



Быстрый старт

Руководство пользователя

DataRate™. Быстрый старт

© ООО «Энергокруг», 2018. Все права защищены.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотографирование, магнитную запись или иные средства копирования или сохранения информации, без письменного разрешения владельцев авторских прав.

Все упомянутые в данном издании товарные знаки и зарегистрированные товарные знаки принадлежат своим законным владельцам.

ООО «Энергокруг»

Адрес : **РОССИЯ, 440028, Пенза, Титова 1**

Тел. : **(841-2) 55-64-95, 55-64-97**

E-mail : общие вопросы : info@scadadatarate.ru, info@energokrug.ru
техническая поддержка: support@scadadatarate.ru

<http://www.ScadaDataRate.ru/>

<http://www.energokrug.ru/>

Обозначение документа: ИГТЯ.20500W-04.20-И2.1.1

СОДЕРЖАНИЕ

	Стр.
ВВЕДЕНИЕ.....	5
1 DATARATE. ОБЪЕКТНЫЙ ПОДХОД К РАЗРАБОТКЕ	6
1.1 Архитектура DataRate	6
1.2 Структура автоматизированной системы	7
2 СОЗДАНИЕ HMI-ПРИЛОЖЕНИЯ «УЗЕЛ УЧЕТА ВОДЫ»	8
2.1 Установка и запуск демо-проекта.....	8
2.2 Структура HMI-приложения Узел учета воды	9
2.3 Последовательность разработки приложения	10
2.4 Библиотеки. Типы тегов и объекты	12
2.4.1 Типы тегов.....	12
2.4.2 Библиотечные объекты.....	12
2.5 Создание объектов	15
2.6 Обмен с внешними источниками данных.....	16
2.6.1 Создание OPC DA коннектора	16
2.6.2 Привязка тегов	18
2.7 Создание мнемосхем	21
2.8 Тестирование графического интерфейса	22
2.9 События, сигнализация и оповещение	23
2.9.1 Протокол событий. Создание и визуализация	23
2.9.2 Сигнализация.....	27
2.9.3 Оповещения.....	33
2.10 Тренды. Создание и визуализация	34
2.10.1 Архивы оперативных трендов.....	35
2.10.2 Просмотр трендов в отдельном окне	37
2.10.3 Экспорт трендов	38
2.11 Отчеты.....	39
2.11.1 Шаблон отчета	39
2.11.2 Просмотр и сохранение отчета в Среде разработки	42
2.11.3 Формирование отчета по расписанию.....	45
2.11.4 Настройки отчета в Менеджере отчетов	47
2.12 Рабочие столы	48
3 ВЫПОЛНЕНИЕ ПРОЕКТА DataRate В СРЕДЕ ИСПОЛНЕНИЯ	50
ЗАКЛЮЧЕНИЕ.....	57
ДОПОЛНИТЕЛЬНАЯ СПРАВОЧНАЯ ИНФОРМАЦИЯ	58
ТЕХНИЧЕСКАЯ ПОДДЕРЖКА.....	60

ВВЕДЕНИЕ

Цель этой книги

SCADA **DataRate™** – универсальное средство мониторинга, контроля и управления производственными процессами.

DataRate™ обеспечивает динамическую визуализацию данных любого уровня управления предприятием.

Использование **DataRate** позволит Вам легко и эффективно:

- Построить автоматизированную систему контроля и управления
- Создать операторский интерфейс для отображения и сохранения информации с устройств, поддерживающих обмен данными по технологии OPC DA /HDA /A & E
- Визуализировать данные из базы данных системы управления производством
- Организовать систему отчетности
- Производить удаленный мониторинг и управление системой с использованием Web-технологий.

С помощью этой книги, на простом примере, мы хотим показать, что процесс разработки в **DataRate**, основанный на объектном подходе, в значительной мере соответствует структуре создаваемой автоматизированной системы и Вашему опыту. Эту возможность **DataRate** обеспечивает за счет готовых проектных решений (системной и предметно-ориентированных библиотек), гибкого создания и связывания объектов, графической визуализации, настраиваемой по событиям, а также за счет встроенных средств регламентации процессов (расписаний), сигнализации аварийных ситуаций, ведения истории процессов и формирования отчетов.

Для кого предназначена эта книга?

Для всех, кто собирается применять **DataRate** для динамической визуализации данных и построения автоматизированных систем:

- Разработчиков, которые хотят использовать **DataRate** для создания систем контроля и управления, учета энергоресурсов и многих других систем промышленной автоматизации
- Менеджеров проектов, инженеров технологов и обслуживающего персонала АСУ ТП, для понимания принципов работы **DataRate**
- Преподавателей и студентов, которые изучают современные программные средства и системы промышленной автоматизации.

Что Вы конкретно узнаете из этой книги?

- Какие объекты и компоненты являются основными в структуре программного обеспечения **DataRate** и для чего они предназначены
- Как построить дерево проекта автоматизированной системы и создать графический интерфейс Пользователя
- Как получить внешние данные, используя OPC-технологию
- Как отладить созданный проект и запустить его на выполнение в Среде исполнения.

1 DATARATE. ОБЪЕКТНЫЙ ПОДХОД К РАЗРАБОТКЕ

1.1 Архитектура *DataRate*

DataRate предоставляет разработчику визуальные и программные средства для создания и интерактивного взаимодействия с автоматизированной системой и ее окружением. Разработка осуществляется на основе готовых и библиотечных объектов, а также объектов созданных разработчиком (рисунок 1.1).

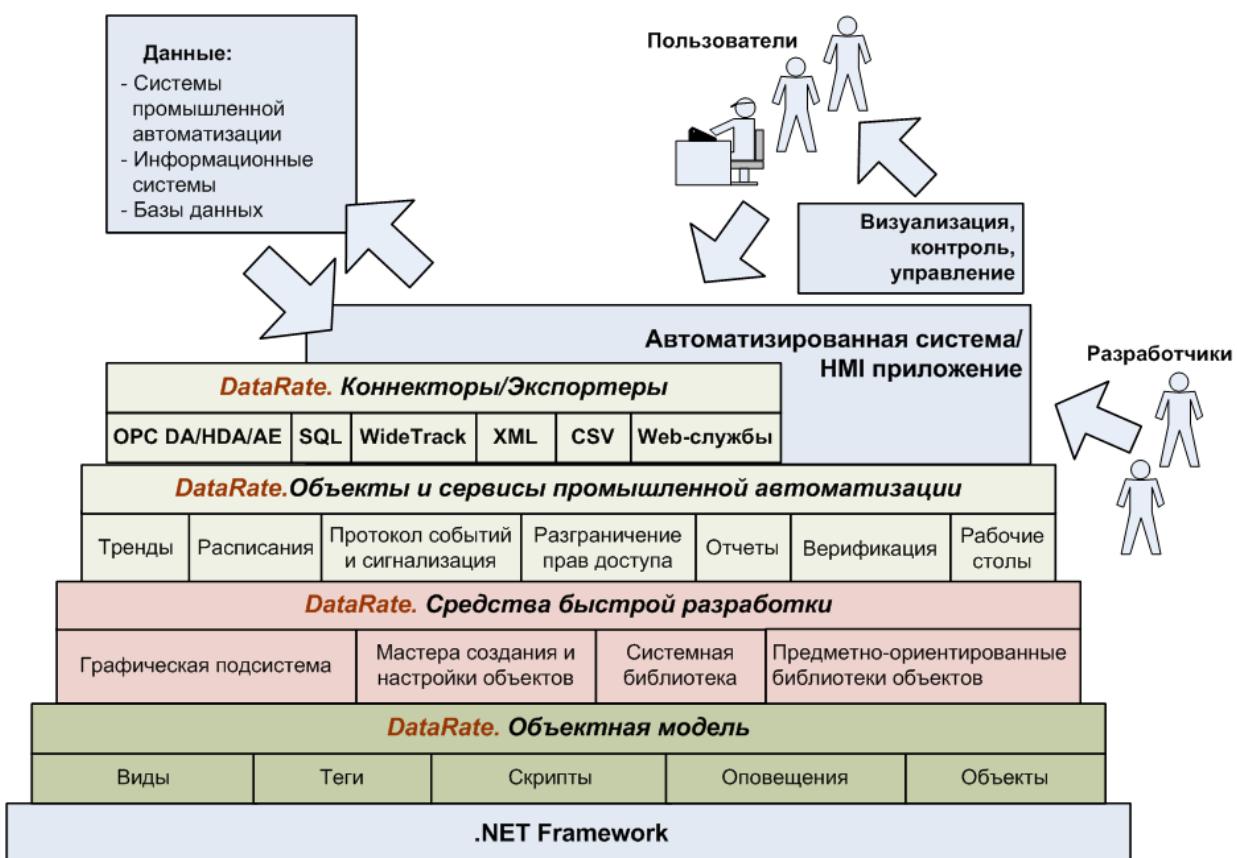


Рисунок 1.1 – Архитектура *DataRate*

Объектно-ориентированная архитектура **DataRate** позволяет реализовать типовой подход к разработке автоматизированной системы контроля и управления, при котором на основе дублирования и объединения объектов формируется законченная система. Использование системной и предметно-ориентированных библиотек технологических объектов в создании автоматизированных систем позволяет значительно сократить затраты на разработку и получить готовую систему в кратчайшие сроки. Преимущества этого подхода заключается в следующем:

- Широкий спектр и повторное использование прикладных объектов
- Быстрое создание систем и сокращение трудозатрат
- Гибкая настройка
- Свободное распространение библиотек и возможность их расширения самим пользователем
- Возможность создания комплексных, интегрированных систем автоматизации.

1.2 Структура автоматизированной системы

Структура автоматизированной системы в **DataRate** базируется на таких понятиях, как **проект, элемент проекта, объект, библиотека**.

Проект – это корневой элемент объектной модели разрабатываемой автоматизированной системы. Разработчик может построить структуру своего проекта в соответствии со структурой реального объекта автоматизации.

Создание, редактирование и отладка проекта осуществляется в **Интегрированной среде разработки**.

Выполняется проект в **Среде исполнения**.

Элемент проекта включает системные сервисы, объекты и всегда имеет библиотечную часть.

Объект DataRate (рисунок 1.2) включает:

- **Вид** – графическое изображение
- **Тег** – входы/выходы (данные с датчиков, сигналы управляющих механизмов, команды оператора и другие)
- **Скрипты** – поведение объекта, алгоритмы работы
- **Оповещение** – действие, выполняемое для оповещения о достижении объектом определенного состояния
- **Вложенные объекты**.

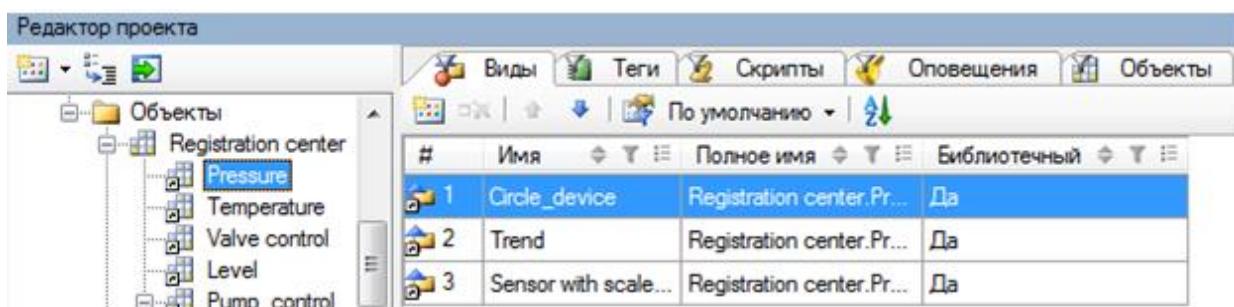


Рисунок 1.2 – Структура объекта **DataRate**

Библиотека – это набор повторно используемых компонентов: тегов, скриптов, объектов.

Объекту, созданному на основе библиотечного объекта, можно добавить новые свойства и переопределить доставшиеся в наследство. Любые изменения в библиотечном объекте будут отражаться и на любой его копии (при необходимости эту возможность можно отключить для конкретного объекта).

Любой созданный объект можно поместить в библиотеку. После этого он станет доступен для тиражирования в любом количестве в любом проекте.

Для быстрой разработки **DataRate** предоставляет системную библиотеку, библиотеку изображений **Image Library**, а также, предметно-ориентированные библиотеки (объекты энергоучета и электрических схем).

2 СОЗДАНИЕ НМІ-ПРИЛОЖЕНИЯ «УЗЕЛ УЧЕТА ВОДЫ»

Познакомиться с разработкой автоматизированных систем в **DataRate** и их работой проще всего на готовом примере – демо-проекте. Рассмотрим автоматизированную систему контроля и управления **Узел учета воды** (проект [Waterflow registration center](#)).

Система **Узел учета воды** моделирует автоматизированную систему узла учёта расхода воды: визуализирует параметры процесса (давление в трубопроводе, температуру и уровень воды в баке), управляет насосом на входе узла учёта и задвижкой – на выходе, а также отображает их состояние (рисунок 2.1).

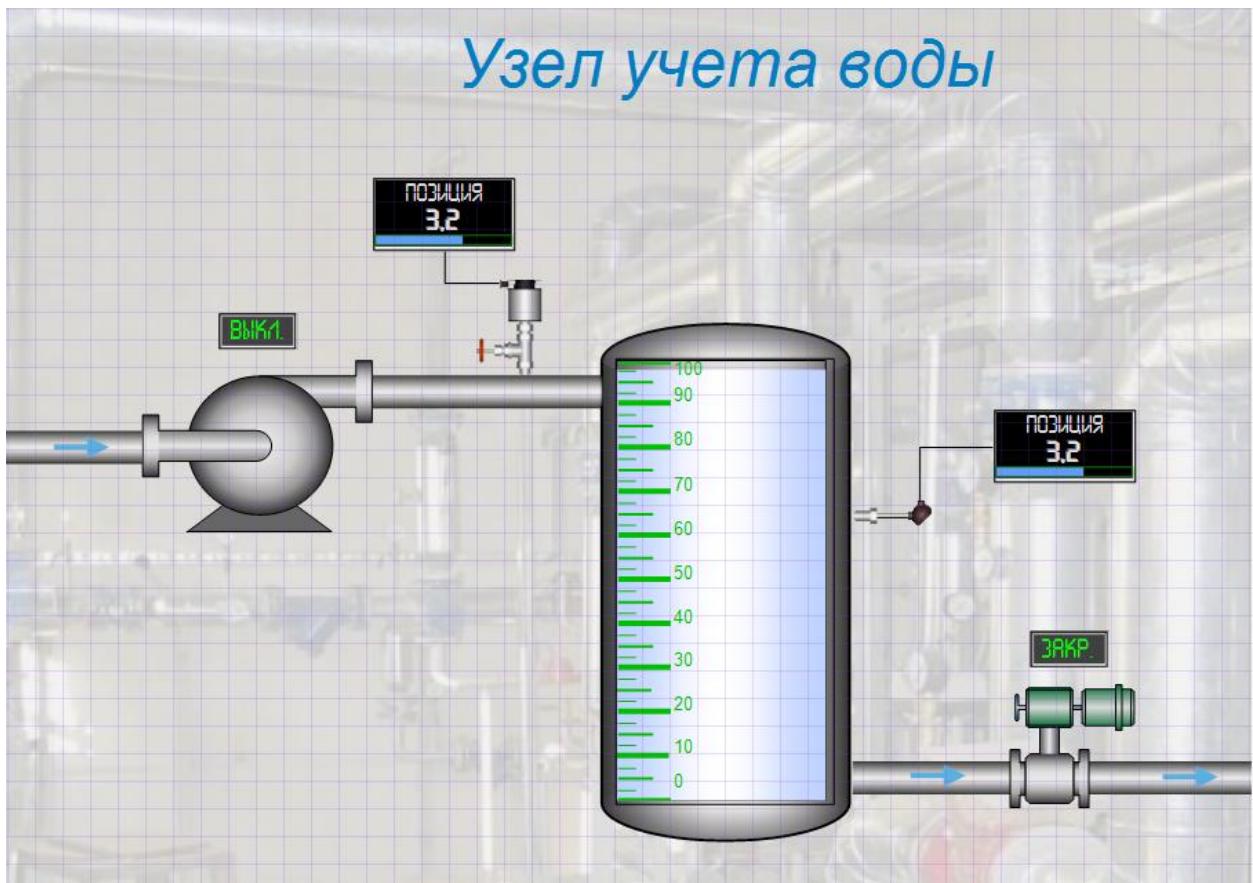


Рисунок 2.1 – Узел учета расхода воды. Главное окно.

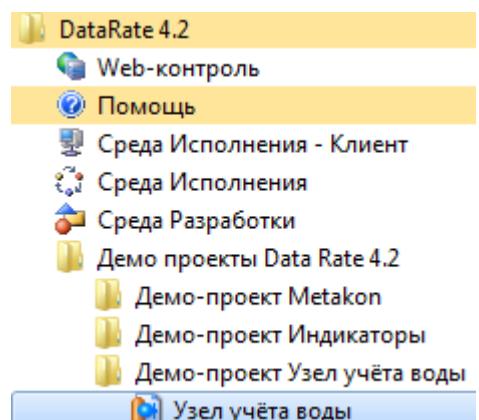
Моделирование источников данных контролируемого процесса осуществляется с помощью «учебного» **OPC DA-сервера**.

2.1 Установка и запуск демо-проекта

После установки **DataRate** выберите пункт **Установить демо-проекты** в меню установочного диска и следуйте далее указаниям инсталлятора.

Для запуска установленного демо-проекта следует :

- 1 В системном меню **Пуск** выбрать



Все программы/ DataRate 4.2/ Демо-проекты DataRate 4.2/ Демо-проект Узел учета воды

- 2 Запустить систему Узел учета воды и в появившемся окне (рисунок 2.2) выбрать Редактировать проект в Среде Разработки или Запустить проект в Среде Исполнения.

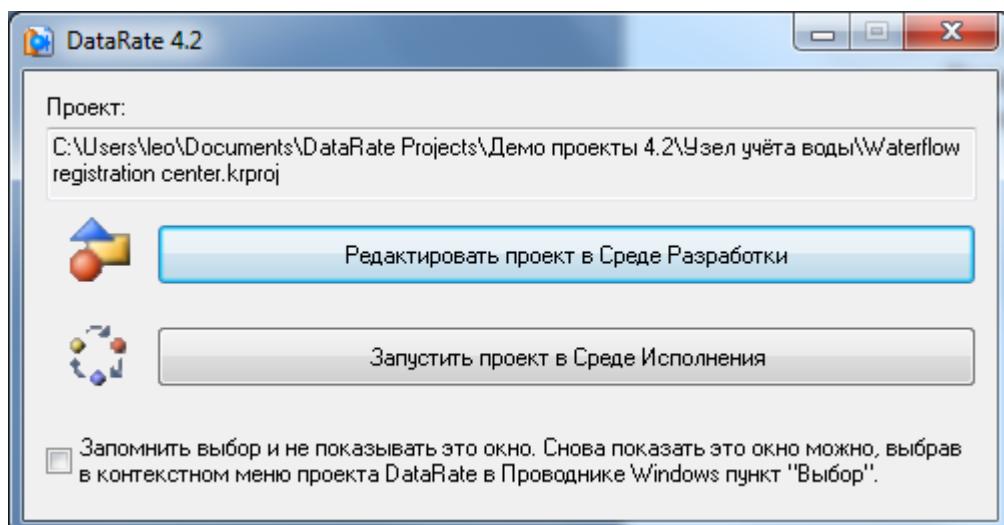


Рисунок 2.2 – Запуск HMI-приложения

2.2 Структура HMI-приложения Узел учета воды

Структура системы Узел учета воды (рисунок 2.3) отображается в окне Редактор проекта и окне Библиотеки.

Системные сервисы и объекты:

- Протокол событий – события управления насосом и задвижкой
- Менеджер трендов – настройка параметров формирования истории процесса
- Рабочие столы – главный рабочий стол
- OPC DA коннектор – обмен данными с учебным OPC DA-сервером.
- Экспортер в CSV – сохранение тренда в формате CSV

Объекты автоматизированной системы:

- Registration center – «главный» объект узла учета
- Pressure – датчик давления
- Temperature – датчик температуры
- Valve control – дискретное управление задвижкой
- Level – индикатор уровня в баке
- Pump_control – дискретное управление насосом (Switcher – переключатель состояния насоса)

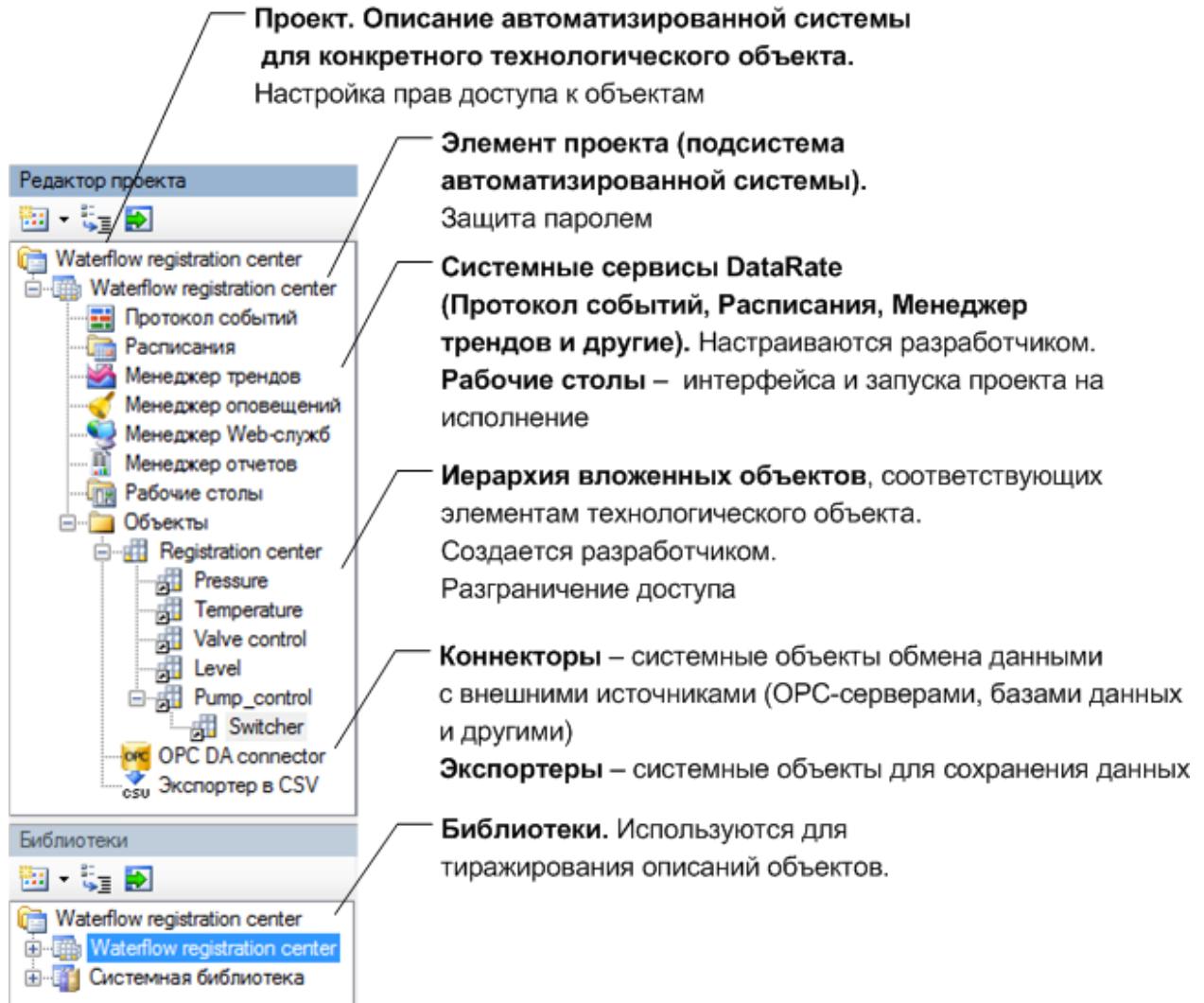


Рисунок 2.3 – Структура проекта

2.3 Последовательность разработки приложения

Последовательность разработки автоматизированной системы **Узел учета воды** в **DataRate** приведена на рисунке 2.4.

С описанием всех этапов разработки можно познакомиться в руководстве пользователя по **DataRate** [<http://www.scadadatarate.ru/download.php>].

В данном руководстве рассмотрим этапы, связанные с использованием библиотечных объектов; созданием и использованием объектов системы; получением внешних данных от OPC DA сервера, их визуализацией и трендированием; формированием отчетов.



Рисунок 2.4 – Последовательность разработки автоматизированной системы в **DataRate**

2.4 Библиотеки. Типы тегов и объекты

В **DataRate** для быстрой разработки используются типы тегов, скрипты и объекты библиотеки элемента проекта (в рассматриваемом приложении – библиотеки **Waterflow registration center**) и Системной библиотеки.

2.4.1 Типы тегов

Основой для создания тегов является **тип тега**.

Используемые в разработке типы тегов **Simple tags** библиотеки проекта содержат 7 основных атрибутов ОРС-технологии и отличаются типом атрибута **Value** (рисунок 2.5 для типа **Boolean**).

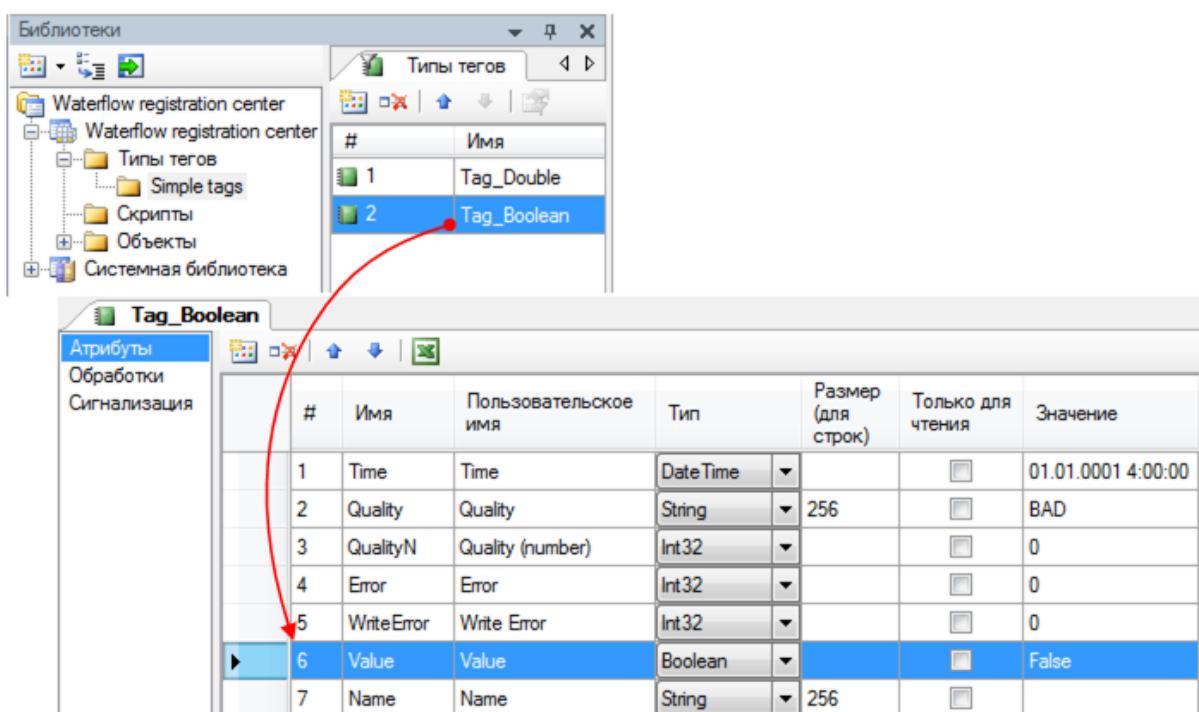


Рисунок 2.5 – Атрибуты тегов

2.4.2 Библиотечные объекты

Объекты библиотеки проекта используются для создания интерфейса пользователя и управления процессами.

В проекте **Waterflow registration center** применяются следующие библиотечные объекты:

- **Pressure** – визуализация значений давления
- **Temperature** – визуализация значений температуры
- **Valve_control** – управление задвижкой
- **Level** – визуализация значений объёма жидкости в резервуаре
- **Pump_control** – **дискретное управление насосом (Switcher – переключатель состояния)**

Рассмотрим подробнее библиотечный объект **Valve_control**.

Его теги созданы на основе библиотечных типов (пример на рисунке 2.6) и предназначены для визуализации состояния и управления задвижкой.

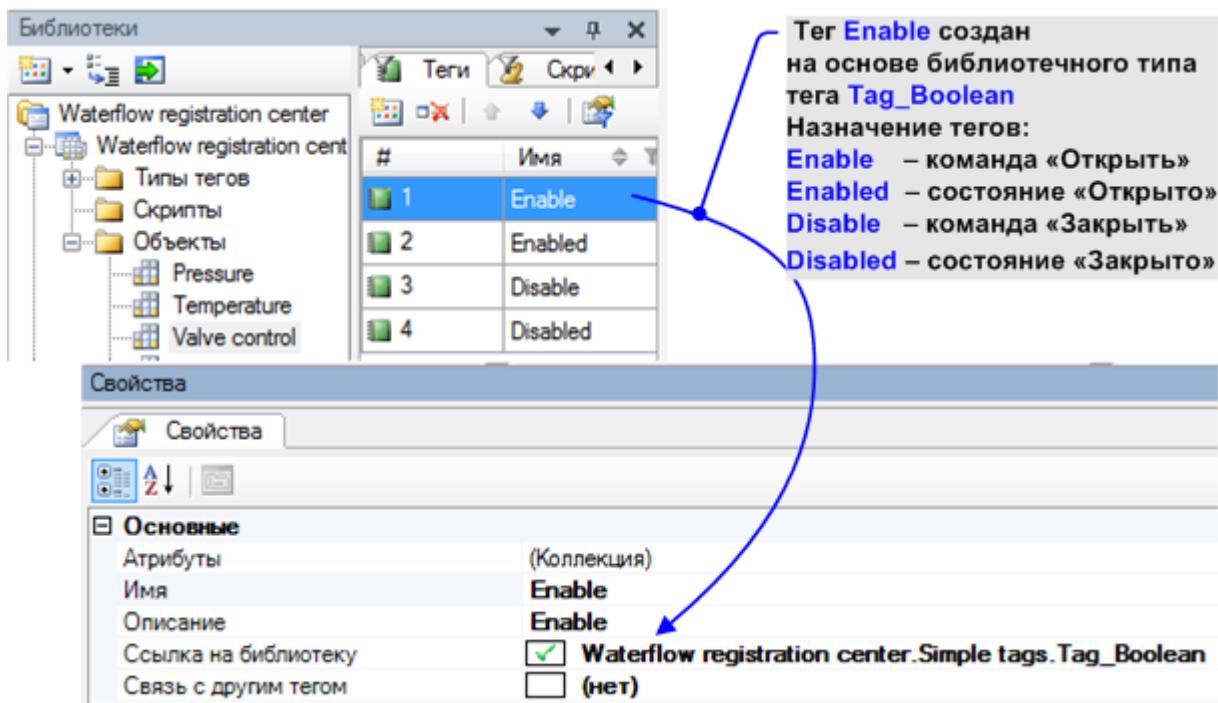


Рисунок 2.6 – Свойства тега **Enable** библиотечного объекта **Valve_control**.

Вид **Valve_control** (рисунок 2.7) используется для управления задвижкой с помощью кнопок **Открыть** и **Закрыть**. Кнопки реализованы как графические примитивы **Объемный прямоугольник** с настроенной анимацией (изменением вида в зависимости от значения указанного атрибута) и реакцией на нажатие левой кнопки мыши. И реакции и анимации используют **аргументы** графических примитивов.

Для задания анимации и реакции графического примитива (в данном случае – **Объемный прямоугольник**) следует:

- 1 Для создания аргумента перетащить в окно **Аргументы** графического примитива необходимые атрибуты тега (рисунок 2.7, комментарий 1)
- 2 Для изменения изображения кнопки (нажата/ отжата) задать в окне **Свойства** графического примитива анимацию **Простое преобразование** (рисунок 2.9, комментарий 2) на свойство **Инверсия границ** аргумента **T3** – текущее значение атрибута **Value** тега **Enable**
- 3 Для установки значения тега **Enable** (команда **Открыть задвижку**) задать в окне **Свойства** графического примитива реакцию **Присвоить значение** (рисунок 2.9, комментарий 3) на событие **Нажатие левой кнопки**.

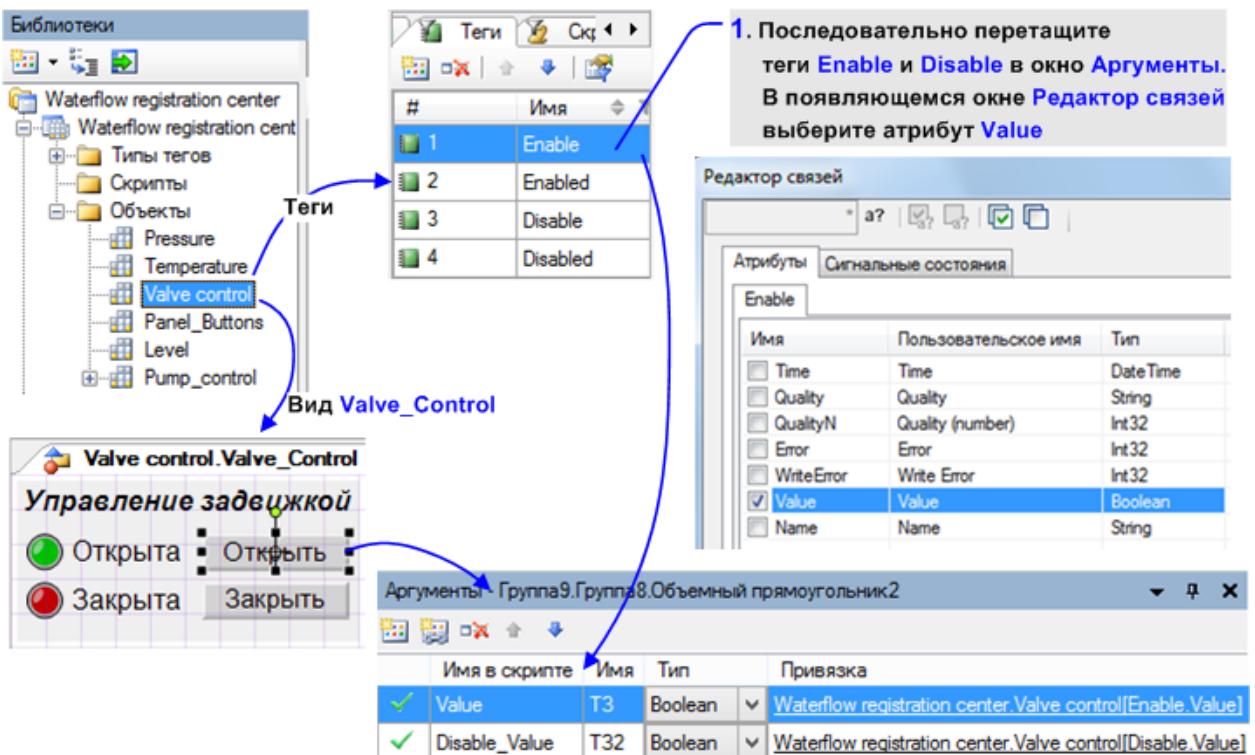


Рисунок 2.7 – Создание аргументов графического примитива **Объемный прямоугольник**

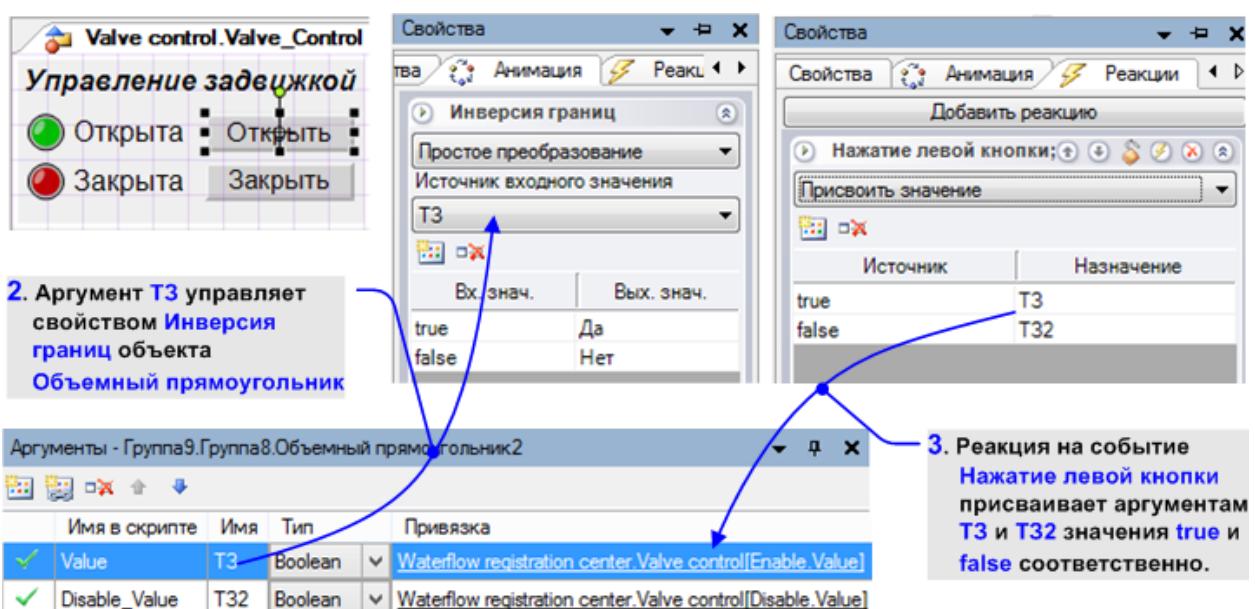


Рисунок 2.8 – Анимация и реакция графического примитива **Объемный прямоугольник**

Таким образом, щелчок левой клавишей мыши по прямоугольнику кнопки **Открыть** присвоит значение **True** атрибуту **T3** тега **Enable**. Так система отреагирует на команду оператора **Открыть задвижку**. При этом кнопка **Открыть** будет изображаться как нажатая.

Аналогично создается и кнопка «**Закрыть**».

2.5 Создание объектов

Самым простым и быстрым способом создания объектов автоматизированной системы является копирование объектов из библиотеки. Для этого следует (рисунок 2.9):

- 1 Выбрать объект в библиотеке элемента проекта
- 2 Перетащить объект в дерево объектов проекта (в данном случае в узел **Registration center**), удерживая нажатой клавишу **Alt**

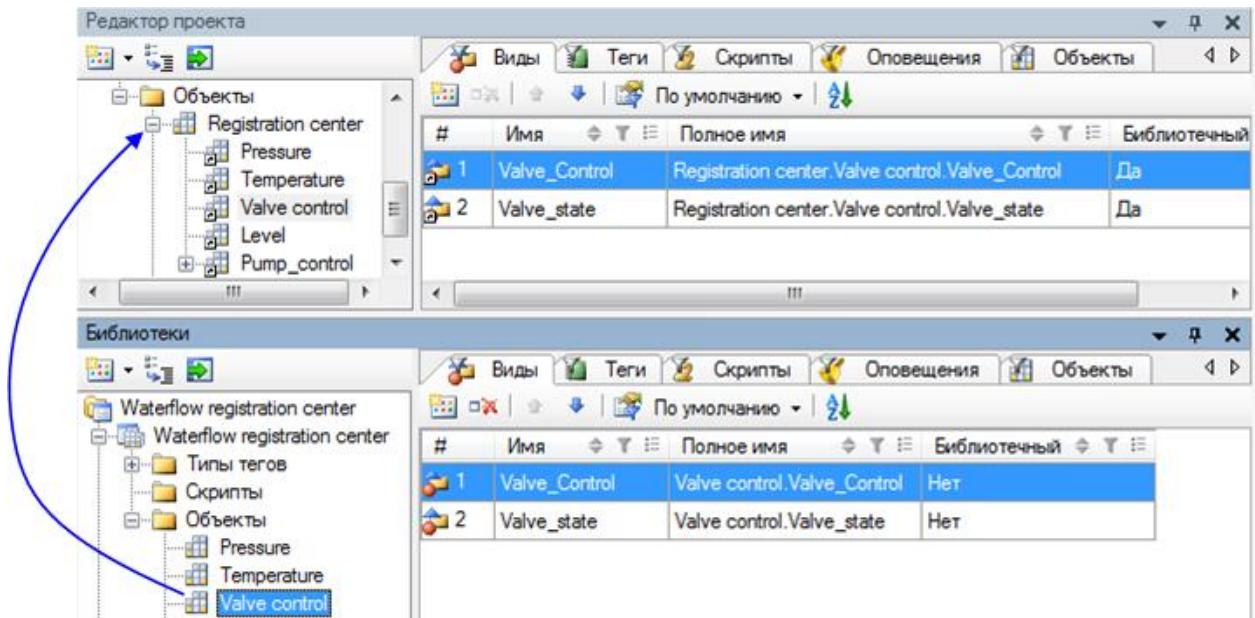


Рисунок 2.9 – Создание объектов из библиотеки

Значение **Да** свойства **Библиотечный** (рисунок 2.9) показывает, что вид данного объекта можно изменять путем изменения этого вида у библиотечного «объекта-родителя».

В проекте **Waterflow registration center** таким способом созданы все «технологические» объекты: **Pressure**, **Temperature**, **Valve Control**, **Level**, **Pump_control**.

Теги созданных объектов – внутренние теги проекта – используются для визуализации соответствующих параметров: давления, температуры, уровня и других.

Анимации и реакции библиотечных объектов настроены на внутренние теги. В примерах на рисунках 2.6 – 2.9 изменение изображения графического примитива и выполнение реакции на событие связаны с аргументом, который привязан к атрибуту **Value** внутреннего тега **Enable** объекта **Valve control**.

Поэтому, после построения графического интерфейса из объектов, созданных на основе библиотечных, следует настроить привязки внутренних тегов объектов проекта к внешним тегам.

2.6 Обмен с внешними источниками данных

В проекте **Waterflow registration center** обмен данными с контролируемым процессом осуществляется по технологии OPC DA (рисунок 2.10).

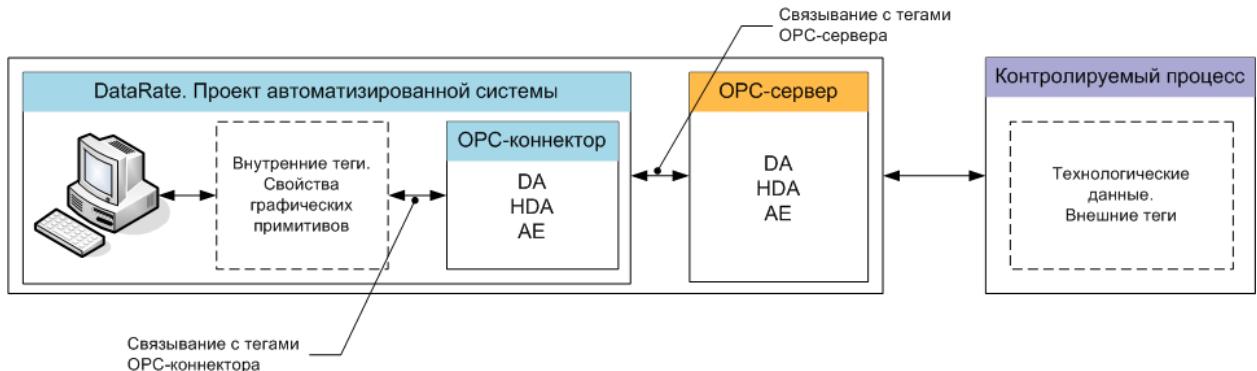


Рисунок 2.10 – Обмен данными по технологии **OPC**

В проекте **Waterflow registration center** источником внешних данных является **учебный OPC DA-сервер – Krug.emUSO.OPCServer.1** – который инсталлируется вместе с демо-проектом **Узел учета воды**.

Функции учебного OPC-сервера:

- Формирует значения температуры, давления, уровня и передает их в проект автоматизированной системы
- Изменяет состояние задвижки по командам «Открыть/Закрыть» и передает состояние задвижки в проект автоматизированной системы
- Изменяет состояние насоса по командам «Включить/Выключить» и передает состояние насоса в проект автоматизированной системы.

2.6.1 Создание OPC DA коннектора

Для создания **OPC DA коннектора** следует:

- 1 В контекстном меню объекта выбрать **Создать/Объект**
- 2 В окне **Добавление нового элемента** (рисунок 2.11) выбрать категорию **Коннекторы**, шаблон **OPC DA коннектор**, задать **Имя** и нажать **OK**
- 3 В появившемся окне **Выберите OPC сервер** выбрать необходимый OPC сервер (в приложении **Waterflow registration center OPC – Krug.emUSO.OPCServer.1**)
- 4 В контекстном меню созданного коннектора выбрать **Добавить/Удалить теги**
- 5 В появившемся окне **Обновление OPC DA тегов** указать теги OPC сервера для обмена данными (рисунок 2.12)

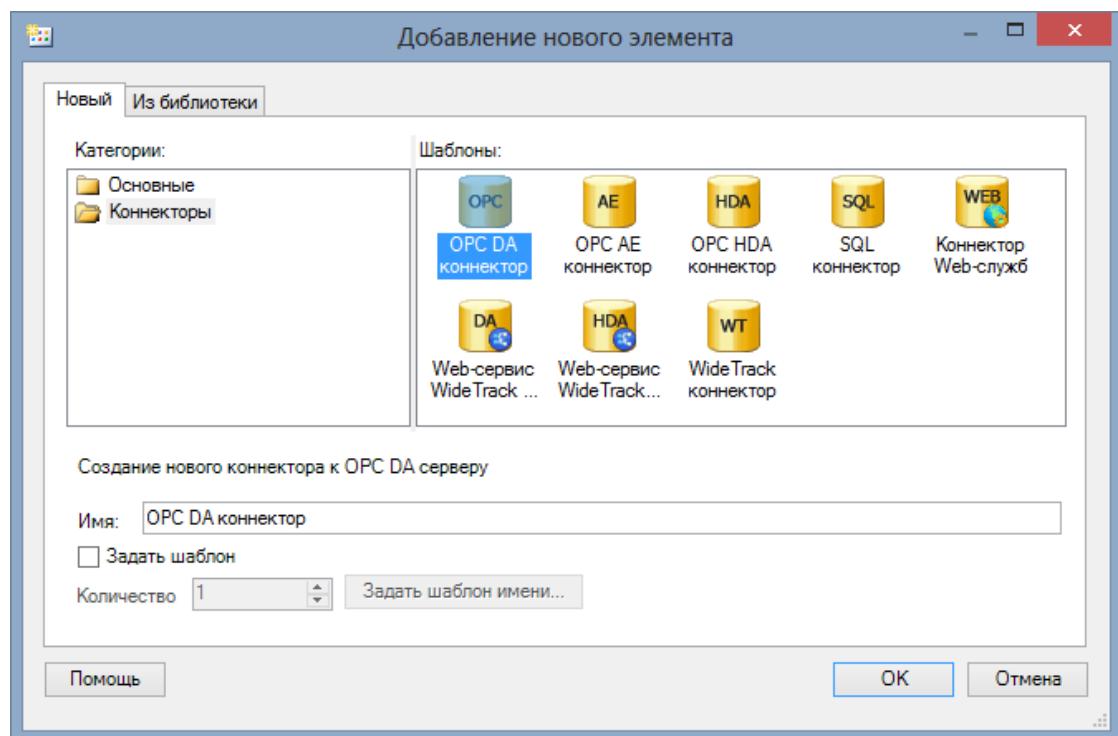


Рисунок 2.11 – Создание OPC DA коннектора

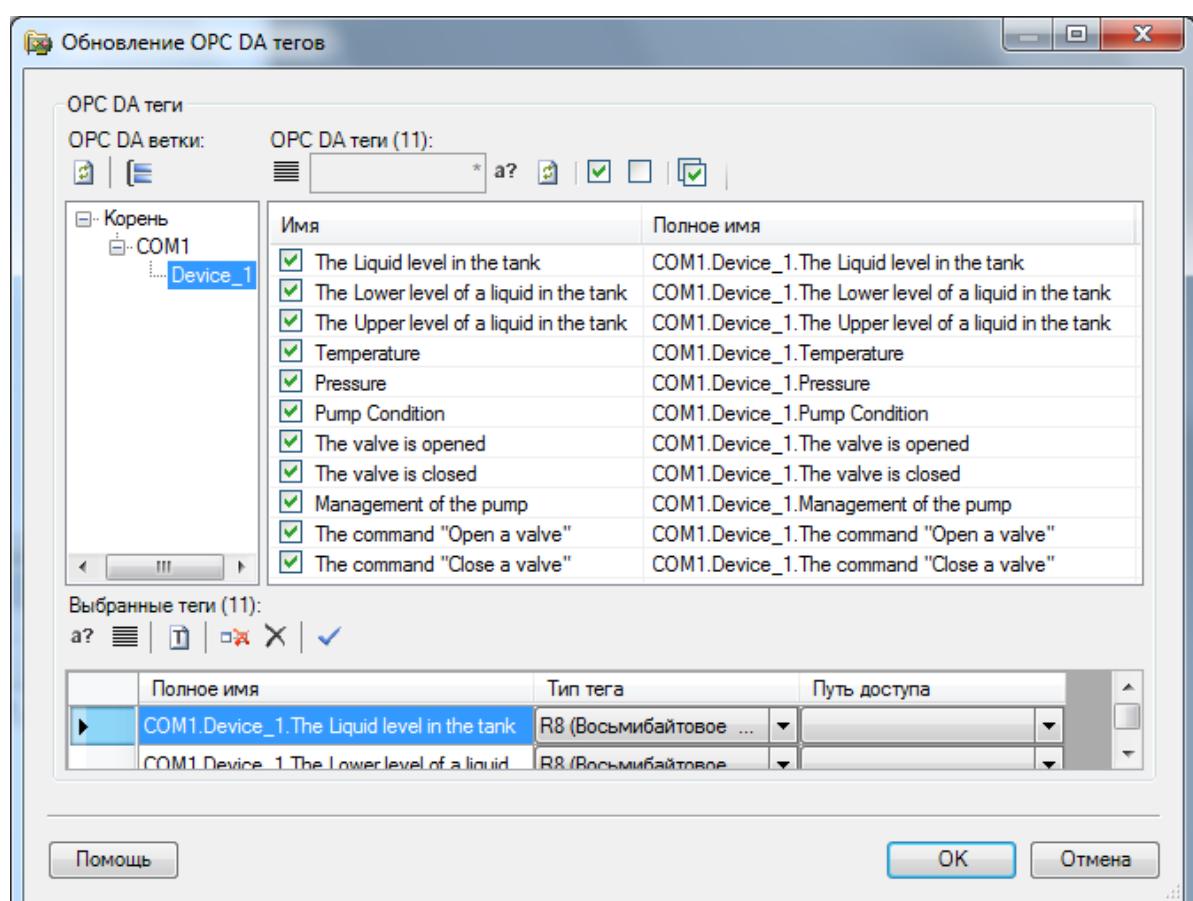


Рисунок 2.12 – Выбор тегов OPC DA сервера для обмена

На рисунке 2.13 показан сформированный список тегов OPC-коннектора, значения которых будет использоваться в работе автоматизированной системы. Тег **System** создается автоматически.

The screenshot shows the 'Редактор проекта' (Project Editor) window. On the left, the project tree for 'Waterflow registration center' is visible, with 'Waterflow registration center' expanded to show 'Протокол событий', 'Расписания', 'Менеджер трендов', 'Менеджер оповещений', 'Менеджер Web-служб', 'Менеджер отчетов', 'Рабочие столы', 'Объекты', 'Registration center', 'OPC DA connector', and 'Экспорт в CSV'. On the right, a table titled 'Теги' (Tags) lists 12 tags:

#	Имя
1	System
2	COM1_Device_1_The Liquid level in the tank
3	COM1_Device_1_The Lower level of a liquid in the tank
4	COM1_Device_1_The Upper level of a liquid in the tank
5	COM1_Device_1_Temperature
6	COM1_Device_1_Pressure
7	COM1_Device_1_Pump Condition
8	COM1_Device_1_The valve is opened
9	COM1_Device_1_The valve is closed
10	COM1_Device_1_Management of the pump
11	COM1_Device_1_The command "Open a valve"
12	COM1_Device_1_The command "Close a valve"

Рисунок 2.13 – Теги OPC DA коннектора

2.6.2 Привязка тегов

После создания и настройки коннектора необходимо привязать к его тегам внутренние теги проекта.

Привязка обеспечивает однозначное соответствие внутренних тегов или свойств графических примитивов проекта внешним тегам.

Настройка привязки тегов выполняется в окне **Свойства** (рисунок 2.14):

- 1 Выберите внутренний тег объекта (на рисунке тег **Enable**)
- 2 В окне **Свойства** выделите его свойство **Связь с другим тегом** и нажмите на кнопку в строке
- 3 В появившемся окне **Выбор тега для создания связи** (рисунок 2.15) выберите тег для связи
- 4 В таблице **Свойство связи** задайте соответствие атрибутов связываемых тегов.

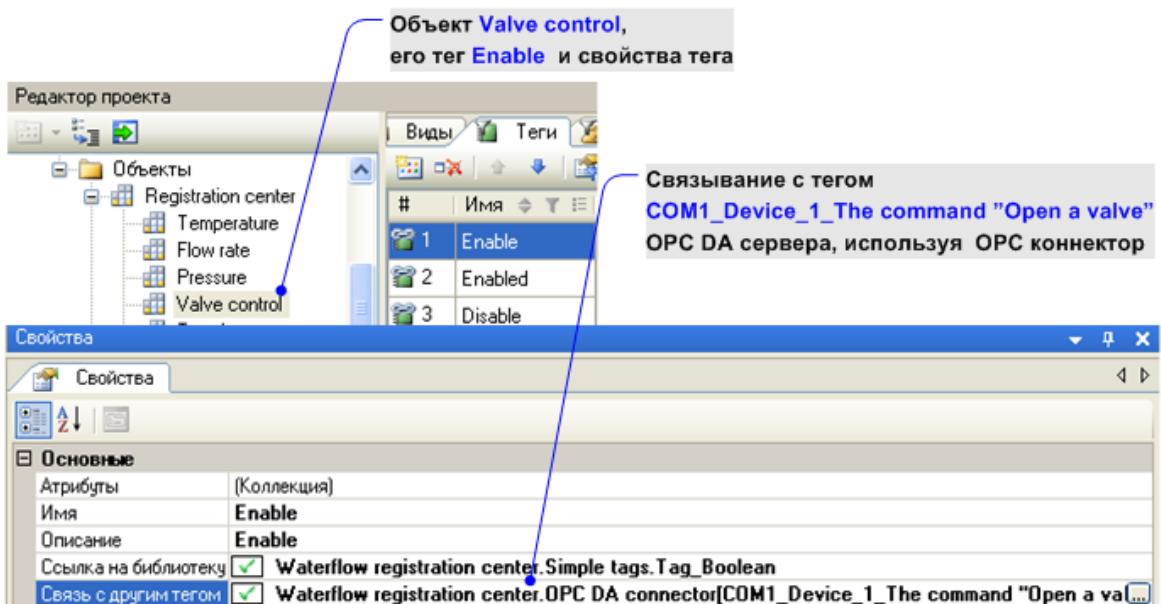


Рисунок 2.14 – Привязка внутреннего тега Valve control. Enable к внешнему тегу

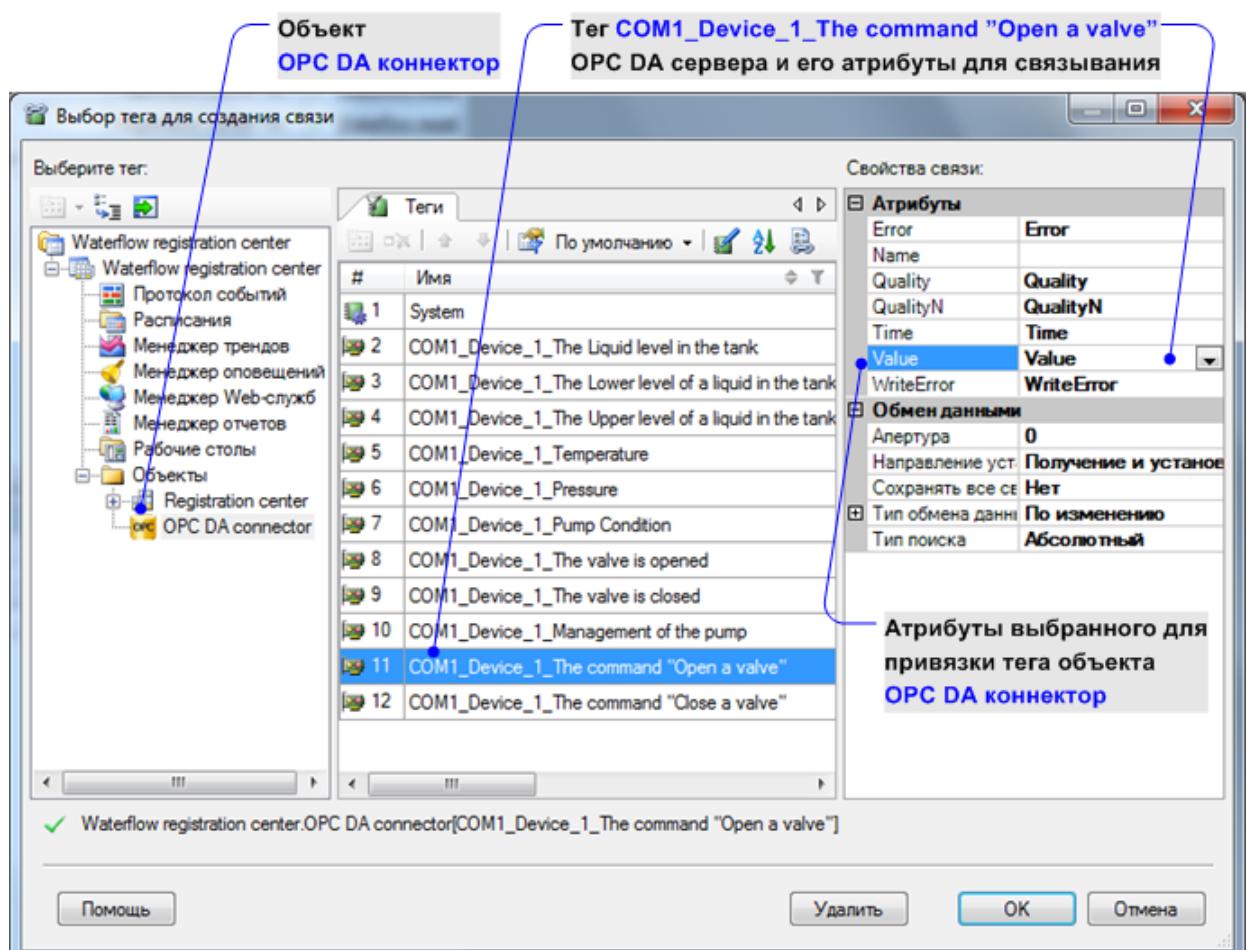


Рисунок 2.15 – Выбор внешнего тега и его атрибутов для привязки

Одноименные атрибуты связываются автоматически!

Свойства **Обмен данными**:

- **Апертура** – определяет минимальную величину изменения значения атрибута связанного тега, в случае равенства или превышения которой изменяется текущее значение связываемого с ним атрибута тега
- **Направление установки значения** – указывает направление обмена данными:
 - **Получение значения** – значения связанных атрибутов будут копироваться в атрибуты тега приемника данных (владельца связи)
 - **Установка значения** – значения связанных атрибутов будут копироваться в атрибуты тега источника данных
 - **Получение и установка значения** – при изменении значений атрибутов тега приемника данных будет происходить установка значений атрибутов тега источника данных (установка значения); при изменении атрибутов тега источника данных значения связанных атрибутов будут копироваться в атрибуты тега приемника данных (получение значений).
- **Сохранять все связанные атрибуты тега** – определяет сохранение значений всех связанных атрибутов тега (**Да**) или сохранение только измененного значения атрибута тега (**Нет**, установлено по умолчанию)
- **Тип обмена данными** – определяет событие, которое приводит к обмену данными:
 - **По изменению** – при любом изменении связанных атрибутов будет происходить обмен данными
 - **По изменению не чаще чем** – обмен данными будет происходить по изменению, но не чаще чем указано в настройке Не чаще чем (мсек), где Вы можете задать частоту обмена в миллисекундах.
- **Тип поиска** – определяет поиск тега источника данных во всем проекте (**Абсолютный**) или поиск относительно положения тега приемника данных (**Относительный**).

Таким образом, изменив привязку внутренних тегов, мы связали графический интерфейс автоматизированной системы с внешними данными. При этом **перенастраивать привязку свойств графических примитивов не надо**, так как они связаны с внутренними тегами.

В проекте **Waterflow registration center** внутренние теги привязаны к тегам объекта **OPC DA коннектор** следующим образом:

Внутренние теги		Теги объекта OPC DA connector
Объект	Тег	
Pressure	Value	COM1_Device_1_Pressure
Temperature	Value	COM1_Device_1_Temperature
Valve Control	Enable	COM1_Device_1_The command "Open a valve"
	Enabled	COM1_Device_1_The valve is opened
	Disable	COM1_Device_1_The command "Close a valve"
	Disabled	COM1_Device_1_The valve is closed
Level	Param	COM1_Device_1_The Liquid level in the tank
Pump_control	Control	COM1_Device_1_Management of the pump
	Pump_State	COM1_Device_1_Pump Condition

2.7 Создание мнемосхем

Для создания изображения схемы автоматизированного процесса следует:

- 1 Создать мнемосхему объекта (на рисунке 2.16 – **Main**, объект **Registration center**) и установить на ней изображения из библиотеки **Image Library** (насос, датчики, бак, задвижку и другие)

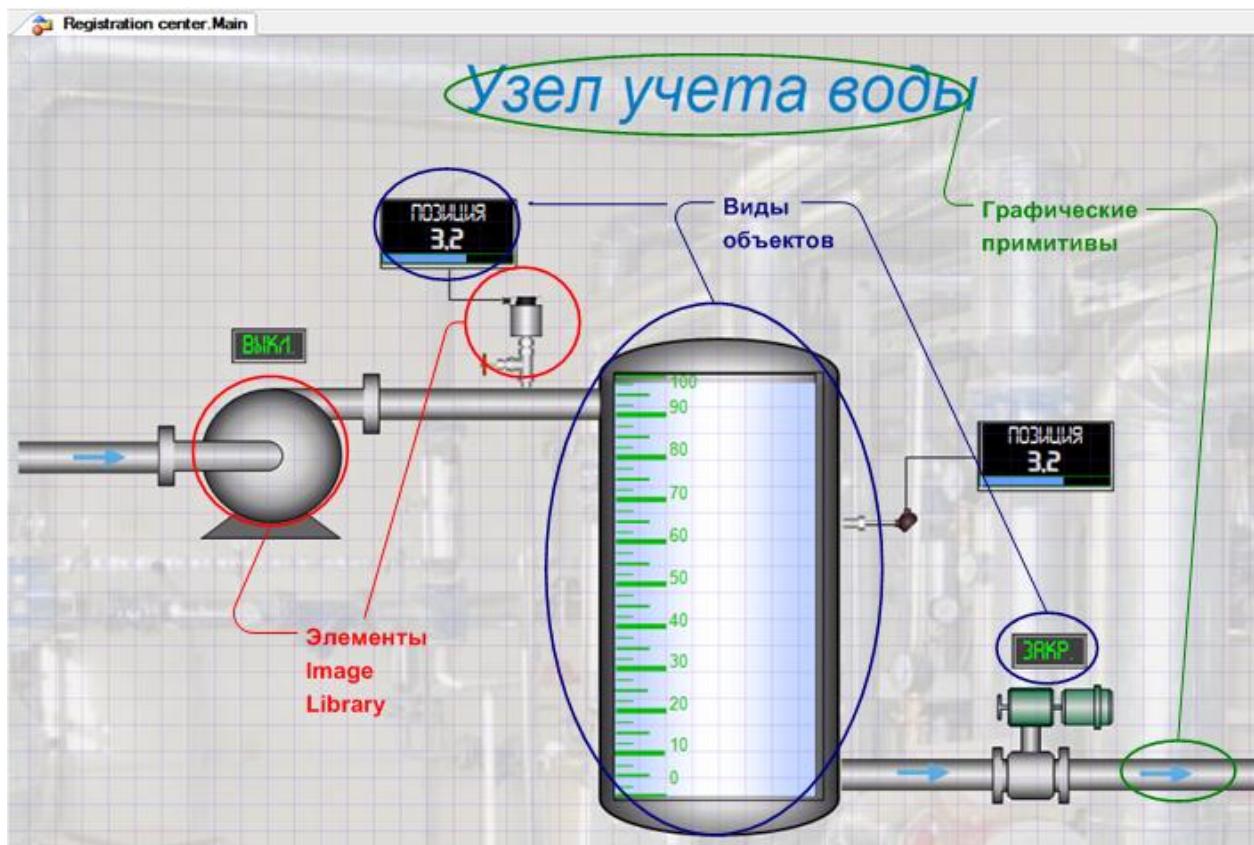


Рисунок 2.16 – Мнемосхема **Main**

- 2 Разместить виды объектов проекта. Для размещения на мнемосхеме достаточно перетащить название вида объекта (из закладки **Виды**) в область редактирования и отладки главного окна **Интегрированной среды разработки DataRate**
- 3 Добавить необходимые примитивы графической подсистемы **DataRate**: текст, линии (стрелки) и другие.

Просмотреть структуру мнемосхемы можно в окне **Дерево примитивов мнемосхемы**.

Мнемосхемы (виды) объекта **Registration center**:

- **Main** — главное окно приложения
- **Panel.Buttons** — панель управления
- **AnalyticalTrend** — аналитический тренд
- **Report** — отчет об изменении температуры, давления и уровня в емкости

2.8 Тестирование графического интерфейса

После привязки тегов объектов **Temperature**, **Pressure**, **Valve Control**, **Level**, **Pump_control** к тегам объекта **OPC DA** коннектор можно выполнить тестирование мнемосхемы **Main** – главного окна приложения.

Для тестирования следует:

- 1 Откройте вид **Main** объекта **Registration center**
- 2 Щелкните по кнопке  в панели управления
- 3 Открытие мнемосхемы сопровождается появлением окна сообщения о подключении OPC-коннектора к OPC-серверу
- 4 После запуска автоматически изменяются уровень и температура воды в емкости, давление в трубопроводе. Состояние насоса и задвижки отображаются на соответствующих табло (рисунок 2.17)..

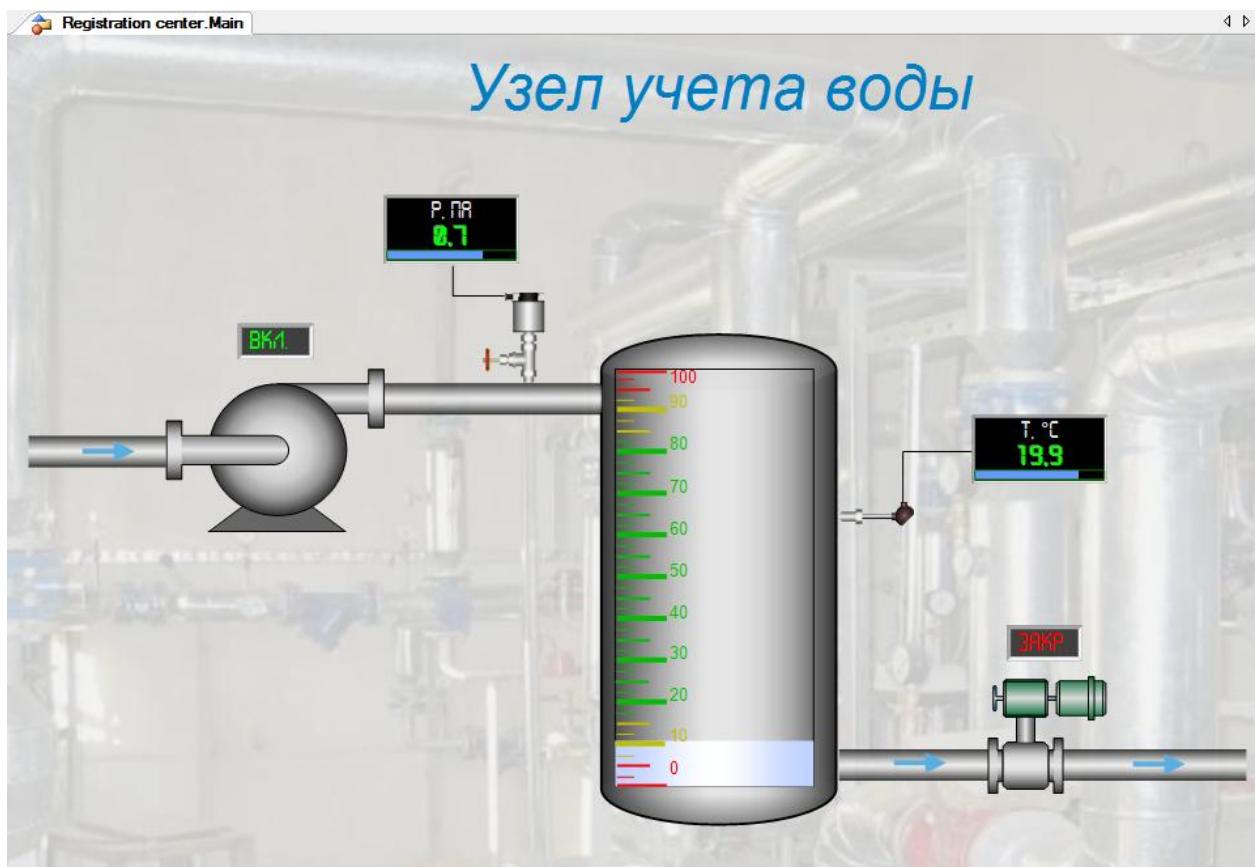


Рисунок 2.17 – Проект Waterflow registration center. Мнемосхема Registration center.Main

- 5 Для управления насосом и задвижкой щелкните по их изображениям на мнемосхеме
- 6 Появившиеся «приборы управления» (**Pump_control**, **Valve_Control** – рисунок 2.19) используйте для ручного управления состоянием насоса (переключатель 1/0) и задвижки (кнопки **Открыть/Закрыть**).

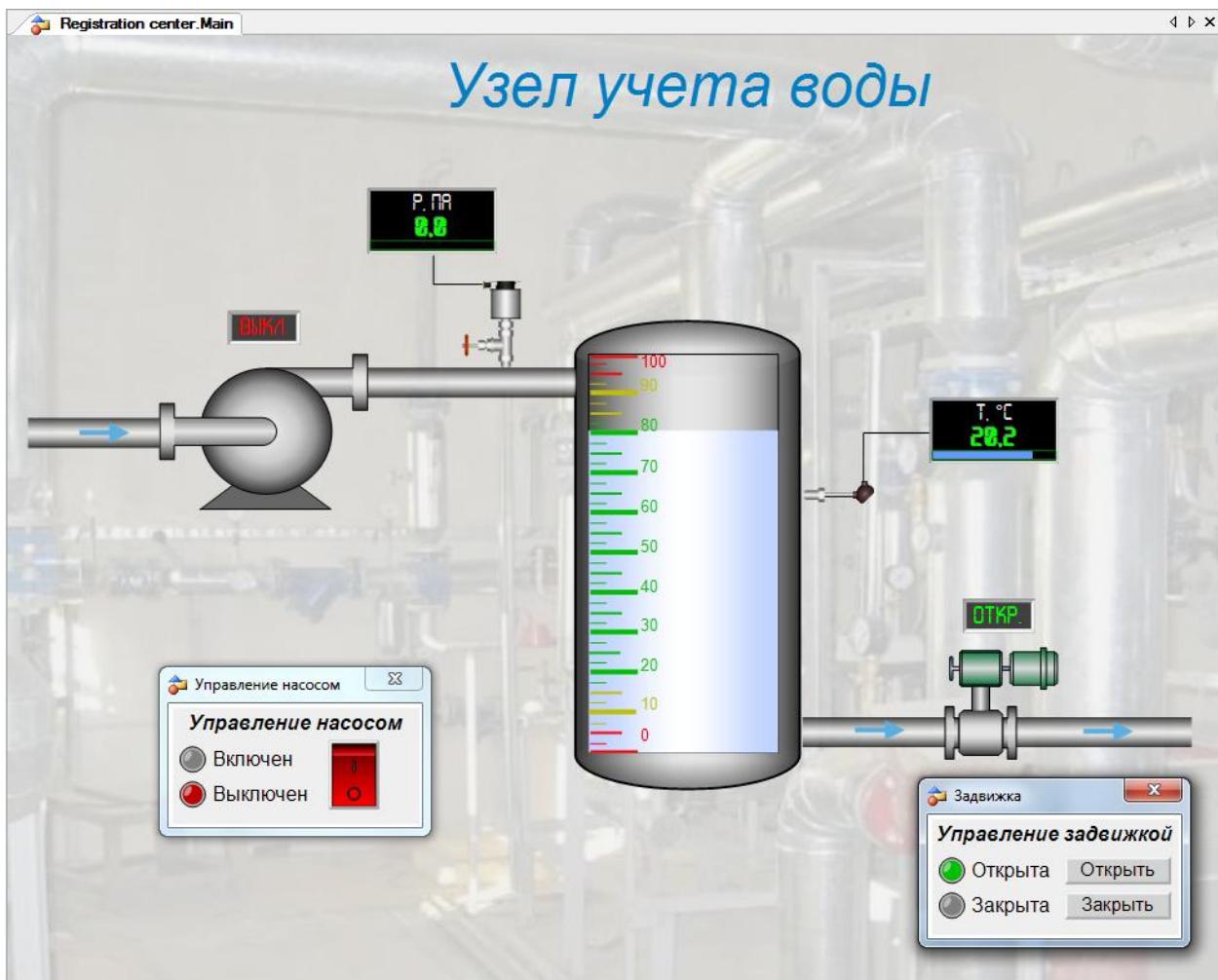


Рисунок 2.18 – Виртуальные «приборы управления» на мнемосхеме технологического процесса

Изменение состояния задвижки сопровождается миганием соответствующей лампочки виртуального прибора.

В случае достижения аварийных границ уровня воды в емкости происходит автоматическое изменение состояний насоса и задвижки.

Для завершения тестирования нажмите на кнопку в панели управления среды разработки.

2.9 События, сигнализация и оповещение

2.9.1 Протокол событий. Создание и визуализация

Для формирования и отображения протокола событий **DataRate** предоставляет:

- **Системный сервис Протокол событий** – конфигурирование, отображение и формирование истории событий (архива)
- **Графический примитив Протокол событий** – настройка и отображение протокола
- **Библиотечный скрипт Event Log** – посылка сообщений в протокол событий

- **Словарь групп событий** – описания групп, события которых соответствуют возможным состояниям автоматизируемого процесса
- **Словарь серьезности** – определяет важности событий
- **Словарь категорий** – определяет основные типы (категории) событий
- **Словарь событий** – описания возможных событий в системе

Для регистрации событий в протоколе событий используются функции посылки сообщений. Эти функции вызываются в скриптах, определенных в описании объектов автоматизированной системы или подключаемых в реакциях графических примитивов.

На рисунке 2.19 показано окно аргументов скрипта **SendMessage_Open** объекта **Valve Control** (задвижка).

Аргументы - Registration center.Valve control.SendMessage_Open					
	Имя в скрипте	Имя	Тип	Запускать скрипт	Привязка
✓	Enabled_Value	Value	Boolean	<input checked="" type="checkbox"/>	Waterflow registration center.Registration center.Valve control[Enabled.Value]
✓	Enable_Name	Name	String	<input type="checkbox"/>	Waterflow registration center.Registration center.Valve control[Enable.Name]

Рисунок 2.19 – Привязка аргумента скрипта к атрибуту тега

Текст скрипта **SendMessage_Open** и взаимосвязь параметров функции регистрации события (**SendSimpleEvent**) со словарями описания событий и отображением протокола событий показана на рисунке 2.20.

Категория событий (2) в вызове функции **SendSimpleEvent** определяет цвет текста и цвет фона отображения сообщения в протоколе событий.

Скрипт запускается только при изменении значения атрибута тега Enabled_Value, к которому привязан аргумент скрипта!

Таким образом, при переходе задвижки (атрибут **Value** внешнего тега **COM1_Device_1_The valve is opened**) из состояния «закрыто» в состояние «открыто» (**COM1_Device_1_The valve is opened.Value = true**) будет сформировано сообщение с параметрами (рисунок 2.20):

- группы событий – 4 («Информационные»)
- серьезность – 500
- текст – «Задвижка открыта».

Для формирования протокола событий разработчик HMI-приложения должен:

- 1 Сформировать словари групп, серьезности, категорий и событий (или использовать системные словари **DataRate**)
- 2 Создать скрипты (на основе системной библиотеки) для посылки сообщений в протокол событий
- 3 Привязать аргументы скриптов к изменяемым атрибутам тегов.

Waterflow registration center.Протокол событий

1. Протокол событий

Время	Качество	Серьезность	Название серьезности	Сообщение	Источник
28.11.2018 14:08:59...	Хорошее	500	Очень низкая	Переменная V, м3 = 83,...	Объем воды в баке
28.11.2018 14:08:58...	Хорошее	500	Системная	Задвижка открыта	Задвижка № 1
28.11.2018 14:08:56...	Хорошее	1000	Предаварийная	Переменная V, м3 = 85,...	Объем воды в баке
28.11.2018 14:07:48...	Хорошее	500	Системная	Задвижка закрыта	Задвижка № 1
28.11.2018 14:07:47...	Хорошее	500	Очень низкая	Переменная V, м3 = 16,...	Объем воды в баке
28.11.2018 14:07:42...	Хорошее	1000	Предаварийная	Переменная V, м3 = 15,...	Объем воды в баке
28.11.2018 14:07:31...	Хорошее	500	Системная	Задвижка открыта	Задвижка № 1
28.11.2018 14:07:19...	Хорошее	500	Очень низкая	Переменная V, м3 = 16,...	Объем воды в баке
28.11.2018 14:07:16...	Хорошее	500	Очень низкая	Переменная T, °C = 19.9...	Температура
28.11.2018 14:07:15...	Хорошее	750	Высокая	Переменная T, °C = 20.5...	Температура
28.11.2018 14:07:14...	Хорошее	500	Очень низкая	Переменная T, °C = 19.8...	Температура

Фильтрация выключена. | Событий: 1188 | Интервал времени: 01.01.0001 3:00:00 - текущее время

Редактор проекта

Valve control.SendMessage_Open

```

if(Enabled_Value)
{
    AlarmServiceHelper.SendSimpleEvent(
        // Отправитель события
        Client,
        // Объект, который сгенерировал уведомление о событии
        Enable_Name,
        // Категория события
        2,
        // Серьезность
        500,
        // Группа событий
        4,
        // Сообщение
        "Задвижка открыта"
    );
}

```

Настройки протокола событий

Номер	Название категории	Цвет текста	Цвет фона
1	Системное		
2	Информационный		
3	Настройка		
4	Диагностика		
5	Предаварийный		
6	Предупреждение		
7	Норма		
8	Недостоверные		
9	Пользовательские		

Рисунок 2.20 – Посылка сообщений в протокол событий

В проекте **Waterflow registration center** протокол сообщений формируется в следующих случаях:

- **Основные события.** Изменилось состояние задвижки. Реализуется с помощью скриптов объекта **Valve_Control**:
 - **SendMessage_Open**
 - **SendMessage_Closed**
- **Сигнализация (тревога/alarm).** Значение давления, температуры или уровня вышло за **предупредительную** или **предаварийную границу**. Реализуется с помощью **сигнализации** тега **Param** объектов **Pressure**, **Temperature** и **Level**.

Просмотр протокола событий

В проекте **Waterflow registration center** для отображения протокола событий на мнемосхеме **Panel-Buttons** используется **графический примитив Протокол событий**, который настроен на отображение **времени события, объекта вызвавшего сообщение и текста сообщения**. Фрагмент протокола событий на мнемосхеме **Panel-Buttons** приведен на рисунке 2.21.

28.11.2018 16:29:45.299	Температура воды в баке	Переменная T, °C = 19.84761. Возврат в норму
28.11.2018 16:29:44.298	Температура воды в баке	Переменная T, °C = 20.57334. Верхняя аварийная граница
28.11.2018 16:29:42.312	Температура воды в баке	Переменная T, °C = 19.80978. Возврат в норму
28.11.2018 16:29:41.307	Объем воды в баке	Переменная V, м³ = 6. Нижняя предупредительная граница
28.11.2018 16:29:41.307	Температура воды в баке	Переменная T, °C = 20.61174. Верхняя аварийная граница

Рисунок 2.21 – Примитив Протокол событий в среде разработки

Для настройки отображения примитива **Протокол событий** используется свойство **Форматирование** (рисунок 2.22).

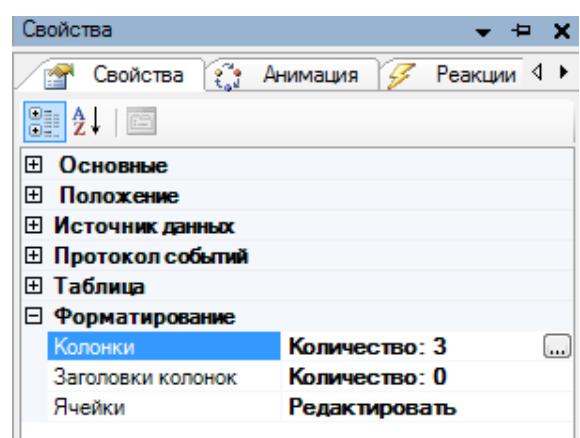


Рисунок 2.22 – Форматирование колонок протокола событий

Для отображения протокола событий в **отдельном окне** используется реакция **Окно протокола событий** на событие **Нажатие левой кнопки** графического примитива **Протокол событий**. Для реализации такого вызова протокола событий следует (рисунок 2.23):

- 1 Разместить на мнемосхеме графический примитив **Протокол событий** (иконка на панели инструментов)
- 2 Назначить для него реакцию **Окно протокола событий**.

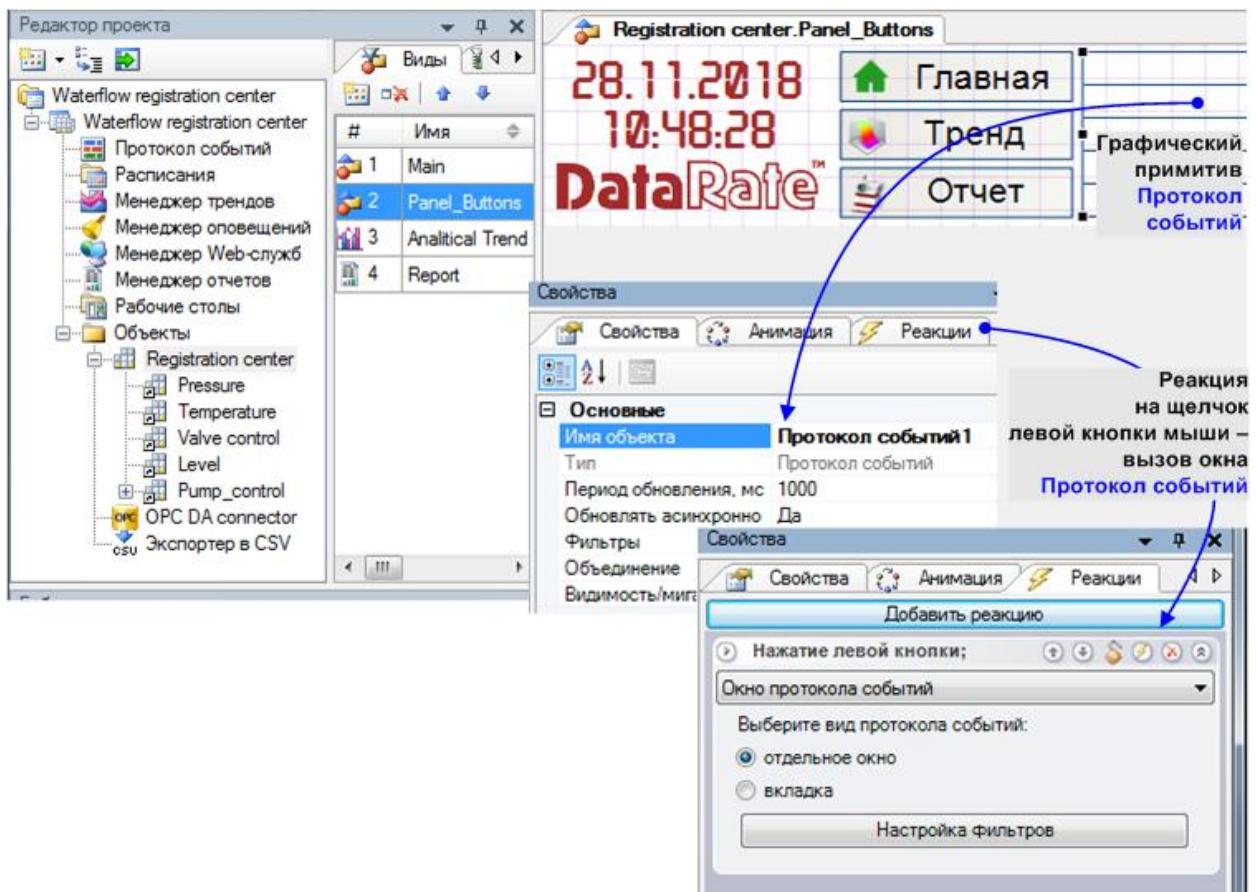


Рисунок 2.23 – Использование примитива **Протокол событий** и его реакции **Окно протокола событий**

2.9.2 Сигнализация

Сигнализация (тревога, alarm) – аварийное состояние, т.е. специальный случай какого-либо состояния, требующего особого внимания.

Сигнализация **DataRate** обеспечивает автоматическое слежение за состояниями объектов проекта, формирование сообщений в протоколе событий и вывод окон с активной сигнализацией для квитирования.

В проекте **DataRate** работа с аварийными состояниями основана на использовании тегов, тип которых включает определенную сигнализацию.

Для включения сигнализации в проект следует:

- 1 Выбрать из библиотеки или создать новый библиотечный тип тега для сигнализации
- 2 Выбрать тип сигнализации
- 3 Если необходимо, настроить словари протокола событий (сигнальное сообщение в протокол, цвета этого сообщения, серьезность) и отредактировать параметры и обработку выбранной сигнализации
- 4 Создать теги объектов проекта на основе сформированных библиотечных типов тегов.

Далее работа с такими тегами ничем не отличается от работы с «обычными» тегами объектов, например, их можно привязать к внешним тегам источников данных или использовать для анимации графических примитивов

5 Настроить оповещение о достижении сигнального состояния.

В проекте **Waterflow registration center** сигнализация используется для слежения за аварийными и предаварийными границами технологических параметров: температуры, давления, уровня.

Для создания и обработки сигнализации используется библиотечный тип тега – **ТЕГ_Single_with_Alarm_Abs**. Атрибуты тега приведены на рисунке 2.24.

The screenshot shows the Tag Library interface with the following details:

- Библиотеки (Libraries):**
 - Waterflow registration center (selected)
 - Системная библиотека
- Типы тегов (Tag Types):**
 - Waterflow registration ce (selected)
 - Скрипты (Scripts)
 - Объекты (Objects)
 - Pressure
 - Temperature
 - Valve control
 - Panel.Buttons
 - Level
 - Pump.control
- Атрибуты (Attributes) for TEG_Single_with_Alarm_Abs:**

#	Имя (Name)	Пользовательское имя (User Name)	Тип (Type)	Размер (для строк) (Size for strings)	Только для чтения (Read-only)	Значение (Value)
1	Time	Время	DateTime		<input checked="" type="checkbox"/>	01.01.000...
2	Quality	Качество	String	256	<input checked="" type="checkbox"/>	0
3	QualityN	Качество (номер)	Int32		<input checked="" type="checkbox"/>	0
4	Error	Ошибка	Int32		<input checked="" type="checkbox"/>	0
5	WriteError	Ошибка записи	Int32		<input checked="" type="checkbox"/>	0
6	Value	Значение	Single		<input checked="" type="checkbox"/>	0
7	Name	Имя	String	256	<input checked="" type="checkbox"/>	0
...	<input checked="" type="checkbox"/>	...
15	UserName	Польз. имя	String	256	<input checked="" type="checkbox"/>	Пользов...
16	HiHi	ВАГ	Single		<input checked="" type="checkbox"/>	0
17	Hi	ВПГ	Single		<input checked="" type="checkbox"/>	0
18	Low	НПГ	Single		<input checked="" type="checkbox"/>	0
19	LowLow	НАГ	Single		<input checked="" type="checkbox"/>	0
20	Alarm	Признак аварии	Byte		<input checked="" type="checkbox"/>	0
21	IsQuit	Сквитирован ли ала...	Boolean		<input checked="" type="checkbox"/>	False
22	Description	Описание параметра	String	256	<input checked="" type="checkbox"/>	Описани...
23	MeasureUnit	Атрибут	String	256	<input checked="" type="checkbox"/>	Единица ...
24	AbsValue	Абсолютное значение	Single		<input checked="" type="checkbox"/>	0
25	Start_scale	Начало шкалы	Double		<input checked="" type="checkbox"/>	0
26	End_scale	Конец шкалы	Double		<input checked="" type="checkbox"/>	100

Рисунок 2.24 – Атрибуты типа тега **ТЕГ_Single_with_Alarm_Abs**

Атрибуты тега включают следующие границы сигнализации: **ВАГ** – верхняя предаварийная граница, **НАГ** – нижняя предаварийная граница, **ВПГ** – верхняя предупредительная граница, **НПГ** – нижняя предупредительная граница. Кроме этого в тип тега добавлены атрибуты верхней и нижней границ шкалы изменения контролируемого параметра – **Начало шкалы/Конец шкалы**.

Обработка типа тега **ТЕГ_Single_with_Alarm_Abs** (рисунок 2.25) вычисляет значение атрибута **AbsValue** (пользовательское имя – **Абсолютное значение**) как абсолютную величину атрибута **Value**.

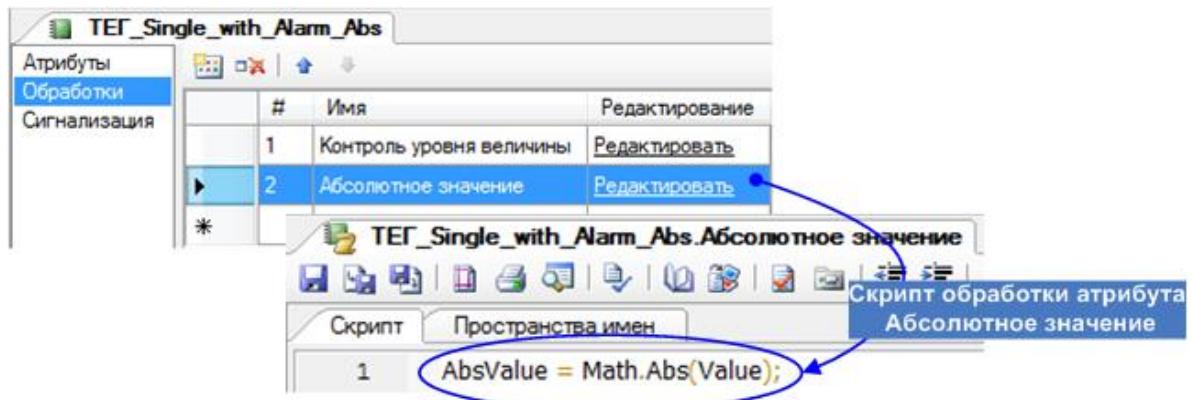


Рисунок 2.25 – Обработка атрибута **Абсолютное значение**

Для типа тега **ТЕГ_Single_with_Alarm_Abs** назначена сигнализация **Контроль уровня** (рисунок 2.26).

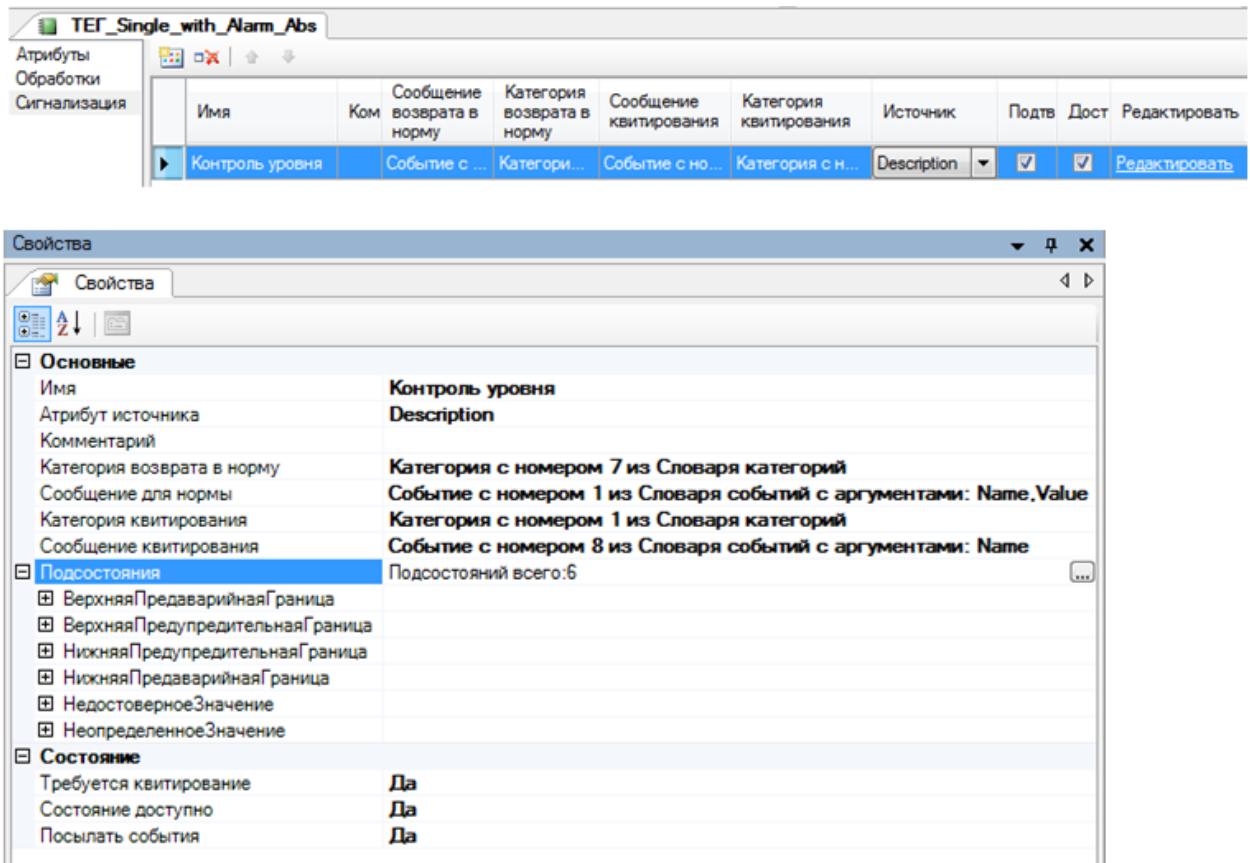


Рисунок 2.26 – Сигнализация **Контроль уровня** типа тега **ТЕГ_Single_with_Alarm_Abs**

Данная сигнализация обеспечивает сравнение **значения выбранного атрибута тега** (по умолчанию **ТЗ** – текущее значение) с **заданными границами**. Если значение этого атрибута превышает границу, то устанавливается определенное подсостояние сигнализации.

Параметры сигнализации **Контроль уровня** для типа тега **ТЕГ_Single_with_Alarm_Abs** приведены на рисунке 2.27.

The screenshot shows the Waterflow registration center interface with the following details:

- Top Bar:** Shows the project name "ТЕГ_Single_with_Alarm_Abs" and the tag type "Контроль уровня".
- Header:** "Подсостояния сигнализации:"
- Table:** A table listing six signalization states:

#	Имя	Серьезность	Категория	Описание	Доступно
1	ВерхняяПредаварийнаяГраница	1000	Категория с номером 5 из Словаря категорий	Событие с номером 4 из Словаря событий с аргументами: Name,Value	<input checked="" type="checkbox"/>
2	ВерхняяПредупредительнаяГраница	750	Категория с номером 6 из Словаря категорий	Событие с номером 2 из Словаря событий с аргументами: Name,Value	<input checked="" type="checkbox"/>
3	НижняяПредупредительнаяГраница	750	Категория с номером 6 из Словаря категорий	Событие с номером 3 из Словаря событий с аргументами: Name,Value	<input checked="" type="checkbox"/>
4	НижняяПредаварийнаяГраница	1000	Категория с номером 5 из Словаря категорий	Событие с номером 5 из Словаря событий с аргументами: Name,Value	<input checked="" type="checkbox"/>
5	НедостоверноеЗначение	500	Категория с номером 8 из Словаря категорий	Событие с номером 6 из Словаря событий с аргументами: Name	<input checked="" type="checkbox"/>
6	НеопределенноеЗначение	500	Категория с номером 8 из Словаря категорий	Событие с номером 7 из Словаря событий с аргументами: Name	<input checked="" type="checkbox"/>
- Section:** "Обработки сигнализации:"
- Table:** A table listing one signalization processing rule:

#	Имя	Редактировать
1	Обработка сигнализации	Редактировать

Рисунок 2.27 – Параметры сигнализации **Контроль уровня** типа тега **ТЕГ_Single_with_Alarm_Abs**

В таблице **Подсостояния сигнализации** (рисунок 2.27) могут быть настроены следующие параметры:

- **Имя** – имя подсостояния, уникальное в пределах сигнализации
- **Серьезность** – важность события, связанного с данным подсостоянием
- **Категория** – тип события (может выбираться из Словаря категорий), связанного с данным подсостоянием
- **Описание** – строка сообщения, автоматически посылаемая в протокол событий при установке данного подсостояния
- **Доступно** – свойство, определяющее автоматическую обработку данного подсостояния
- **Выражение** – выражение, которое определяет условие установки данного подсостояния.

Алгоритм обработки сигнализации **Контроль уровня** реализуется скриптом **Обработка сигнализации**, текст которого (рисунок 2.28) вызывается по ссылке **Редактировать**.

Для проекта **Waterflow registration center** наименования по умолчанию в скрипте **Контроль уровня. Обработка сигнализации** (рисунок 2.28) были отредактированы в соответствии с наименованиями атрибутов типа тега **ТЕГ_Single_with_Alarm_Abs** (рисунок 2.24). Например, в качестве **Контролируемого значения** (строка 1 в тексте скрипта), которое сравнивается с границами сигнализации, используется атрибут **AbsValue** (номер 24 в таблице на рисунке 2.24), а для анализа качества значения атрибута тега (что особенно важно для ОРС-тегов) используется атрибут **QualityN** (номер 3 в таблице на рисунке 2.24).

```

1 double КонтролируемоеЗначение = AbsValue;
2
3 if (OPCQualityUtilities.IsGoodQuality(QualityN))
4 {
5     if (HiHi <= КонтролируемоеЗначение && HiHi!=0)
6     {
7         Condition.SetActiveSubCondition(ВерхняяПредаварийнаяГраница);
8     }
9     else if (Hi <= КонтролируемоеЗначение && КонтролируемоеЗначение < HiHi && Hi!= 0)
10    {
11        Condition.SetActiveSubCondition(ВерхняяПредупредительнаяГраница);
12    }
13    else if (LowLow < КонтролируемоеЗначение && КонтролируемоеЗначение <= Low && Low!=0)
14    {
15        Condition.SetActiveSubCondition(НижняяПредупредительнаяГраница);
16    }
17    else if (КонтролируемоеЗначение <= LowLow && LowLow!=0)
18    {
19        Condition.SetActiveSubCondition(НижняяПредаварийнаяГраница);
20    }
21    else
22    {
23        Condition.SetActiveSubCondition(Норма);
24    }
25
26 }
27 else
28 {
29     if (OPCQualityUtilities.IsUncertainQuality(QualityN)) Condition.SetActiveSubCondition(Uncertain);
30     if (OPCQualityUtilities.IsBadQuality(QualityN)) Condition.SetActiveSubCondition(BAD);
31 }

```

Рисунок 2.28 – Скрипт обработки сигнализации **Контроль уровня**

В проекте **Waterflow registration center** тегами, которые созданы на основе типа **ТЕГ_Single_with_Alarm_Abs**, являются теги **Param** (пример на рисунке 2.29) объектов **Pressure**, **Temperature**, **Level**.

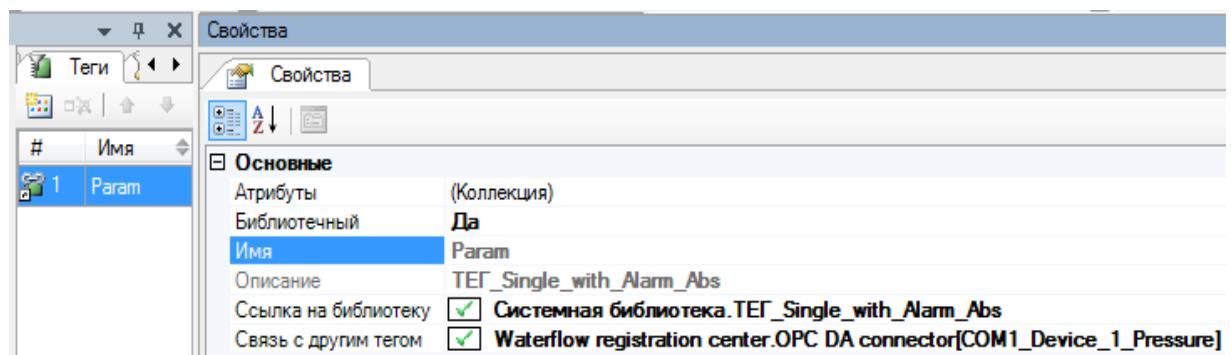


Рисунок 2.29 – Тег **Value** объекта **Pressure**

Конкретные значения границ сигнализации можно установить с помощью скрипта, который вызывается при старте HMI-приложения. Пример инициализации границ сигнализации **Контроль уровня** для объекта **Level** показан на рисунке 2.30.

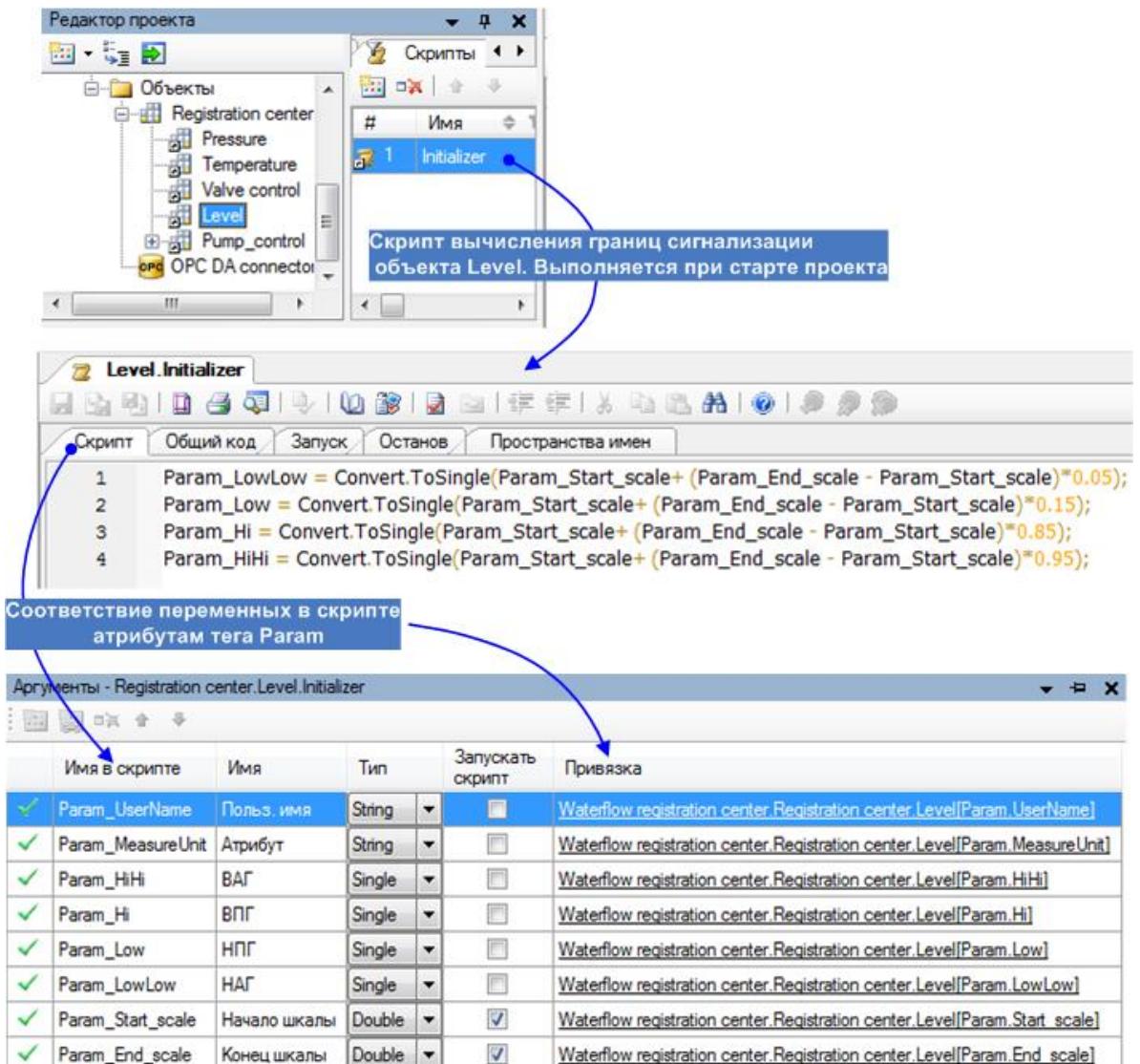


Рисунок 2.30 – Вычисление границ сигнализации

Из привязки переменных скрипта (рисунок 2.30) видно, что границы сигнализации (НАГ, ВАГ, НПГ, ВПГ) рассчитываются на основе значений нижней и верхней границ шкалы (**Start_Scale**, **End_Scale**). Это позволяет автоматически корректировать границы сигнализации при смене шкалы (в данном проекте – шкалы уровня жидкости в емкости). Начальные значения шкалы задаются в атрибутах тега **Param**.

Таким образом, создав тег **Param** на основе типа **ТЕГ_Single_with_Alarm_Abs** и «привязав» данный тег по атрибуту **Value** к тегу OPC-коннектора (на рисунке 2.29 – Связь с другим тегом), мы фактически получили обработку сигнализации **Контроль уровня** по значению внешнего технологического параметра, полученного от OPC-сервера.

Следовательно, все объекты, содержащие тег **Param**, будут обладать сигнализацией **Контроль уровня** с автоматической посылкой сообщений в протокол событий в случае установки подсостояний сигнализации, определенных типом **ТЕГ_Single_with_Alarm_Abs**.

2.9.3 Оповещения

Оповещение **Звук** для объекта **Registration center** и привязка проигрывания звукового файла к сигнальному состоянию объекта (на примере объекта **Pressure**) показано на рисунках 2.31, 2.32.

Аналогично осуществляется привязка проигрывания звукового файла к сигнальному состоянию объектов **Temperature** и **Level**.

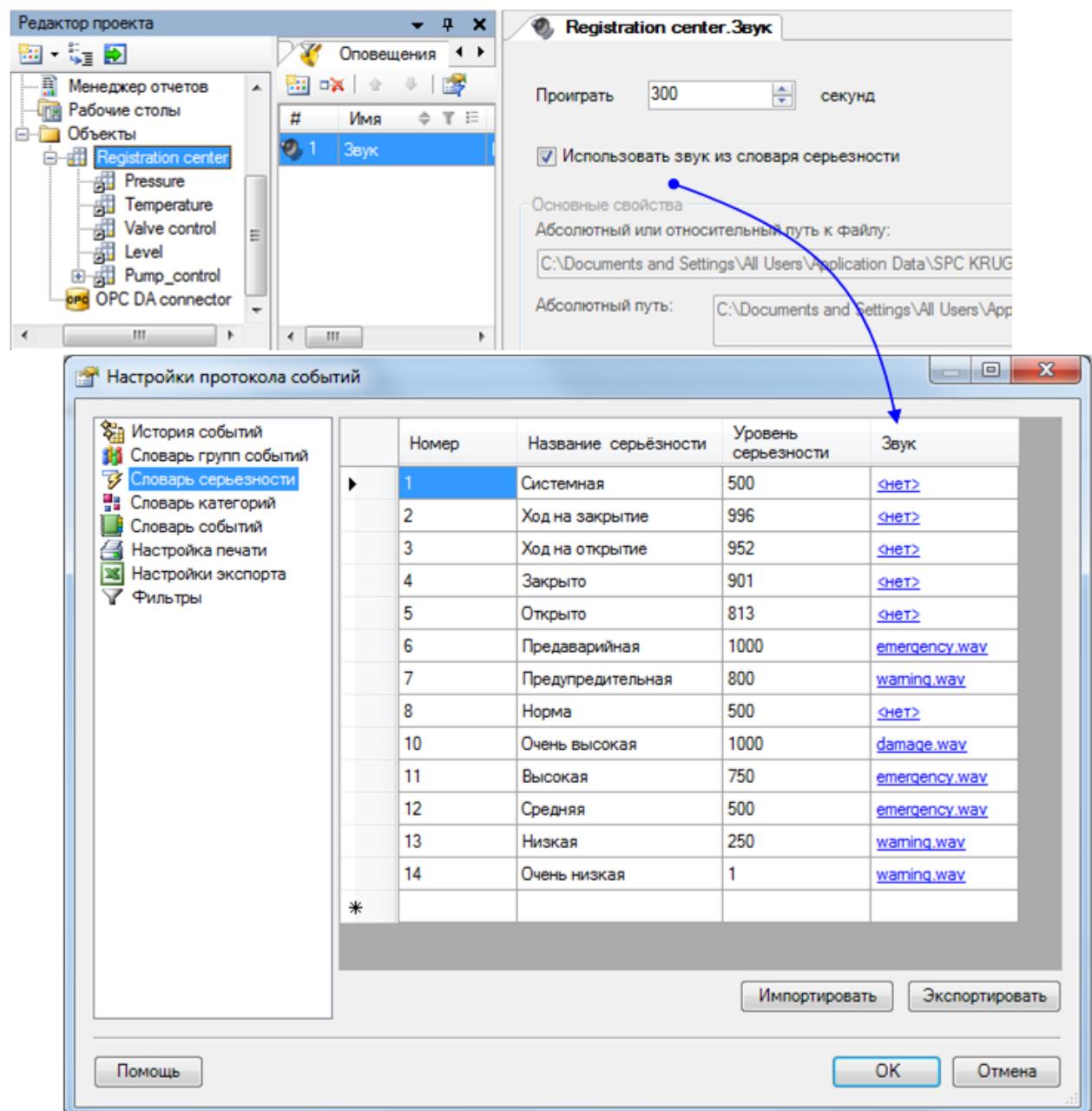


Рисунок 2.31 – Оповещение **Звук** объекта **Registration center**

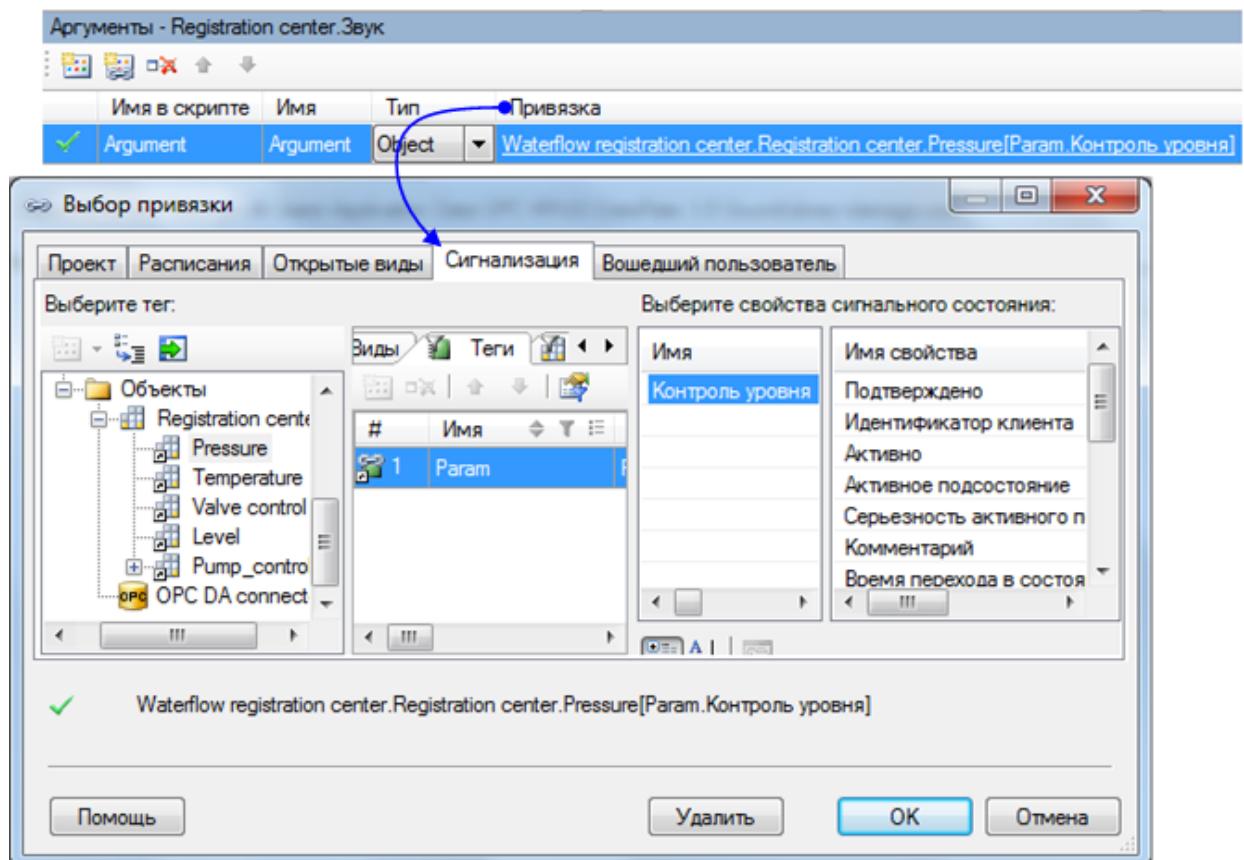


Рисунок 2.32 – Привязка оповещения **Звук** к сигнальному состоянию тега **Param** объекта **Pressure**

2.10 Тренды. Создание и визуализация

Для формирования и отображения трендов **DataRate** предоставляет:

- Системный сервис **Менеджер трендов** – конфигурирование, просмотр и формирование архива трендов
- Графические примитивы **Тренд**, **Табличный тренд** – настройка и отображение тренда
- Шаблон **Аналитический тренд** вида объекта – настройка и отображение тренда
- Объекты библиотек – настройка, отображение и управление трендом

Средства **DataRate** обеспечивают ведение оперативных трендов, а также хранение истории процесса (архивов оперативных трендов).

Отличие в создании и визуализации трендов заключается в следующем:

- Оперативные тренды** – привязка перьев тренда к атрибутам внутренних или внешних тегов
- Архивы оперативных трендов** – привязка перьев тренда к перьям **Менеджера трендов** с использованием **Групп настроек истории**.

2.10.1 Архивы оперативных трендов

Для хранения истории процесса оперативные тренды должны создаваться на основе **перьев Менеджера трендов**:

- 1 Выберите в контекстном меню **Менеджер трендов** пункт **Группы настроек**
- 2 В появившемся окне **Группы настройки истории** добавьте новую группу (на рисунке 2.33 – **SettingsGroup**) и задайте ее параметры.

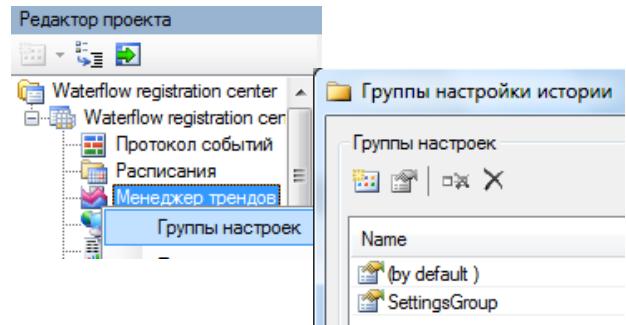


Рисунок 2.34 – Создание группы настроек

Группа настройки хранения истории содержит название группы, путь и имя файла хранения истории, максимальное количество точек тренда в истории, глубину хранения и другие параметры

- 3 Выберите в контекстном меню объекта **Менеджер трендов** пункт **Перья** и в появившемся окне щелкните по иконке – **Новые перья**
- 4 В окне **Добавить новые перья** выберите **Новые перья для тега** и нажмите **OK** (рисунок 2.34)

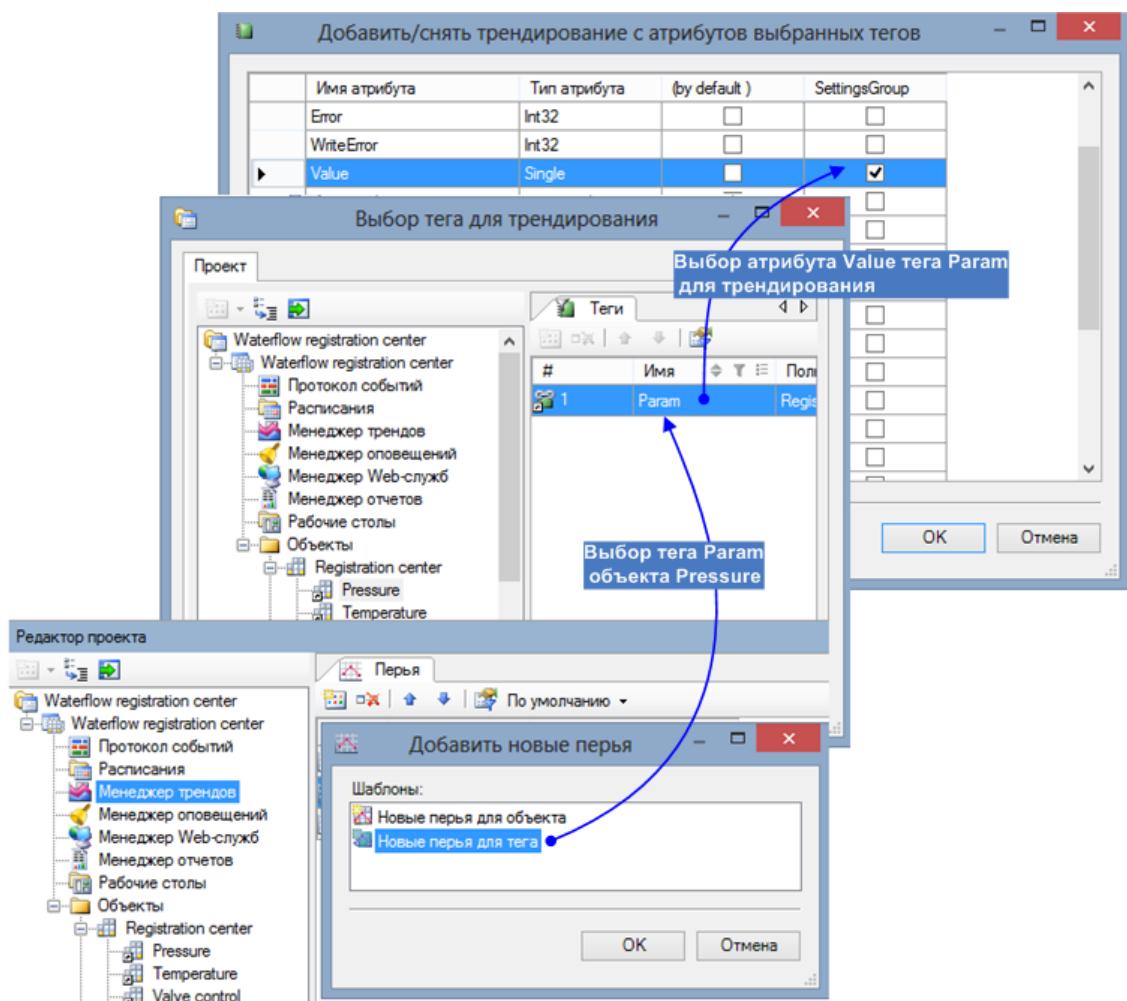


Рисунок 2.34 – Выбор атрибута тега для трендирования

- 5 В окне **Выбор тега для трендирования** укажите объект и его тег, затем нажмите **OK** (рисунок 2.34)
- 6 В появившемся окне **Добавить/снять трендирование с атрибутов выбранных тегов** укажите атрибут тега и сделайте отметку в колонке **Settings Group** для формирования архива тренда (рисунок 2.34).

Таким образом, сформируйте все необходимые перья **Менеджера трендов** для ведения трендов температуры, давления и уровня жидкости в емкости (рисунок 2.35).

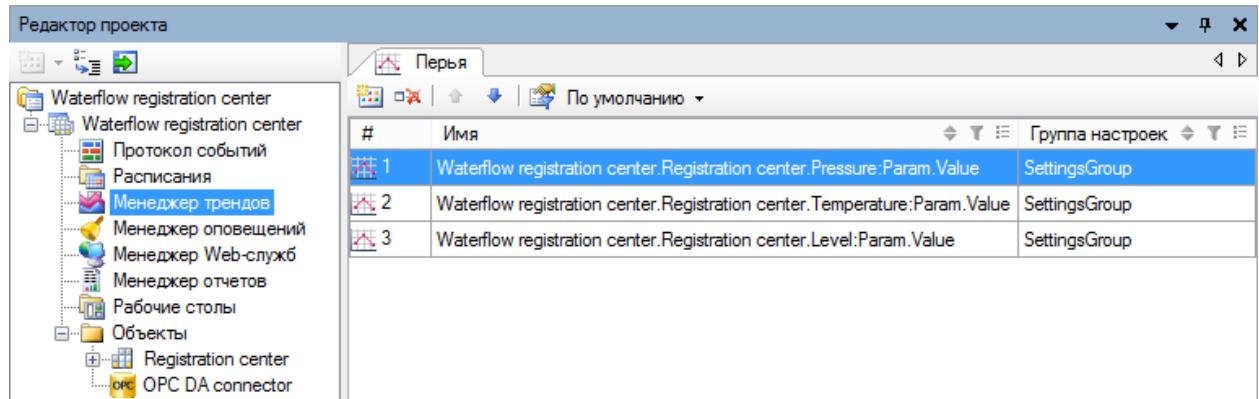


Рисунок 2.35 – Менеджер трендов и его перья

- 7 Привяжите перья **Pressure**, **Temperature**, **Level** аналитического тренда (объект **Registration center**, вид **Analytical Trend**, свойство пера **Основные/Привязка**) к соответствующим перьям **Менеджера трендов** (пример результата привязки пера **Pressure** на рисунке 2.36).

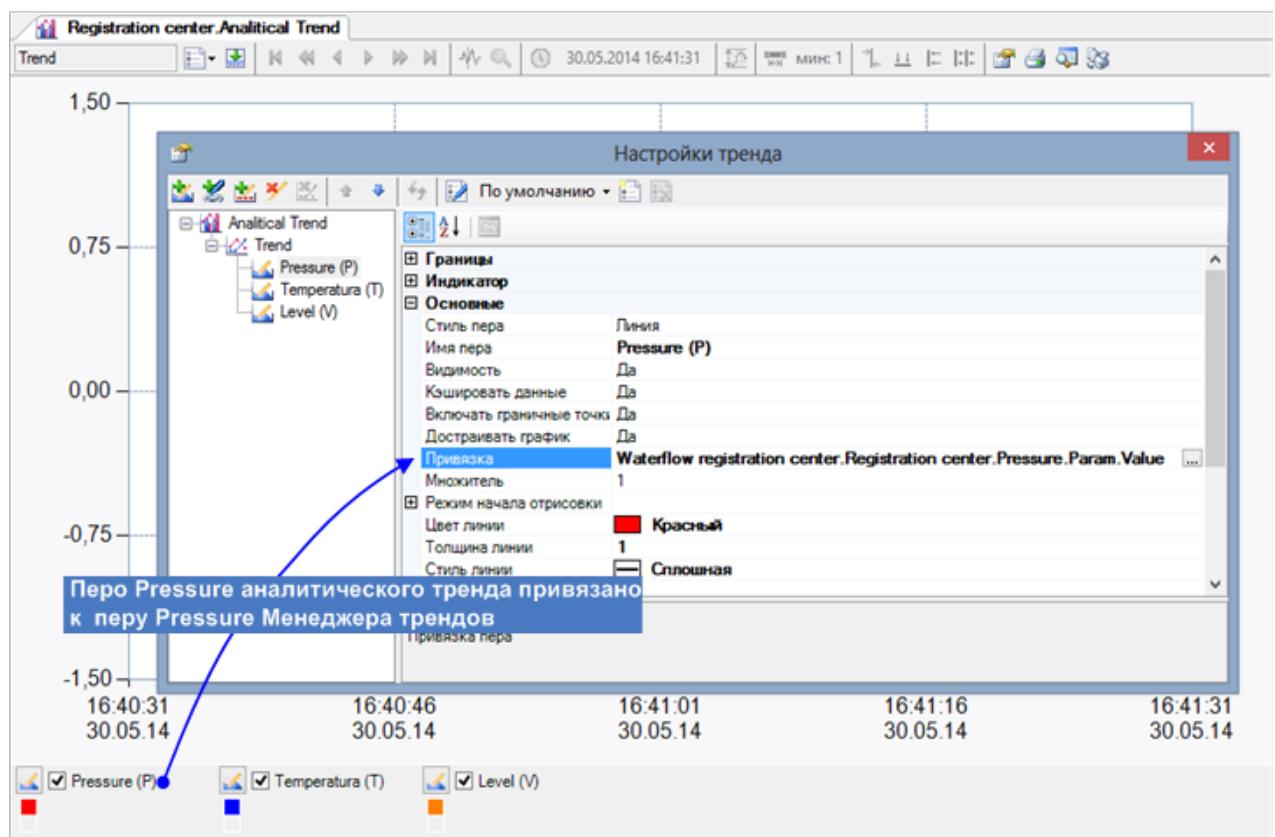


Рисунок 2.36 – Привязка пера **Pressure**

2.10.2 Просмотр трендов в отдельном окне

Для просмотра тренда назначена реакция **Переход** на событие **Нажатие левой кнопки мыши** (рисунок 2.37) для графического примитива **Прямоугольник** вида **Panel_Buttons** объекта **Registration center**.

Реакция **Переход** настроена на переход к изображению вида **Analitical Trend** объекта **Registration center**. Вид открывается в рабочей области **Main desktop** (главном окне HMI-приложения).

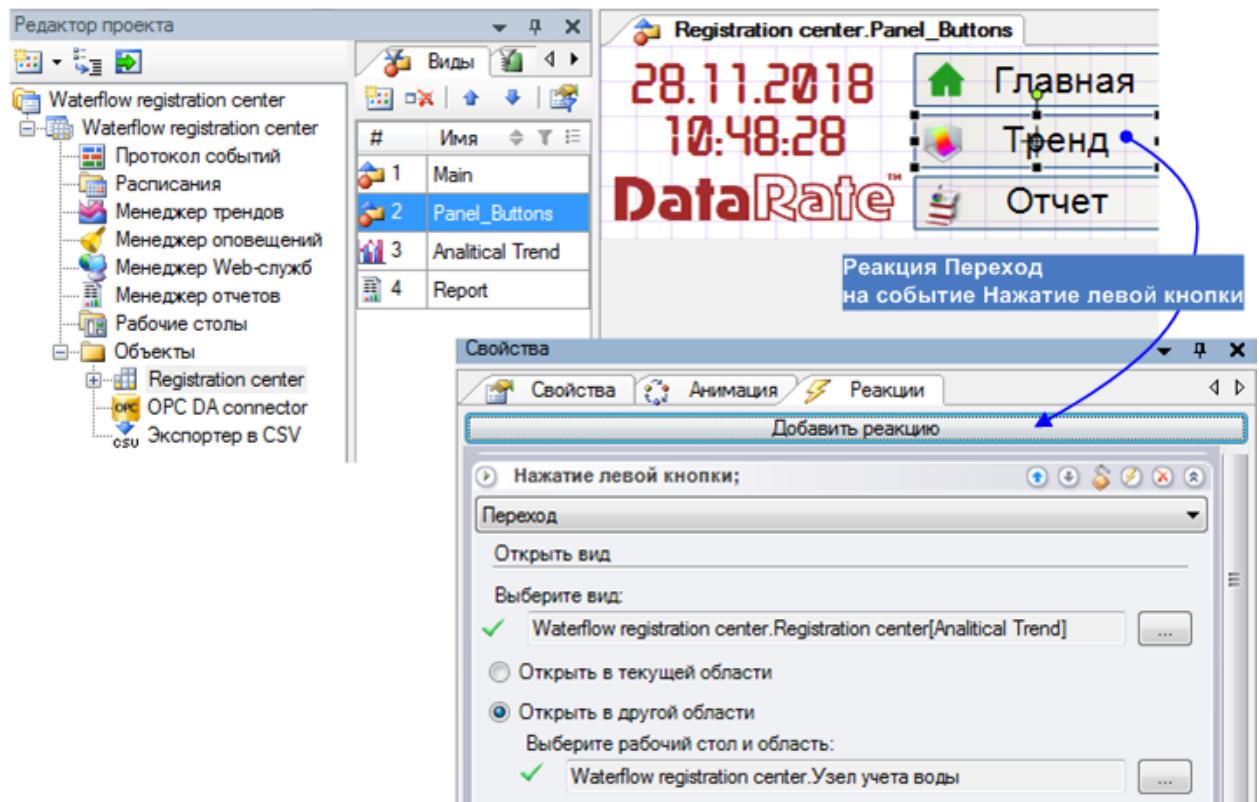


Рисунок 2.37 – Вид **Analitical Trend** объекта **Registration center**. Реакция на щелчок мыши для вызова окна тренда

В режиме отладки или выполнения HMI-приложения щелчок по «кнопке» **Тренд** вызывает окно отображения трендов температуры, давления и уровня жидкости в емкости (рисунок 2.38).

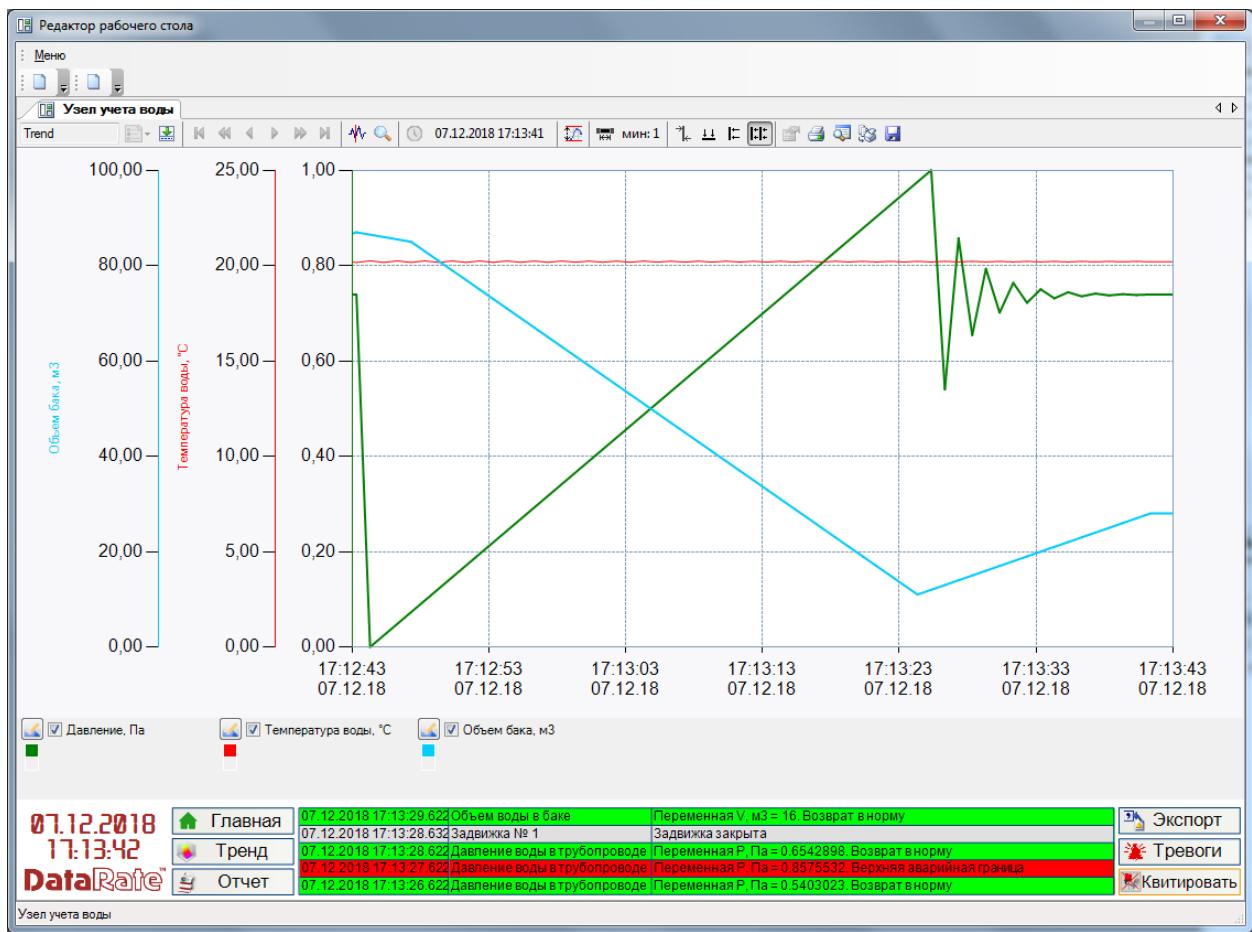


Рисунок 2.39 – Окно Аналитический тренд и окно Активная сигнализация

2.10.3 Экспорт трендов

DataRate обеспечивает **экспорт данных** – выгрузку данных из хранилища трендов в файлы форматов XML и CSV.

Экспорт данных осуществляется с помощью объектов-экспортеров: **Экспортер в XML** и **Экспортер в CSV**.

В проекте **Waterflow registration center** используется **Экспортер в CSV**, настроенный на экспорт данных давления, температуры и объема жидкости в баке. Управление работой экспортера осуществляется с помощью атрибута **Make** тега **SourceSetup** (рисунок 2.40).

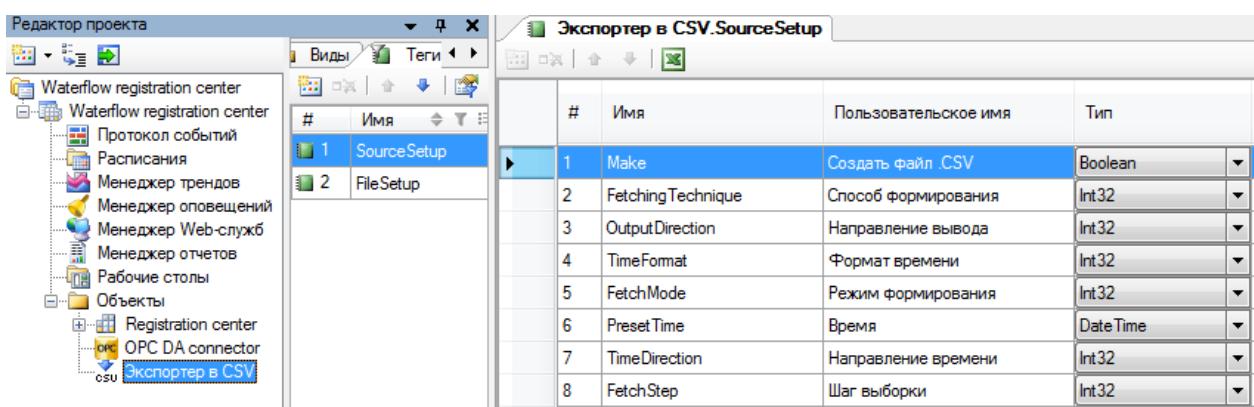


Рисунок 2.40 – Фрагмент определения тега SourceSetup

Установка значения этого атрибута в значение **true** инициирует создание **CSV**-файла. В проекте это реализуется с помощью реакции на нажатие кнопки **Экспорт** на мнемосхеме **Panel_Button** (рисунок 2.41)

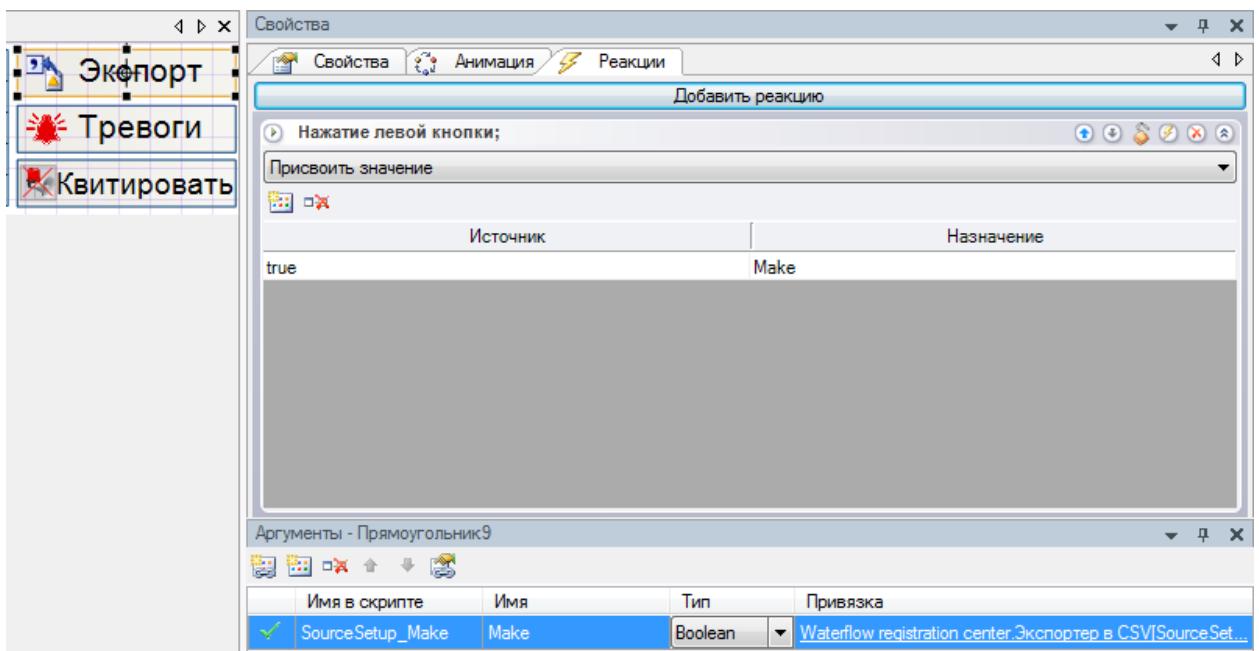


Рисунок 2.41 – Реакция на нажатие левой кнопки для формирования CSV-файла

2.11 Отчеты

DataRate обеспечивает визуальное создание шаблонов любых необходимых отчетов, а также хранение отчетов, их рассылку по электронной почте и печать по событиям и расписаниям.

Менеджер отчетов DataRate предназначен для создания отчета по заданному шаблону. Менеджер отчетов конфигурируется в Среде разработки и интегрируется в Среду исполнения **DataRate**.

2.11.1 Шаблон отчета

Шаблон отчета – это макет документа, создаваемый в **дизайнере отчетов**. В объектной модели **DataRate** шаблон отчета является видом объекта (например, как мнемосхема). Шаблон сохраняется как **XML**-документ (файл с расширением **rdlc**). Это обеспечивает работу с отчетами, созданными не только с помощью дизайнера отчетов **DataRate**, но и с помощью других средств.

Для создания и формирования отчета следует:

- 1 Создать шаблон отчета в дизайнере отчетов (или выбрать существующий шаблон)
- 2 Привязать поля шаблона к источникам данных
- 3 Настроить параметры формирования отчета.

В проекте **Waterflow registration center** шаблон отчета создан как вид **Report** (рисунки 2.42, 2.43) объекта **Registration center**.

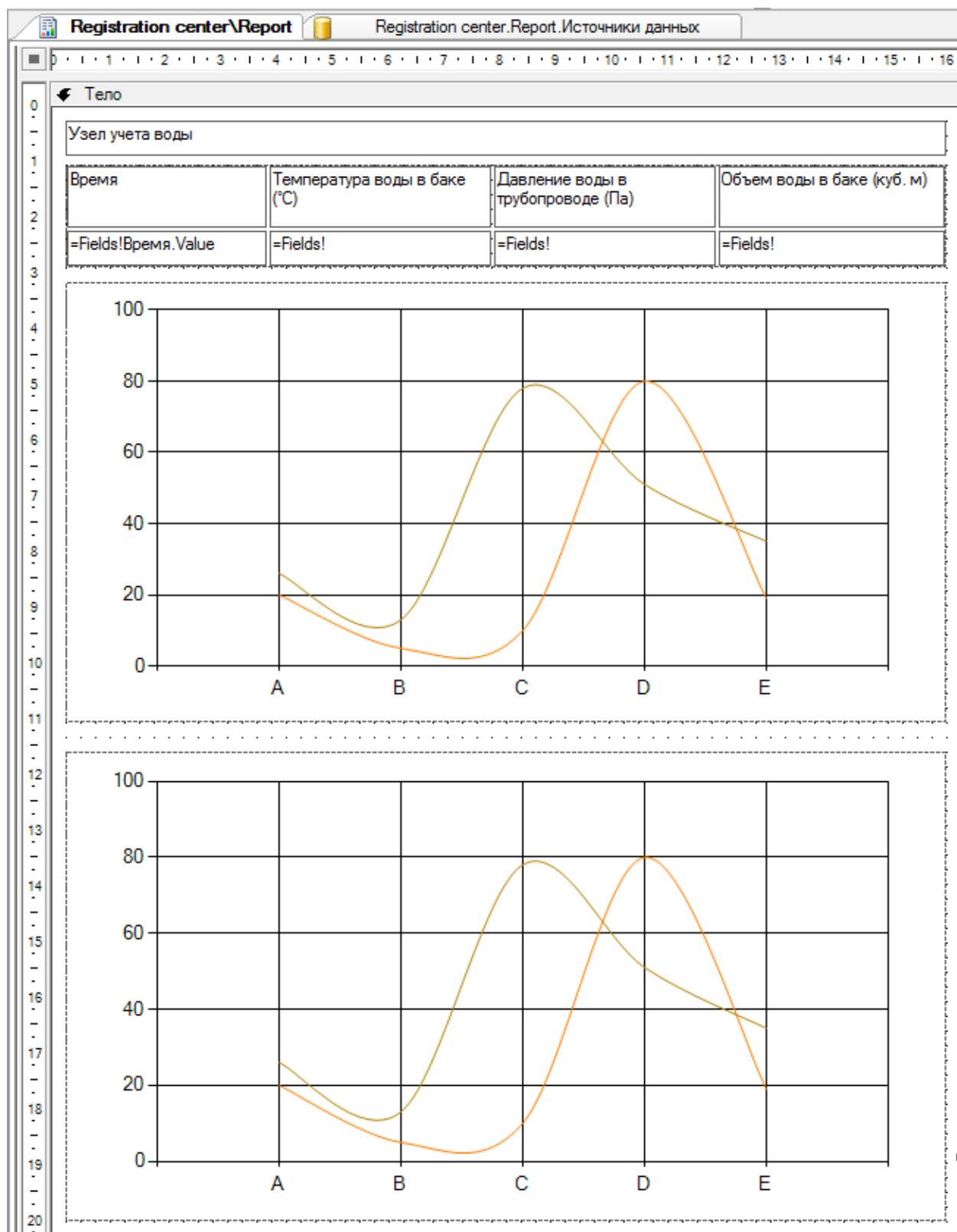


Рисунок 2.42 – Шаблон отчета Report

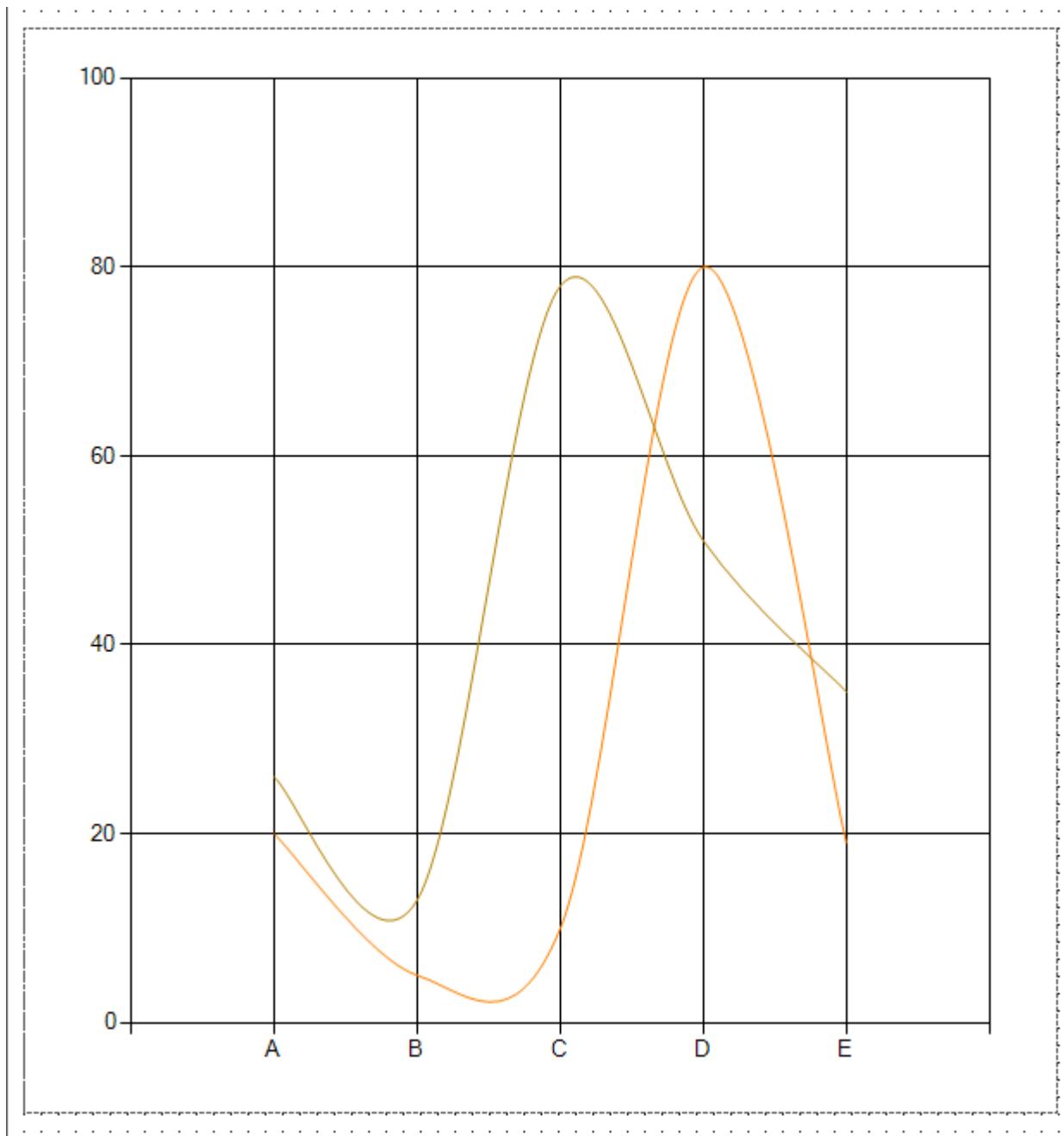


Рисунок 2.43 – Шаблон отчета **Report** (продолжение)

Текстовые поля отчета предназначены для отображения значений температуры, давления и уровня жидкости в баке.

Для формирования значений этих полей следует:

- 1 В контекстном меню вида **Report** выбрать **Источники данных и параметры**
- 2 Настроить источник данных (рисунок 2.44) шаблона отчета. Источником данных являются перья (значения тегов **Param** объектов **Temperature**, **Pressure**, **Level**) группы **SettingsGroup** Менеджера трендов

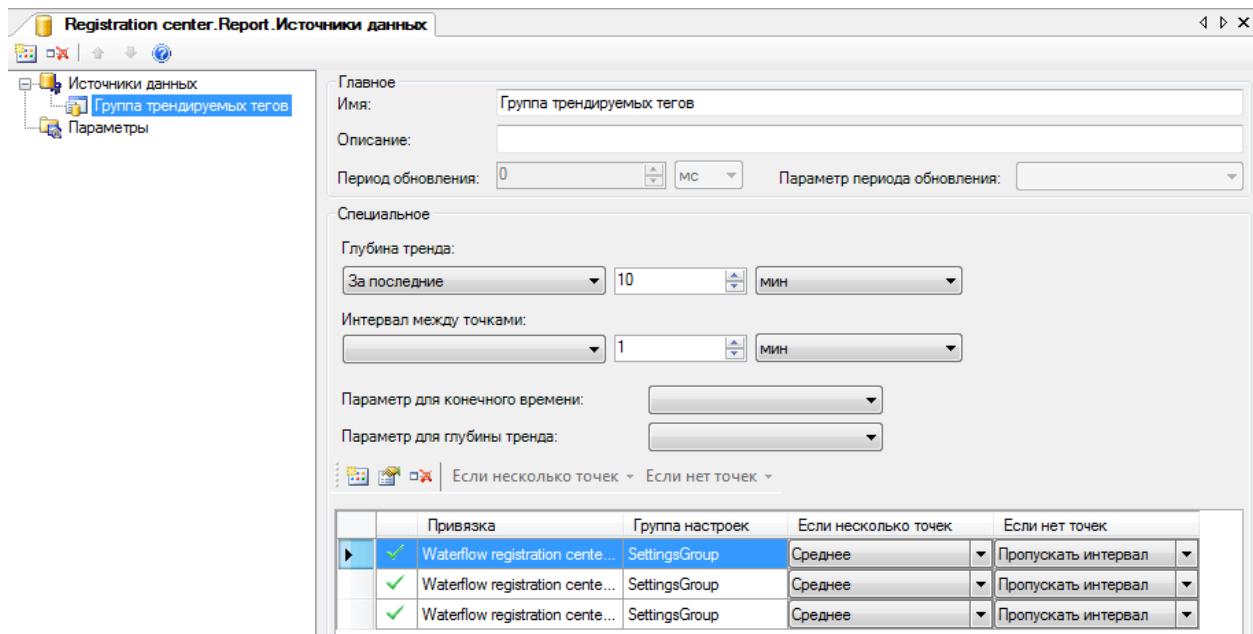


Рисунок 2.44 – Настройка источника данных и параметров формирования отчета **Report**

- 3 Настроить свойства шаблона **Report**. В свойстве **Данные/Имя источника данных** указать имя настроенного источника – **Группа трендируемых тегов**.

2.11.2 Просмотр и сохранение отчета в Среде разработки

Для просмотра отчета в Среде разработки **DataRate** следует:

- 1 Открыть шаблон отчета (в данном проекте вид **Report** объекта **Registration center**) в области редактирования
- 2 Запустить проект на имитацию
- 3 Перейти на вкладку **Предварительный просмотр** (рисунки 2.45 -2.47).

Для просмотра отчета в отдельном **плавающем окне** назначена реакция **Переход** на событие **Нажатие левой кнопки мыши** для графического примитива **Прямоугольник** (имя – **Отчет**) вида **Panel.Buttons** объекта **Registration center** (аналогично просмотру тренда, рисунок 2.37).

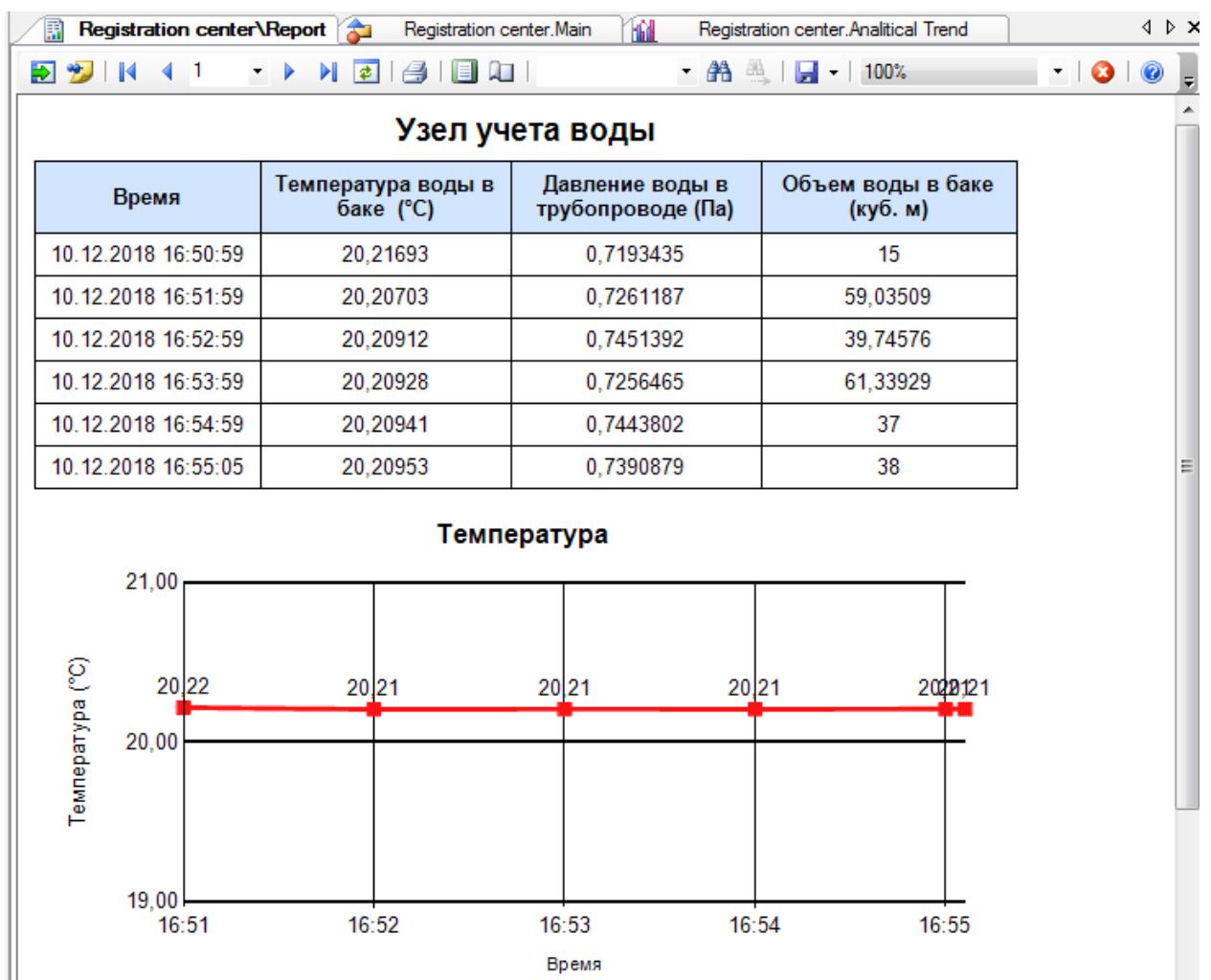


Рисунок 2.45 – Просмотр отчета в Среде разработки

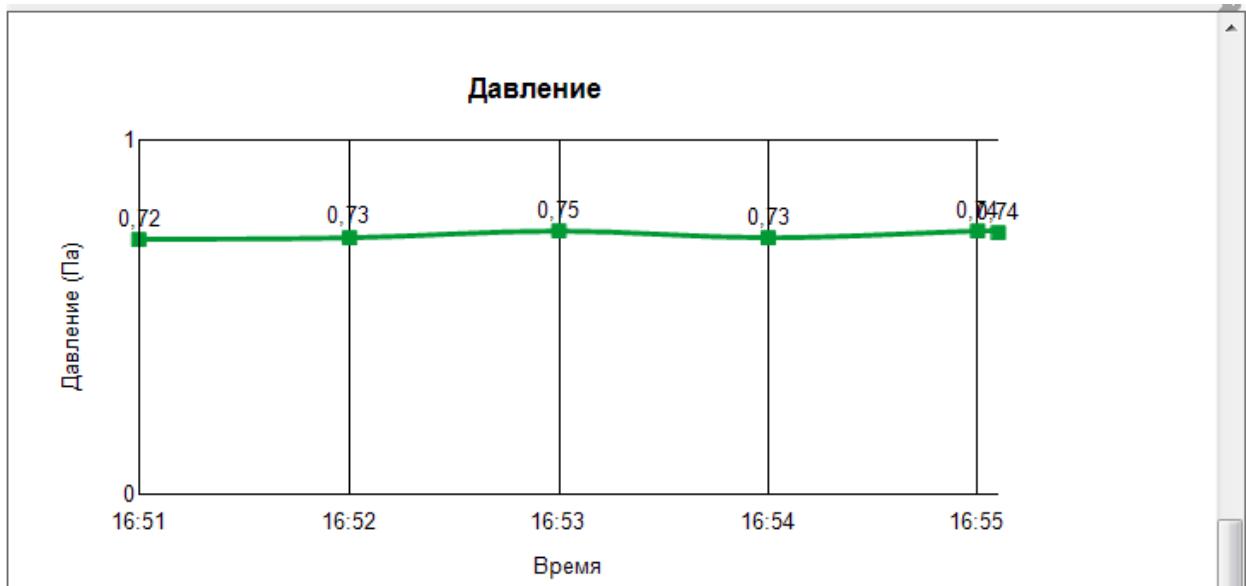


Рисунок 2.46 – Просмотр отчета в Среде разработки (продолжение)

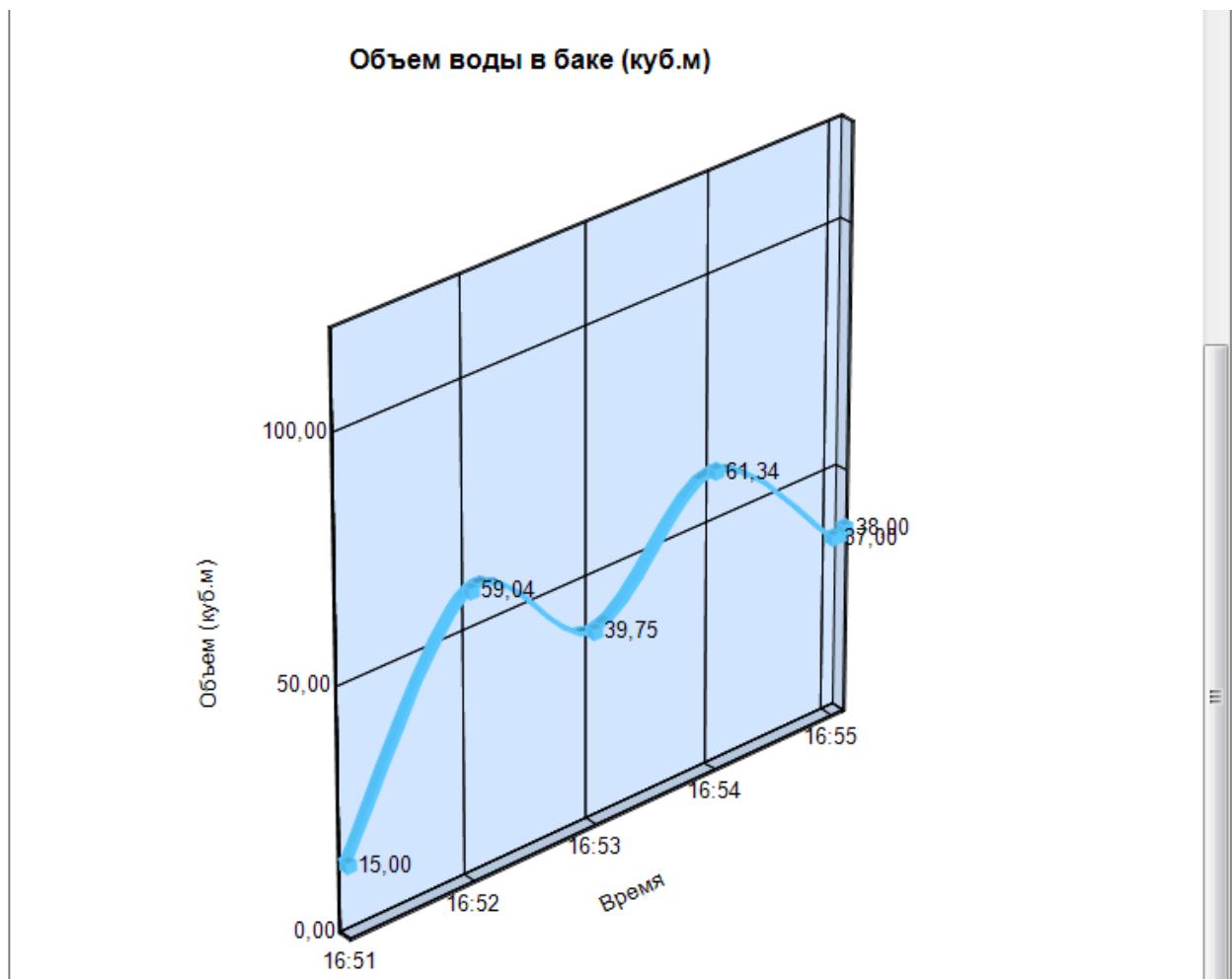
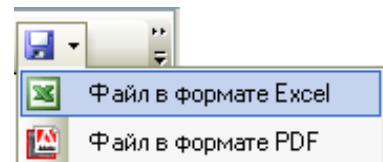


Рисунок 2.47 – Просмотр отчета в Среде разработки (окончание)

Для сохранения отчета:

- 1 Щелкните по иконке сохранения файла на панели инструментов (рисунок 2.45)
- 2 Выберите формат файла – **xls** (Microsoft Excel) или **pdf**.



Отчет в формате **xls** показан на рисунке 2.48.

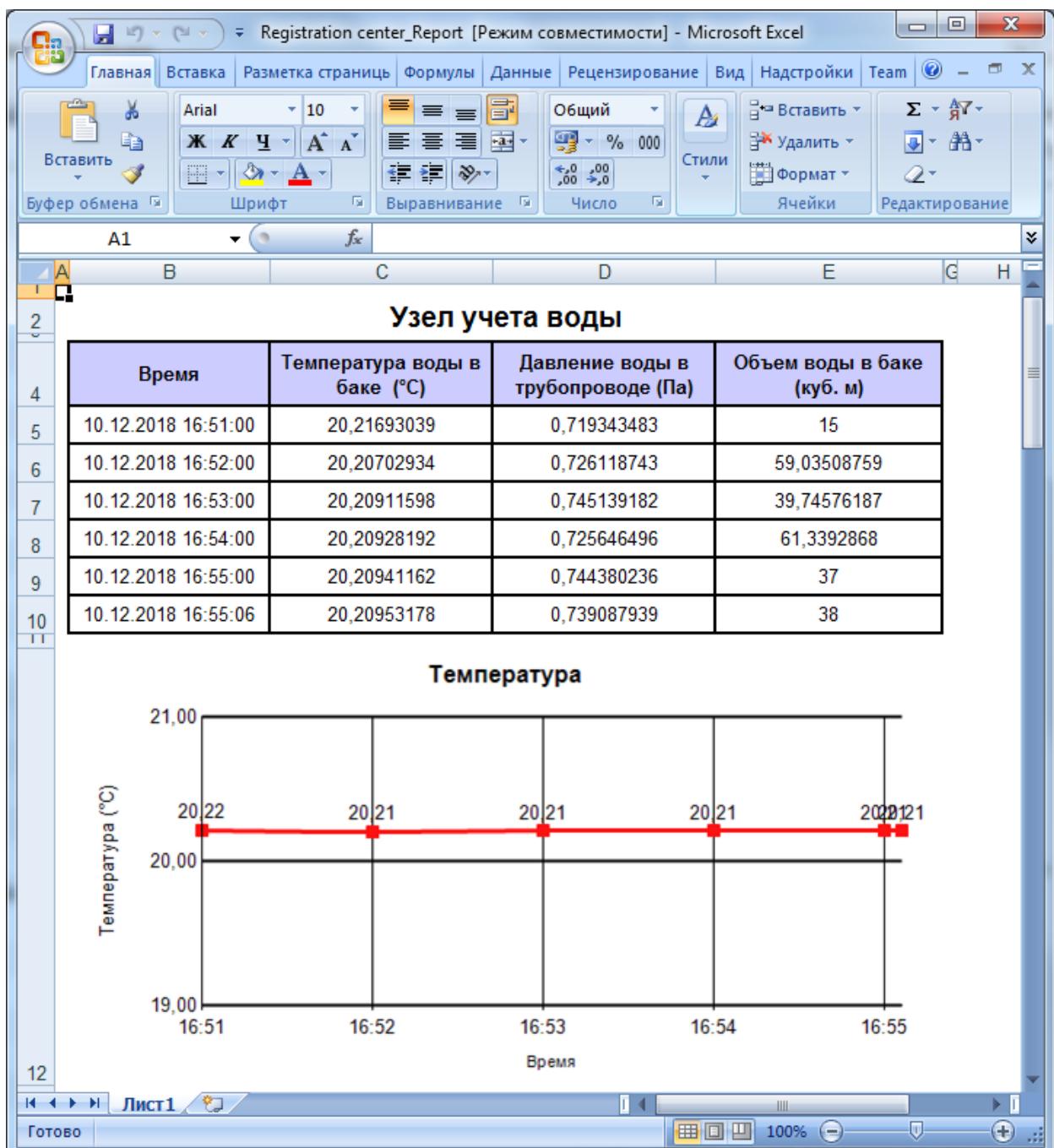


Рисунок 2.48 – xls-файл отчета

2.11.3 Формирование отчета по расписанию

В **DataRate** момент непосредственного формирования отчета можно задать с помощью расписания.

Для создания расписания следует:

- 1 В дереве элемента проекта щелкнуть левой клавишей мыши по системному объекту **Расписание**
- 2 В открывшейся закладке **Расписания** выбрать (или создать –) расписание

3 Двойным щелчком по наименованию расписания открыть окно **Расписание** и задать необходимые настройки (рисунок 2.49)

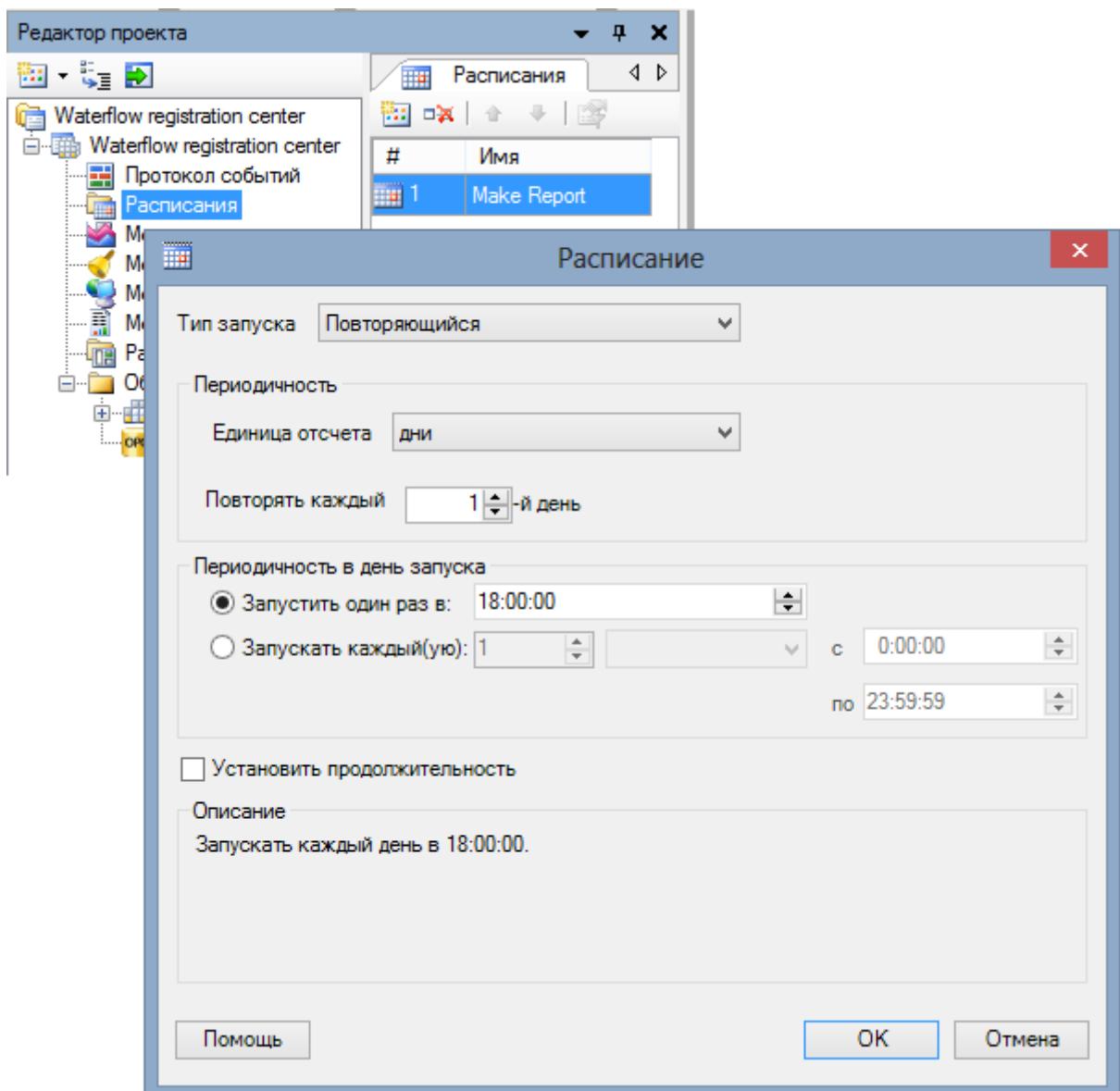


Рисунок 2.49 – Настройка расписания **Make Report**

Для задания момента формирования отчета по расписанию следует установить свойство отчета **Генерация/Расписание**.

Например, если в свойствах отчета **Report** выбрать расписание **Make Report** (рисунок 2.49), то отчет будет формироваться один раз в сутки в 18.00 (в конце рабочего дня, рисунок 2.49) при этом будет формироваться сообщение в протокол событий (свойство **Публикация/Отправлять сообщение в протокол при публикации – Да**, рисунок 2.50).

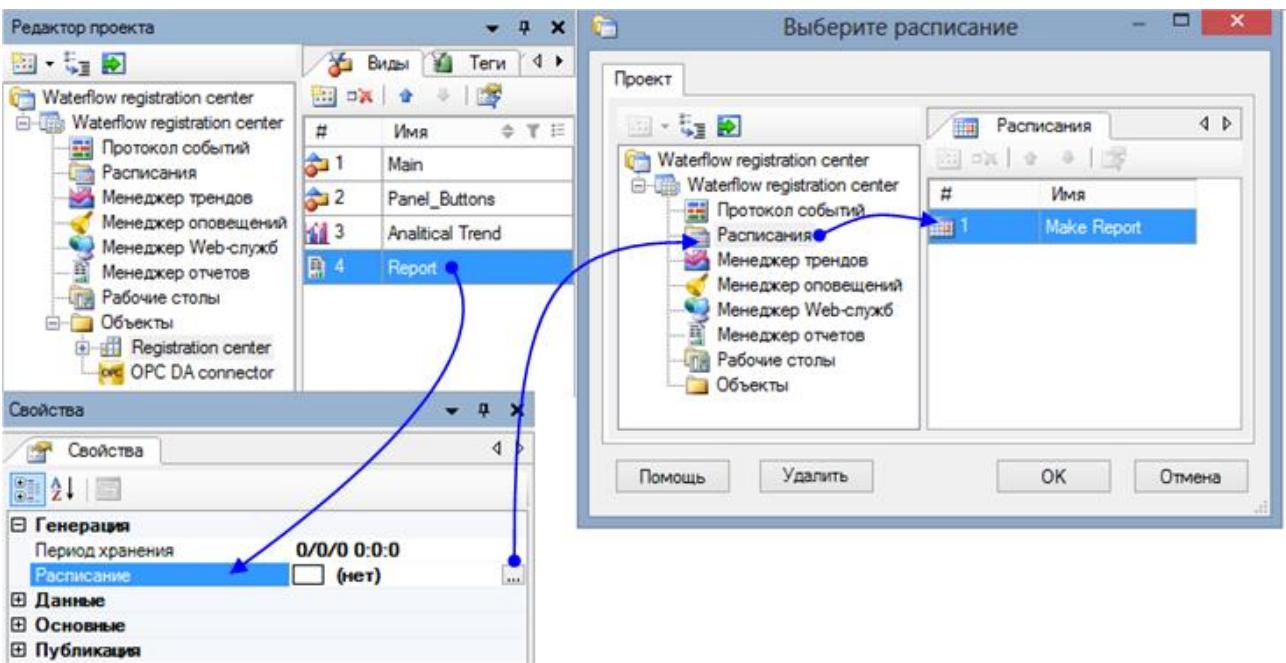


Рисунок 2.50 – Выбор расписания Make Report для отчета Report

2.11.4 Настройки отчета в Менеджере отчетов

Менеджер отчетов обеспечивает настройку параметров хранения, печати и рассылки отчетов.

Пример сохранения созданного отчета в файл, параметры которого указываются в окне **Настройки менеджера отчетов/ Публикации в папку** приведен на рисунке 2.51.

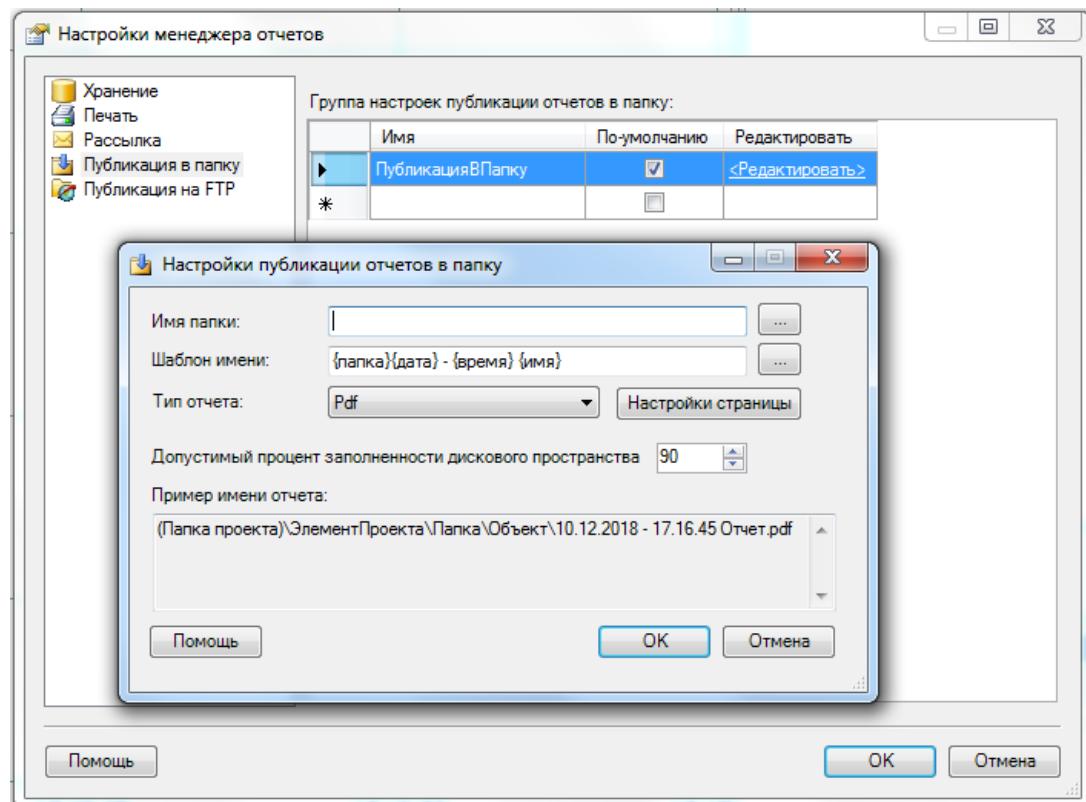


Рисунок 2.51 – Настройка параметров сохранения отчета

2.12 Рабочие столы

Для исполнения HMI-приложения необходимо создать хотя бы один рабочий стол!

В HMI-приложении **Waterflow registration center** рабочий стол **Main desktop** содержит две мнемосхемы (рисунок 2.52):

- **Main** – схема контролируемого процесса
- **Panel.Buttons** – панель управления с кнопками: **На главную**, **Аналитический тренд**, **Отчет**.

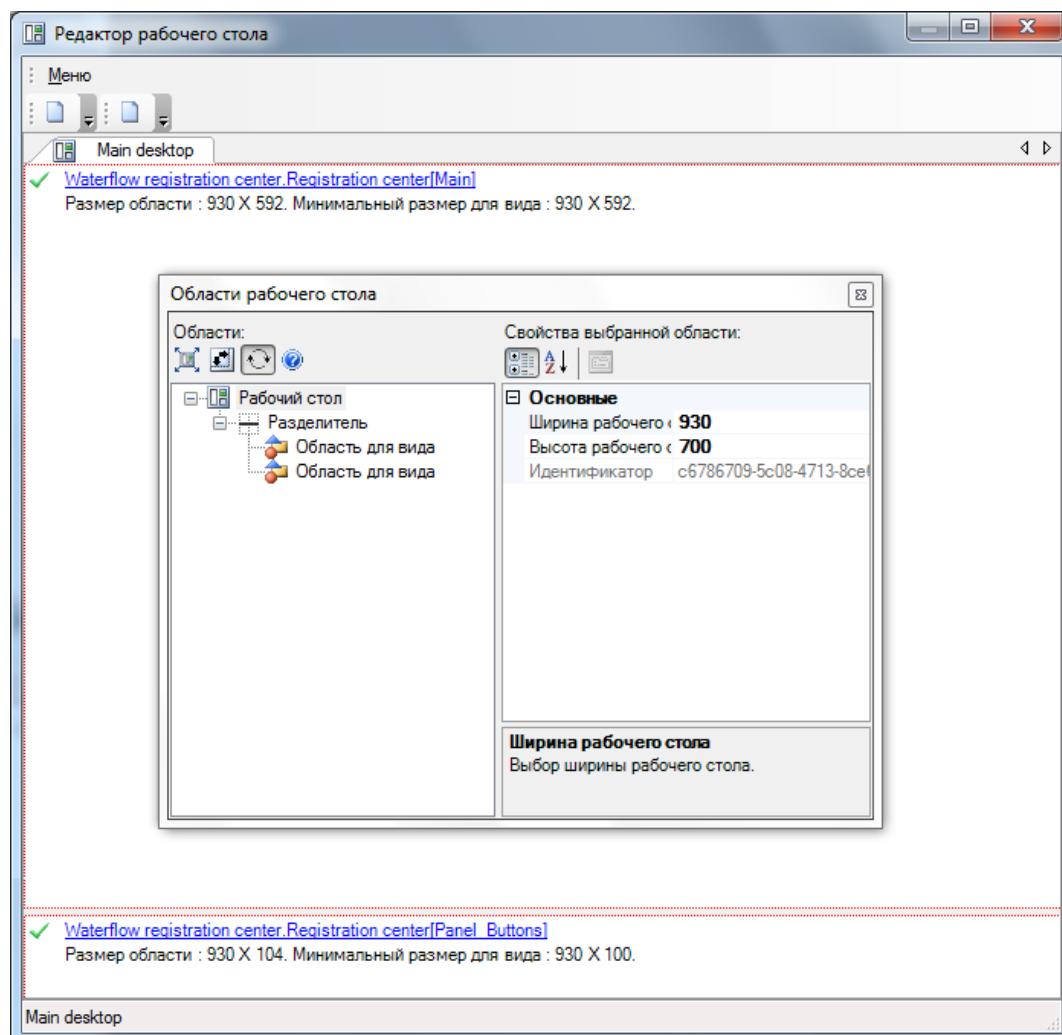


Рисунок 2.52 – Рабочий стол **Main desktop**

Для запуска HMI-приложения из интегрированной среды разработки:

- 1 Щелкните по кнопке в панели инструментов. Проект переключается в режим выполнения
- 2 Выберите системный объект **Рабочие столы** в дереве элементов проекта и выполните двойной щелчок по названию рабочего стола (**Main desktop** в закладке **Рабочие столы**)
- 3 Рабочий стол откроется в режиме исполнения (рисунок 2.53).

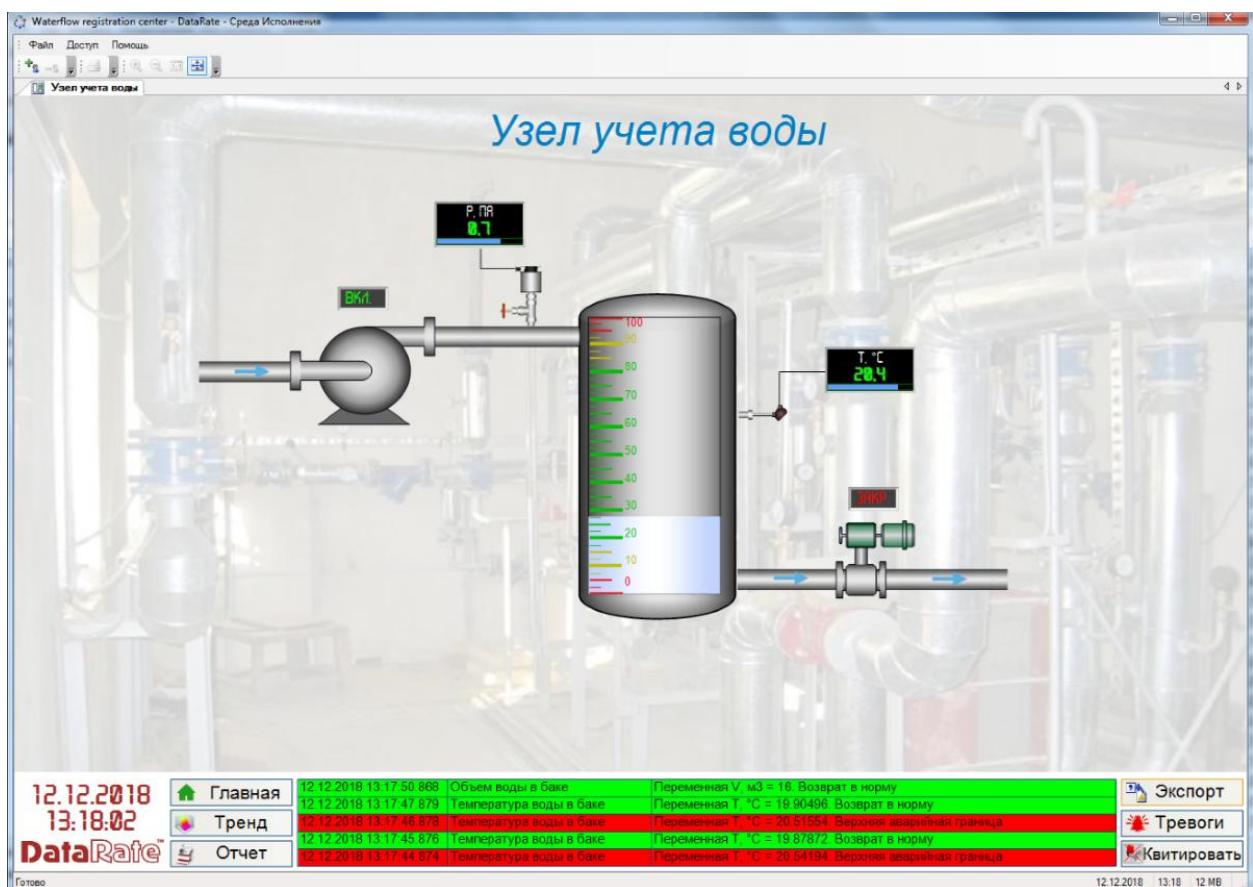


Рисунок 2.53 – Рабочий стол **Узел учета воды** в режиме выполнения

3 ВЫПОЛНЕНИЕ ПРОЕКТА *DataRate* В СРЕДЕ ИСПОЛНЕНИЯ

Для запуска проекта в Среде исполнения *DataRate* следует:

- 1 Выбрать **Все программы/ DataRate 4.2** в системном меню **ПУСК**
- 2 Выбрать **Среда исполнения** (рисунок 3.1)

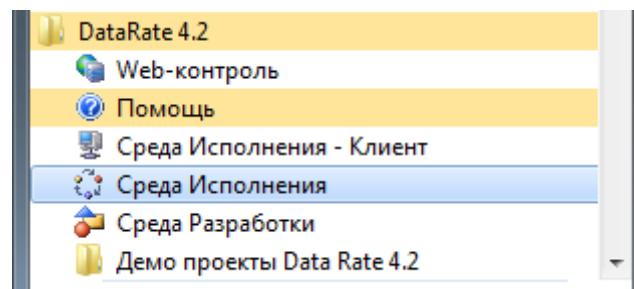


Рисунок 3.1 – Запуск проекта в Среде исполнения *DataRate*

- 3 Среда исполнения *DataRate* (рисунок 3.2) запускается в **trial**-режиме. Если необходимо, выполните регистрацию. Для продолжения щелкните по кнопке **OK**

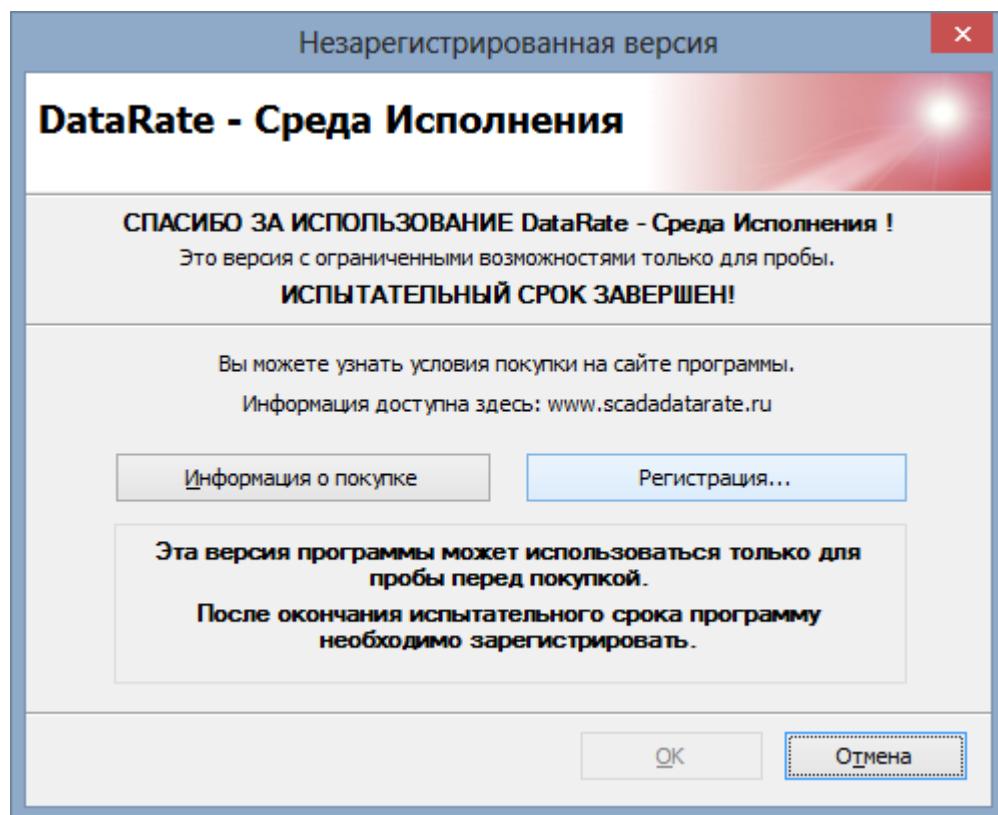


Рисунок 3.2 – Запуск Среды исполнения в **trial**-режиме

- 4 В появившемся окне **Открыть** выберите файл проекта **Waterflow registration center.krproj**. Приложение начинает работать (рисунок 3.3).

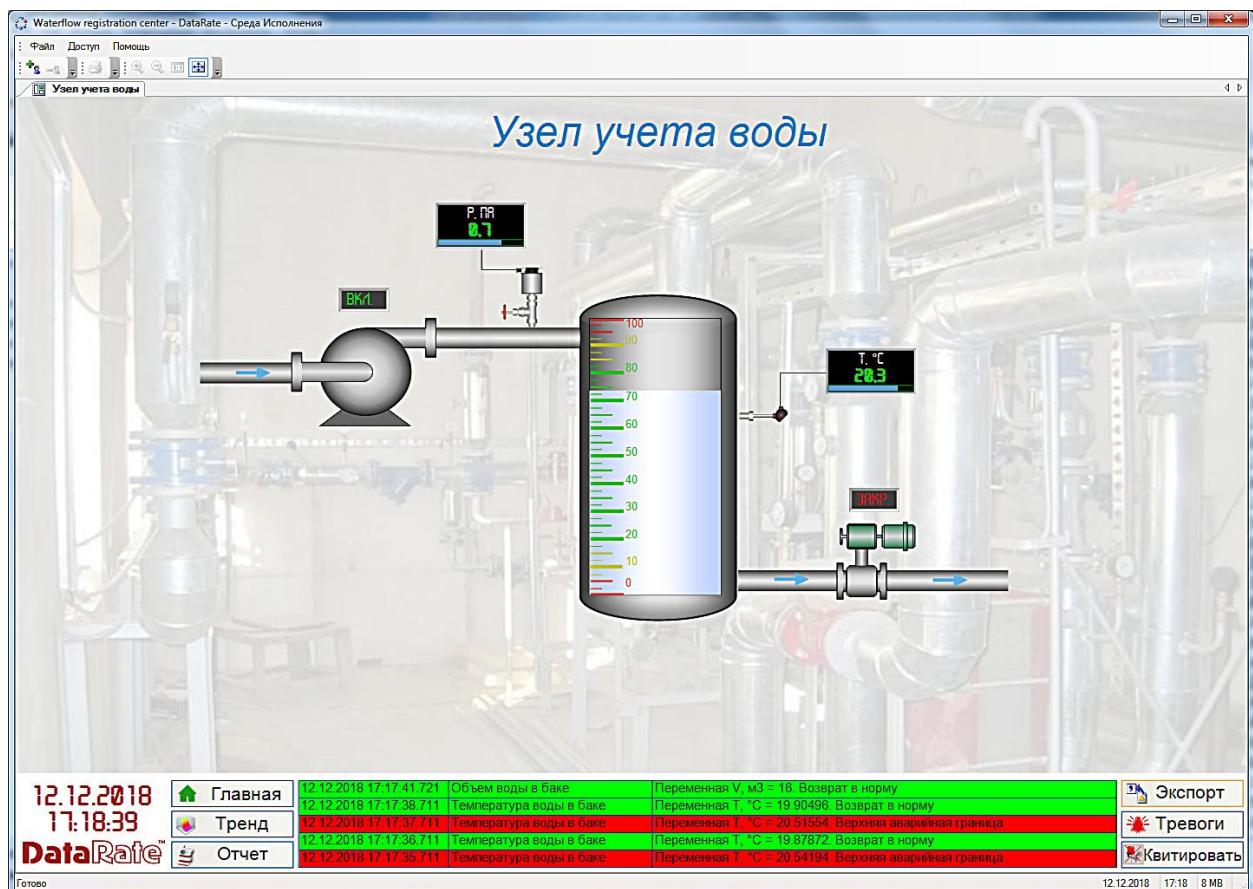


Рисунок 3.3 – Приложение Waterflow registration center в Среде исполнения DataRate

Тестирование приложения Waterflow registration center

Выполним тестирование приложения Waterflow registration center. Для этого последовательно изменим состояние насоса и задвижки, и затем убедимся в правильном формировании трендов и протокола событий:

- 1 Двойным щелчком по панели визуализации давления (рисунок 3.3., сверху) откройте вид Circle_device объекта Pressure (рисунок 3.4)
- 2 Двойным щелчком по панели визуализации температуры (рисунок 3.3., справа) откройте вид Vertical_termometr объекта Temperature (рисунок 3.5)
- 3 Для управления насосом и задвижкой щелкните по их изображению в окне приложения (рисунок 3.6). В открывшихся виртуальных приборах управления с помощью переключателя включите/выключите насос и с помощью кнопок Открыть/Закрыть измените состояние задвижки.

Не забывайте, что работает автоматическое управление системой. Поэтому Ваша команда может быть замена на команду системы в случае достижения подсостояния сигнализации (ВАГ, НАГ, ВПГ, НПГ).

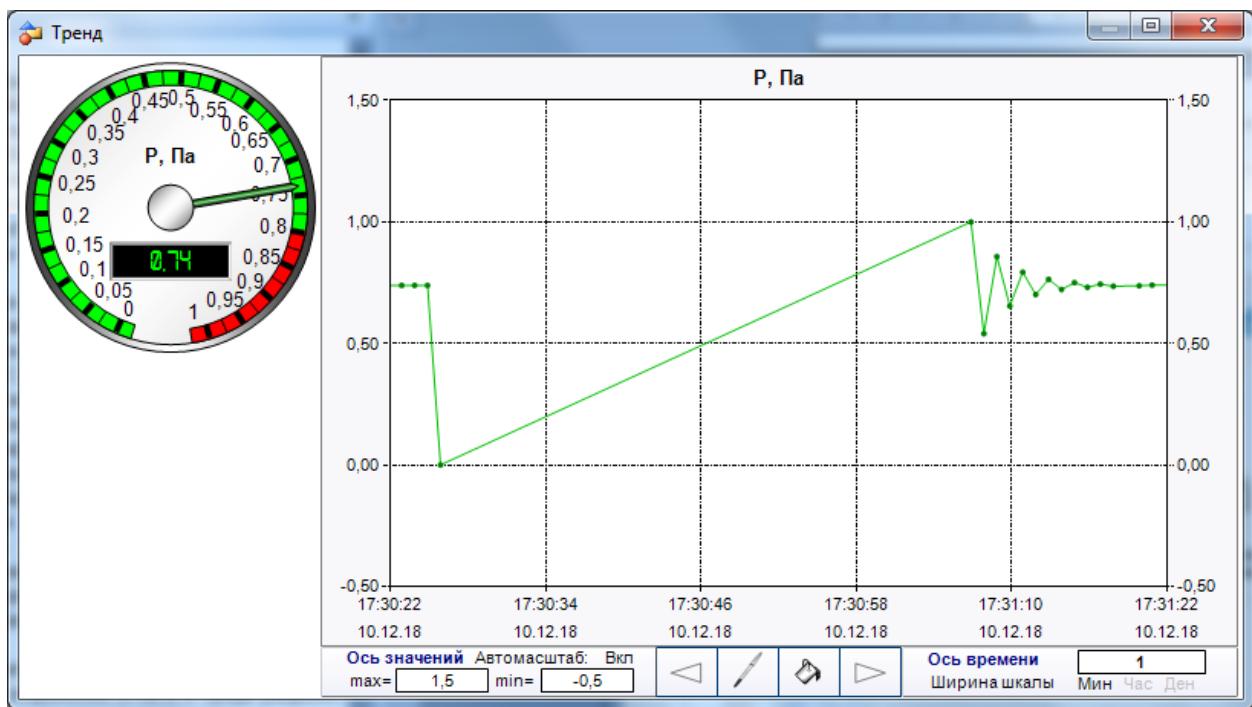


Рисунок 3.4 – Виртуальный прибор и тренд давления

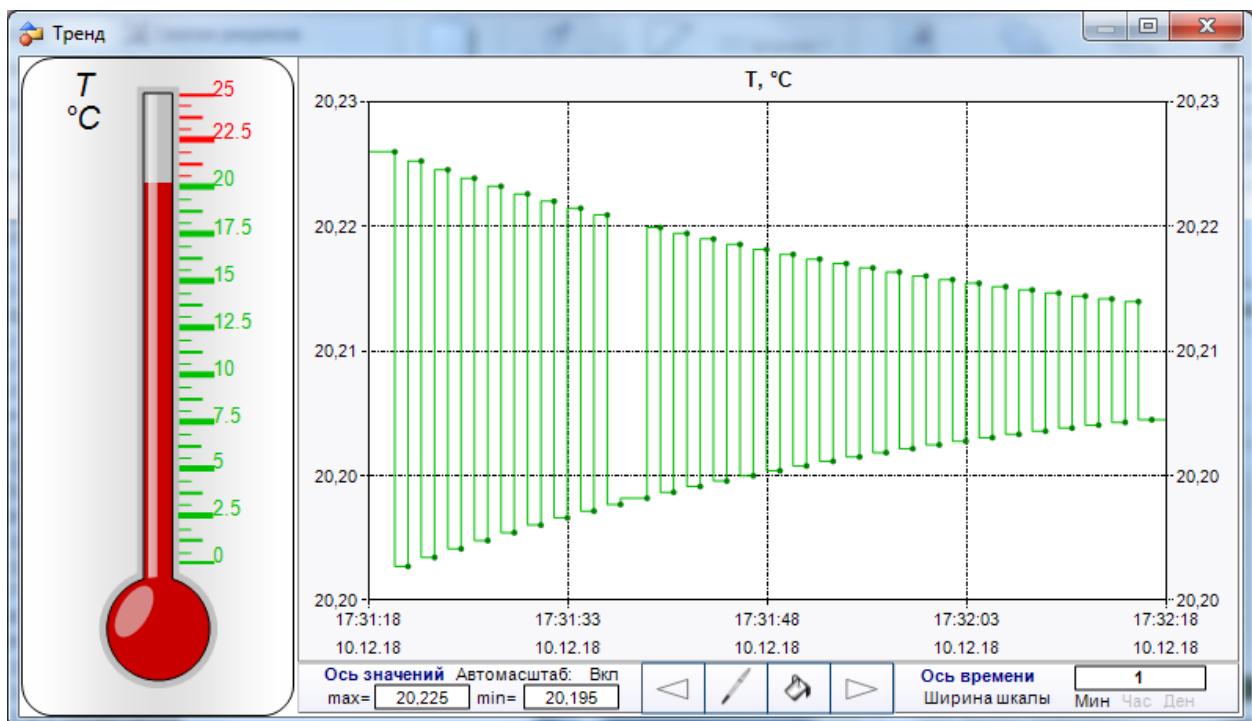


Рисунок 3.5 – Виртуальный прибор и тренд температуры

- 4 Щелчком по кнопке **Аналитический тренд** вызовите окно трендов. Измените состояние насоса и/или задвижки и убедитесь, что тренды параметров изменяются в соответствие с ее состоянием (рисунок 3.7)

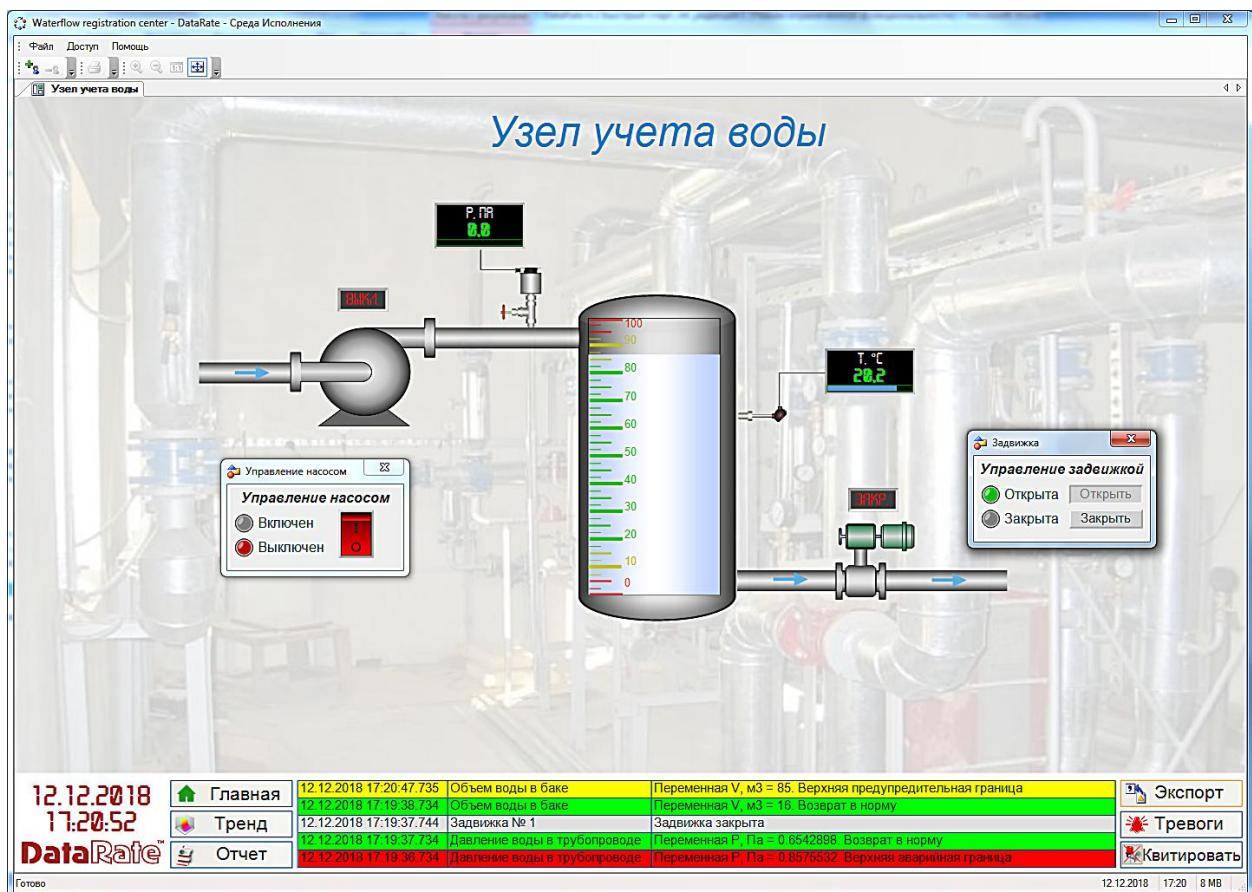


Рисунок 3.6 – Виртуальные приборы управления насосом и задвижкой



Рисунок 3.7 – Среда исполнения. Тренд значений параметров

- 5 Щелчком по элементу **Протокол событий** на панели кнопок переходов вызовите системное окно протокола событий (рисунок 3.8).

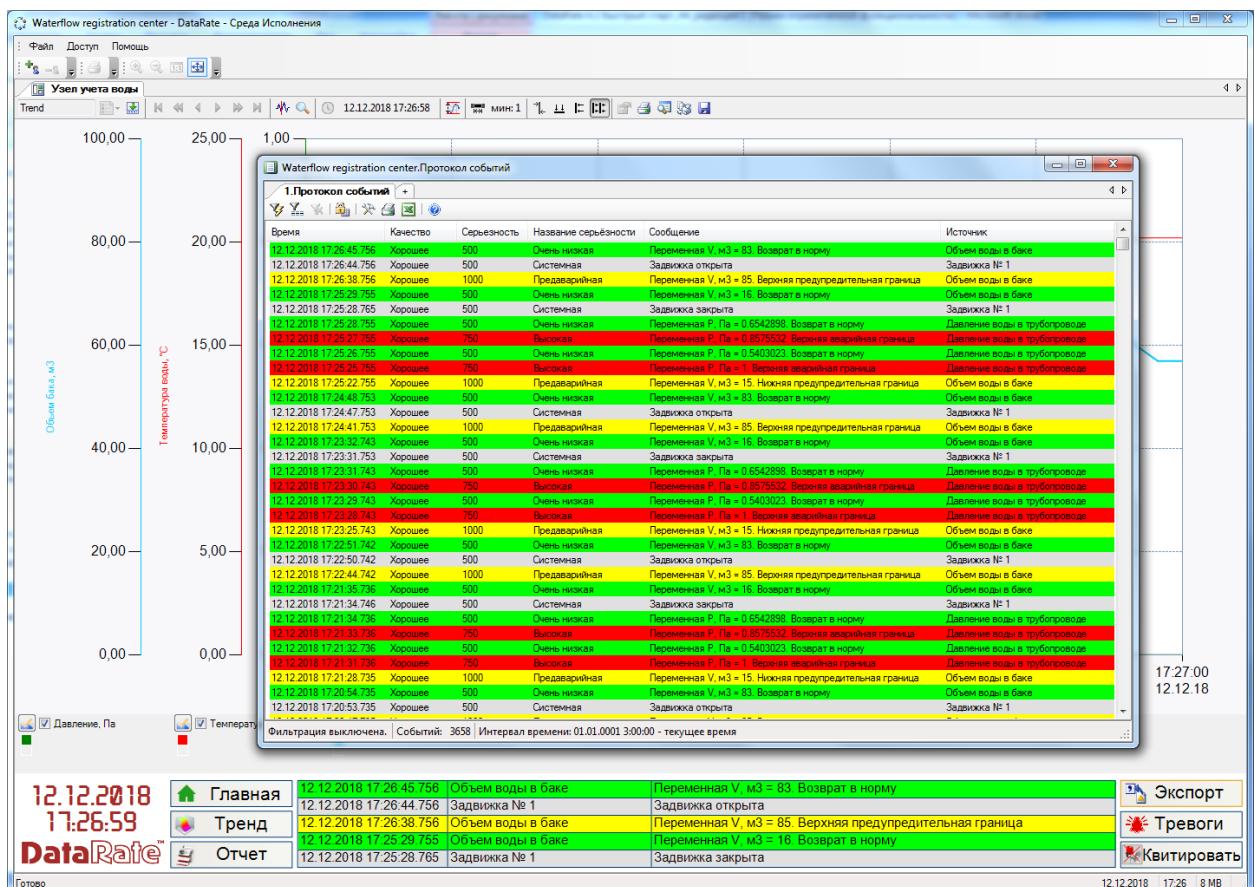


Рисунок 3.8 – Среда исполнения. Протокол событий в отдельном окне

Сообщения в окне Протокол событий соответствуют событиям автоматизированного процесса

- 6 Щелчком по кнопке **Экспорт** инициируем создание CSV-файла с данными об изменениях значений давления, температуры и объема жидкости в баке. Фрагмент сформированного файла (<системный диск>:\Users\<Пользователь>\Documents\ DataRate Projects\Демо проекты 4.2\ Узел учёта воды\Export\ Экспортер в CSV_2018_12_12_10_55_32_063.csv) показан на рисунке 3.9. Стока файла содержит время и пары «значение/качество» (качество 192 – Хорошее) для давления (колонки B, C), температуры(колонки D, E) и объема(колонки F, G).

Экспортер в CSV_2018_12_12_10_55_32_063 - Mi...

Главная Вставка Разметка Формул Данные Рецензия Вид Надстрока Team Редактировать

Шрифт Выравнивание

Число Стили Ячейки Редактировать...

A1 10:55:31 12 декабря 2018

	A	B	C	D	E	F	G	H
1	10:55:31 1	0,739085	192	20,11732	192	55	192	
2	10:55:30 1	0,739085	192	20,30276	192	54	192	
3	10:55:29 1	0,739085	192	20,11277	192	53	192	
4	10:55:28 1	0,739085	192	20,30734	192	52	192	
5	10:55:27 1	0,739085	192	20,10795	192	51	192	
6	10:55:26 1	0,739085	192	20,31219	192	50	192	
7	10:55:25 1	0,739085	192	20,10285	192	49	192	
8	10:55:24 1	0,739085	192	20,31732	192	48	192	
9	10:55:23 1	0,739085	192	20,09745	192	47	192	
10	10:55:22 1	0,739085	192	20,32274	192	46	192	
11	10:55:21 1	0,739085	192	20,09173	192	45	192	
12	10:55:20 1	0,739085	192	20,3285	192	44	192	
13	10:55:19 1	0,739085	192	20,08564	192	43	192	
14	10:55:18 1	0,739085	192	20,33461	192	42	192	
15	10:55:17 1	0,739085	192	20,07916	192	41	192	
16	10:55:16 1	0,739085	192	20,34111	192	40	192	
17	10:55:15 1	0,739085	192	20,07226	192	39	192	
18	10:55:14 1	0,739085	192	20,34803	192	38	192	
19	10:55:13 1	0,739085	192	20,06488	192	37	192	

Экспортер в CSV_2018_12_12_10_5 Готово 100%

ЗАКЛЮЧЕНИЕ

DataRate предоставляет богатые возможности для создания HMI-приложения. Насколько гибко и эффективно сможет использовать их разработчик зависит во многом от его опыта и, конечно, знания **DataRate**.

В рассмотренных примерах мы дали описание далеко не всех возможностей графической подсистемы **DataRate**. Но даже эти простые примеры показывают, что свойства анимации и реакции объектов в **DataRate** позволяют реализовать графический интерфейс практически любой сложности и функциональности.

Обмен данными в HMI-приложении **Waterflow registration center** с помощью **OPC DA коннектора** демонстрирует, как получить доступ к внешнему источнику данных – устройству, которое имеет свой **OPC DA сервер**.

В примерах не рассматривается чтение и обработка исторических трендов с помощью OPC HDA коннектора, так как, с точки зрения **DataRate**, работа с OPC HDA коннектором аналогична работе с OPC DA коннектором. Вопросы обработки исторических трендов требуют, прежде всего, знания спецификации OPC HDA и конкретного OPC HDA сервера.

Предлагаемый в примерах способ создания приложения продиктован использованием библиотечных объектов. Это позволяет, с одной стороны, быстро тиражировать функционально подобные объекты из библиотеки, а с другой – легко адаптировать работу приложения к различным внешним источникам данных (достаточно перепривязать внутренние теги к тегам необходимого коннектора).

Однако, также легко можно привязать свойства графических примитивов непосредственно к тегам коннектора, не создавая внутренние теги.

Выбор остается за разработчиком.

За разработчиком остается и выбор последовательности действий при создании HMI-приложения.

Сначала можно создать внутренние теги и графический интерфейс, затем создать коннекторы к источникам данным и привязать внутренние теги к внешним.

Можно начинать разработку с создания коннекторов.

Если приложение содержит много однотипных объектов (например, датчиков), то разработку HMI-приложения следует начинать с создания библиотечных типов тегов и библиотечных объектов (как их использовать показано в наших примерах).

Удачи и успехов Вам на пути освоения **DataRate** – дорогу осилит идущий!

ДОПОЛНИТЕЛЬНАЯ СПРАВОЧНАЯ ИНФОРМАЦИЯ

Если Вы хотите получить больше информации о том, как ...		Смотрите раздел справки <i>DataRate</i>
Установить <i>DataRate</i>		Введение/ Инсталляция и начало работы
ОБЪЕКТНАЯ МОДЕЛЬ /		
Создать проект		Проект
Создать элемент проект		Элемент проекта
Защитить элемент проекта		Элемент проекта/ Защита элемента проекта
Использовать библиотеки <i>DataRate</i>		Библиотеки
Создать объект		Объекты
Создать и привязать теги		Объекты/ Теги
Создать и использовать скрипт		Объекты/ Скрипты
Использовать аргументы		Объекты/ Аргументы
ДИНАМИЧЕСКАЯ ВИЗУАЛИЗАЦИЯ ДАННЫХ /		
Создать мнемосхему		Графическая подсистема/ Мнемосхема
Использовать графические примитивы для визуализации		Графическая подсистема/ Графические примитивы
Создать рабочий стол		Графическая подсистема/ Рабочие столы
Настроить анимацию графических элементов		Анимация Примеры использования реакций и анимаций
Использовать реакции для управления динамическими свойствами объектов		Реакции Примеры использования реакций и анимаций
СОБЫТИЯ И ТРЕВОГИ /		
Сформировать и настроить протокол событий		Протокол событий
Визуализировать протокол событий		Протокол событий/ Примитив протокол событий Протокол событий/ окно Протокол событий элемента проекта
Назначить и настроить сигнализацию		Сигнализация; Оповещения
ИСТОРИЯ ПРОЦЕССА. ТРЕНДЫ /		
Организовать трендирование		Модуль Менеджер трендов
Визуализировать тренды		Графический элемент Тренд Графический элемент Табличный тренд;
Отобразить данные в виде «гладких» графиков		Аналитический тренд

Если Вы хотите получить больше информации о том, как ...	Смотрите раздел справки <i>DataRate</i>
ОБМЕН ДАННЫМИ /	
Использовать <i>DataRate</i> коннекторы для обмена данными	OPC DA-коннектор; OPC HDA-коннектор; OPC АЕ-коннектор SQL-коннектор
Как экспорттировать данные	Экспорт данных
РАСПИСАНИЕ /	
Создать и настроить расписание	Расписание
МОДУЛЬ ОТЧЕТОВ /	
Создать и использовать шаблон отчета	Шаблон отчета
Управлять хранением, печатью, рассылкой и публикацией отчетов	Менеджер отчетов
Сформировать отчет по расписанию	Генерация отчетов
СРЕДА ИСПОЛНЕНИЯ / СРЕДА ИСПОЛНЕНИЯ КЛИЕНТ / WEB-КОНТРОЛЬ /	
Выполнить проект в Среде исполнения или в Web-браузере	Среда исполнения; Среда исполнения Клиент Web-контроль

ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

Если у Вас есть вопросы или проблемы использования **DataRate**, пожалуйста, обратитесь в службу технической поддержки ООО «Энергокруг»:

E-mail:

Общие вопросы: info@scadadatarate.ru

Техническая поддержка: support@scadadatarate.ru

Web:

<http://www.ScadaDataRate.ru>