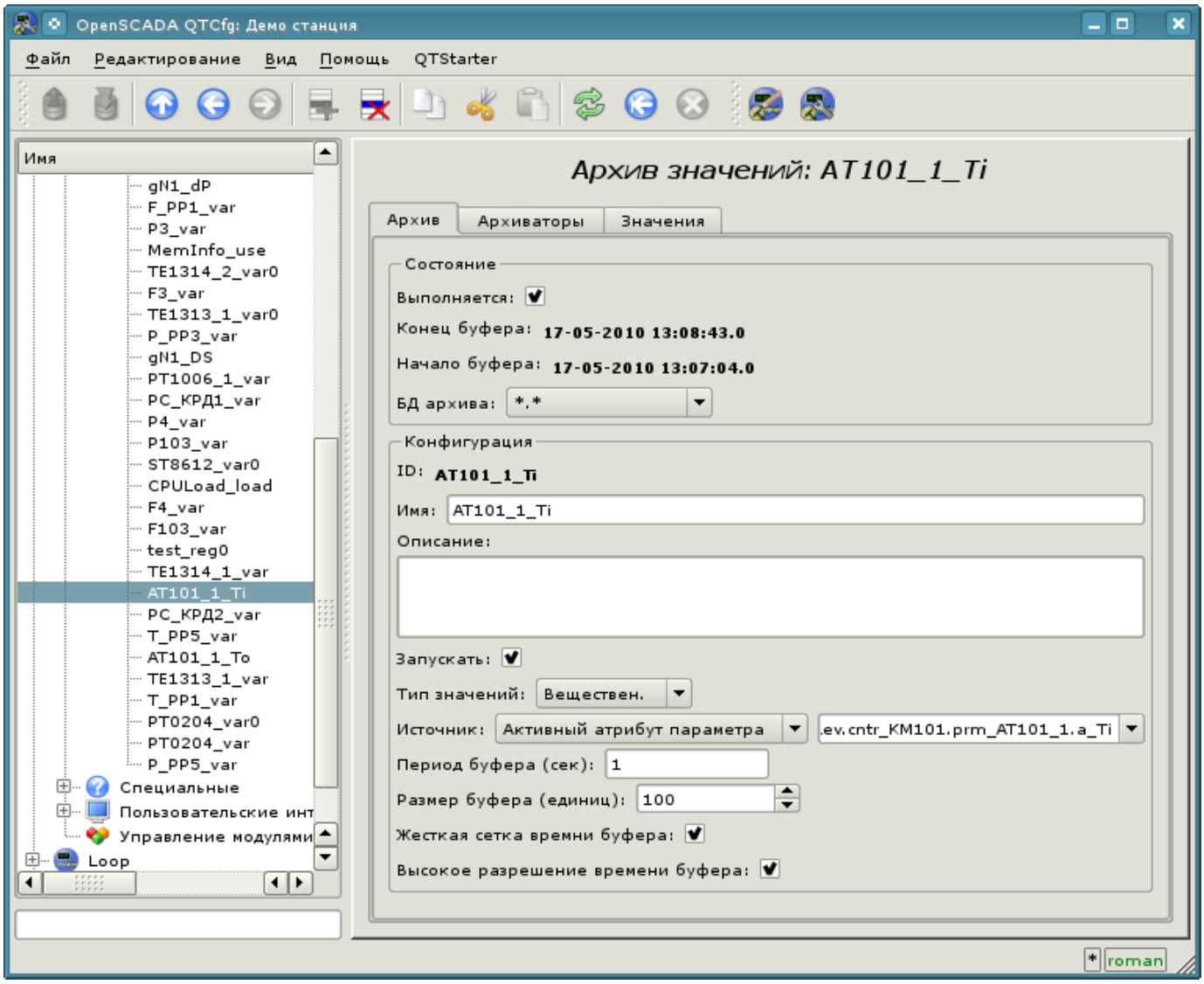

Рис. 4.3.2. Страница объекта архива атрибута "Ti" параметра AT101_1.

Обычно настройки архива менять не нужно, однако, если потребуется особая конфигурация, то её можно сделать на указанной выше странице. Чаще может понадобиться получение информации об архиве. Например, узнать размер архива как по времени, так и на носителе, а также взглянуть на график параметра (рис.4.3.3).

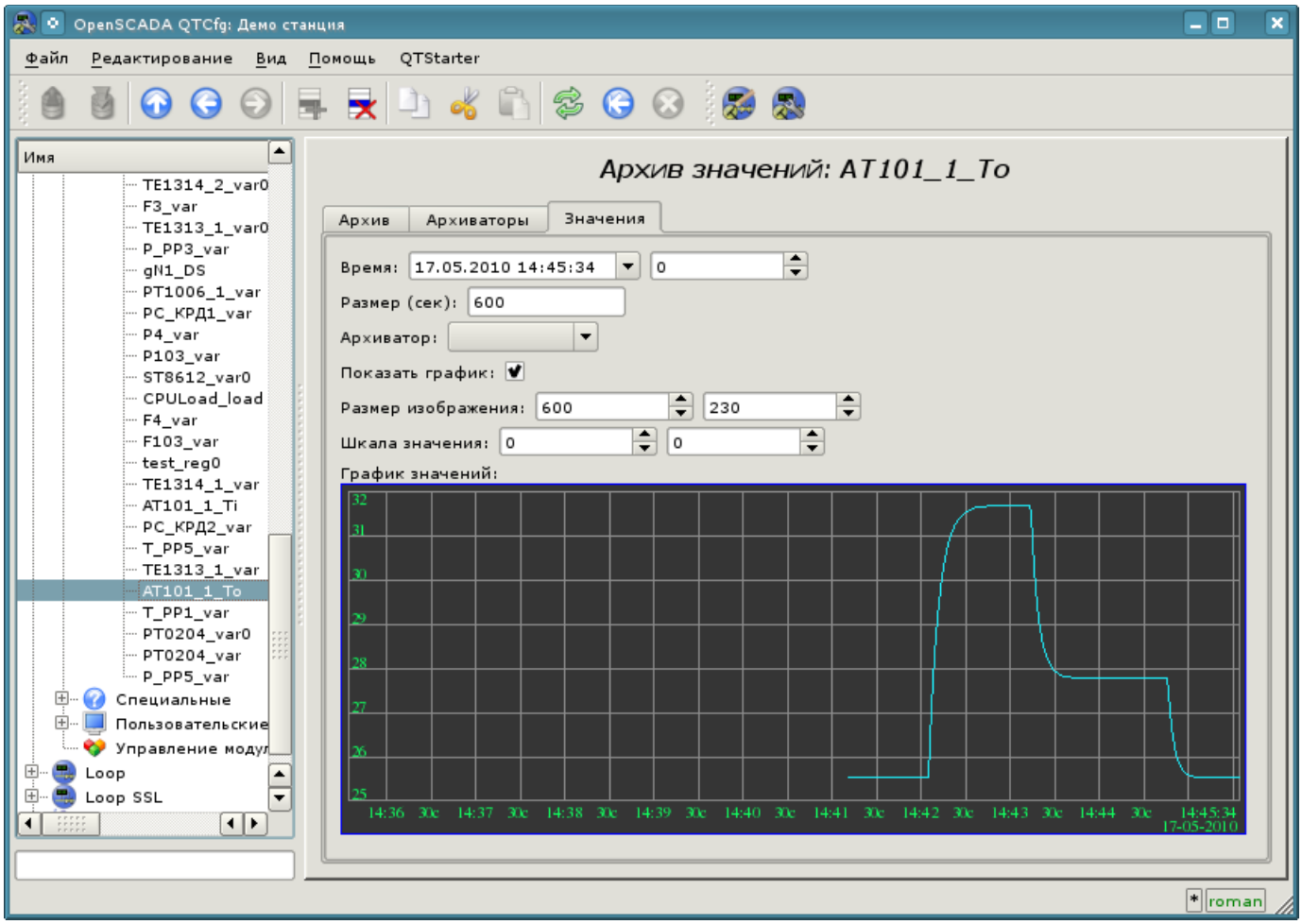

Рис. 4.3.3. Вкладка "Значения" страницы объекта архива атрибута "Ti" параметра AT101_1.

5. Формирование визуального представления

Формирование визуального представления может выполняться на трёх уровнях сложности и пользователь может выбрать любой из них, в зависимости от уровня своих знаний и наличия библиотек с готовыми образами и шаблонами.

Первый уровень требует минимальной квалификации пользователя, но подразумевает наличие библиотек шаблонных кадров, нужных для решения его задачи. В рамках первого уровня пользователю достаточно знать, как подключить динамику к страницам шаблонных кадров и как добавить новые страницы шаблонных кадров.

Второй уровень предусматривает дополнительную способность создавать новые кадры на основе готовых комплексных элементов, путём простого их размещения в области кадра. Для обеспечения этого квалификационного уровня пользователю понадобятся библиотеки комплексных элементов, нужных для решения его задач.

И третий уровень предусматривает владение всеми инструментами среды разработки визуальных интерфейсов OpenSCADA, включая создание новых комплексных элементов и разработки новых интерфейсов пользователя в проекте.

Все работы над интерфейсом визуализации будем выполнять в окружении модуля "Vision" подсистемы "Пользовательские интерфейсы". Для открытия окна интерфейса "Vision" нажмём вторую иконку справа на панели инструментов конфигуратора. В результате получим окно, ранее изображённое на рис.3.3.

5.1. Добавление шаблонной страницы в проект и подключение динамики

Рассмотрим задачу первого уровня сложности, когда в уже разработанном интерфейсе нужно подключить динамику к шаблонной странице. Понятие "Шаблон страницы" подразумевает страницу, на основе которой путём наследования может создаваться множество конечных страниц визуализации с индивидуальным перечнем динамики. Примером таких страниц являются: "Группа графиков", "Группа контуров", "Обзорный кадры" и "Сводная таблица". На рис.5.1.1 представлена шаблонная страница "Группа графиков" в дереве проекта "Группы сигнализаций (шаблон)".

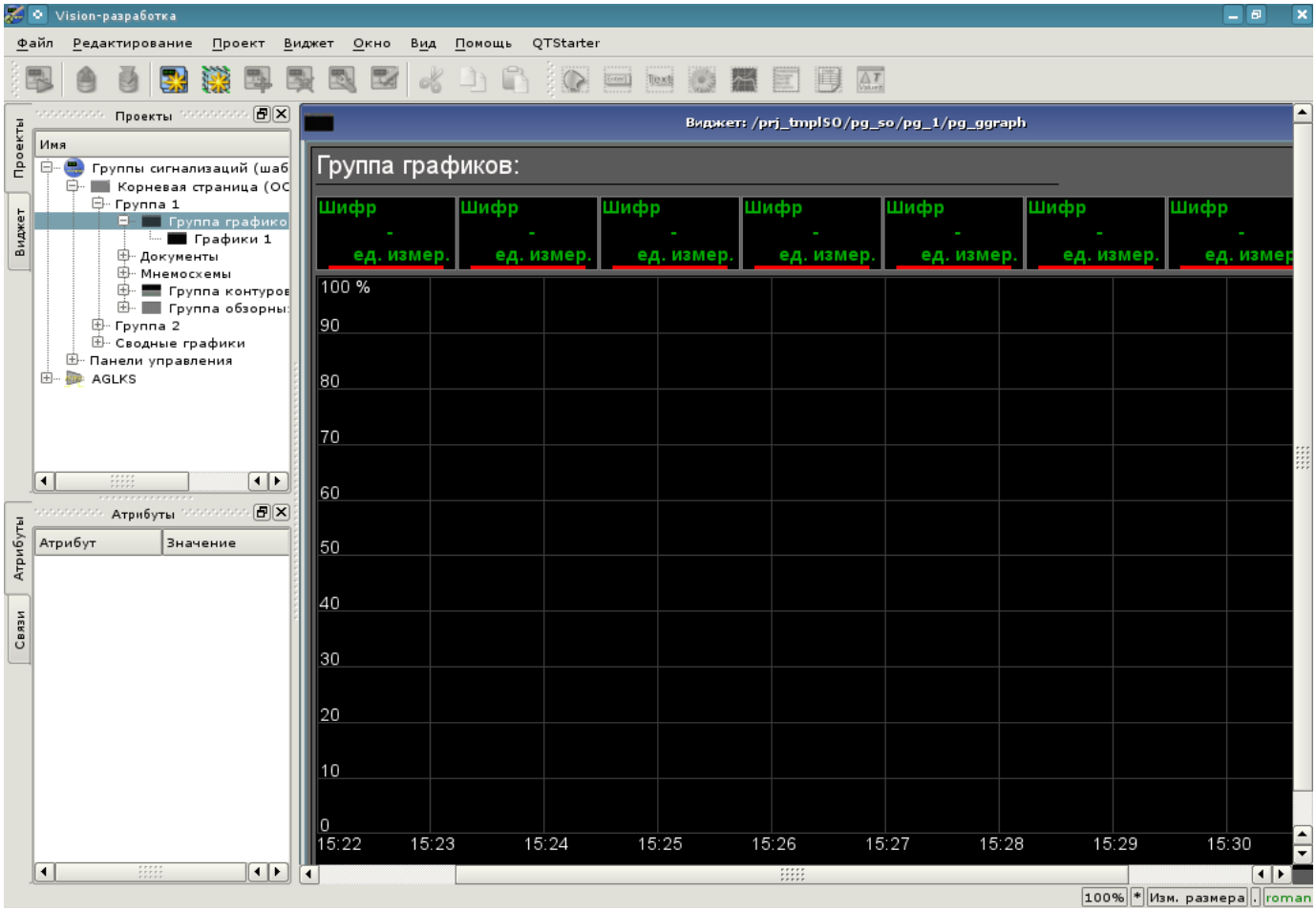

Рис. 5.1.1. Шаблонная страница "Группа графиков".

Шаблонная страница "Группа графиков" предоставляет возможность подключить до восьми сигналов для одновременного их отображения на графике. Элементы отображения значения сверху автоматически скрываются для неустановленных ссылок.

Создадим новую группу графиков "Графики 2" в шаблонном контейнере "Группа графиков" первой группы корневой страницы проекта "Группы сигнализаций (шаблон)". Для этого в контекстном меню пункта "Группа графиков" выберем "Добавить визуальный элемент" (рис.5.1.2). Для ввода идентификатора и имени нового визуального элемента появится диалог для их ввода (рис.5.1.3). Введём идентификатор "2" и имя "Графики 2".

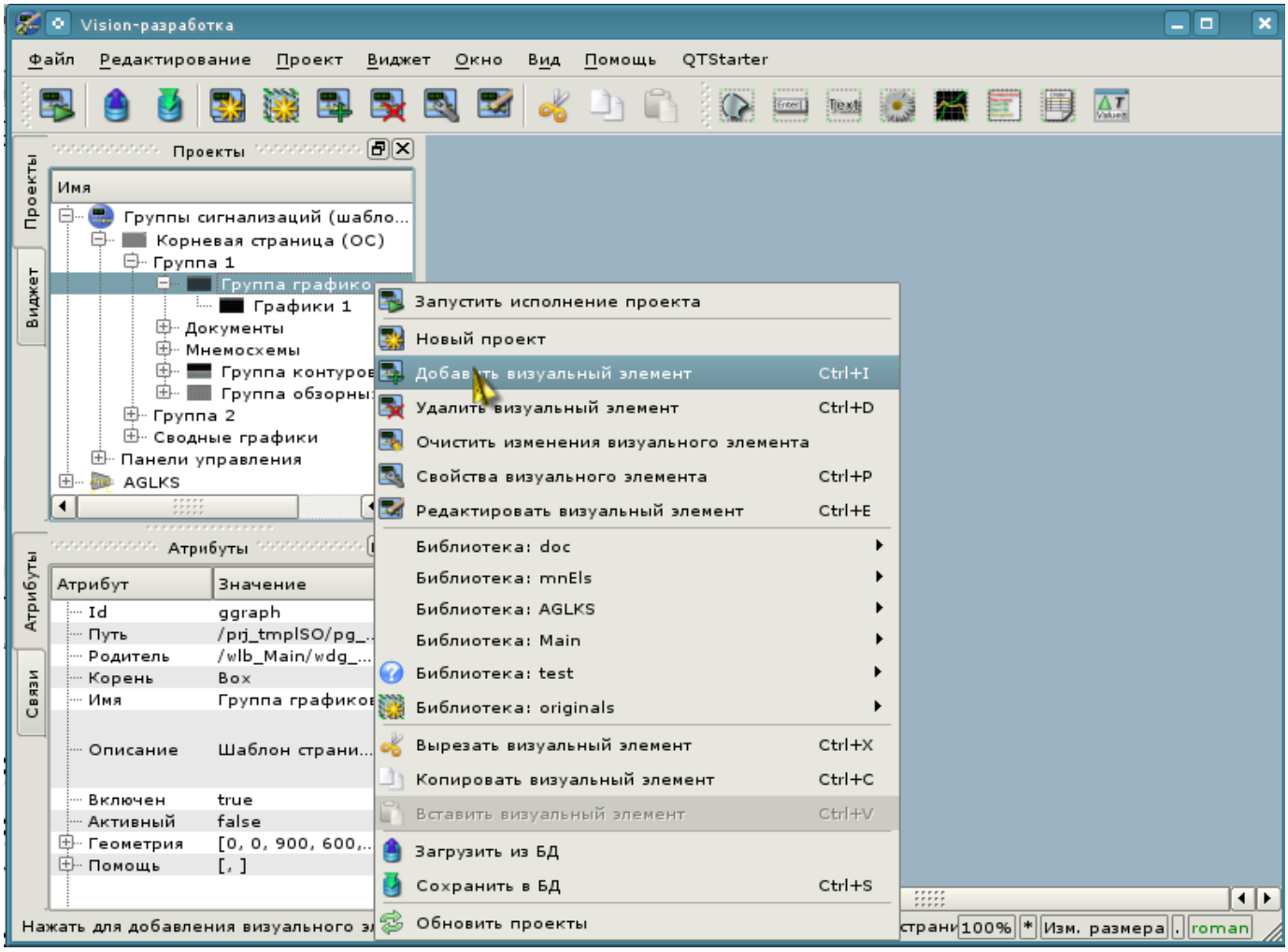

Рис. 5.1.2. Добавление группы графиков "Графики 2".

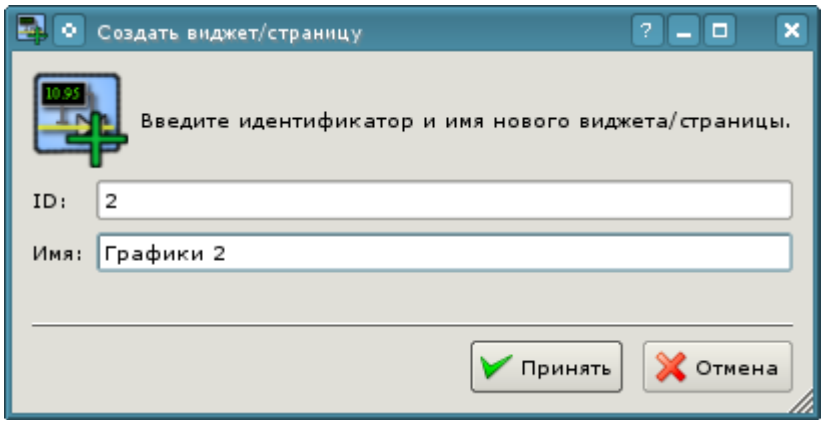

Рис. 5.1.3. Диалог ввода идентификатора и имени.

После подтверждения ввода имени будет создана новая страница. Однако, для её активации нам понадобится её включить. Включить страницу можно в диалоге редактирования свойств страницы (рис.5.1.4). Открыть эту страницу можно посредством выбора пункта меню "Свойства визуального элемента" в контекстном меню вновь созданной страницы.

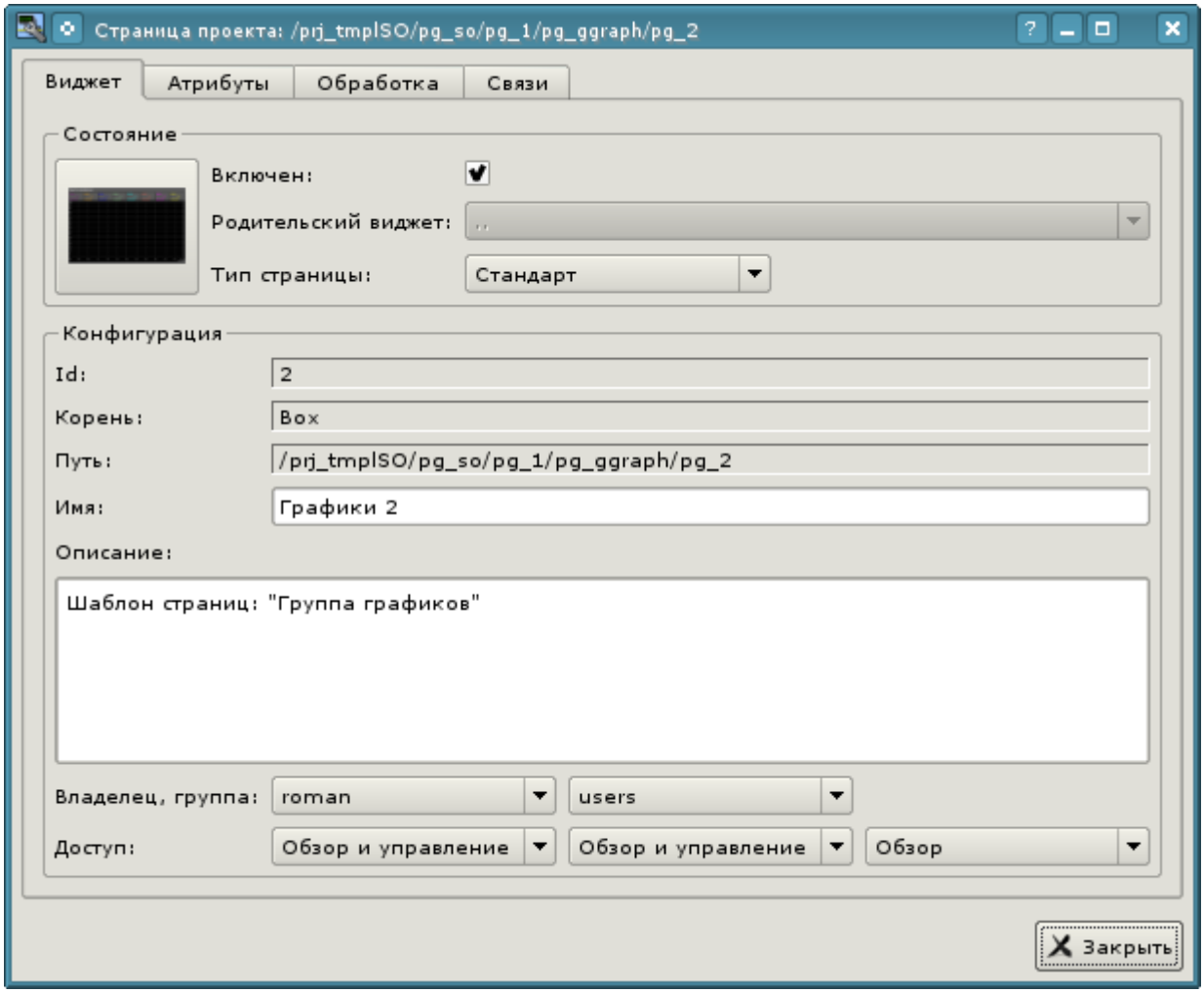

Рис. 5.1.4. Диалог редактирования свойств визуального элемента.

После включения страницы можно приступать к установке связей на созданные в предыдущей главе параметры контроллеров. Для этого, не покидая диалога редактирования свойств вновь созданной страницы (рис.5.1.4), перейдём на вкладку "Связи" (рис.5.1.5). На этой вкладке мы увидим дерево с элементами "el1" ... "el8". Развернув любой из элементов мы увидим ветку "Parameter", вот в ней мы и должны указать или выбрать адрес наших атрибутов "Ti" и "To". Итого заполним четыре элемента. При заполнении элементов часть свойств нужно указывать как постоянные. Например, обязательно нужно указать:

- *name* - "val:AT101_1 Ti"
- *ed* - "val:град.C"
- *max* - "val:150" (для Ti) и "val:100" (для To)
- *min* - "val:0"

Если заранее предусмотреть наличие атрибутов, указанных постоянными в шаблоне параметра контроллера, то можно будет указывать только параметр, а атрибуты расставятся автоматически.

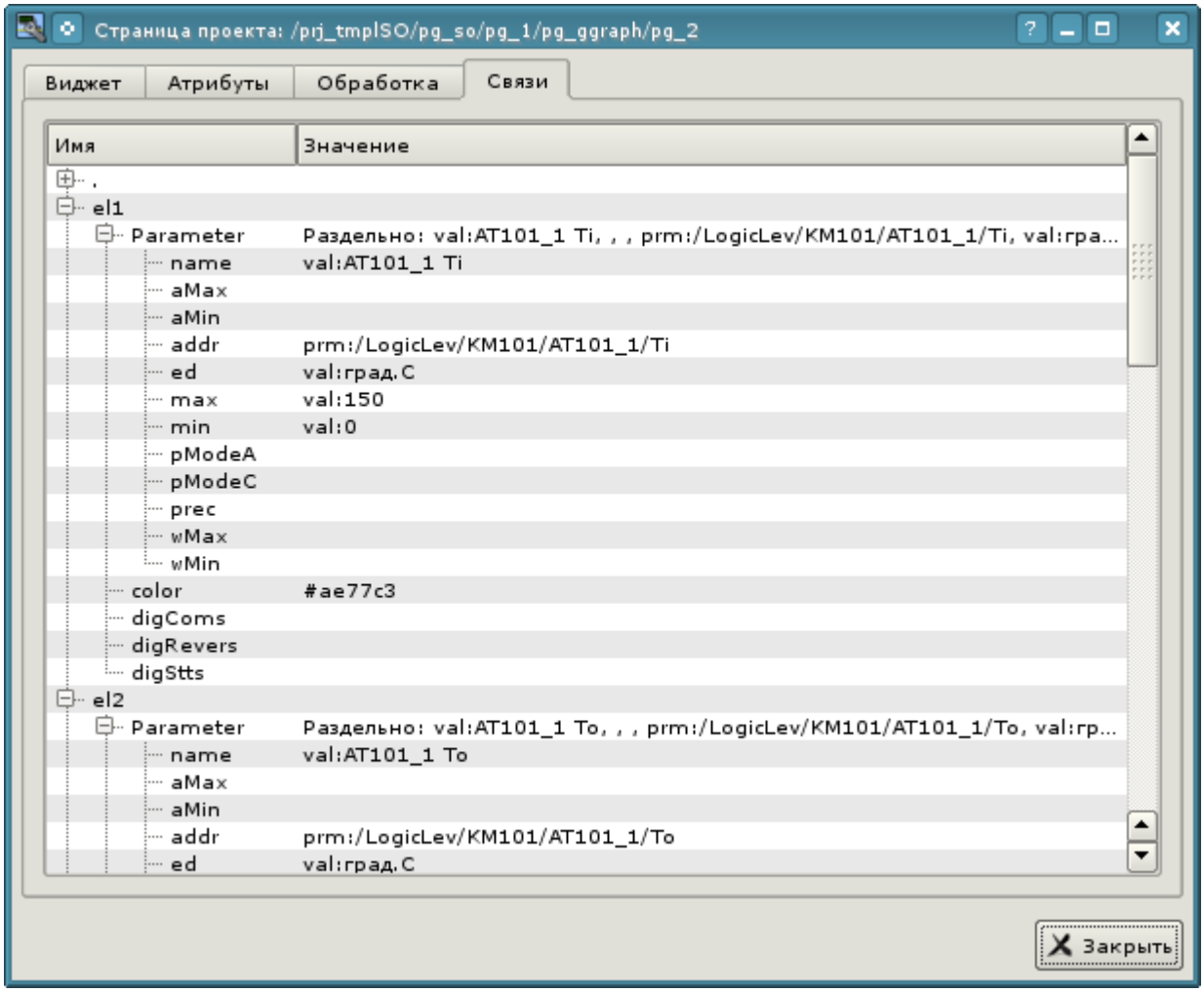

Рис. 5.1.5. Вкладка "Связи" диалога редактирования свойств визуального элемента.

Закончив ввод связей, можем проверить, что получилось в результате наших усилий. Для этого закроем оно диалога свойств и запустим проект "Группы сигнализаций (шаблон)" на исполнение, про кнопку запуска мы помним из первых глав. Затем выберем графики и переключимся на вторую страницу. При безошибочной конфигурации мы должны увидеть что-то подобное изображённому на рис.5.1.6.

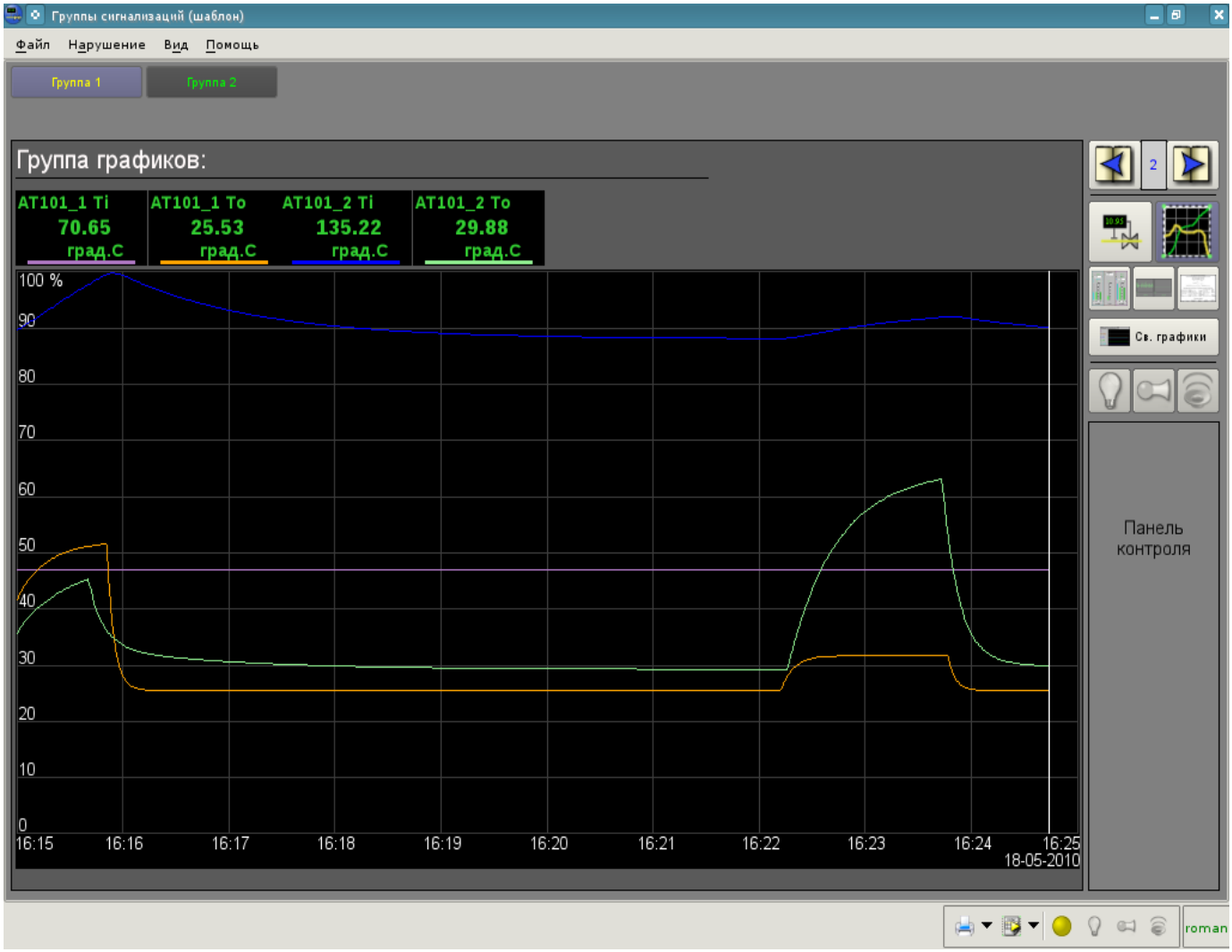

Рис. 5.1.6. Созданная группа графиков с четырьмя подключенными сигналами.

5.2. Создание нового кадра, мнемосхемы

Поднимем планку и создадим новый кадр, куда поместим базовые элементы отображения значений параметров наших контроллеров. Такие кадры обычно называются мнемосхемами и кроме отображения динамики, и даже в первую очередь, содержат статическое изображение технологического процесса в мнемоническом представлении. Мы же не будем акцентировать внимание на создание статики, а добавим элементы динамики и подключим к ней параметры наших контроллеров. Ну и поместим созданный кадр в дерево уже известного нам проекта.

Новые кадры, предназначенные в последствии для помещения в проект, принято создавать в библиотеке виджетов. Создадим новую библиотеку виджетов "KM101"; выбрав вертикальную вкладку "Виджет" и в контекстном меню окна библиотек виджетов выберем пункт "Новая библиотека" (рис.5.2.1). В диалоге ввода имени укажем идентификатор "KM101" и имя "KM 101", а затем подтвердим.

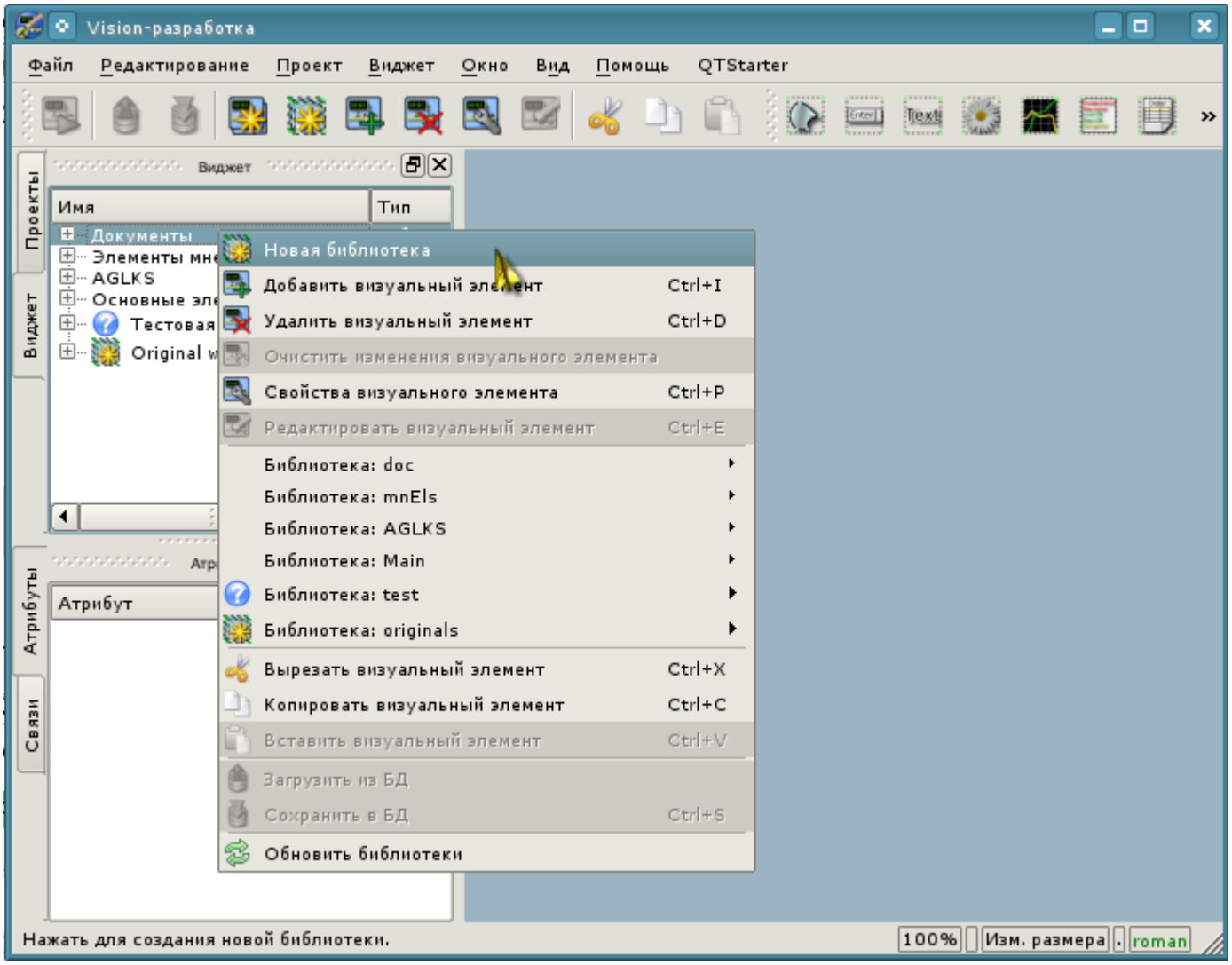

Рис. 5.2.1. Добавление новой библиотеки виджетов.

Далее добавляем новый кадр "AT101", выбрав пункт "Библиотека: originals"->"Группа элементов" в контекстном меню созданной библиотеки "KM101" (рис.5.2.2). В диалоге ввода имени укажем идентификатор "AT101" и имя "AT 101", а затем подтвердим. В основе любого кадра и страницы должен лежать элемент "Группа элементов" ("Box"), поэтому мы его и выбрали.

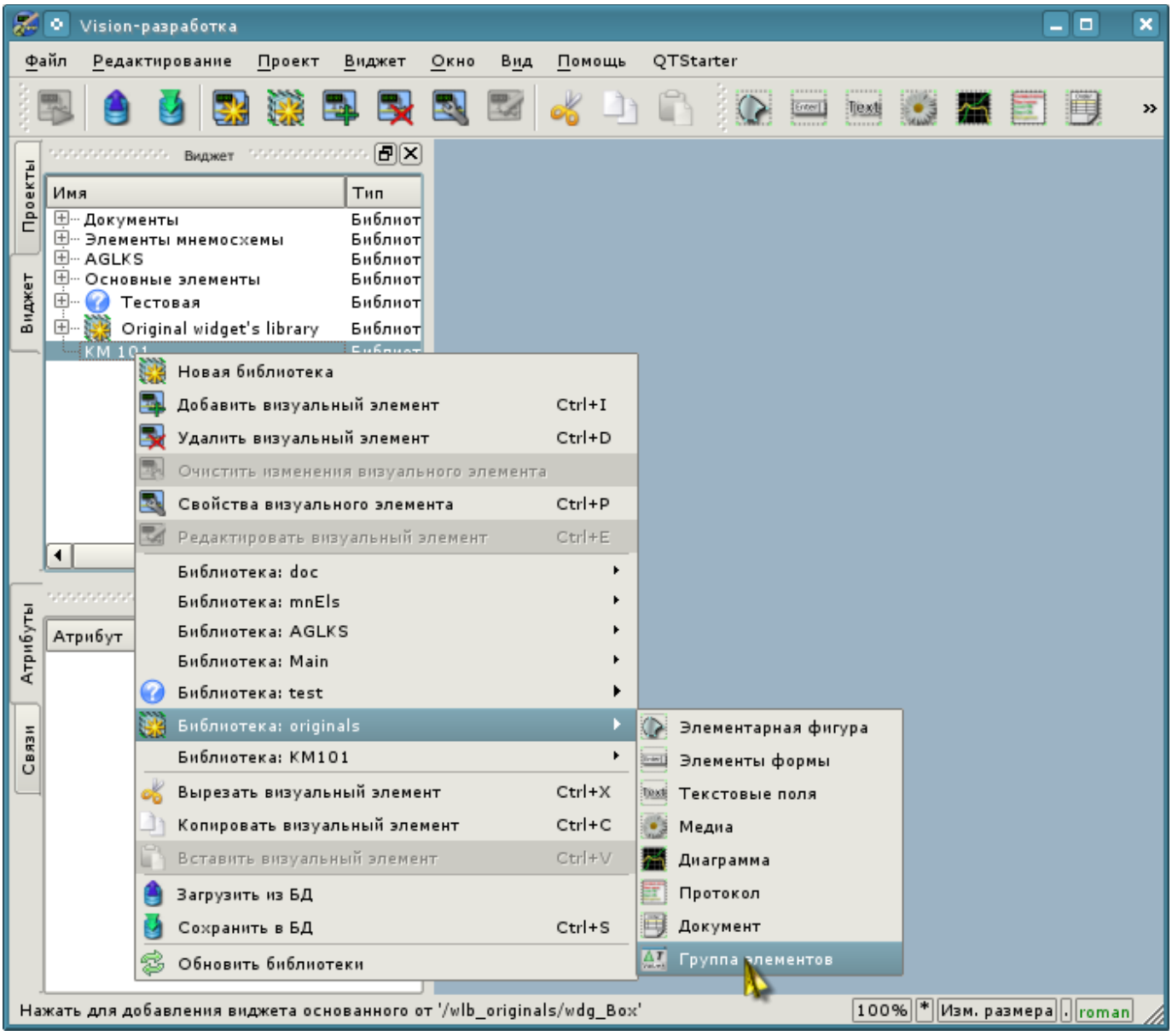

Рис. 5.2.2. Добавление нового кадра.

Сразу после создания элемента нового кадра нужно установить его базовые свойства, характерные для кадра мнемосхемы. Свойства или атрибуты любого визуального элемента можно указать в панели инструментов "Атрибуты", предварительно выбрав нужный визуальный элемент. Выберем созданный кадр "AT 101" и установим следующие свойства:

- *Геометрия:ширина* - 900;
- *Геометрия:высота* - 600;
- *Фон:цвет* - "#5A5A5A";
- *Граница:ширина* - 1;
- *Граница:цвет* - "black".

В результате получим пустой кадр (рис.5.2.3), готовый для добавления элементов на него. Для редактирования или просмотра вида кадра необходимо в контекстном меню кадра выбрать пункт "Редактировать визуальный элемент".

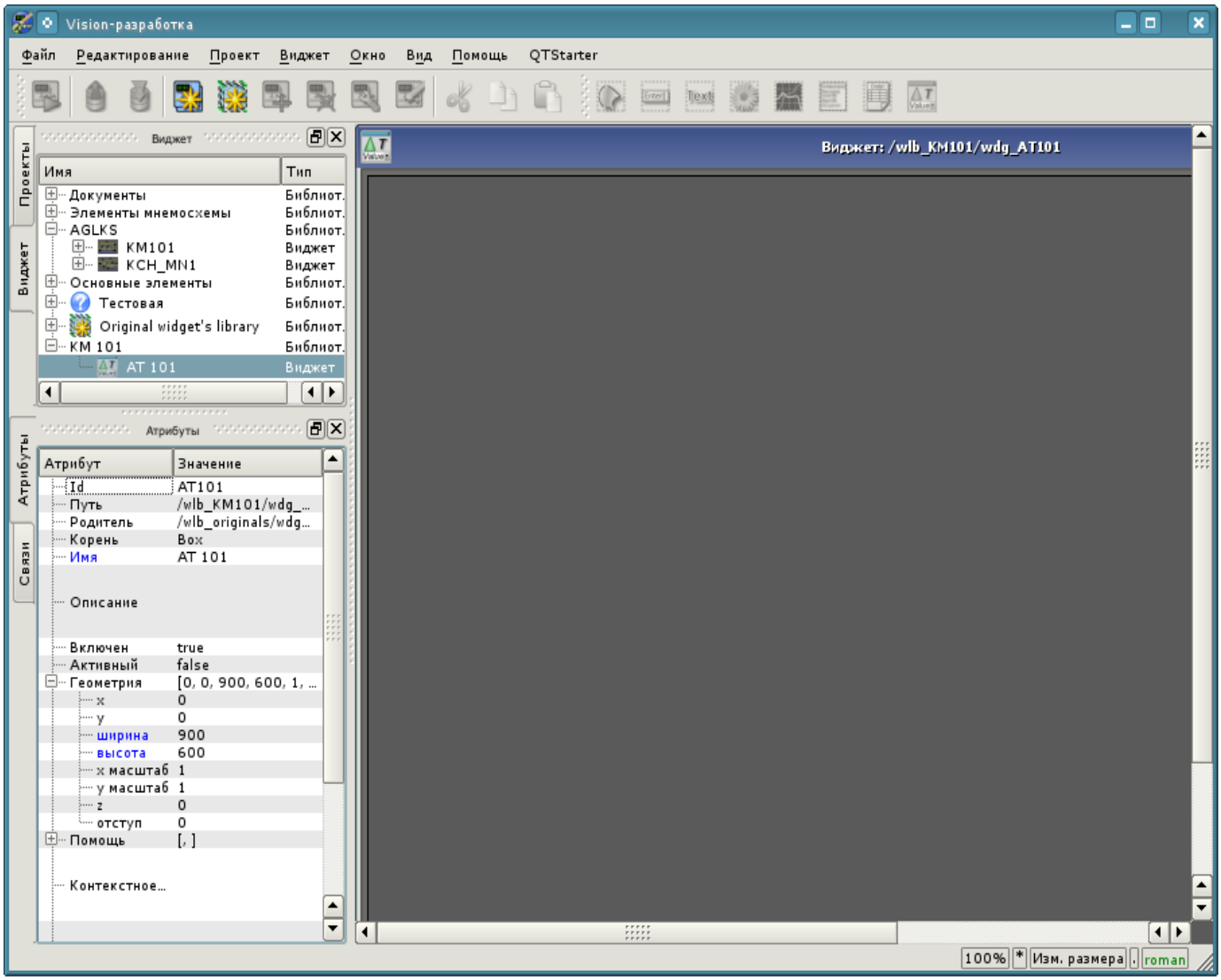

Рис. 5.2.3. Вид нового кадра и установленных атрибутов для мнемосхемы.

Теперь добавим на кадр элементы отображения значения аналогового параметра для наших четырёх сигналов. Для помещения на мнемосхему элемента отображения аналогового сигнала нужно выбрать нашу мнемосхему, а затем в меню окна выбрать пункт меню "Виджет"->"Библиотека: Main"->"Отобр аналог"; после чего появится курсор с образом этого элемента, который нужно подвести в желаемую область мнемосхемы и нажать левую кнопку мыши. В момент добавления появится диалог с запросом имени нового элемента. Добавлять подобным образом будем четыре элемента, которые назовём: "A1_Ti", "A1_To", "A2_Ti" и "A2_To". Добавленные элементы можно в последствии расположить как нужно, просто выделяя и перетаскивая мышью. После выполнения подобных манипуляций у нас должна получиться мнемосхема с видом, похожим представленной на рис.5.2.4.

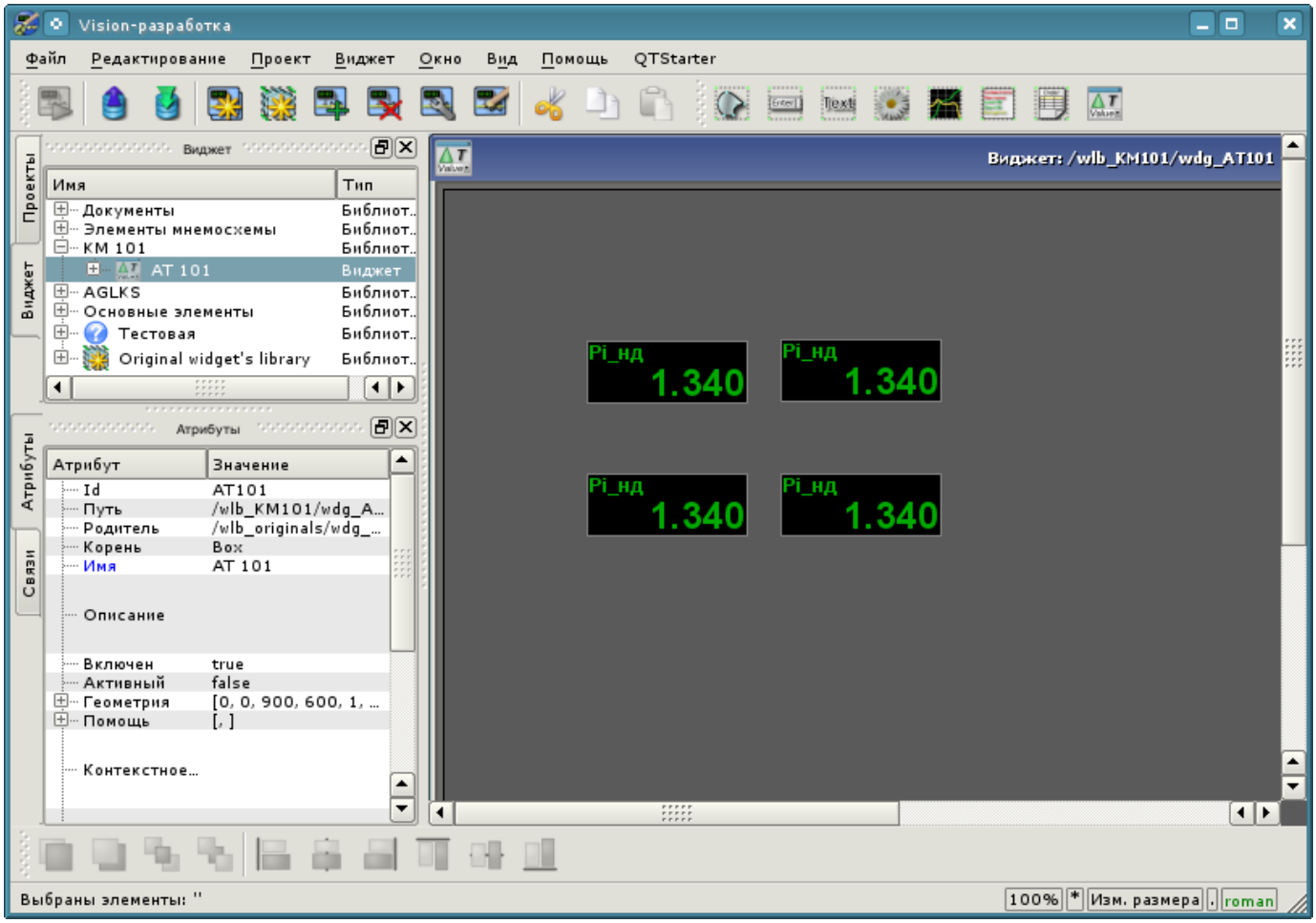

Рис. 5.2.4. Вид нового кадра и установленных атрибутов для мнемосхемы.

На этом процедуру создания мнемосхемы будем считать законченной. Сохраним новую библиотеку виджетов "KM101" и приступим к этапу размещения нашей мнемосхемы в дереве проекта "Группы сигнализаций (шаблон)".

Поместим нашу мнемосхему в ветвь "Группы сигнализаций (шаблон)"->"Корневая страница"->"Группа 1"->"Мнемосхемы" путём выбора в контекстном меню для пункта "Мнемосхемы" пункта "Библиотека: KM101"->"AT 101". Идентификатор для новой мнемосхемы установим в "2", при этом поле имени оставим пустым. Сразу после добавления нужно установить базовое свойство мнемосхемы "Страница: группа" в значение "so".

Далее нужно произвести уже знакомую нам операцию по предыдущей главе, а именно установку связей на созданные в предыдущей главе параметры контроллеров. Для этого откроем диалог редактирования свойств мнемосхемы на вкладке "Связи" (рис.5.2.5). На этой вкладке мы увидим дерево с элементами "A1_Ti", "A1_To", "A2_Ti" и "A2_To". Развернув любой из элементов, мы увидим ветку "Parameter", вот в ней мы и должны указать или выбрать адрес значений наших атрибутов "Ti" и "To" соответственно. При заполнении элементов часть свойств нужно указывать как постоянные. Например, обязательно нужно указать:

- *pName* - "val:AT101_1 Ti"

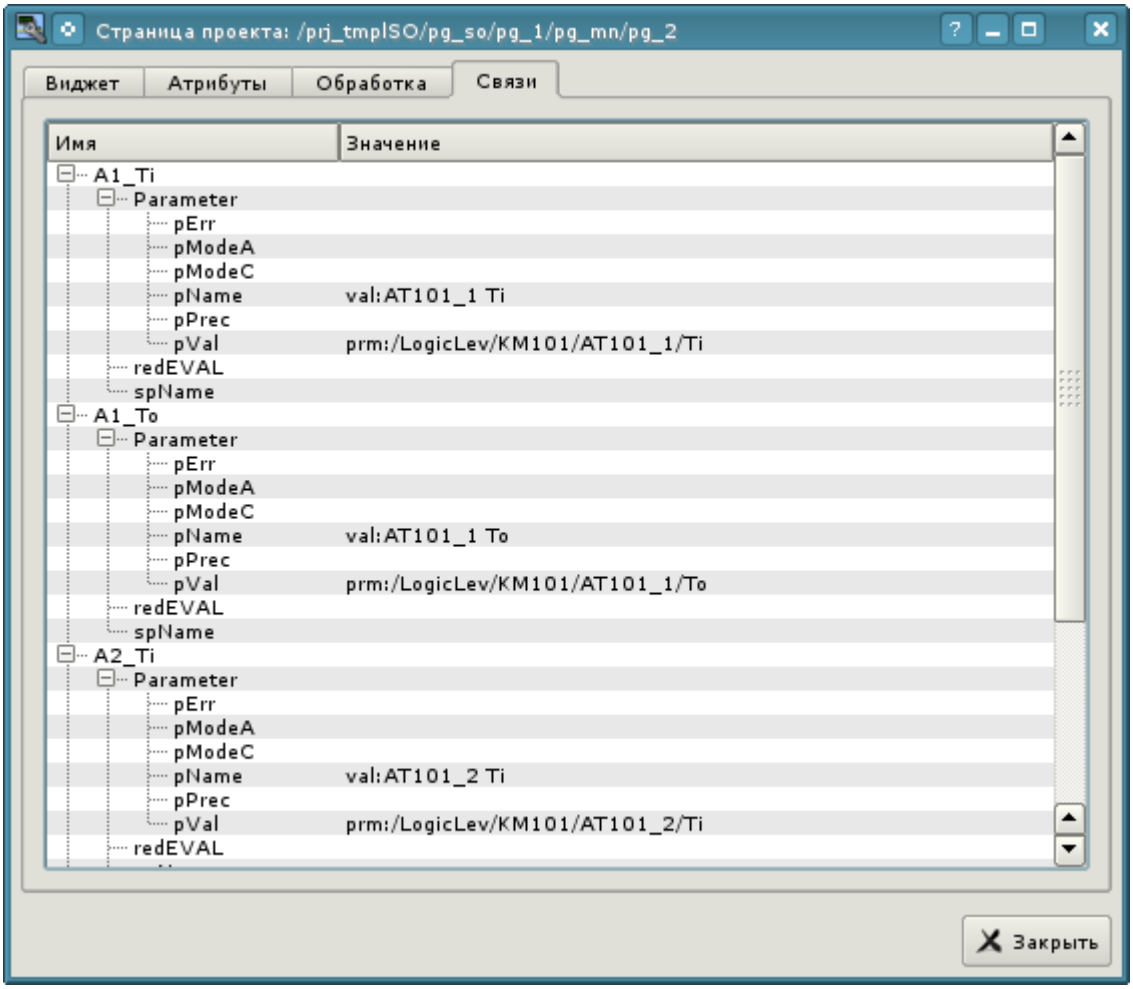

Рис. 5.2.5. Вкладка "Связи" диалога редактирования свойств мнемосхемы.

Теперь можем сохранить нашу мнемосхему и проверить, что получилось. Для этого закроем окно диалога свойств и запустим проект "Группы сигнализаций (шаблон)" на исполнение. Затем переключимся на вторую мнемосхему кнопками листания. При безошибочной конфигурации мы должны увидеть что-то подобное изображённому на рис.5.2.6.

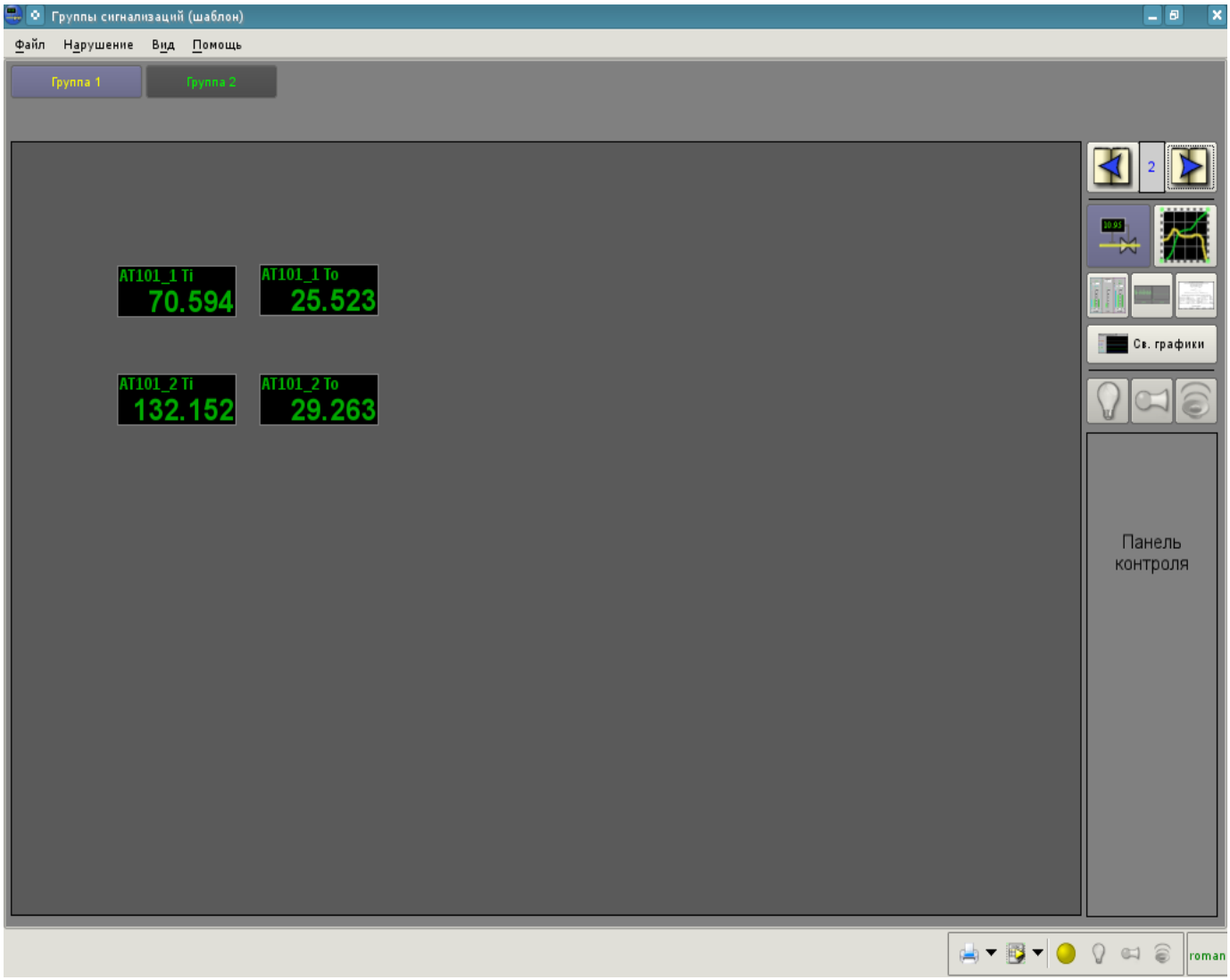

Рис. 5.2.6. Созданная мнемосхема с четырьмя подключенными сигналами.

5.3. Создание нового комплексного элемента

Приступим к рассмотрению задач третьего уровня сложности, а именно к созданию комплексного элемента. Создание нового комплексного элемента, включающего в себя комбинацию различных базовых примитивов, может осуществляться в несколько этапов. В качестве примера рассмотрим задачу, состоящую из двух этапов:

- Создание виджета "Воздушный холодильник" на основе примитива "Элементарная фигура".
- Создание финального скомпонованного виджета "Холодильник" на основе примитива "Группа элементов".

5.3.1. Создание виджета "Воздушный холодильник" на основе примитива "Элементарная фигура".

Виджет будем создавать в ранее нами созданной библиотеке "KM101". Для этого кликаем правой кнопкой манипулятора "мышь" по пункту этой библиотеке и выбираем пункт "Библиотека: originals"->"Элементарная фигура", как это показано на рисунке 5.3.1.1. Для нового элемента указываем идентификатор "air_cooler" и имя "Воздушный холодильник".

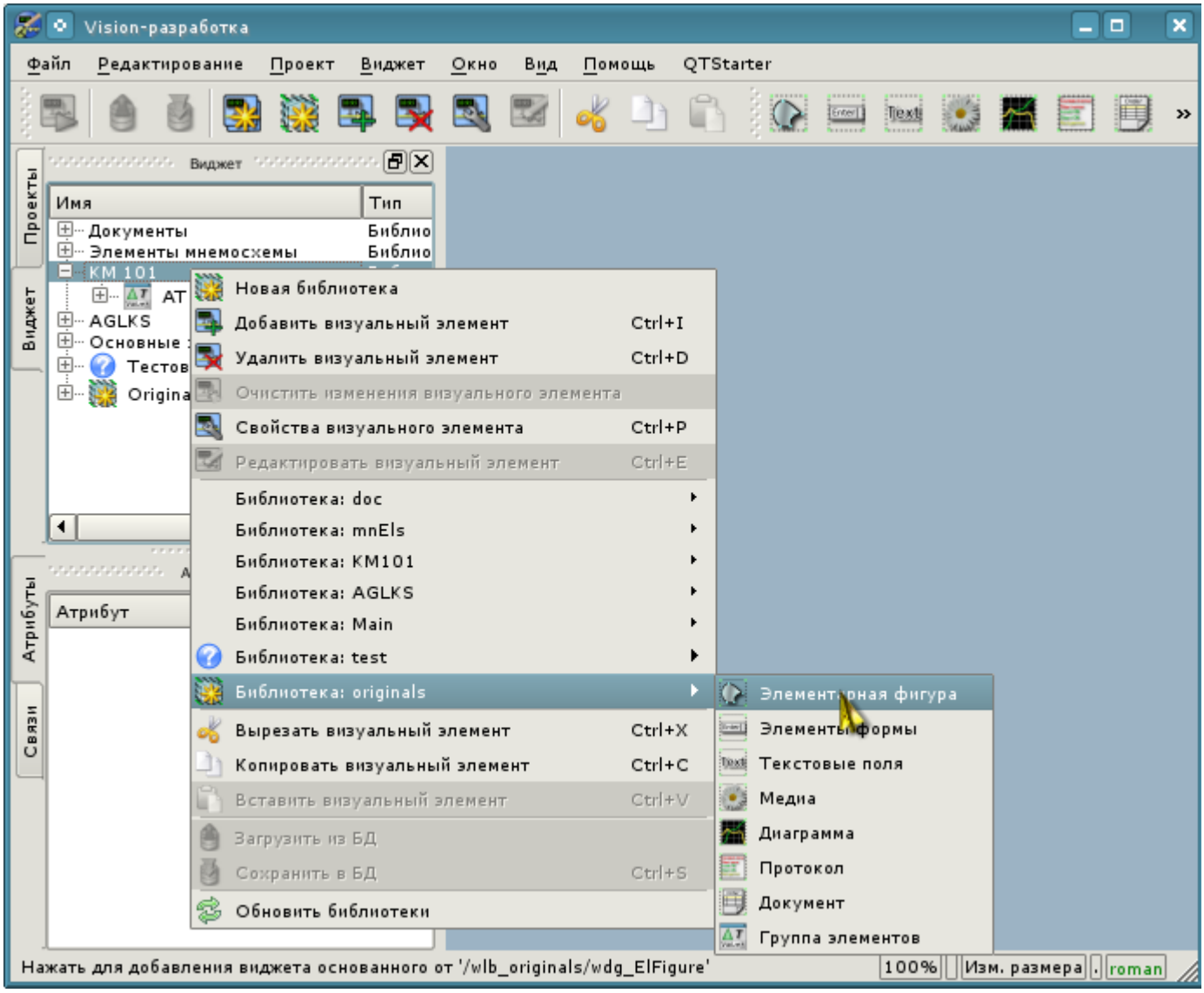

Рис. 5.3.1.1. Добавление виджета на основе примитива "Элементарная фигура" в библиотеку "KM101".

После подтверждения у нас появится объект нового виджета с именем "Воздушный холодильник". Выберем его в списке виджетов библиотеки "KM101" и откроем для редактирования посредством контекстного меню нового элемента. Зададим теперь во вкладке "Атрибуты" в пункте "Геометрия" ширину и высоту виджета в 200 пикселов (рис. 5.3.1.2).

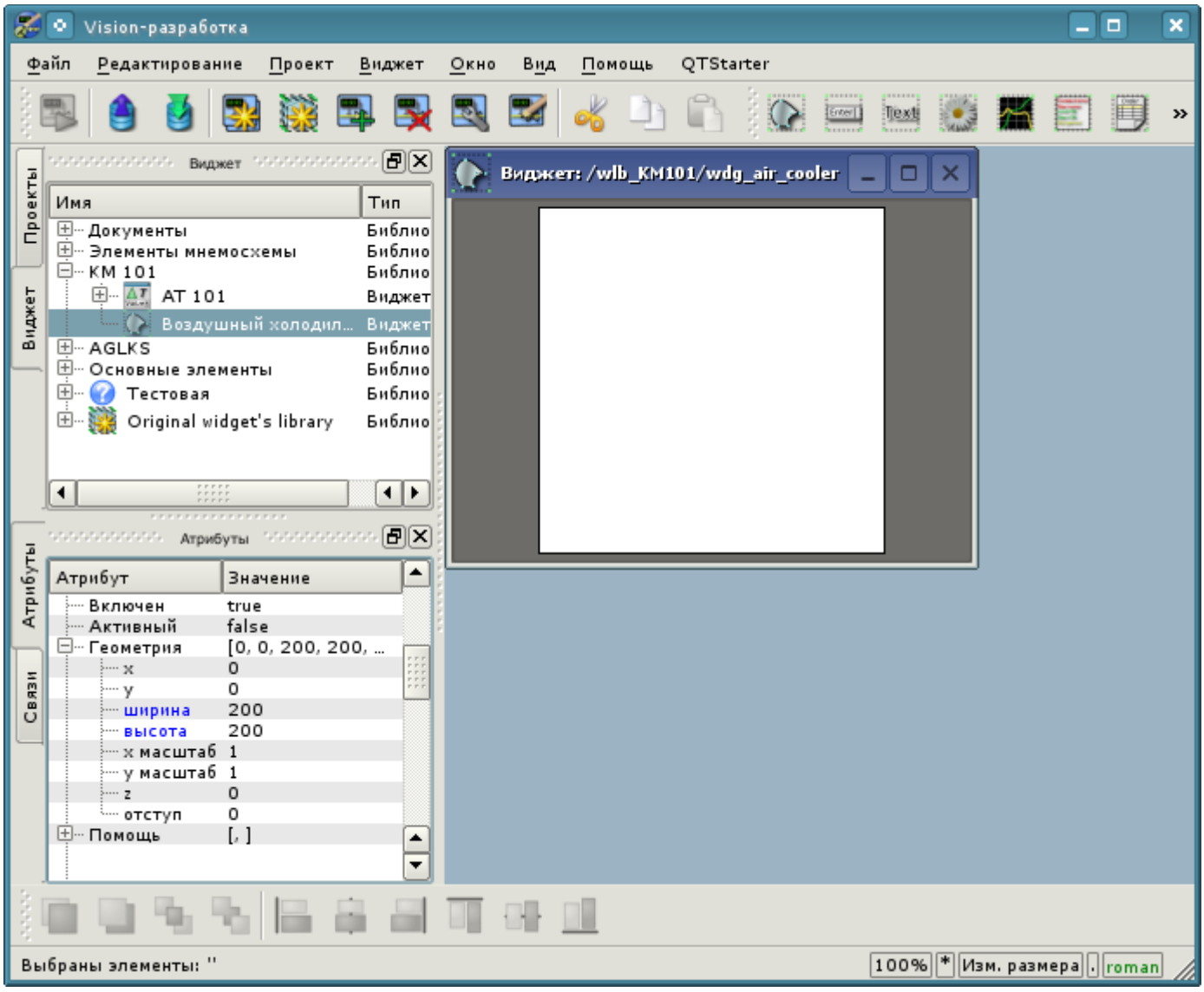

Рис. 5.3.1.2. Задание геометрических размеров виджета.

Теперь нарисуем визуальное представление виджета. Эту процедуру можно проделать двумя нижеописанными способами:

- Нарисовать желаемое изображение манипулятором "мышь", 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 рис. 5.3.1.3, либо двойным нажатием левой кнопки манипулятора "мышь" на теле виджета.
- Вручную заполнить поле "Список элементов", введя перечень необходимых элементов и координат точек.

Дополнительную информацию о редакторе можно прочитать здесь: http://wiki.oscada.org/Doc/Vision/ElFigure

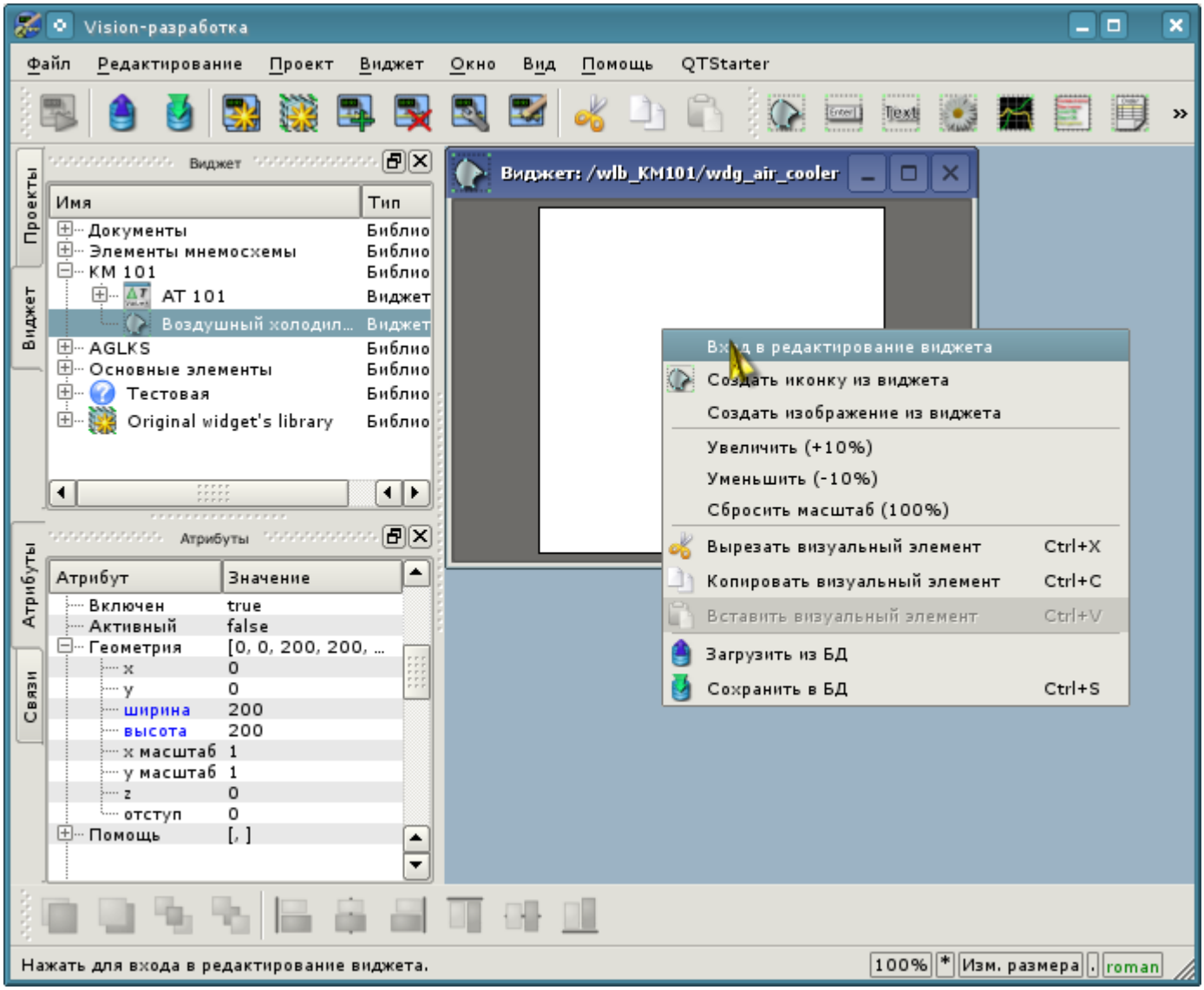

Рис. 5.3.1.3. Вход в режим рисования виджета, основанного на примитиве "Элементарная фигура".

В нашем примере мы воспользуемся вторым способом. Для этого в поле "Список элементов" инспектора атрибутов введем нижеприведенный перечень и нажмем "Ctrl"+"Enter".

```
line:(20|80):(100|20)
line:(100|20):(180|80)
line:(180|80):(100|140)
line:(100|140):(20|80)
line:(100|20):(100|140)
line:(20|80):(180|80)
line:(50|165):(100|140)
line:(100|140):(150|165)
line:(150|165):(50|165)
fill:(20|80):(100|20):(180|80):(100|140)
fill:(50|165):(100|140):(150|165)
```

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 рис. 5.3.1.4.

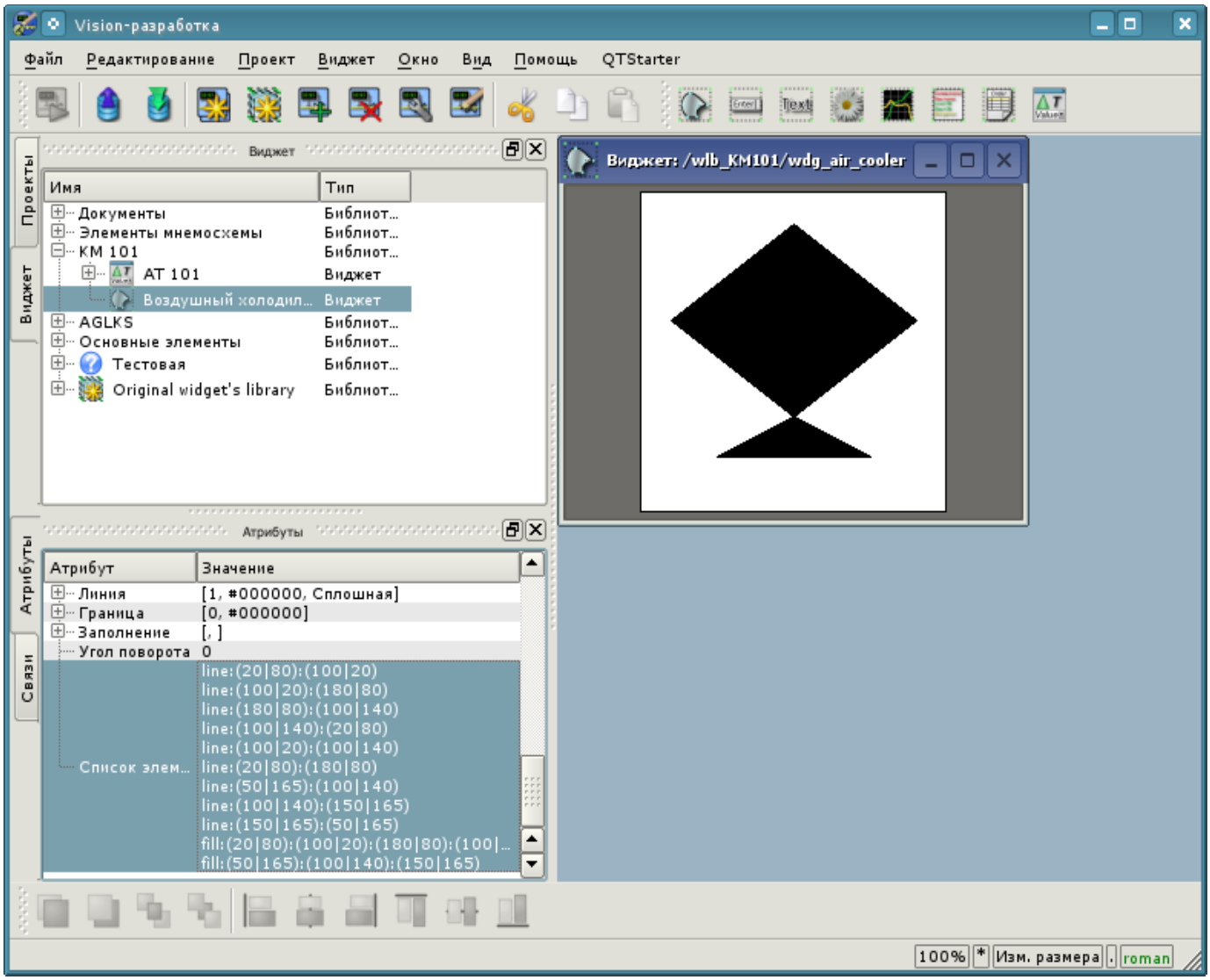

Рис. 5.3.1.4. Изображение, соответствующее "Списку элементов" виджета.

Теперь поменяем цвет заливки (черный, если не указан никакой другой (по умолчанию)) во вкладке "Атрибуты" в пункте "Заполнение" на "lightgrey" (рис. 5.3.1.5). Цвет можно задавать, как с помощью названий цветов, так и в формате #RRGGBB(#RRGGBB-AAA).

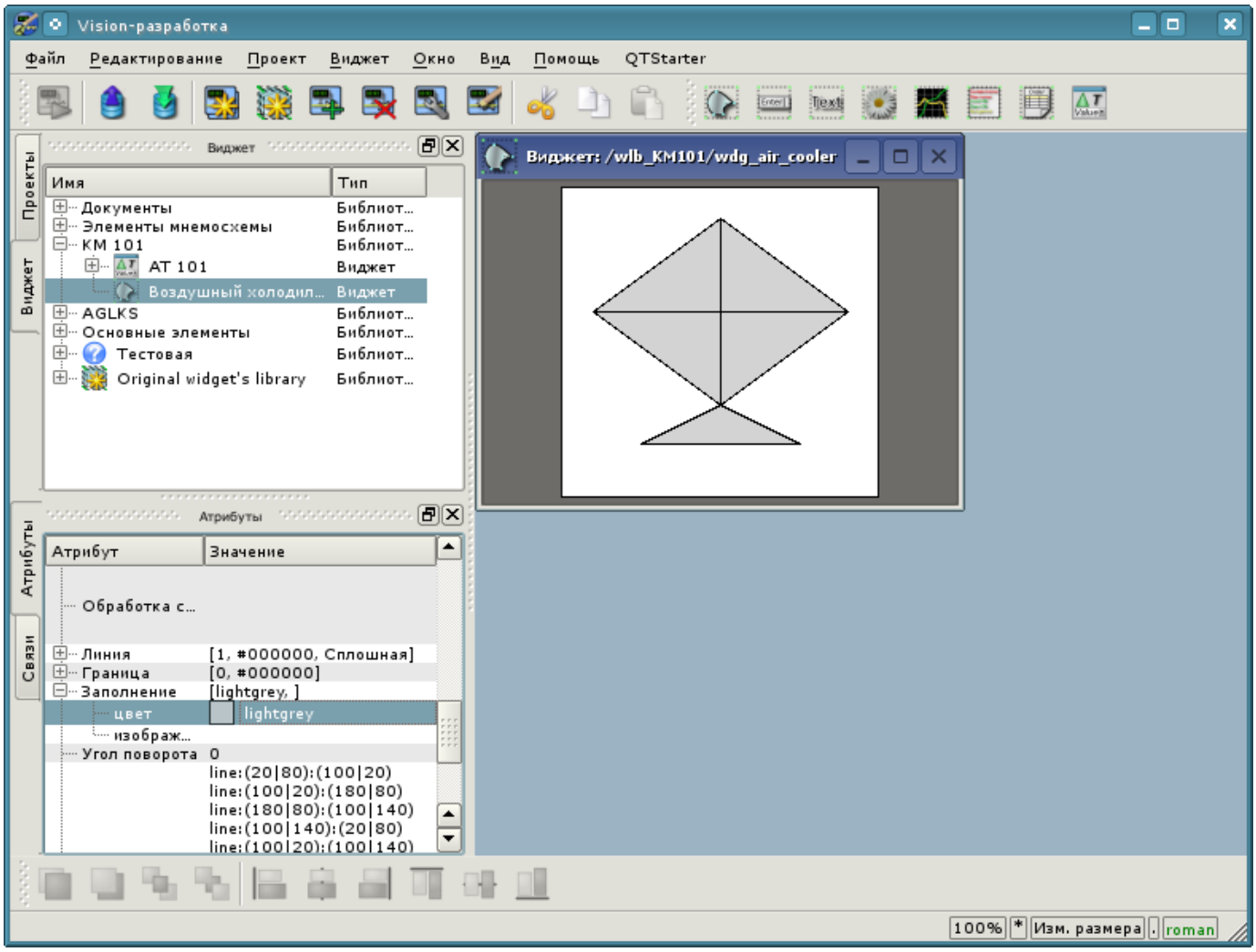

Рис. 5.3.1.5. Изменение цвета заполнения (заливки).

Создадим иконку для нашего виджета, которая будет видна в дереве виджетов библиотеки "KM101" (рис. 5.3.1.6).

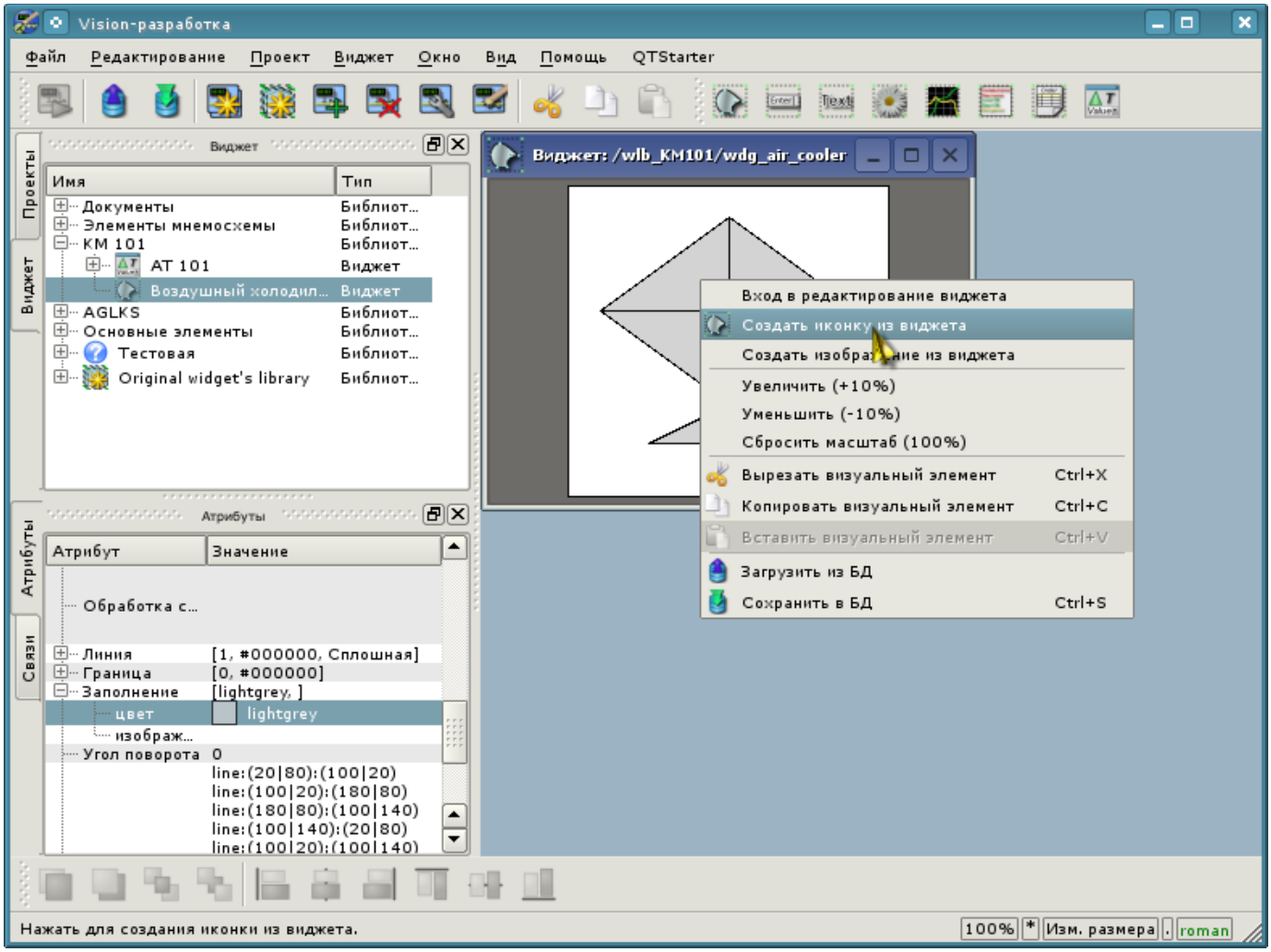

Рис. 5.3.1.6. Создание иконки для виджета.

На этом процесс создания первого виджета можно считать завершенным. Перейдем теперь к этапу компоновки и созданию результирующего виджета.

5.3.2. Создание финального скомпонованного виджета "Холодильник" на основе примитива "Группа элементов"

Результирующий виджет будем создавать в библиотеке "KM 101". Для этого кликаем правой кнопкой манипулятора "мышь" по этой библиотеке и выбираем примитив "Группа элементов", как это показано на рисунке 5.3.2.1. Для нового элемента указываем идентификатор "elCooler" и имя "Холодильник".

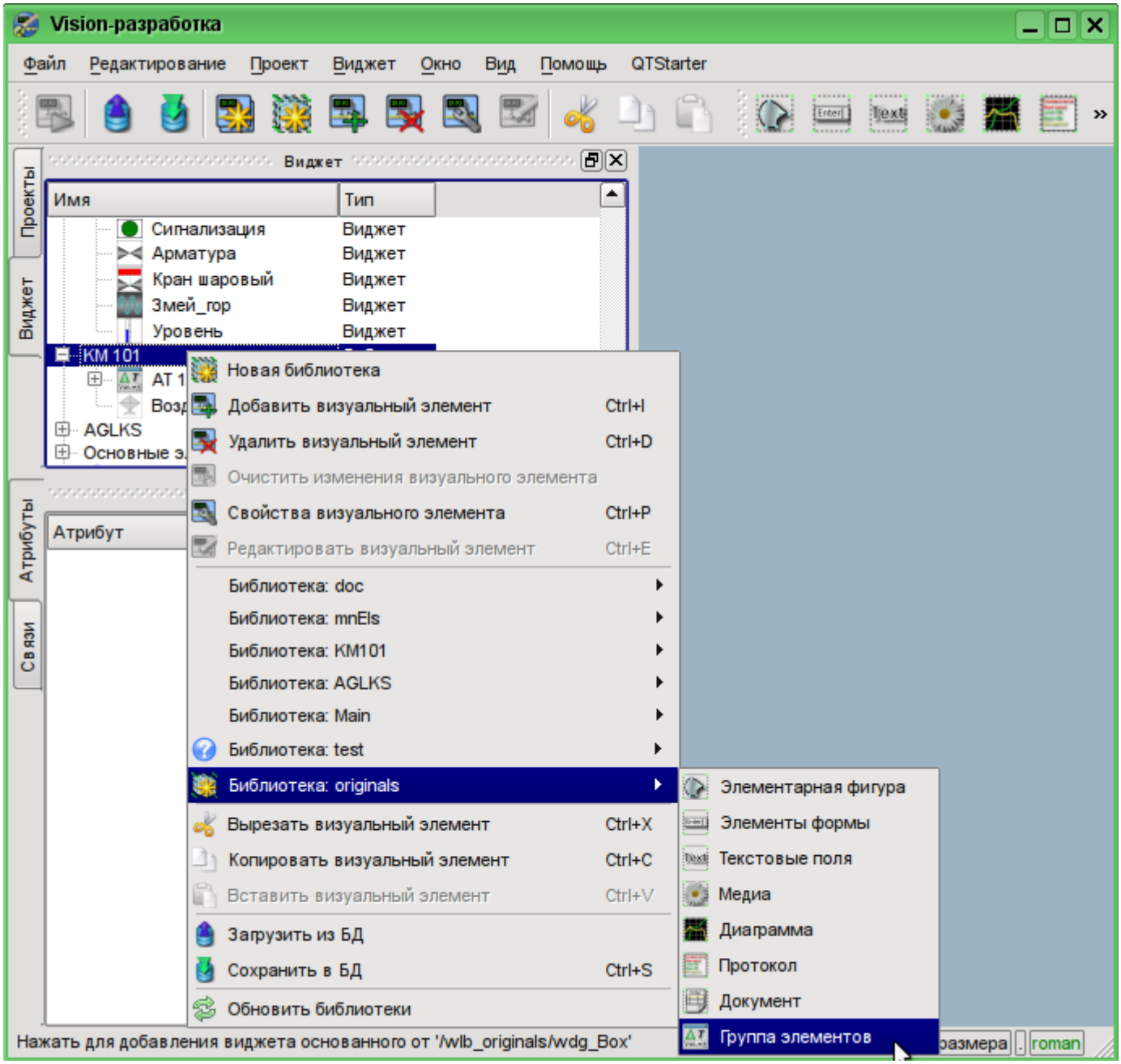

Рис. 5.3.2.1. Добавление виджета на основе примитива "Группа элементов" в библиотеку "KM 101".

После подтверждения у нас появится объект нового виджета с именем "Холодильник". Выберем его в списке виджетов библиотеки "KM 101" и открываем для редактирования. Зададим теперь во вкладке "Атрибуты" в пункте "Геометрия" ширину и высоту виджета в 250 и 200 пикселов соответственно.

Возьмём ранее созданный элемент "Воздушный холодильник" (air_cooler) и перетащим его (нажав на нем левую кнопку манипулятора "мышь", и перемещая курсор "мыши" до области виджета; после этого нужно отпустить кнопку) на вновь созданный нами виджет (рис. 5.3.2.2).

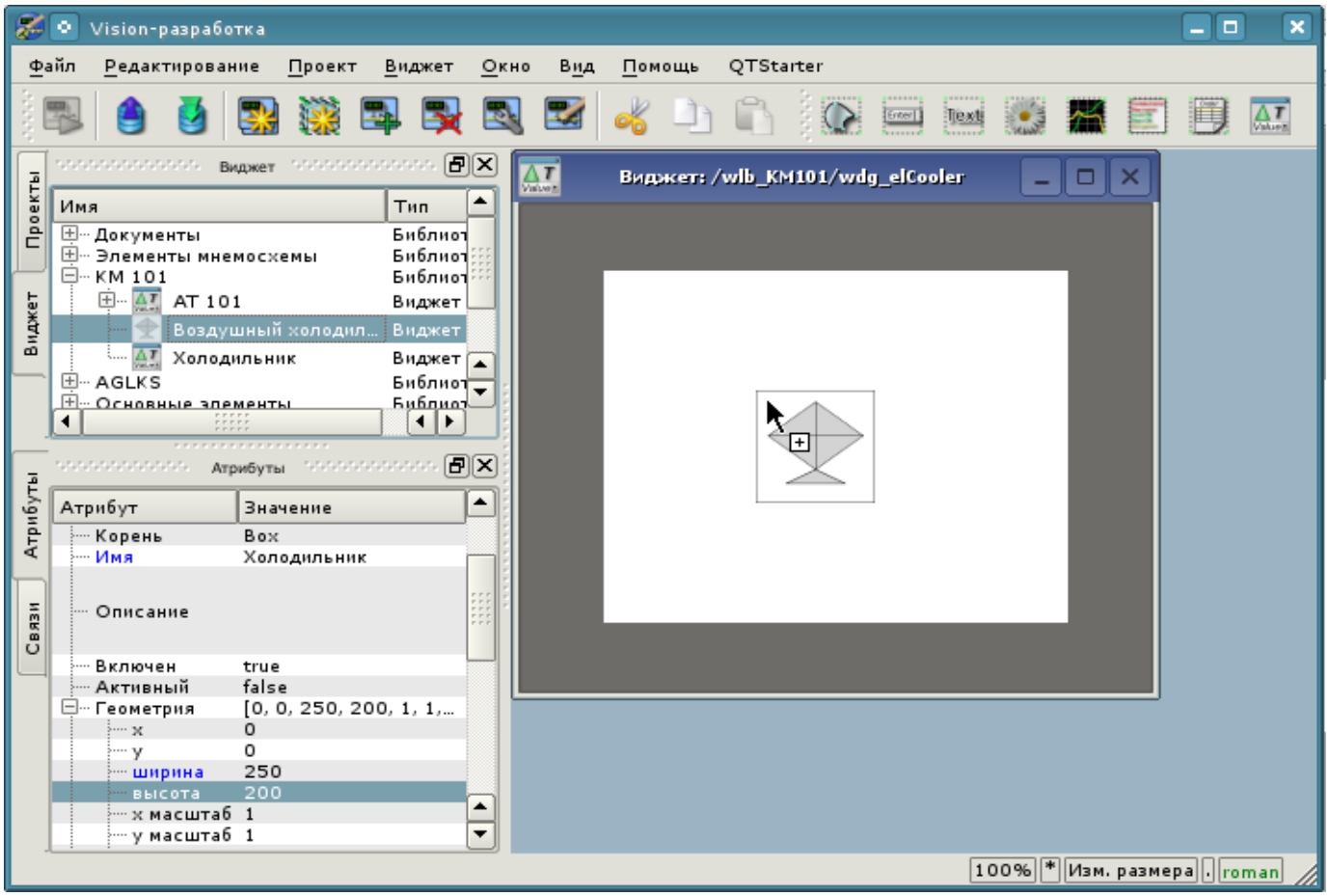

Рис. 5.3.2.2. Перетаскивание(Drag&Drop) виджета "air_cooler" в виджет-контейнер "elCooler".

В результате появится окно диалога с предложением ввести идентификатор и имя нового виджета. Идентификатор и имя могут быть заданы произвольно. Мы введём идентификатор "air_cooler", а имя оставим пустым (оно унаследуется от родителя - элемента "air_cooler"). Таким образом, вновь созданный виджет внутри контейнера "elCooler" унаследует элемент - "Воздушный холодильник" ("air_cooler"). После подтверждения ввода идентификатора и имени виджет "Воздушный холодильник" ("air_cooler") добавится в наш виджет-контейнер "elCooler" (рис. 5.3.2.3)

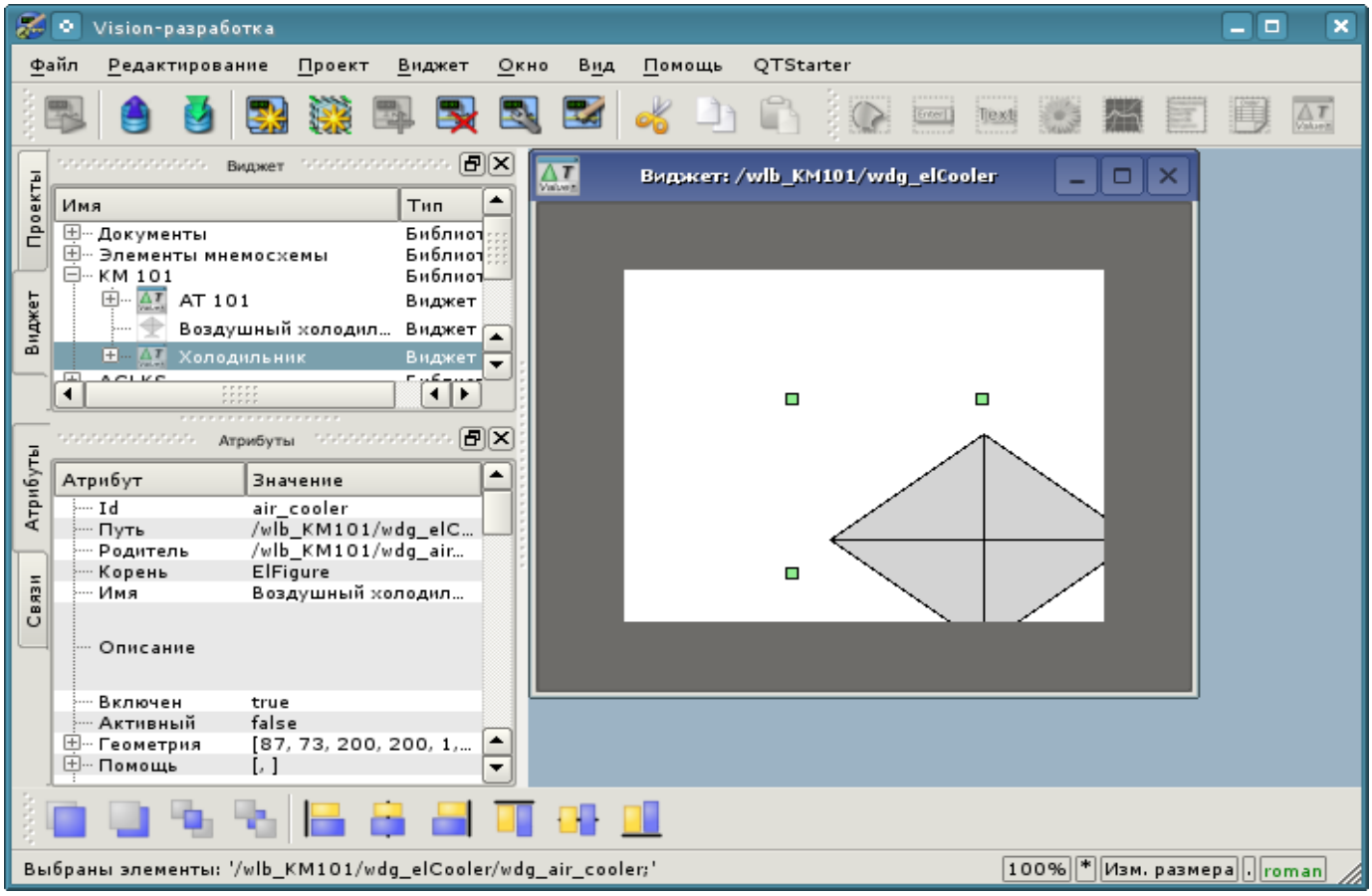

Рис. 5.3.2.3. Добавление унаследованного виджета "air_cooler".

Зададим на пенели атрибутов виджета в пункте "Геометрия" координаты "x" и "y" верхнего левого угла виджета в 25 и 0 пикселов соответственно.

Далее развернем библиотеку "Элементы мнемосхемы", найдем там элемент "Вентилятор"(cooler2) и перетащим его на виджет-контейнер. Этот элемент будет динамически отображать интенсивность работы воздушного холодильника. В результате появится окно диалога для ввода идентификатора и имени нового виджета. Введём идентификатор "cooler2", а имя снова оставим пустым. Таким образом, вновь созданный виджет внутри контейнера "elCooler" унаследует элемент библиотеки "Элементы мнемосхемы" - "Вентилятор" ("cooler2"). После подтверждения ввода идентификатора и имени виджет "Вентилятор" ("cooler2") добавится в наш виджет-контейнер "elCooler". Если необходимо, "поднимем" виджет "cooler2" наверх относительно виджета "air_cooler" внутри виджета-контейнера "elCooler" с помощью панели инструментов размещения снизу. Зададим во вкладке "Атрибуты" в пункте "Геометрия" координаты "x" и "y" верхнего левого угла виджета "Вентилятор" в 75 и 30 пикселов соответственно. Изменим у унаследованного виджета "Вентилятор" альфа канал (прозрачность) цветов заливок (заполнений). Для этого во вкладке "Атрибуты" в полях "Цвет1" и "Цвет2" изменим значения цветов, добавив к ним "-200", где 200 - значение прозрачности ("0" - полностью прозрачный, а "255" - полностью непрозрачный), как показано на рис. 5.3.2.4.

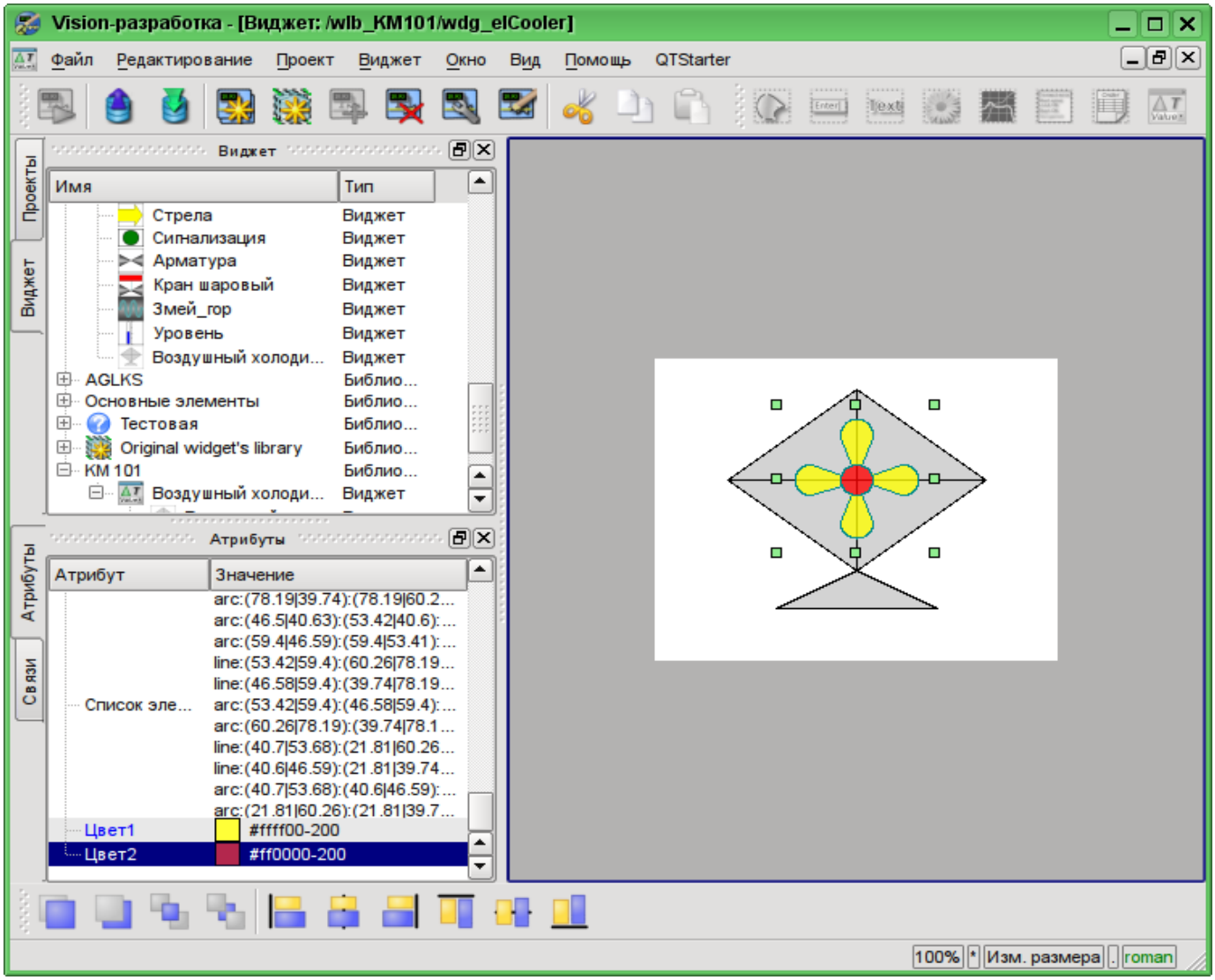

Рис. 5.3.2.4. Изменение прозрачности цветов заполнений у унаследованного виджета "cooler2".

Теперь добавим в виджет-контейнер "elCooler" два текстовых поля, основанных на примитиве "Текст", с целью отображения входной и выходной температур потока. Для этого в библиотеке "KM 101" выделим виджет "холодильник" и нажмем на панели визуальных элементов на иконку примитива "Текст", как это показано на рис 5.3.2.5.

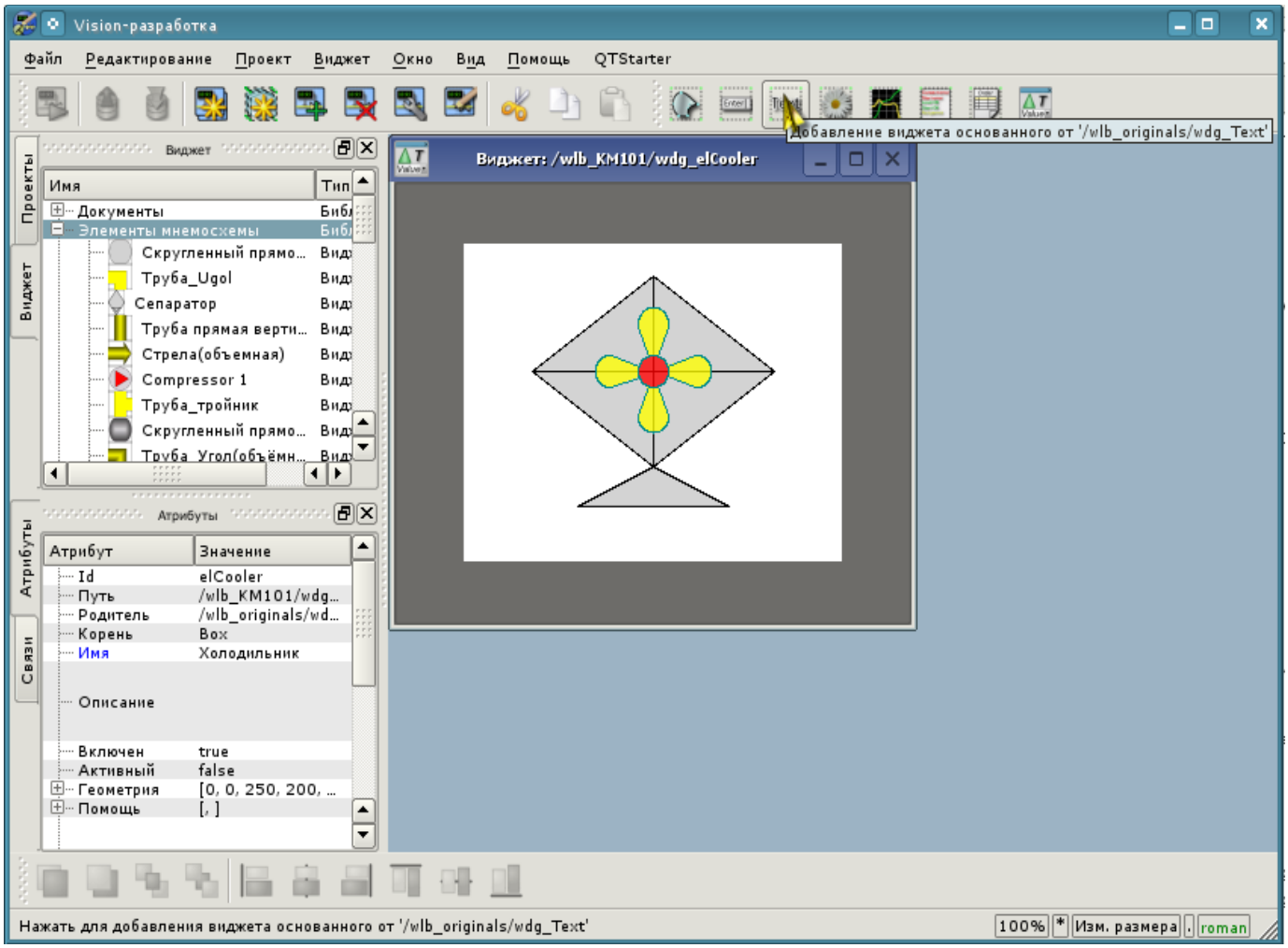

Рис. 5.3.2.5. Добавление в контейнер нового элемента, основанного на примитиве "Текст".

В результате появится диалог ввода идентификатора и имени вновь создаваемого элемента. Введем идентификатор "Ti" для первого текстового поля, а имя оставим пустым. Зададим геометрические размеры и координаты верхнего левого угла виджета, как это показано на рис. 5.3.2.6

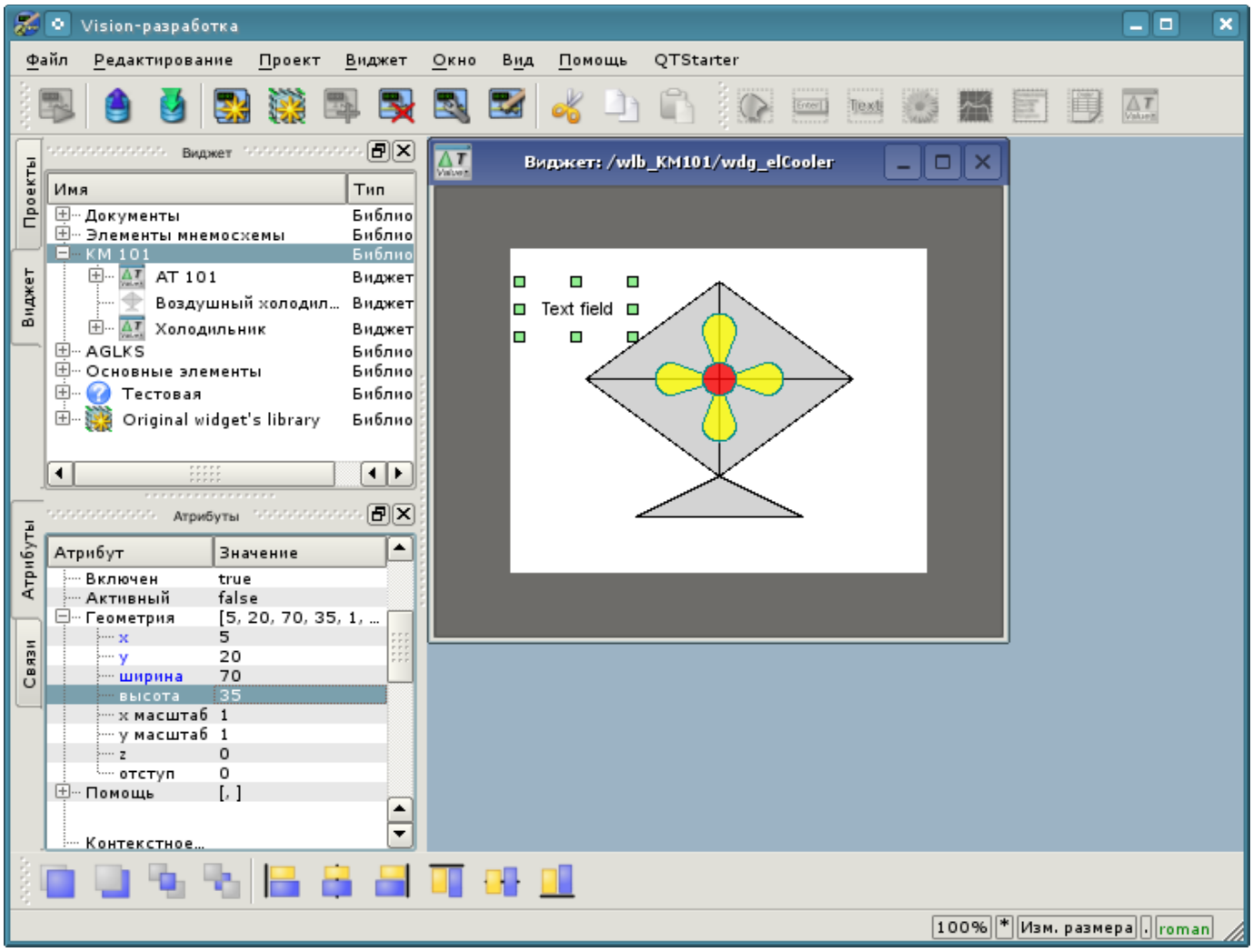

Рис. 5.3.2.6. Задание геометрии виджета "Ti".

Изменим размер и установим усиление шрифта для этого элемента (рис. 5.3.2.7). Обратим внимание, что измененные поля у унаследованных виджетов подсвечиваются синим цветом для удобства отслеживания изменений и последующей их "очистки"(отката) с помощью клика правой кнопкой манипулятора "мышь" по измененному атрибуту.

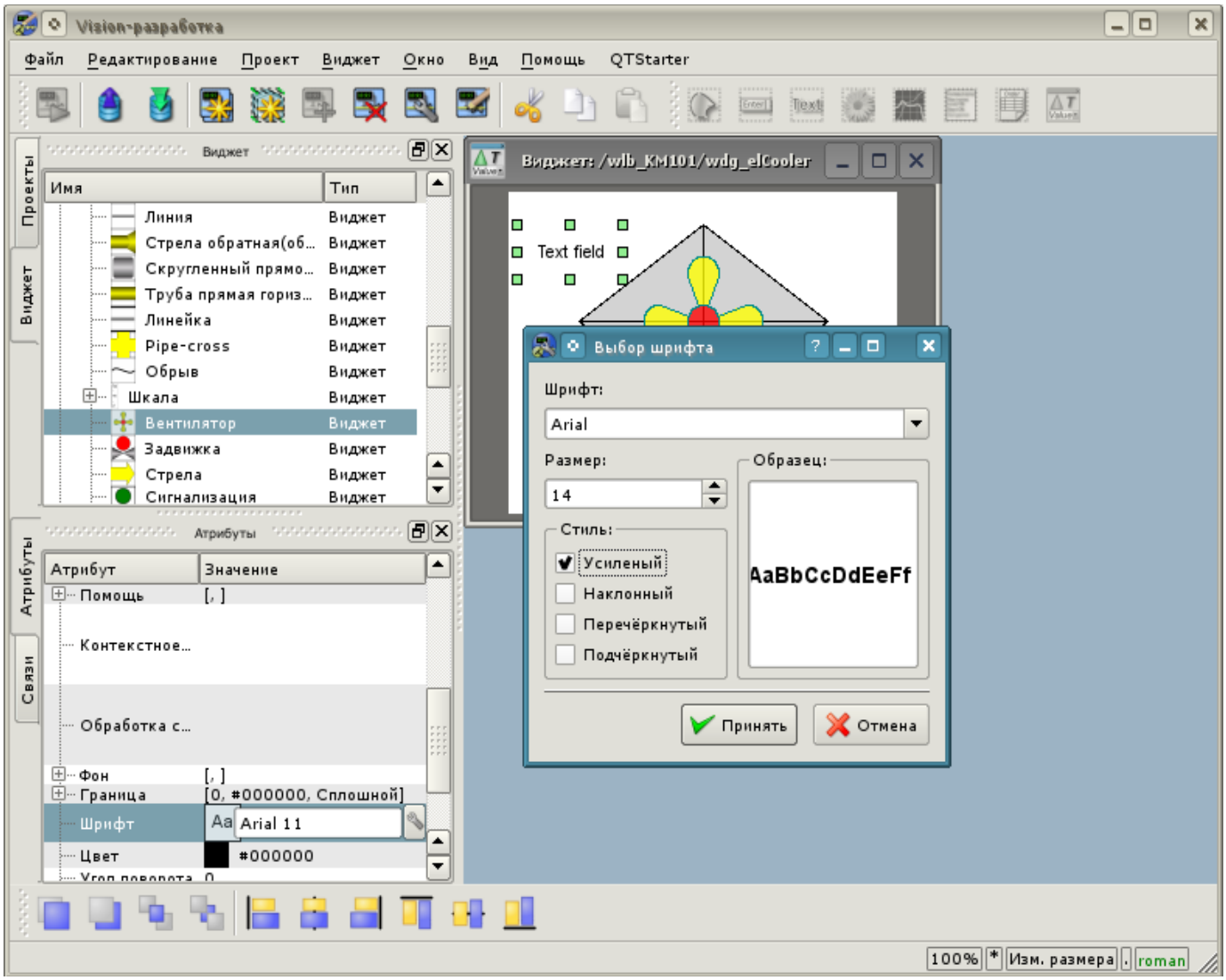

Рис. 5.3.2.7. Изменение размера шрифта для виджета "Ti".

Теперь изменим поле "Текст" виджета "Ti", указав в нем наличие аргумента "%1", в который впоследствии будет передаваться реальное значение входной температуры (рис. 5.3.2.8).

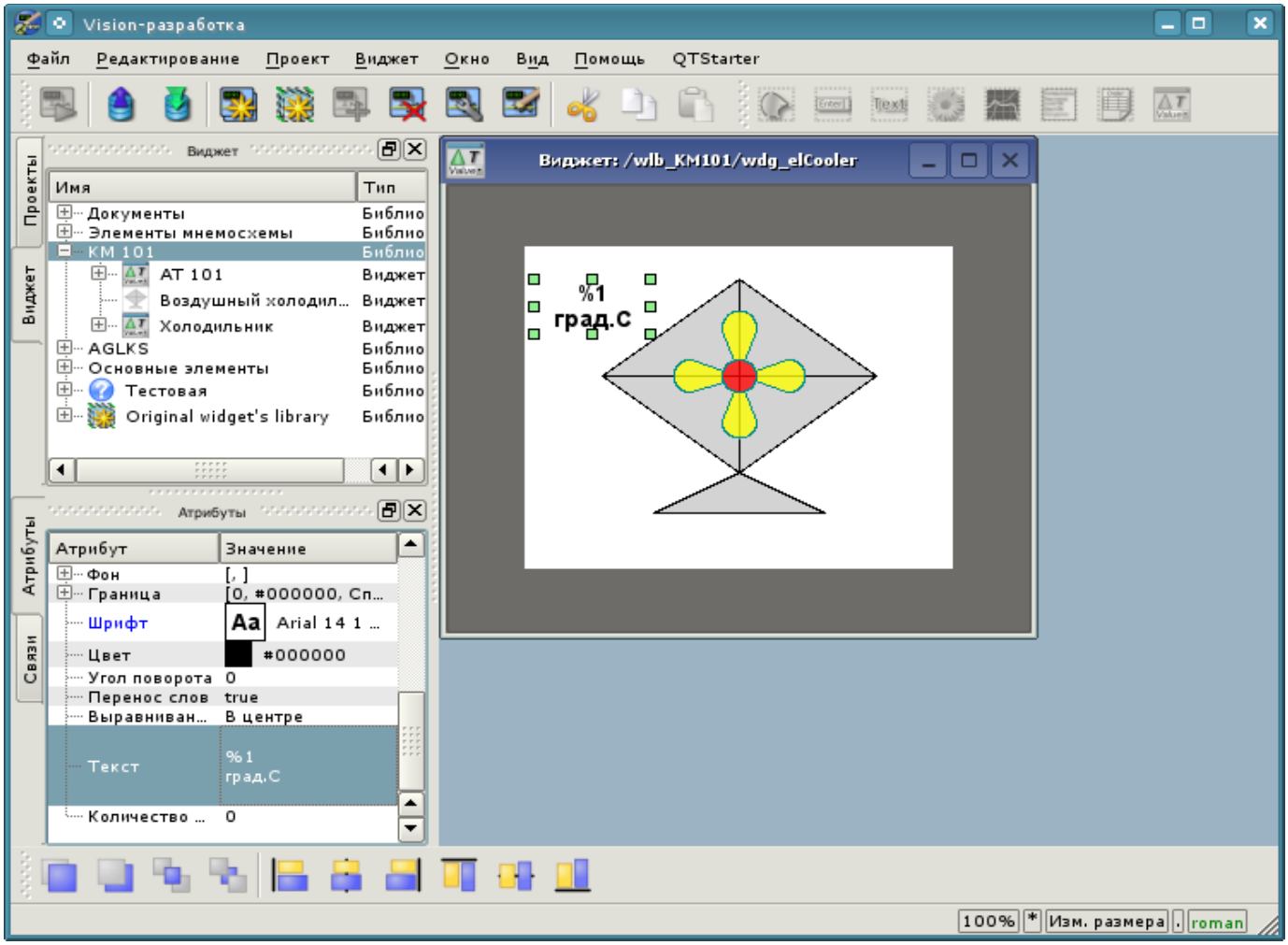

Рис. 5.3.2.8. Изменение поля "Текст" и указание в нем наличия атрибута для виджета "Ti".

Далее в перечне атрибутов виджета "Ti" в пункте "Количество аргументов" укажем "1" и сконфигурируем аргумент (рис. 5.3.2.9). Число "300.25" задано лишь только с целью наглядности, в режиме исполнения оно будет подменяться реальным значением входной температуры.

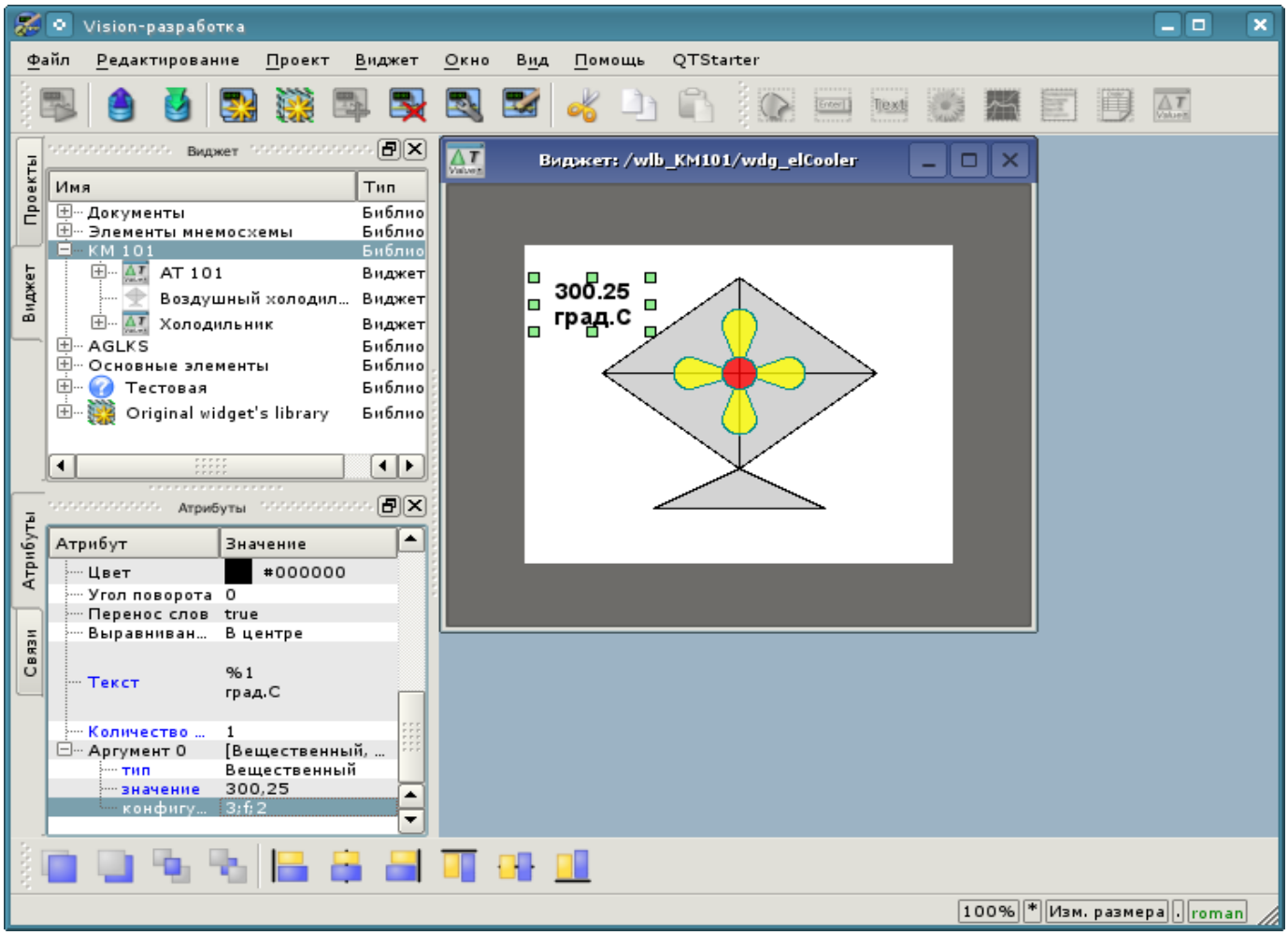

Рис. 5.3.2.9. Конфигурация аргумента для виджета "Ti".

Теперь скопируем виджет "Ti" с целью создания аналогичного ему виджета "To" (выходная температура), как это показано на рисунке 5.3.2.10.

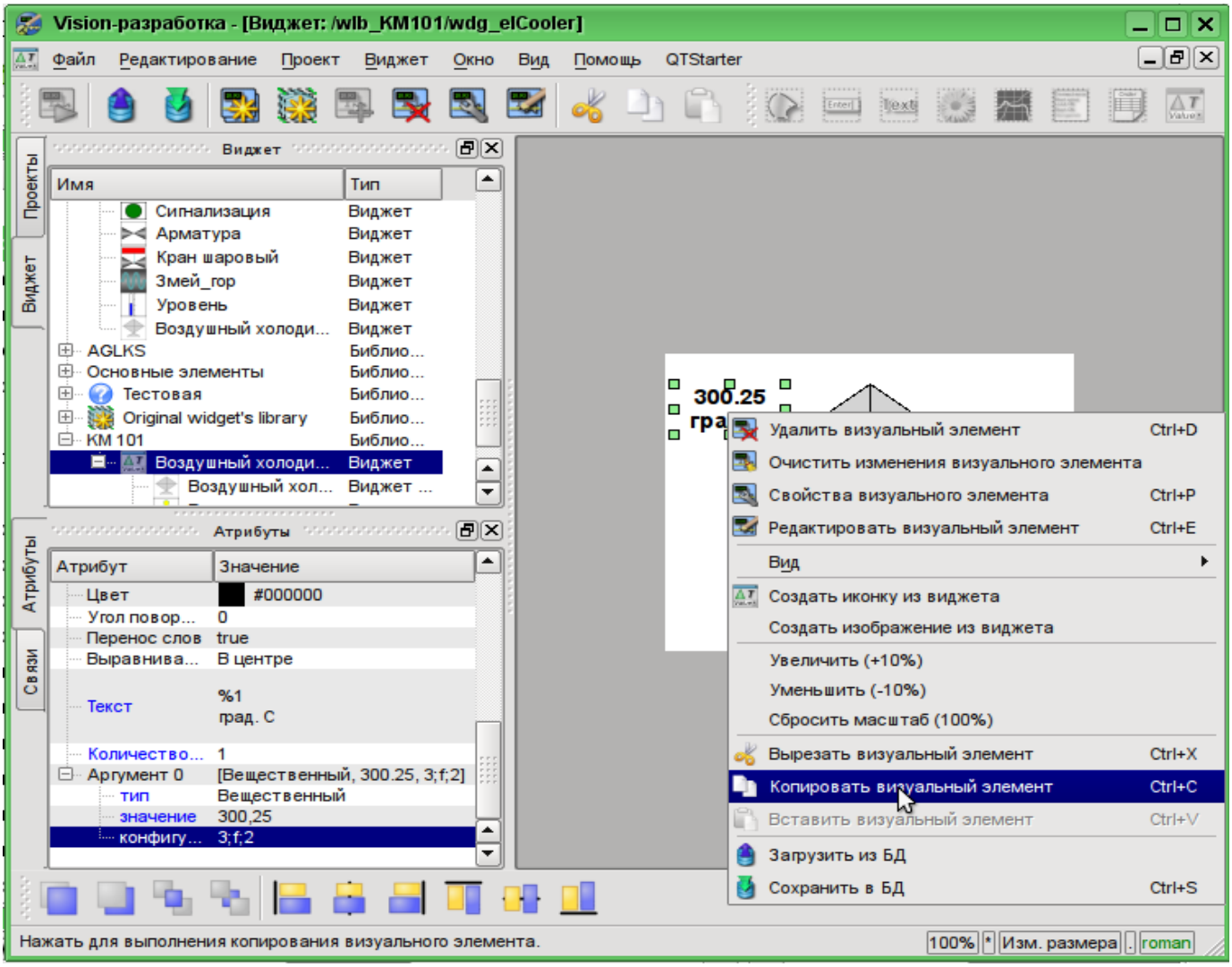

Рис. 5.3.2.10. Копирование виджета "Ti".

Вставим скопированный виджет в виджет-контейнер "Холодильник" в библиотеке "KM 101" (рис. 5.3.2.11). В диалоге ввода идентификатора и имени для вновь созданного виджета в поле "ID" укажем "To", а имя оставим пустым.

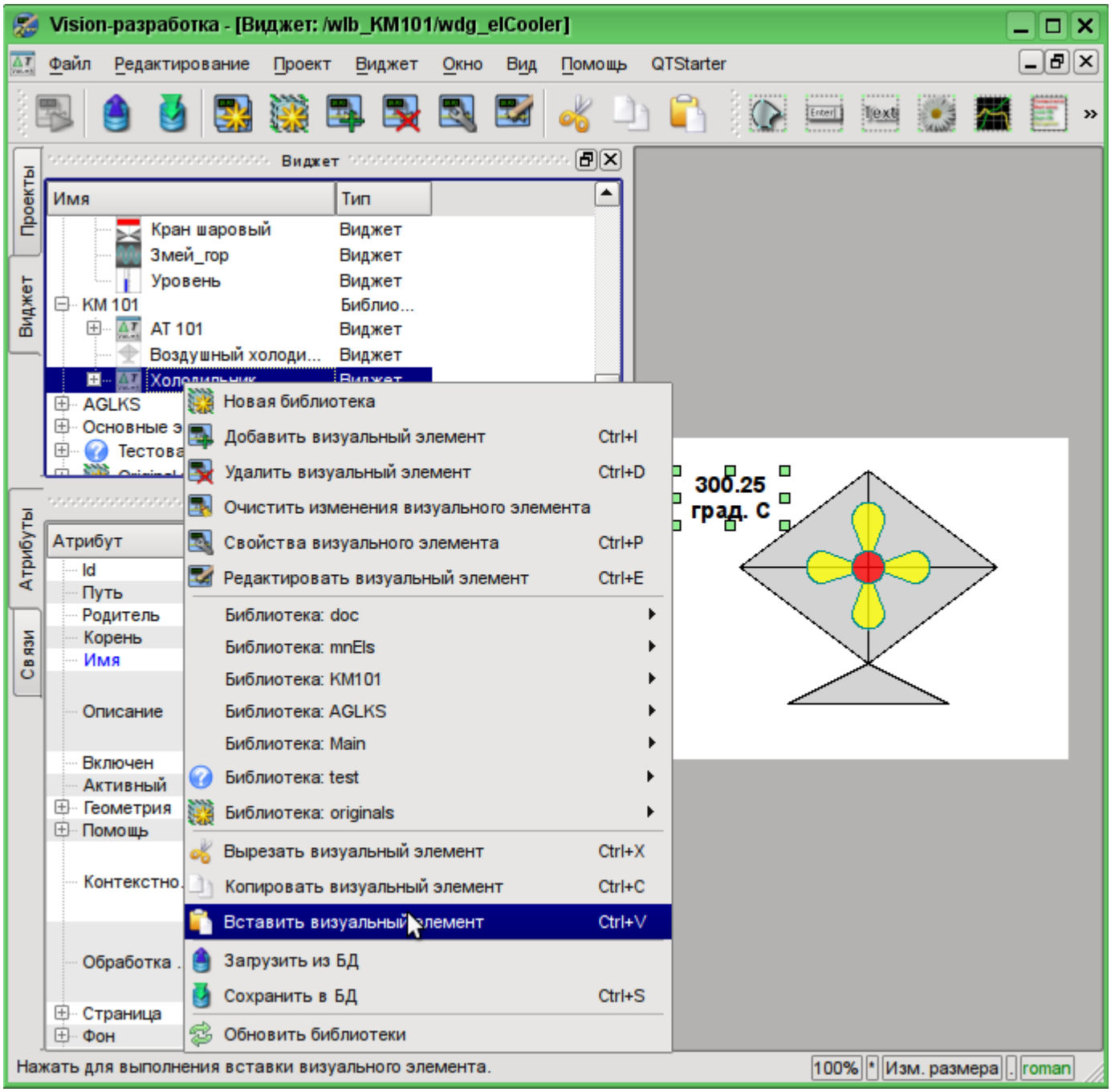

Рис. 5.3.2.11. Вставка скопированного виджета в виджет-контейнер "Воздушный холодильник" библиотеки "KM 101".

Изменим геометрию виджета "To", как показано на рис. 5.3.2.12.

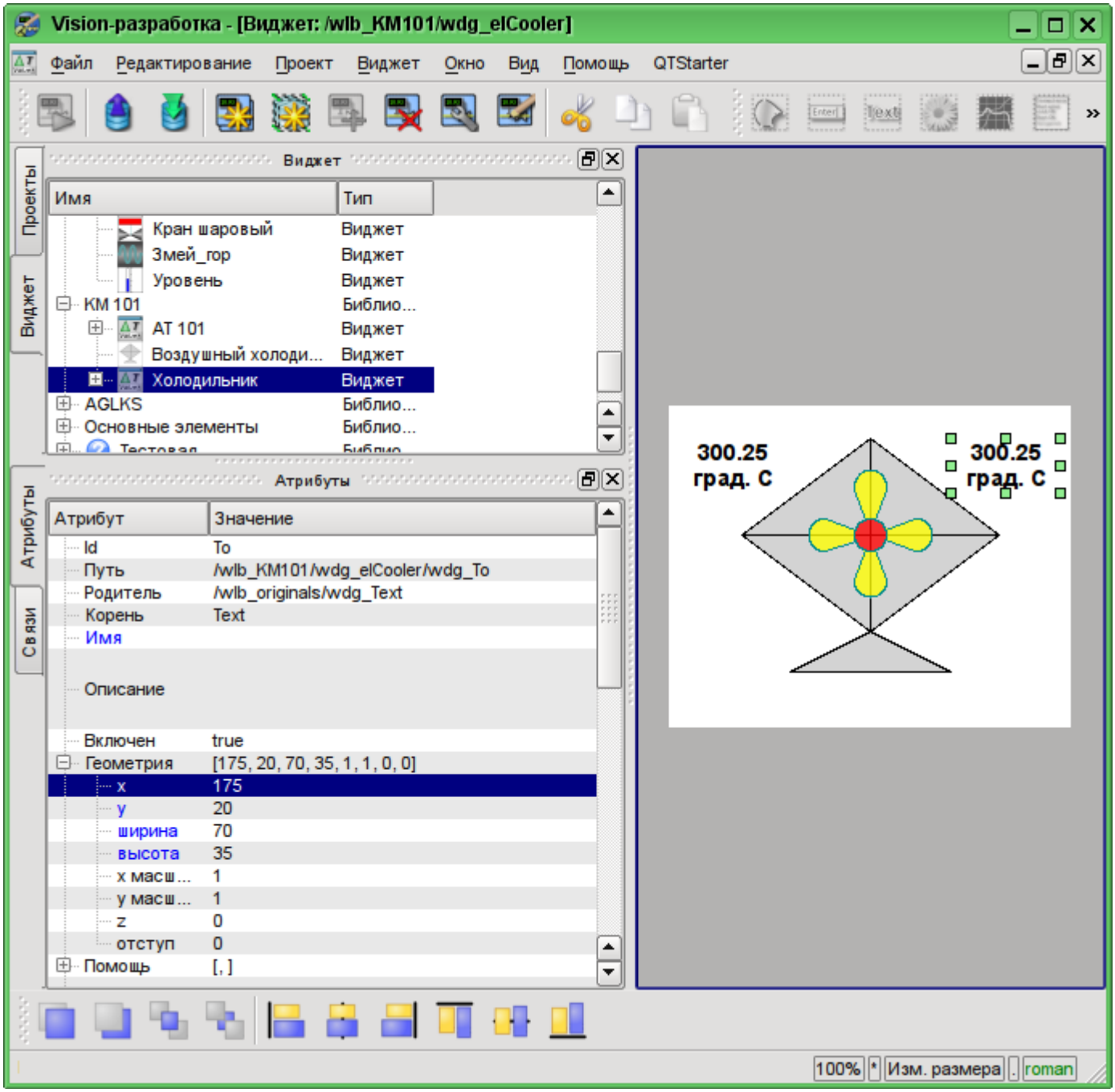

Рис. 5.3.2.12. Изменение геометрии виджета "To".

Теперь добавим виджет, основанный на примитиве "Элементы формы", который будем использовать в качестве ComboBox для выбора заданий производительности холодильника. В качестве идентификатора укажем "cw", и поле "Имя" оставим пустым. (рис. 5.3.2.13)

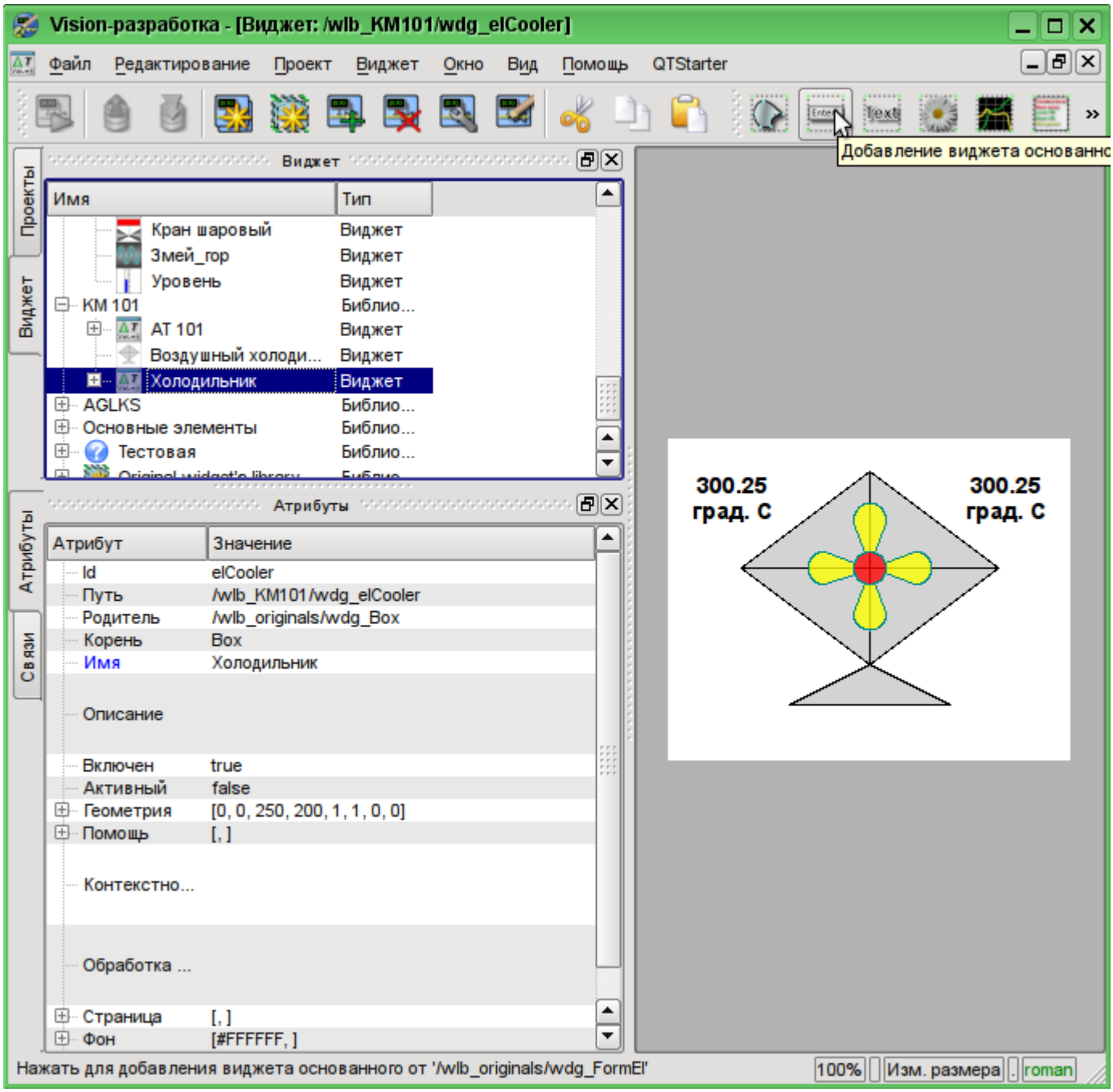

Рис. 5.3.2.13. Добавление виджета, основанного на примитиве "Элементы формы".

Зададим параметры пункта "Геометрия" во вкладке "Атрибуты" для вновь добавленного виджета: координаты "x", "y" верхнего левого угла, ширину и высоту в 60, 158, 60, 40 соответственно. Изменим "Тип элемента" в Combo Box, как это показано на рис. 5.3.2.14.

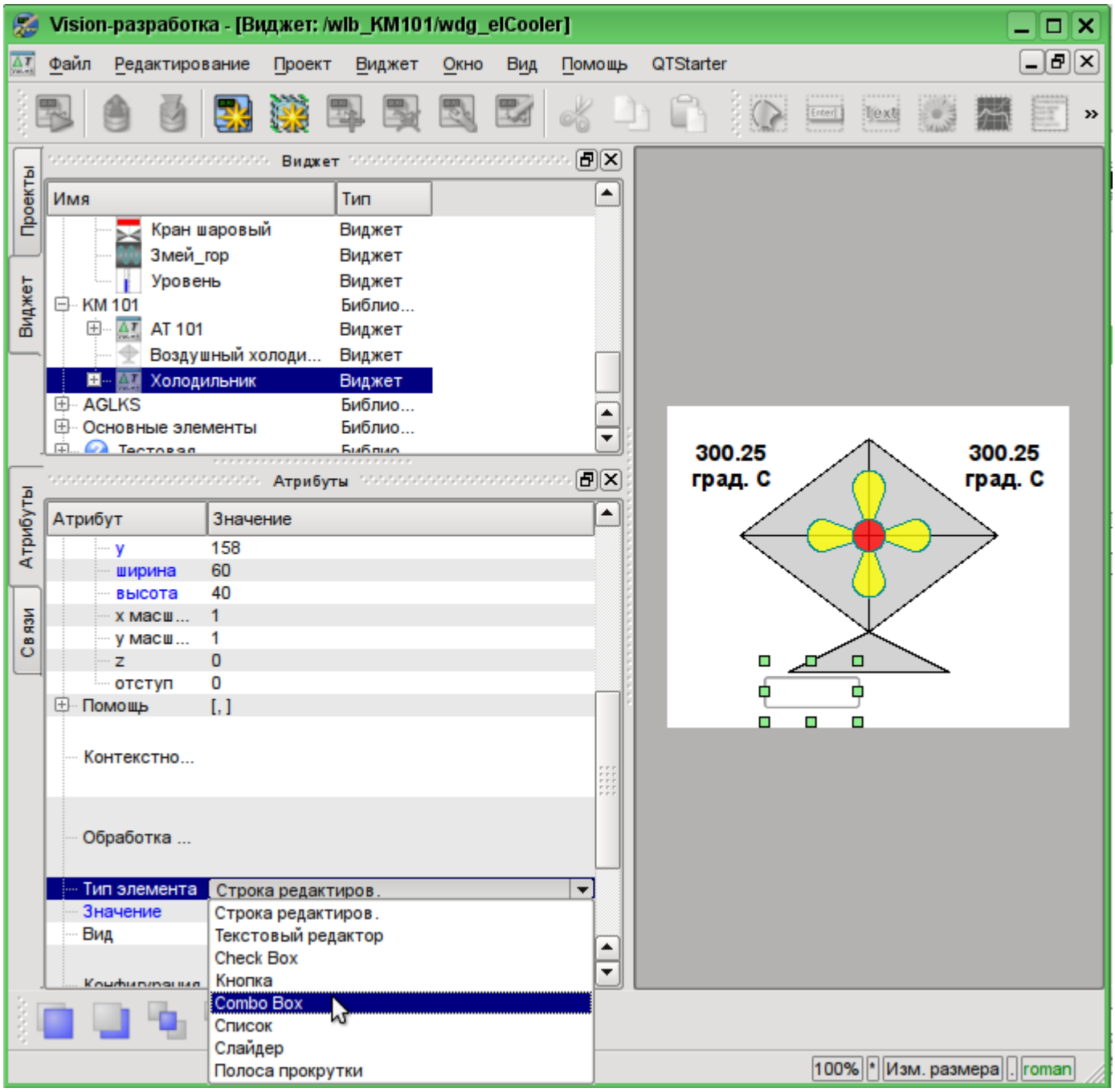

Рис. 5.3.2.14. Изменение "Геометрии" и "Типа элемента" для вновь созданного виджета.

Заполним поля: "Значение", "Элементы" и "Шрифт", как это показано на рис. 5.3.2.15. Кроме этого важно комбобокс поднять над всеми и сделать активным. Для активизации виджета комбобокса нужно установить соответствующее свойство для него.

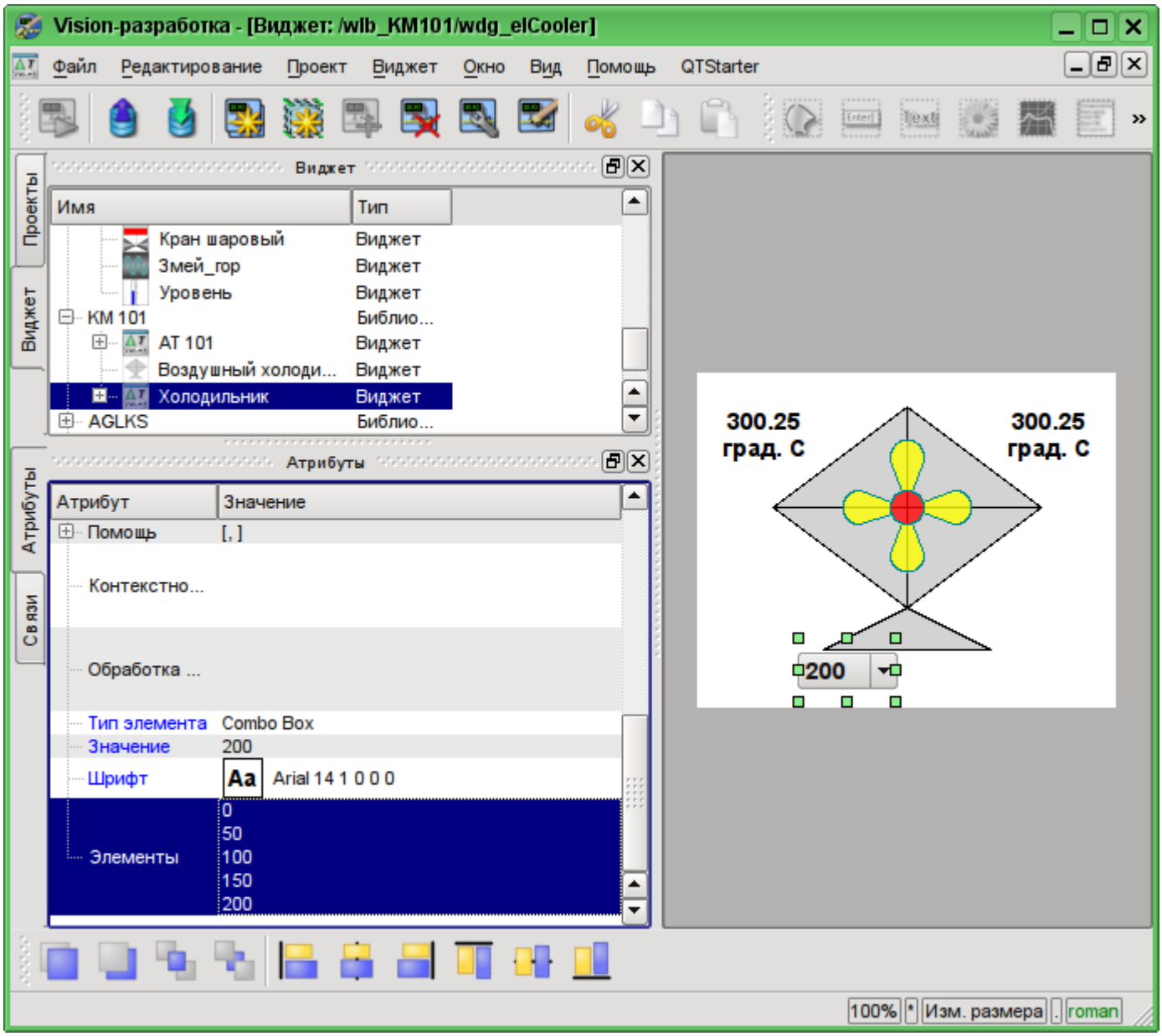

Рис. 5.3.2.15. Заполнение параметров ComboBox "cw".

Для отображения единицы измерения производительности холодильника добавим виджет на основе примитива "Текст". Делаем ту же процедуру, что и для виджета "Ti". Идентификатором вновь созданного виджета сделаем "dimension", изменим геометрию: координаты верхнего левого угла "x", "y" укажем в 125, 168 соответственно, а ширину и высоту - в 60 и 20 соответственно. Поменяем размер шрифта на "14 усиленный", а в поле "Текст" впишем "об./мин.", что и будет нашей единицей измерения (рис. 5.3.2.16).

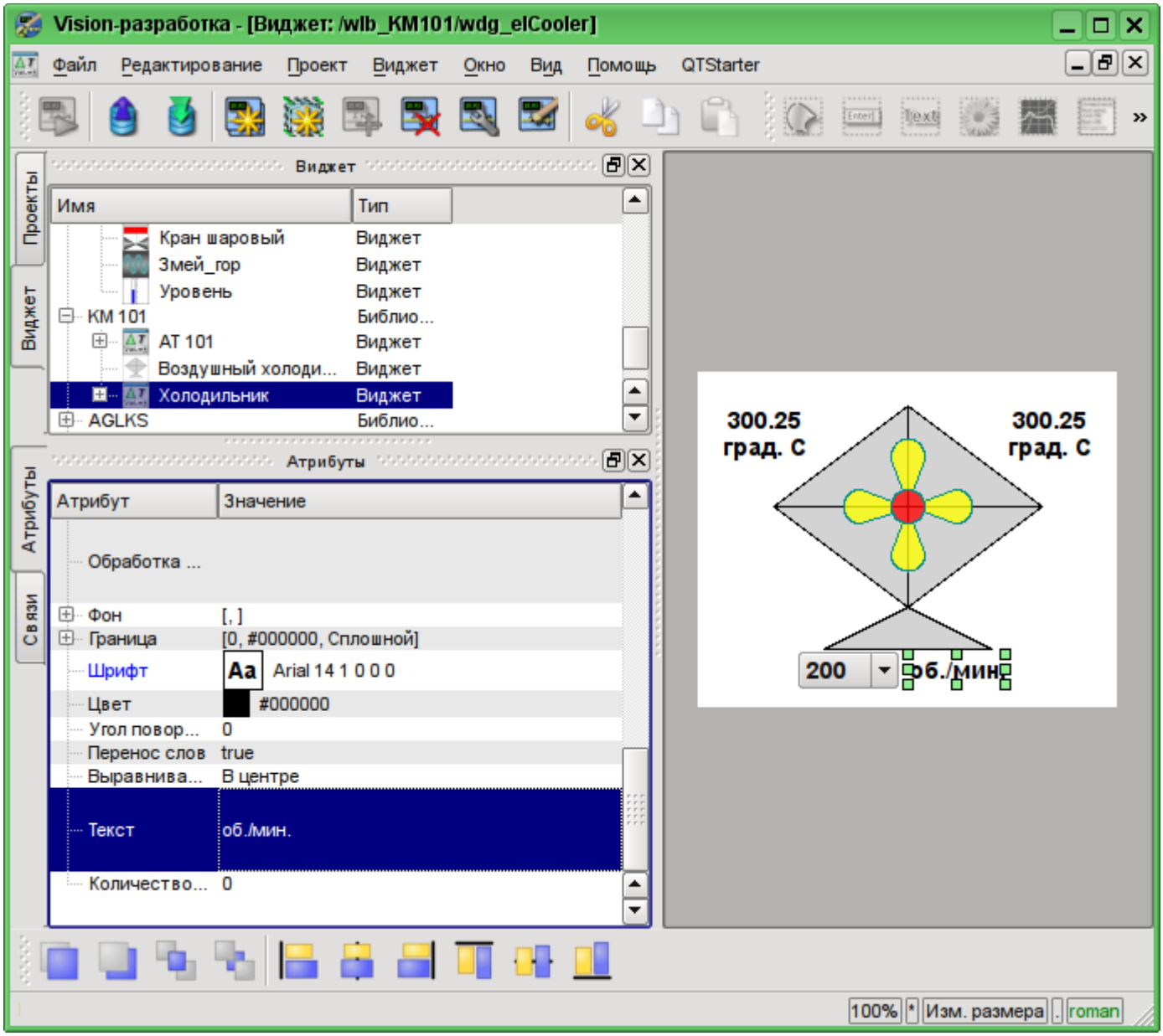

Рис. 5.3.2.16. Добавление виджета "dimension", основанного на примитиве "Текст" и изменение его параметров.

Для добавления логики обработки виджета "Холодильник"(elCooler) откроем диалог редактирования свойств этого визуального элемента и перейдём на вкладку "Обработка". На этой вкладке мы увидим дерево атрибутов виджета и поле текста программы, для обработки атрибутов. Для решения нашей задачи нужно добавить три атрибута: Ti, To, Cw (рис. 5.3.2.17). Для добавления атрибута нужно развернуть корневой элемент ".", выбрать любой элемент внутри корневого и нажать кнопку "Добавить атрибут" снизу.

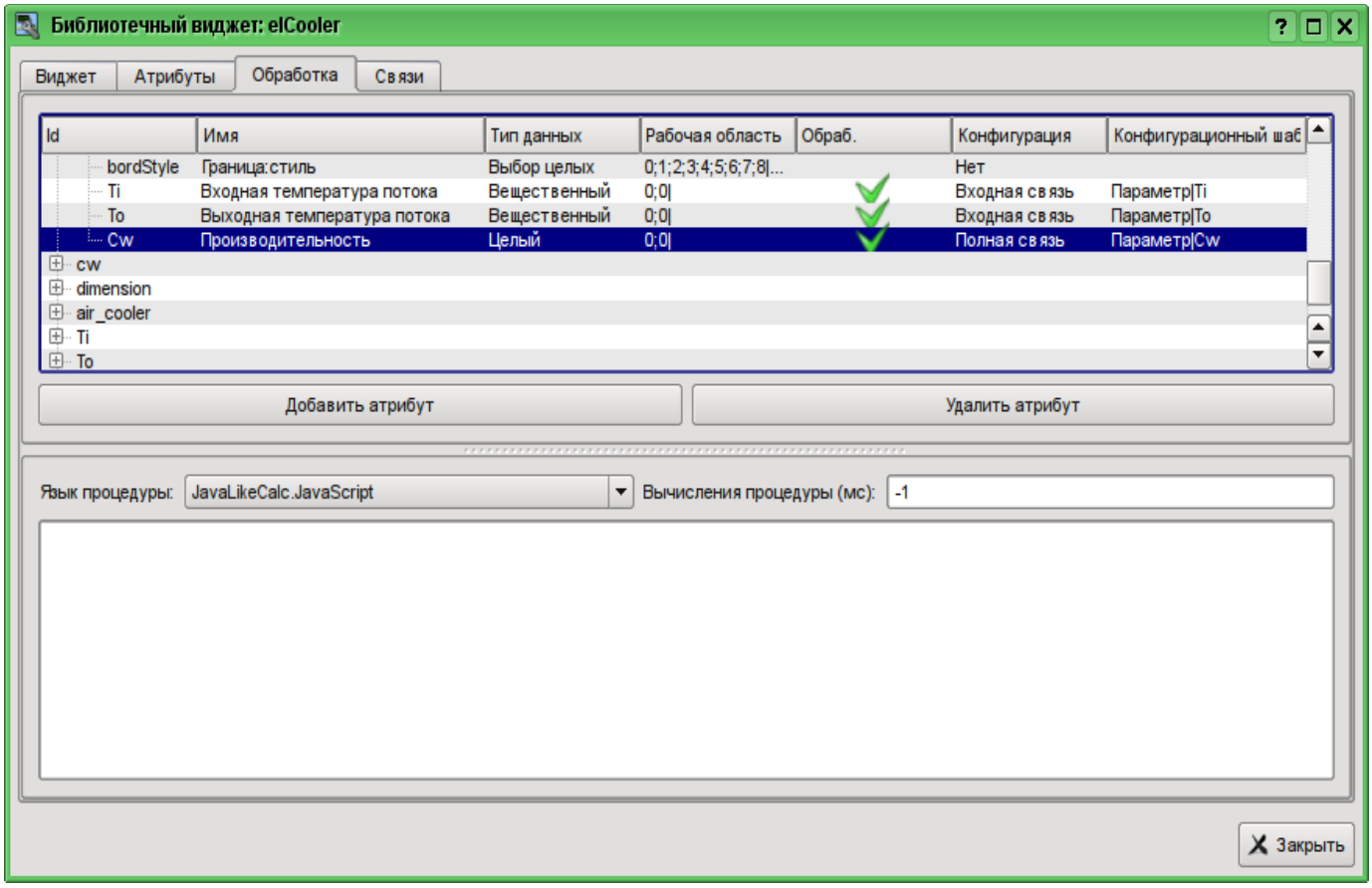

Рис. 5.3.2.17. Добавление трех атрибутов для элемента "elCooler" библиотеки "КМ 101".

Далее включим в обработку атрибут "value" комбобокса "cw", как это показано на рис. 5.3.2.18. Аналогично включим в обработку атрибут "arg0val" для Ti и To, а также атрибут "speed" у элемента "cooler2".

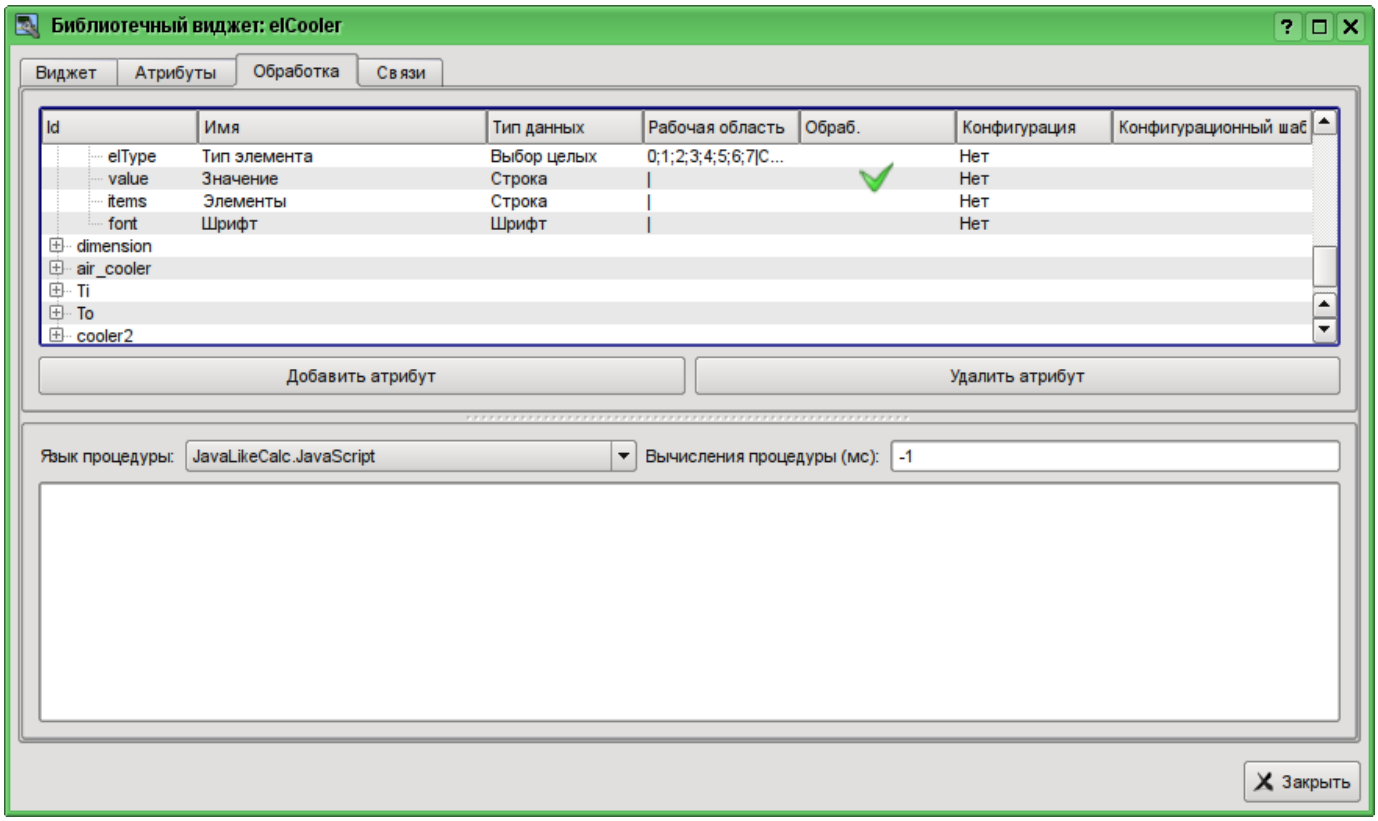

Рис. 5.3.2.18. Включение в обработку атрибута "value" комбобокса "cw".

В завершении установим язык пользовательского программирования для процедуры в "JavaLikeCalc.JavaScript" и напишем программу обработки этого виджета:

```
Ti_arg0val = Ti;
To_arg0val = To;

ev_wrk = ev_rez = "";
off = 0;
while(true)
{
  ev_wrk = Special.FLibSYS.strParse(event,0,"\n",off);
  if(ev_wrk == "") break;
  if(ev_wrk == "ws_CombChange:/cw") Cw = cw_value;
  else ev_rez += ev_wrk+"\n";
}
cw_value = Cw;
cooler2_speed = Cw/5;
```

Внимание! Помещение или редактирование программы виджета не приводит к непосредственной её компиляции, а значит не будет сообщений об ошибках в программе, если они имеют место быть. Это связано с тем, что непосредственное исполнение программы, а значит и её компиляция, осуществляется в окружении и в момент запуска на исполнение проекта визуализации. При этом все ошибки, возникшие при компиляции, выводятся в виде сообщений OpenSCADA, а виджеты с ошибками не исполняются. Посмотреть архив сообщений OpenSCADA можно в главной вкладке подсистемы "Архивы" или в терминале запуска OpenSCADA, если запуск был из терминала или его эмулятора.

Результирующий вид вкладки обработка виджета "elCooler" библиотеки "KM 101" будет иметь вид, показанный на рис. 5.3.2.19.

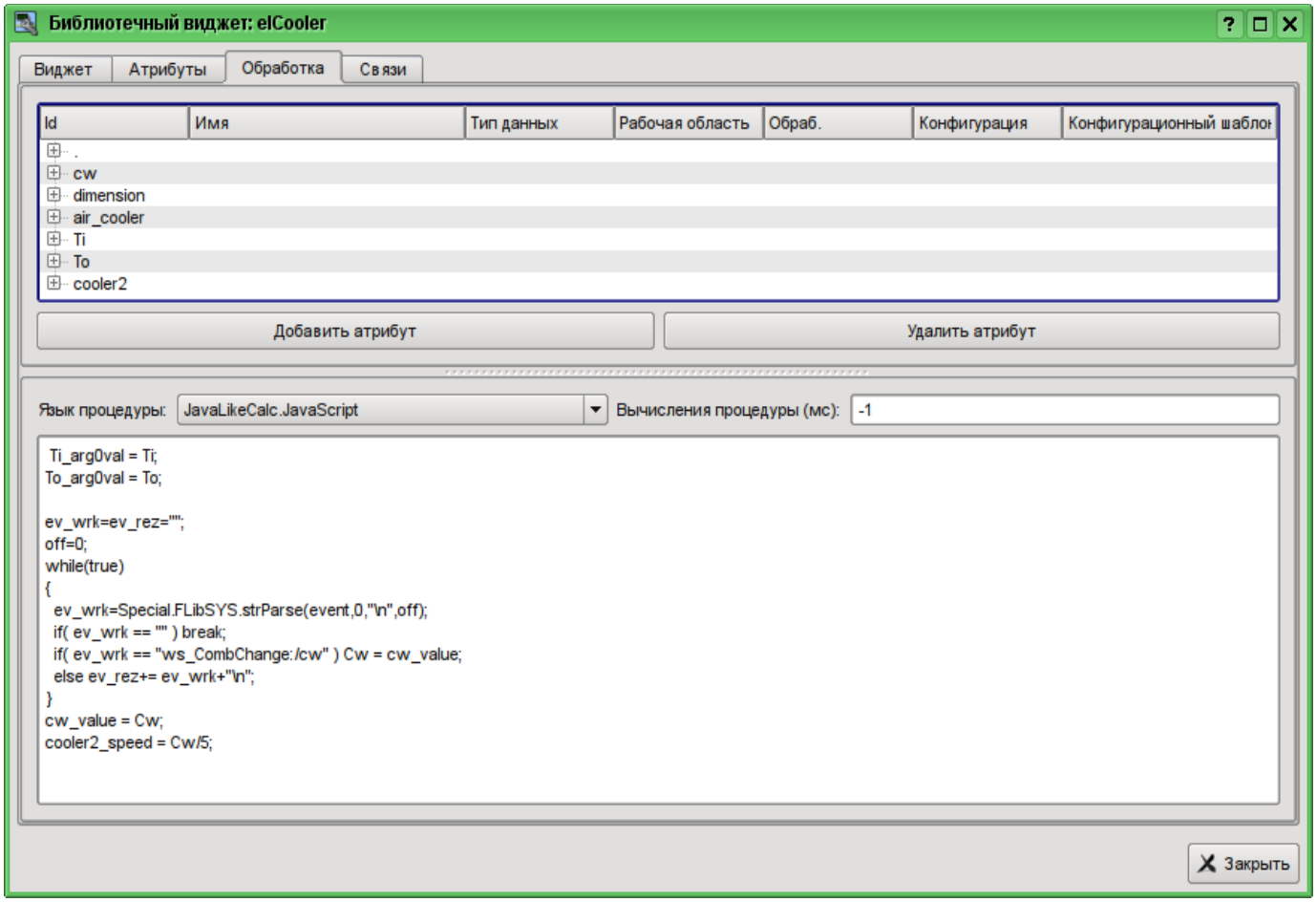

Рис. 5.3.2.19. Результирующий вид вкладки обработка виджета "elCooler" библиотеки "KM 101".

Закроем диалог редактирования свойств визуального элемента, создадим иконку на основе нашего элемента, закроем внутреннее окно редактирования и сохраним это всё.

На этом разработку комплексного элемента можно считать законченной.

5.3.3. Добавление комплексного элемента на мнемосхему

Для проверки работоспособности и оценки результатов наших усилий добавим созданный виджет на мнемосхему, разработанную в разделе 5.2. Выполним эту операцию для двух холодильников "AT101_1" и "AT101_2".

Для этого откроем кадр мнемосхемы "AT 101" на редактирование. После чего хватаем "мышью" наш комплексный элемент и тащим на мнемосхему, где отпускаем в нужной нам позиции. В диалоге запроса имени вводим идентификаторы "AT101_1" и "AT101_2" соответственно. Поле имени опускаем. Добавленные элементы располагаем как нам удобно. После выполнения подобных манипуляций у нас должна получиться мнемосхема с видом, похожим представленной на рис.5.3.3.1.

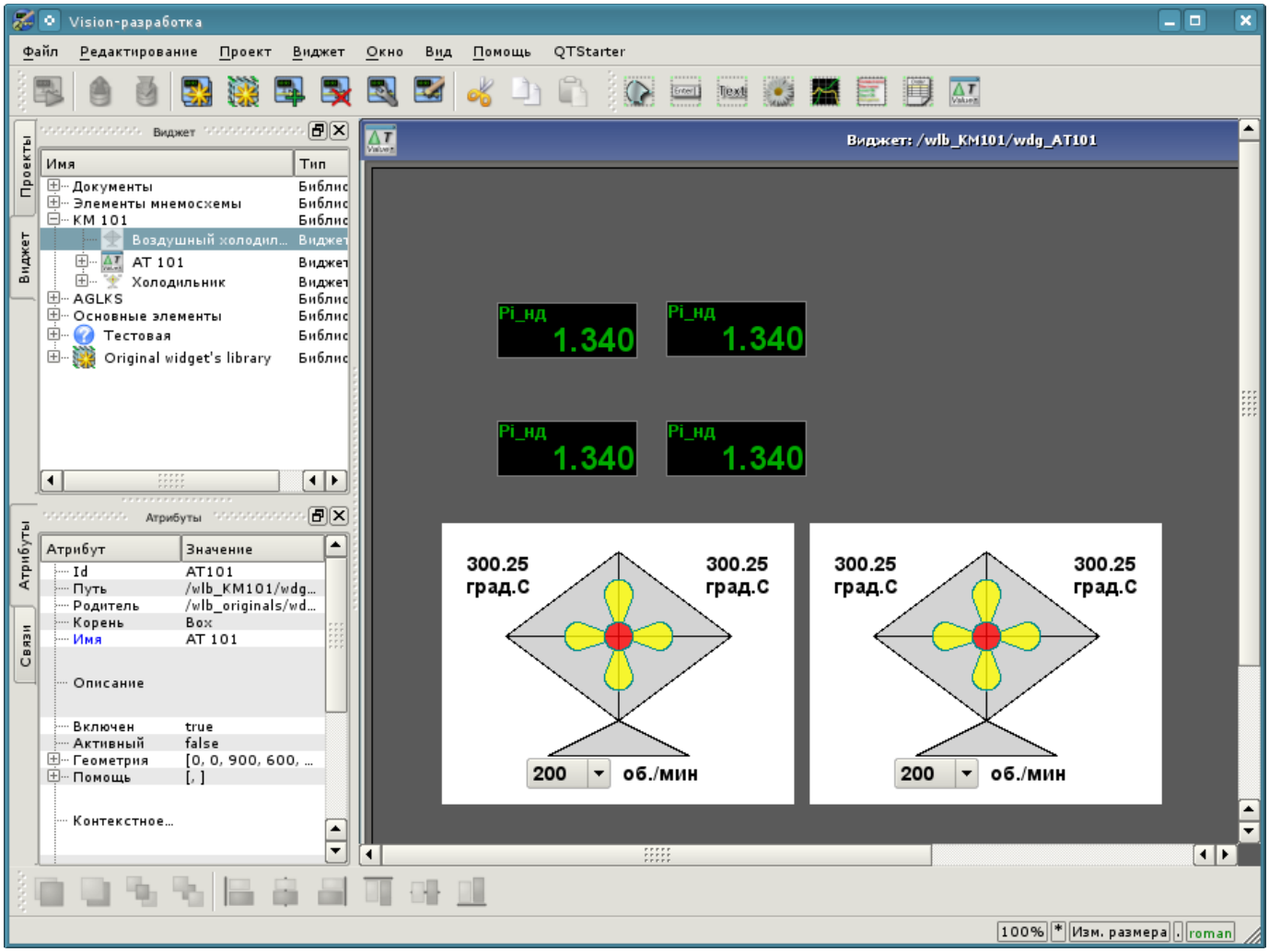

Рис. 5.3.3.1. Вид мнемосхемы с комплексными элементами.

Сохраним новую мнемосхему и закроем её окно. Далее перейдём в проект и откроем эту мнемосхему в дереве проекта "Группы сигнализаций (шаблон)"->"Корневая страница"->"Группа 1"->"Мнемосхемы"->"AT 101". Как можно заметить, наши новые элементы появились здесь автоматически. И нам осталось только подключить связи к новым элементам. Для этого откроем диалог редактирования свойств мнемосхемы на вкладке "Связи" (рис.5.3.3.2). На этой вкладке мы увидим дерево с элементами "AT101_1" и "AT101_2". Развернув любой из элементов, мы увидим ветку "Параметр" как раз с атрибутами "Ti", "To" и "Cw", Таким образом, мы можем просто указать адрес параметра "prm:/LogicLev/KM101/AT101_1" в поле "Параметр", а атрибуты будут расставлены автоматически.

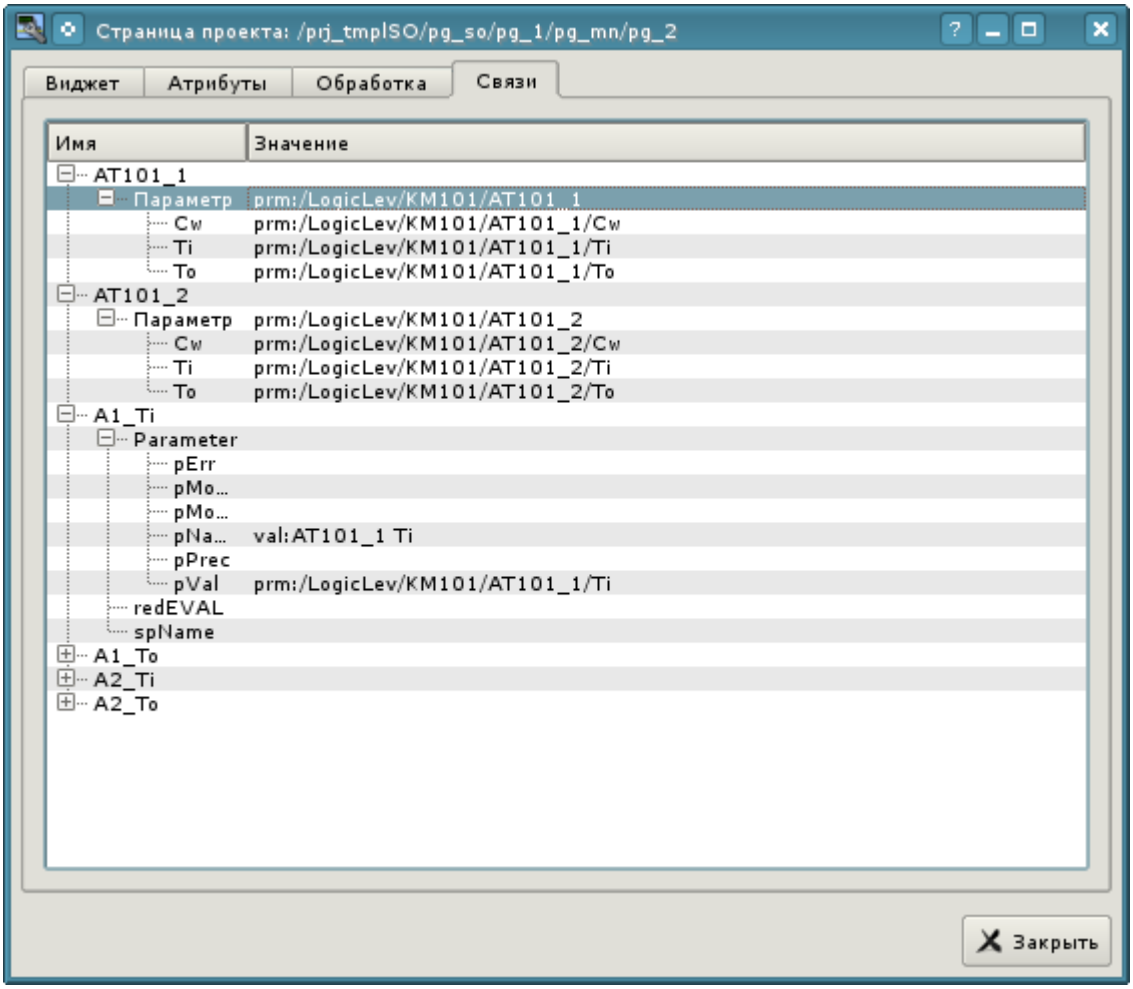

Рис. 5.3.3.2. Вкладка "Связи" диалога редактирования свойств мнемосхемы.

Сохраним нашу мнемосхему и проверим, что получилось. Для этого закроем окно диалога свойств и запустим проект "Группы сигнализаций (шаблон)" на исполнение. Затем переключимся на вторую мнемосхему кнопками листания. При безошибочной конфигурации мы должны увидеть что-то подобное изображённому на рис.5.3.3.3.

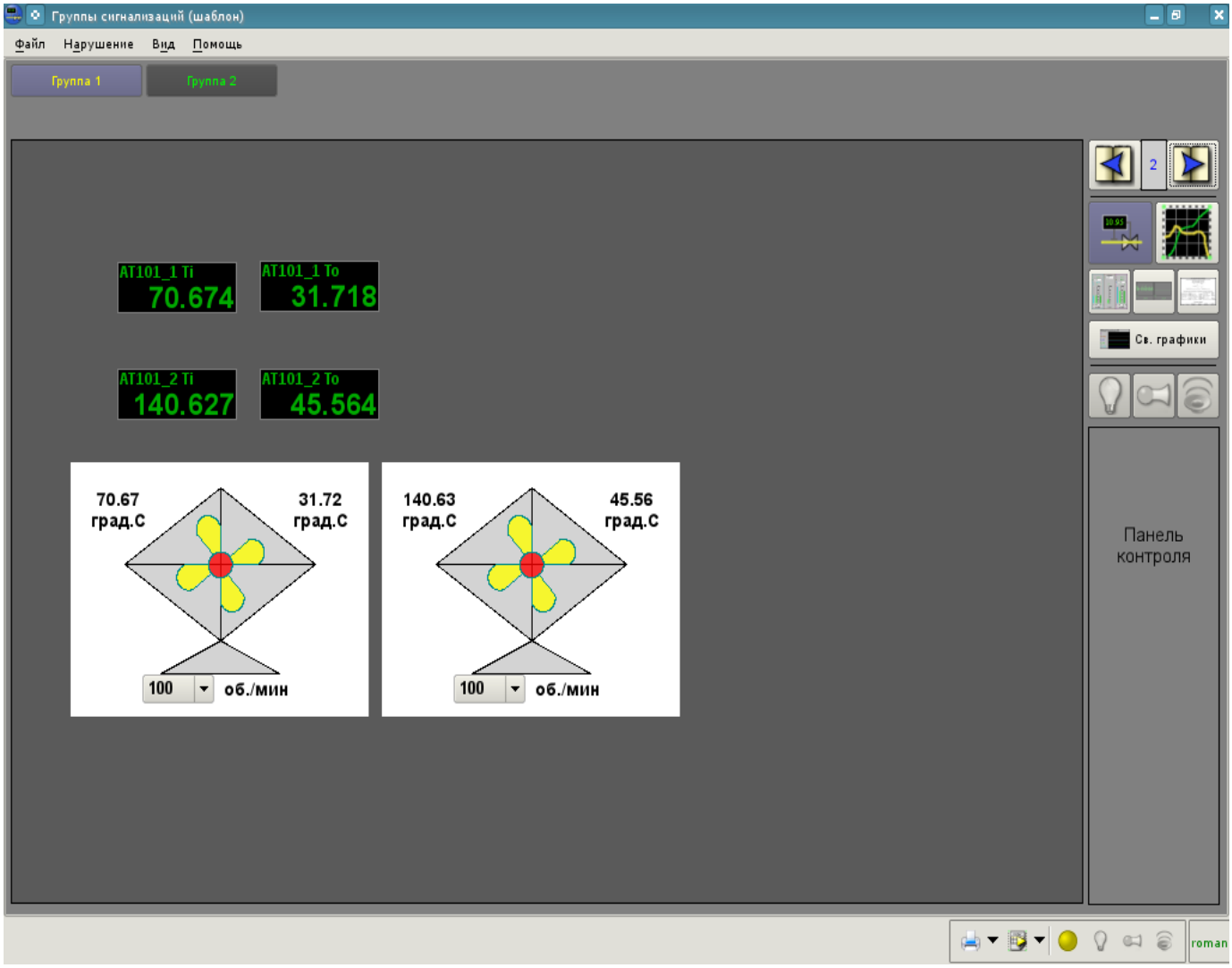

Рис. 5.3.3.3. Результирующая мнемосхема.

На этой мнемосхеме посредством наших комплексных элементов мы можем не только наблюдать но и управлять производительностью холодильников, просто меняя значение в комбобоксе. Меняя производительность, мы можем заметить и изменение температуры. Историю изменения мы можем увидеть на созданной нами в разделе 5.1 группе графиков.

6. Рецепты

Данный раздел предназначен для предоставления описания рецептов решения часто встречающихся проблем и задач пользователя. Рецепты решения задач и проблем для помещения в этот раздел могут предлагаться пользователями.

6.1. Перенос конфигураций OpenSCADA из одного проекта в другой

Часто востребованной бывает задача переноса конфигурации из одного проекта OpenSCADA в другой. Причём, чаще всего нужно осуществить частичный перенос, например, перенос отдельных наработок, которые могут пригодиться в новом проекте.

Вообще, нужно отметить, что любые наработки с малейшим намёком и перспективой вторичного использования нужно стараться унифицировать и сохранять в отдельные, собственные, библиотеки и БД. Крайне не рекомендуется непосредственно менять стандартные конфигурации и элементы стандартных библиотек, а также сохранять собственные, новые, библиотеки и элементы в базах данных стандартных библиотек. Это позволит Вам впоследствии безболезненно обновлять стандартные библиотеки, а также просто использовать наработки ваших предыдущих проектов.

Простой перенос БД с библиотеками и конфигурацией

Если Вы учли вышеуказанные рекомендации и все Ваши унифицированные наработки содержаться в отдельной БД, то весь процесс переноса будет заключаться в копировании БД и подключении её в новом проекте.

Процедура копирование БД отличается для различных типов БД и с ней нужно будет ознакомиться из документации к БД. Для сопутствующего распространения с дистрибутивами OpenSCADA обычно используется БД SQLite, в виде отдельных файлов *.db. Копирование БД SQLite соответственно заключается в простом копировании нужного файла БД из директории баз данных старого проекта в директорию баз данных нового.

Подключение осуществляется путём создания нового объекта БД в модуле нужного типа БД подсистемы БД и последующей его конфигурации (детальнее). После создания, конфигурации и включения БД можно сразу загрузить конфигурацию из неё, нажав кнопку "Загрузить систему из этой БД" на форме объекта БД.

Выделение нужной конфигурации

В случае, если нужная конфигурация содержится в общей БД или БД стандартных библиотек, то предварительно нужно осуществить выделение её в отдельную БД. Выделить конфигурацию можно или в отдельную БД с Вашими библиотеками или в экспортную БД. Экспортная БД, в отличие от библиотечной, служит только для переноса конфигурации и будет впоследствии удалена. В любом случае нужно создать новую БД для нужного типа БД, подобно процедуре подключения выше. Для переноса нужно использовать тип БД, который планируется использовать в новом проекте. Обычно для переноса лучше использовать тип БД SQLite, ввиду простой процедуры копирования. Однако, если использовать сетевую СУБД, эта процедура может превратиться в простое подключение библиотечной или экспортной БД в новом проекте.

Далее необходимо выделить конфигурацию в унифицирующие или экспортные библиотеки, если она не может быть прямо сохранена в БД. Например, отдельные шаблоны параметров или параметры контроллеров сбора данных, визуальные элементы библиотек виджетов и т.д. выделить можно путём создания библиотеки экспорта или унификации соответствующего элемента, например, библиотека шаблонов или контроллер параметров сбора данных, библиотека виджетов и т.д. Для вновь созданной библиотеки в качестве БД нужно указать ранее созданную унифицирующую или экспортную БД. Далее осуществляется копирование нужных элементов из исходной библиотеки в унифицирующую/экспортную посредством стандартной функции копирования. После копирования унифицирующую/экспортную библиотеку нужно сохранить.

В случае необходимости переноса элемента конфигурации с отдельным свойством БД или целых библиотек операцию создания промежуточной библиотеки и последующего копирования можно опустить. Достаточно в поле БД указать ранее созданную унифицирующую или экспортную БД и сохранить элемент.

Дальнейшие действия, а именно простой перенос БД, осуществляются в соответствии с предыдущим разделом.

При переносе конфигурации путём экспортирования необходимо осуществить обратный процесс копирования из экспортных библиотек в локальные библиотеки нового проекта и удаление экспортной БД.

Низкоуровневое копирование содержимого БД

Для переноса можно осуществить избирательное копирование таблиц БД с конфигурацией путём выбора объектов таблиц в объекте БД; команды копирования, выбора объекта новой БД и команды вставки (детальнее). Однако, для этого нужно знать структуру БД, про которую изложено по этой ссылке.

Заключение

Данный документ детально описывает основной процесс создания элементов пользовательского интерфейса с предварительной подготовкой и конфигурацией источника данных. В целом это позволяет быстро получить представление о работе с системой OpenSCADA, а также целенаправленно искать решения сопутствующих проблем.