

Министерство науки и высшего образования Российской Федерации

Рязанский институт (филиал)  
федерального государственного бюджетного образовательного учре-  
ждения высшего образования  
«Московский политехнический университет»

Кафедра «Механико-технологические дисциплины»

**Лопатин Е.И. Мельников А.Ю.**

**Автоматизация технологических процессов  
(в электроэнергетике)**

Методичное указание по выполнению лабораторных работ

**Рязань  
2020**

**УДК 621.002(075.8)**

**ББК 34.5Я73**

**Е.И. Лопатин, А.Ю. Мельников**

**«АТП»: методические указания / Е.И. Лопатин., А.Ю. Мельников  
Рязань: Рязанский институт (филиал) Московского политехнического университета, 2020. – 94 с.**

Пособие предназначено для студентов всех форм обучения специальностей 13.04.02 «Электроэнергетика и электротехника».

**Печатается по решению методического совета Рязанского института (филиала) Московского политехнического университета.**

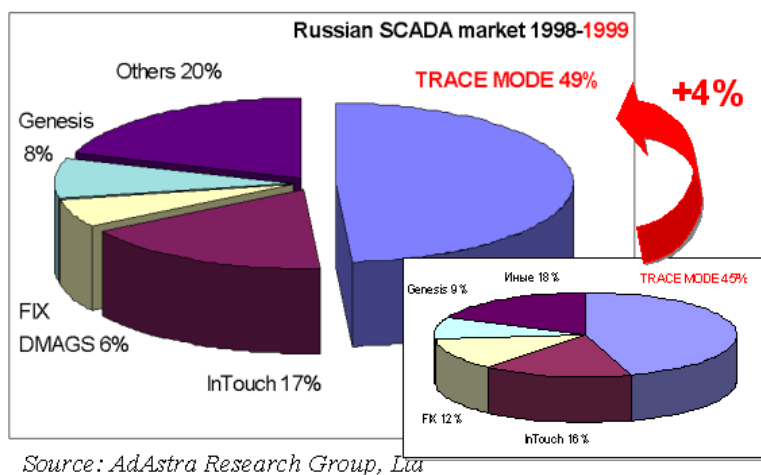
**Версия программы ТРЕЙС МОУД является учебной и не требует приобретения лицензий, может быть установлена на личном ПК студента, для обеспечения проведения работ в дистанционном форме.**

© Рязанский институт (филиал)  
Московского политехнического  
университета, 2020

## ВВЕДЕНИЕ

Внедрение автоматизированных систем контроля и учета энергоресурсов (АСКУЭ) позволяет потребителю снизить финансовые затраты как за счет снижения затрат энергии и ее сверхлимитного расходования, так и за счет перехода на более выгодные для потребителя тарифные системы.

Выбор пакета **ТРЕЙС МОУД**, как инструмента реализации АСКУЭ, обусловлен: наличием русской версии программы; положительной динамикой числа инсталляций по России; качеством проработки программного продукта сравнимого с зарубежными аналогами (*Genesis*); экономическими факторами.



Версия программы **ТРЕЙС МОУД 6.05** дает интегрированный инструмент для программирования контроллеров (*Softlogic*), разработки операторского интерфейса (*SCADA – Supervisory Control And Data Acquisition*), управления производственными заданиями (*MES – Manufacturing Execution System*), управления персоналом (*HRM – Human Resources Management*) и основными фондами (*EAM – Enterprise Asset Management*).

Версия программы **ТРЕЙС МОУД** является учебной и не требует приобретения лицензий, может быть установлена на личном ПК студента, для обеспечения проведения работ в дистанционном форме.

Лабораторные работы подготовлены на основе справочного материала фирмы **AdAstra**. Он был подвергнут редактированию, заключающемуся в уменьшении объема и исправлении обнаруженных ошибок. В них были заменены некоторые данные для выполняемых заданий.

Виртуальный компьютерный практикум включает в себя вопросы построения графических экранов оператора АРМ (автоматизированное рабочее место), автопостроения каналов, программирования, создание отчета тревог, архивирования данных, связи оболочки со средствами *MS Office* по *DDE (Dynamic Data Exchange)* протоколу.

Данный виртуальный лабораторный практикум охватывает основные разделы дисциплины «АТП» направления 13.04.02 – «Электроэнергетика и электротехника»

## ЛАБОРАТОРНЫЕ РАБОТЫ

### Лабораторная работа №1

#### СОЗДАНИЕ ЭЛЕМЕНТАРНОГО АРМ

Цель работы: создание элементарного АРМ, выполняющего функции сканирования, модификации и хранения информации периферийных датчиков.

#### Указания к выполнению работы

В общем случае, система АСУ на предприятии включает в себя несколько уровней:

- уровень контрольно-измерительных приборов и аппаратуры,
- уровень контроллеров (традиционные ПЛК и *Softlogic*),
- уровень рабочих станций (АРМ).

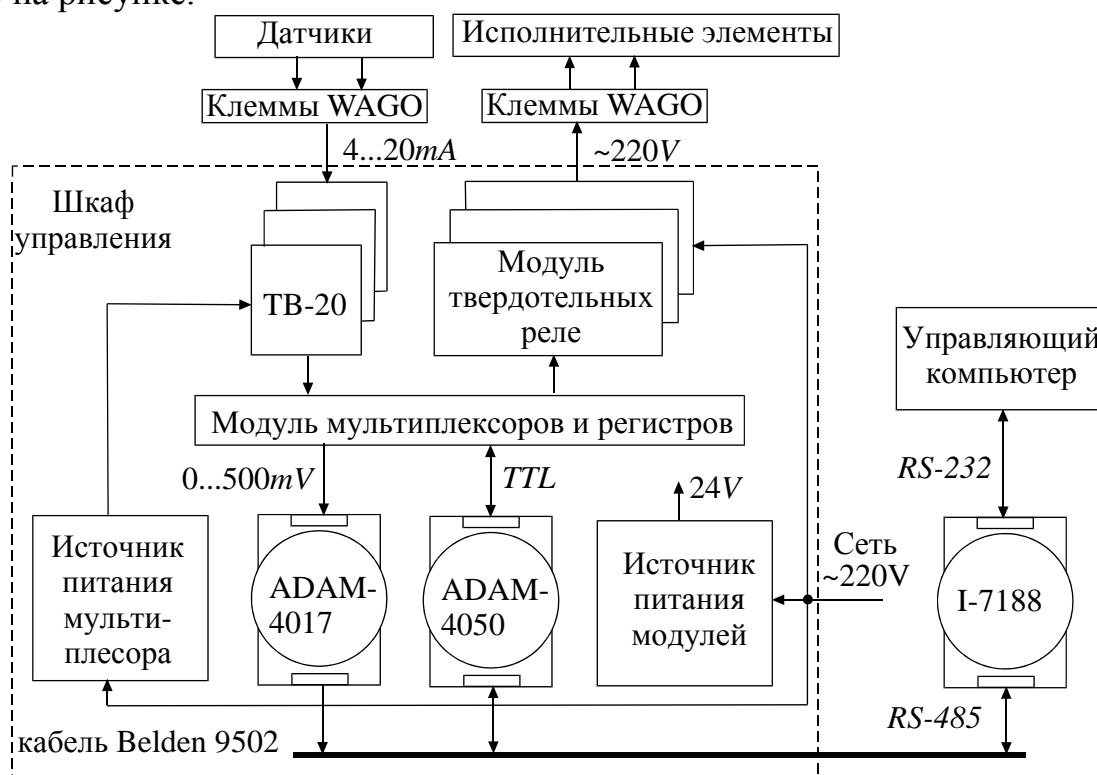
Программное обеспечение (ПО) уровня *SCADA* выполняет следующие функции:

- Визуализация – отображение информации о процессе на мнемосхемах в виде числовой информации, трендах, анимации, *ActiveX* компонентов и т.д. Также обеспечивается получение управляющих воздействий от операторов;
- Обмен с контроллерами в реальном времени – обмен информации с контроллерами по последовательному интерфейсу, сети *Ethernet*;
- Архивирование – сохранение информации;
- Документирование процесса – создание отчётов по заранее созданным шаблонам с последующей их передачей по разным каналам обмена;
- Получение и передача информации в другие программные пакеты – обмен оперативными и архивными данными со специализированным ПО;
- Неоперативная обработка информации – вычисление статистической информации, управление небыстрыми процессами;
- Управление тревогами – список тревог, фильтр событий, квитирование.

Термин *Softlogic* подразумевает использование контроллеров с *PC*-совместимой архитектурой. Программирование таких контроллеров осуществляется на языках типа ассемблер или *C++*, а может осуществляться с помощью инструментальных пакетов на визуальных языках верхнего уровня, например, на

языке функциональных блоков или языке инструкций, совместимых со стандартом МЭК61131-3.

Структурная схема системы управления может быть представлена так, как показано на рисунке.



Управляющие функции выполняет контроллер связи *I-7188*, который имеет ОЗУ 256 кбайт, флэш-память 512 кбайт, порт консоли, последовательные порты *RS-232* и *RS-485*, сторожевой таймер, часы реального времени и совместим с операционной системой *ROM-DOS*. По каналу *RS-232* контроллер соединен с управляющим компьютером, а по каналу *RS-485* (кабель *Belden 9502*) – со шкафом управления.



В шкафу установлены модули аналогового ввода *ADAM-4017* и цифрового ввода-вывода *ADAM-4050* фирмы *Advantech*, модуль мультиплексора и регистров, модули твердотельных реле (*Cosmo*), источники питания. Для подключения аналоговых сигналов к модулю мультиплексора используются клеммные платы *TB-20* (*Fastwel*). Шкаф управления соединен кабелями с датчиками и исполнительными механизмами.

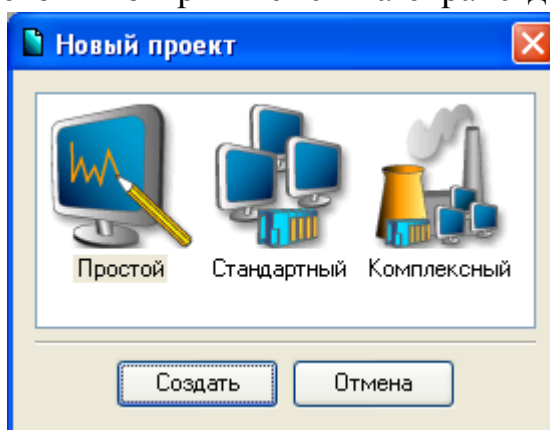
Модуль мультиплексора и регистров позволяет увеличить количество аналоговых и цифровых входов

Во флеш памяти контроллера при монтаже системы записываются параметры технологического оборудования, параметры ПИД-регуляторов, нормализаторов входных сигналов, цифровых фильтров, таблица логических каналов, ключи доступа и т.д. Программное обеспечение контроллера может быть написано на

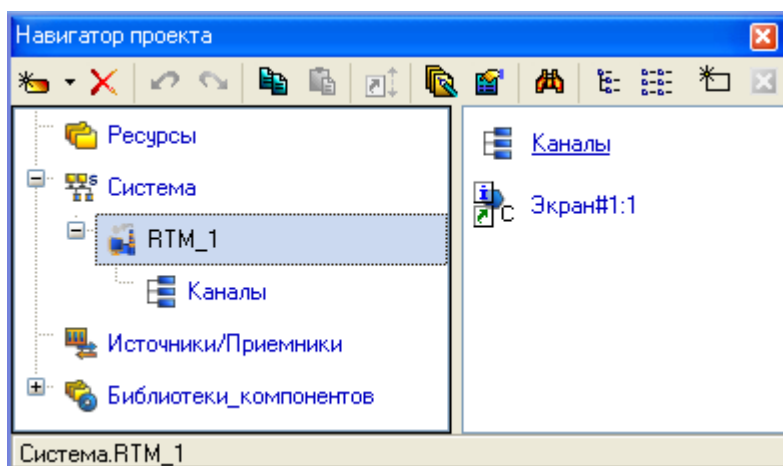
языке C++. Управляющий компьютер под управлением SCADA системы обеспечивает цветное графическое отображение работы оборудования.

## Описание виртуальной лабораторной установки

Загрузить инструментальную систему двойным щелчком левой клавиши (ЛК) мыши по иконке  *TRACE MODE IDE 6 Base*, расположенной на рабочем столе Windows. С помощью иконки **Создать новый проект** инструментальной панели  выполнить это действие, при этом в открывшемся на экране диалоге выберем **Простой** стиль разработки (либо выбор меню **Файл – Настройки – Уровень сложности – простой**). После нажатия ЛК мыши на экранной кнопке **Создать**, в левом окне навигатора проекта появится дерево проекта с заданным узлом **АРМ RTM\_1**. В правом окне Навигатора проекта отобразится держимое узла – пустая группа **Каналы** и один канал класса **Вызов Экран #1:1**, предназначенный для отображения на узле **АРМ** графического экрана.



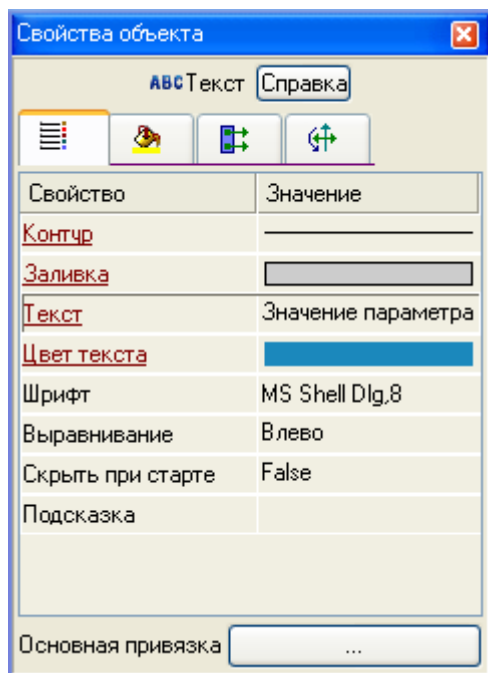
ИС  
По-  
  
ра  
со-  
  
со-  
и




Для перехода в режим редактирования шаблона по двойному нажатию ЛК на канале **Вызов** необходимо в общих настройках интегрированной среды разработки (меню **Файл – Настройки ИС**) в разделе **Уровень сложности** активировать пункт **Открывать шаблон**.

Для автоматического вывода окна свойств Графического Элемента (ГЭ) по завершению его размещения необходимо в общих настройках интегрированной среды разработки в разделе **РПД** (редактор представления данных) активизировать пункт **«Открывать свойства автоматически»**.


Выделить компонент «Экран#1:1» в узле **RTM\_1** и по нажатию **ПК** мыши



вызвать контекстное меню, в нем выбрать **ЛК** пункт **Редактировать шаблон**, после чего откроется окно графического редактора.

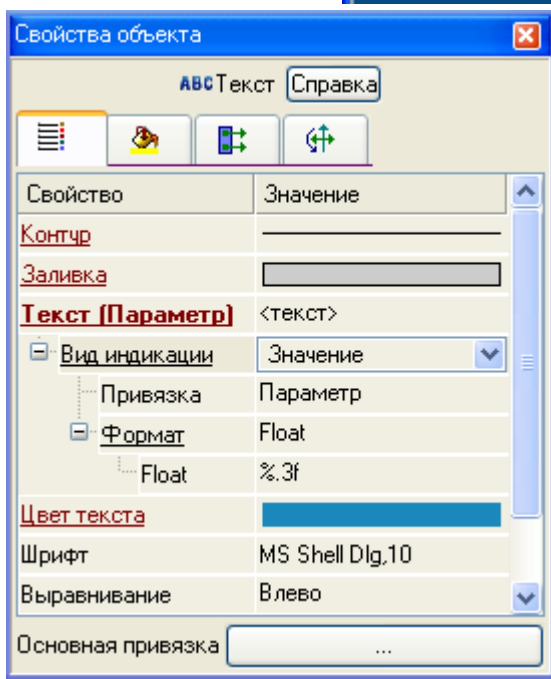
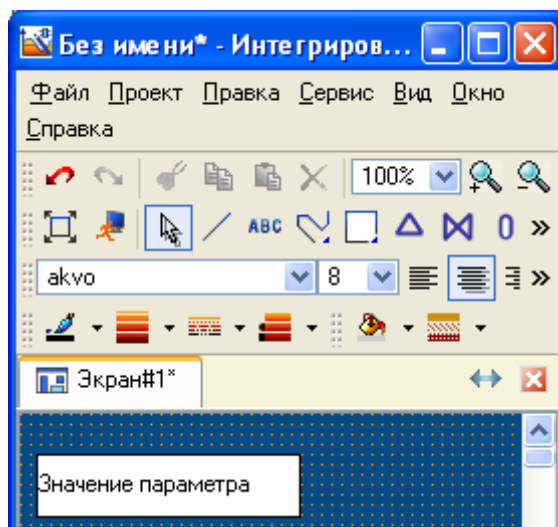
На панели графического редактора выделить иконку ГЭ «Текст» ; на поле редактора установить прямоугольник ГЭ, для чего зафиксировать **ЛК** «точку привязки»; развернуть прямоугольник движением курсора и зафиксировать выбранный ГЭ.

Для перехода в режим редактирования элемента выделить на панели инструментов


иконку  с подсказкой **Режим редактирования**; двойным щелчком **ЛК** по размещенному ГЭ открыть окно его свойств и в правом поле строки «Текст» набрать «Значение параметра».

Закрывать окно свойств, ГЭ будет иметь вид:

Закрывать окно свойств, ГЭ будет иметь вид:



Если введенный Вами текст не уместился в прямоугольнике ГЭ, выделите его и растяните до нужного размера с помощью мыши.

Создать и разместить новый ГЭ  справа от ГЭ «Значение параметра»:

двойным щелчком **ЛК** на строке «Текст» вызвать меню **Вид индикации**. В правом поле

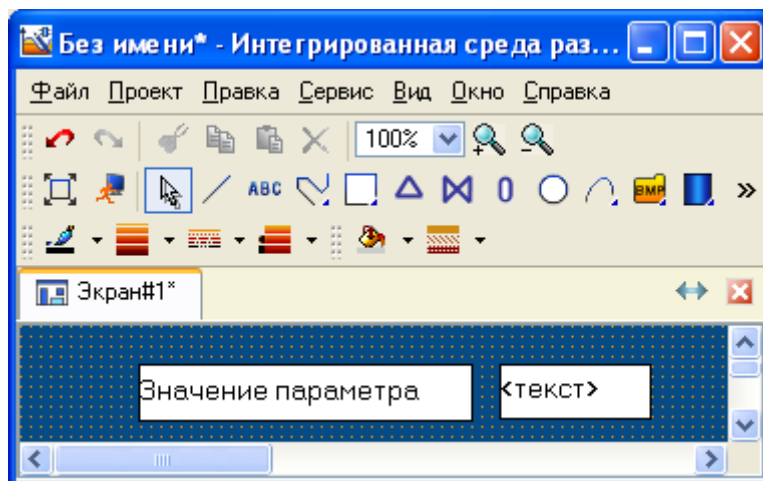
- в открывшемся меню настройки параметров динамизации выбрать свойство «Привязка» и в открывшемся окне «Свойства привязки» нажать кнопку

- двойным щелчком **ЛК** выделить имя аргумента и изменить его, введя с клавиатуры «Параметр» (завершить ввод нажатием клавиши *Enter*);



[illegible]

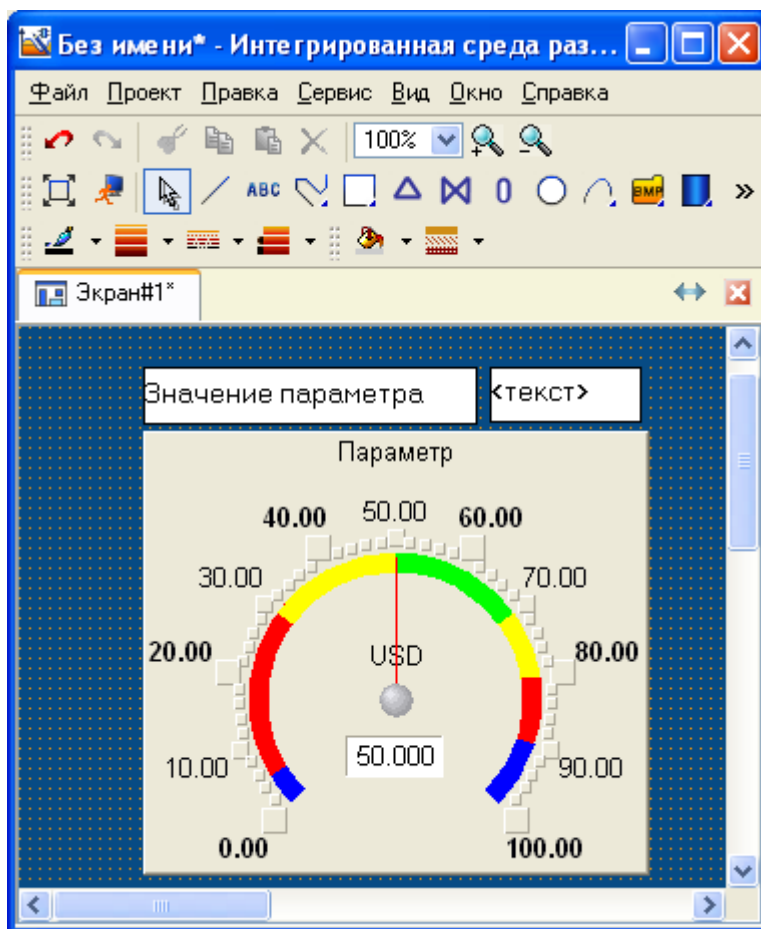
8






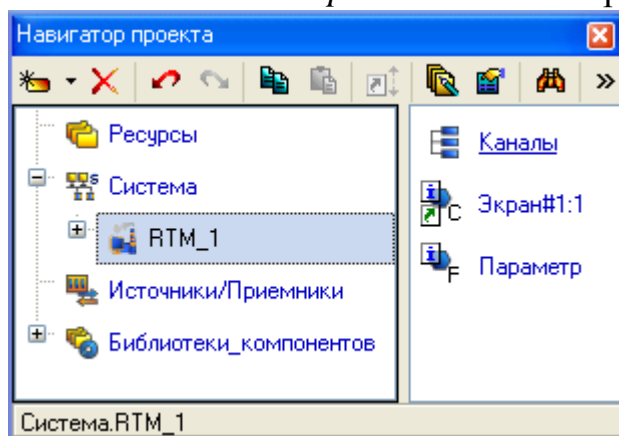
Создадим образ стрелочного прибора на экране, для этого необходимо:

- выделить двойным щелчком **ЛК** на инструментальной панели редактора графики иконку с подсказкой «**ползунок**»  и выбрать в появившемся меню иконку стрелочного прибора ;
- установить ГЭ «**Стрелочный прибор**», выбрав его размер таким образом, чтобы все элементы графики и текста на нем были разборчивы и симметричны;
- перейти в режим редактирования и открыть окно свойств ГЭ «**Стрелочный прибор**»;
- двойным щелчком **ЛК** на свойстве «**Отображаемая величина**» открыть окно настроек, выбрать свойство «**Привязка**» и в открывшемся окне вызвать диалог «**Свойства привязки**», в котором выделен уже имеющийся аргумент. Подтверждением связи (кнопка **Готово**) закрыть окно «**Свойства привязки**»;
- двойным щелчком **ЛК** открыть свойство **Заголовок** и в строке «**Текст**» ввести слово – Параметр;
- двойным щелчком **ЛК** открыть свойство **Единицы** и в строке «**Текст**» ввести сокращение «**USD**»;
- закрыть окно свойств. Экран примет вид, указанный на рисунке.



Рассмотрим процесс автоматического порождения канала класса *Float* «Параметр»;




- В слое «Система» навигатора проекта открыть узел **RTM\_1**;
- Выделить в правой половине навигатора проекта компонент «Экран#1:1» при помощи **ПК** вызвать контекстное меню, где выбрать ЛК позицию «Свойства»;
- Во вкладке «Аргументы» выделить ЛК аргумент «Параметр», и с помощью иконки  создать канал *Float* типа *Input* с именем «Параметр».



Создадим генератора синуса и привяжем его к каналу **Параметр**, для этого необходимо:

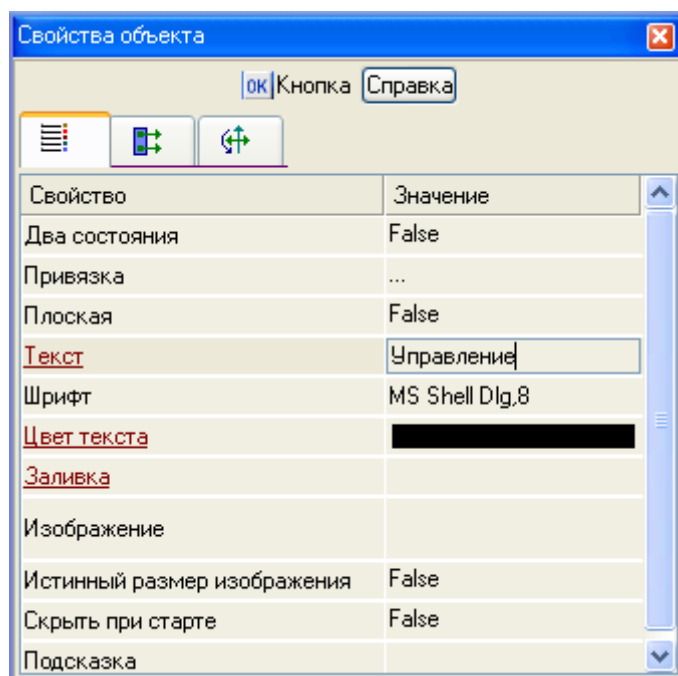
- открыть слой **Источники/Преемники** и через **ПК** создать в нем группу **Генераторы**;
- открыть группу **Генераторы** и через **ПК** создать в ней компонент **Синусоида**;
- захватить с помощью **ЛК** компонент **Синусоида**, и, не отпуская **ЛК**, перетащить курсор на узел **RTM\_1**, откроется палитра компонентов узла, а курсор установить на канал **Параметр**. Отпустить **ЛК**.

Запустим проект следующим образом:


- Сохраним проект с помощью иконки «**Сохранить текущий проект**» , задав имя проекта латинским шрифтом, и расположив файл в папке *DEMO*;
- на инструментальной панели выбрать команду «**Сохранить для МРВ**» (монитор реального времени)  и скомпилировать проект для запуска в реальном времени;
- **ЛК** выделить в слое система узел **RTM\_1**, выбрать иконку,  соответствующую команде «**Запустить профайлер**» на инструментальной панели, и активировать режим исполнения.

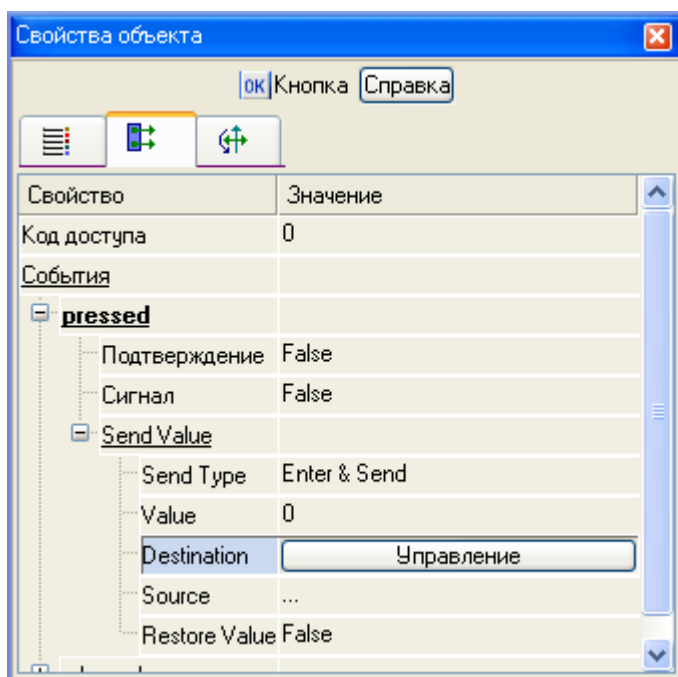
Осуществим редактирование экрана, для этого необходимо:

- выбрать на инструментальной панели графического редактора иконку ГЭ

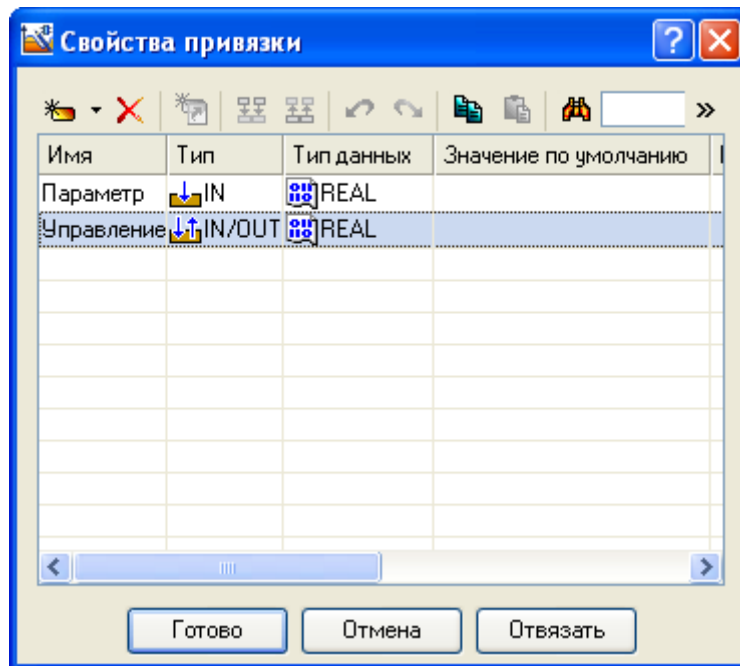


**Кнопка** –  ;

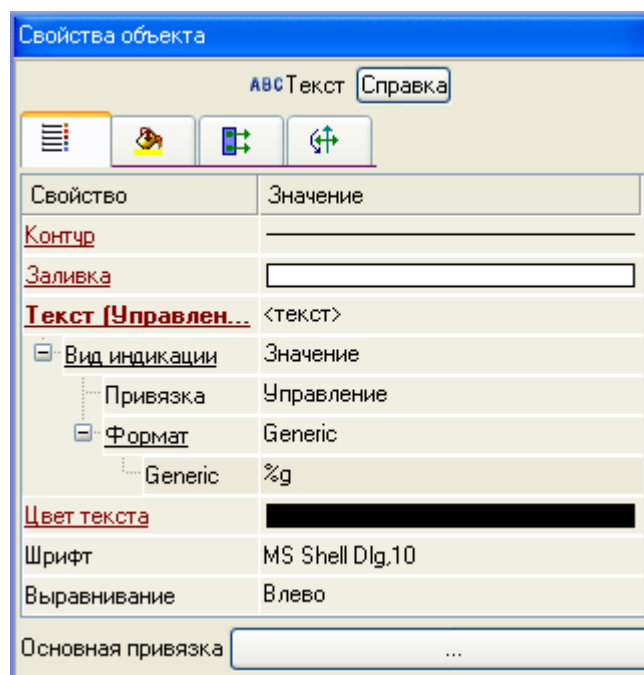
- с помощью мыши разместить его в поле графического экрана под ГЭ «**Стрелочный прибор**»;
- перейти в режим редактирования и вызвать окно свойств ГЭ «**Кнопка**»;
- в поле «**Текст**» ввести «**Управление**»;
- открыть бланк «**События**»  и ПК раскрыть меню «**По нажатию**» (*pressed*);
- выбрать из списка команду «**Послать значение**» (*Send Value*), раскрыть меню настроек выбранной команды, в поле «**Тип посылки**» (*Send Type*) выбрать из списка «**Ввести и послать**» (*Enter & Send*);




- ЛК в поле «**Назначение**» (*Destination*) вызвать табличный редактор аргументов. Создать еще один аргумент и задать ему имя «**Управление**»;
- изменить тип аргумента на *IN/OUT*, кнопкой «**Готово**» подтвердить привязку атрибута ГЭ к этому аргументу;

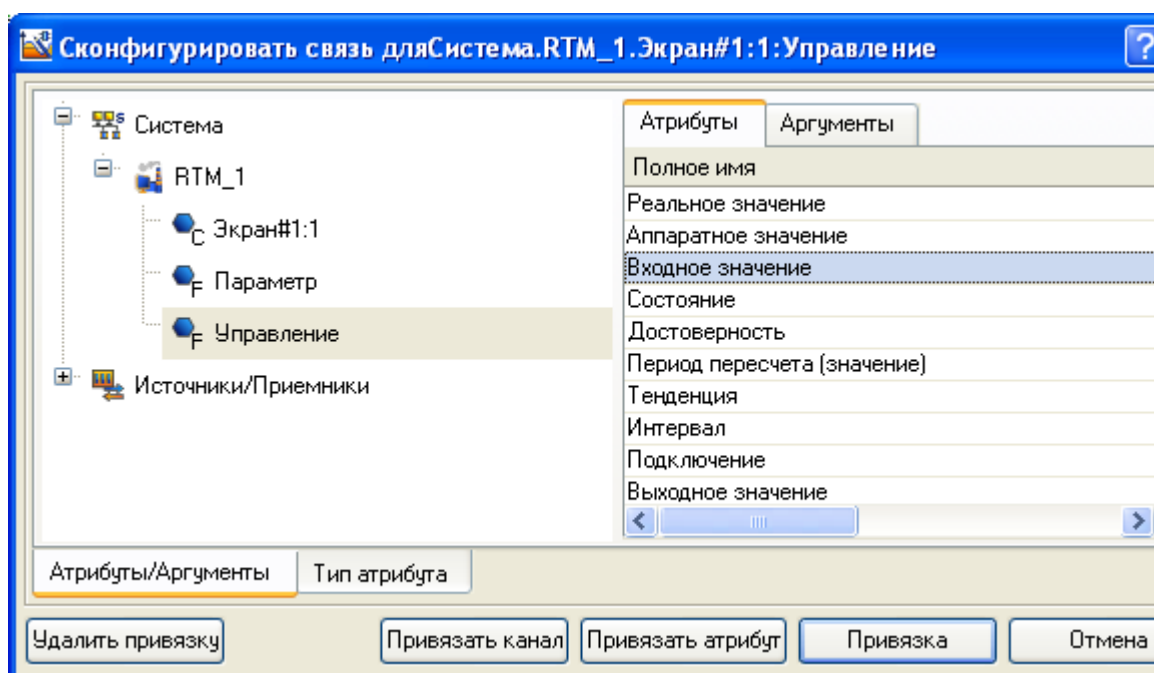
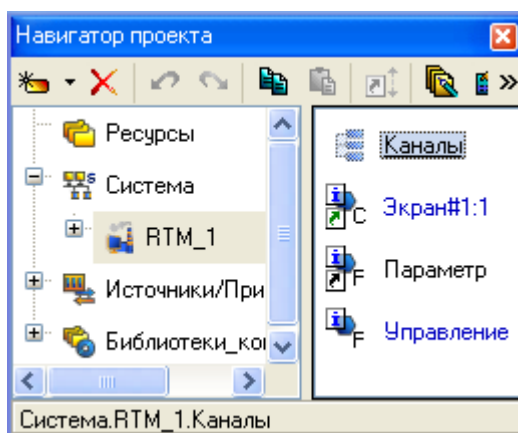


- закрыть окно свойств ГЭ;
- выделить **ЛК** имеющийся ГЭ, отображающий значение «Параметра»;
- с помощью контекстного меню **ПК** скопировать выделенный ГЭ в буфер обмена;
- с помощью контекстного меню **ПК** извлечь копию ГЭ из буфера обмена на графический экран;
- переместить копию ГЭ справа от кнопки и снять с него выделение клавишей *ESC*;
- выделить **ЛК** ГЭ и открыть окно его свойств;
- двойным щелчком **ЛК** на строке «Текст» перейти к настройке его атрибутов, где через правое поле строки «Привязка» открыть табличный редактор аргументов экрана;
- выделить в списке аргумент «Управление» и кнопкой «Готово» подтвердить привязку атрибута ГЭ к данному аргументу;
- закрыть окно свойств ГЭ и окно графического редактора.




Привяжем второй аргумент экрана к каналу **Управление**, для этого необходимо:


- В слое **Система** открыть узел **RTM\_1**;
- с помощью **ПК** вызвать через контекстное меню свойства компонента «Экран#1:1»;
- выбрать вкладку **Аргументы**, **ЛК** выделить аргумент «Управление» и с помощью иконки  создать новый канал;
- в узле **RTM\_1** образуется новый канал с именем «**Управление**»;
- двойным щелчком в поле «**Привязка**» аргумента «Управление» вызвать окно настройки связи, выбрать в нем атрибут «**Входное значение**»;
- кнопкой **Привязка** подтвердить связь аргумента экрана «Управление» с атрибутом «Входное значение» канала «Управление»;

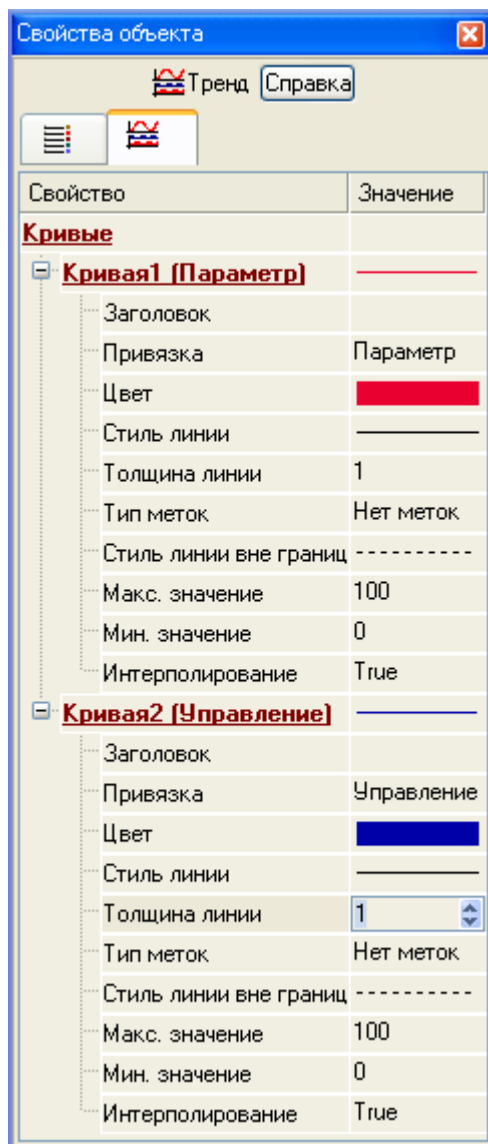


- закрыть окно свойств компонента «Экран#1:1».

В группу графики входит единственный элемент **Тренд**. ГЭ **Тренд** предназначен для отображения на графике изменения значения привязанных аргументов во времени.


В правой части графического экрана разместим ГЭ **Тренд**  для вывода значений каналов «Параметр» и «Управление». Основные свойства ГЭ оставим


заданными по умолчанию. Перейдем во вкладку «**Кривые**»  и, выделив ЛК строку **Кривые**, с помощью ПК создадим еще одну линию. Настроим их привязки к аргументам, толщину и цвет линий (цвет линий должен совпадать с цветом числовых обозначений соответствующих аргументов).



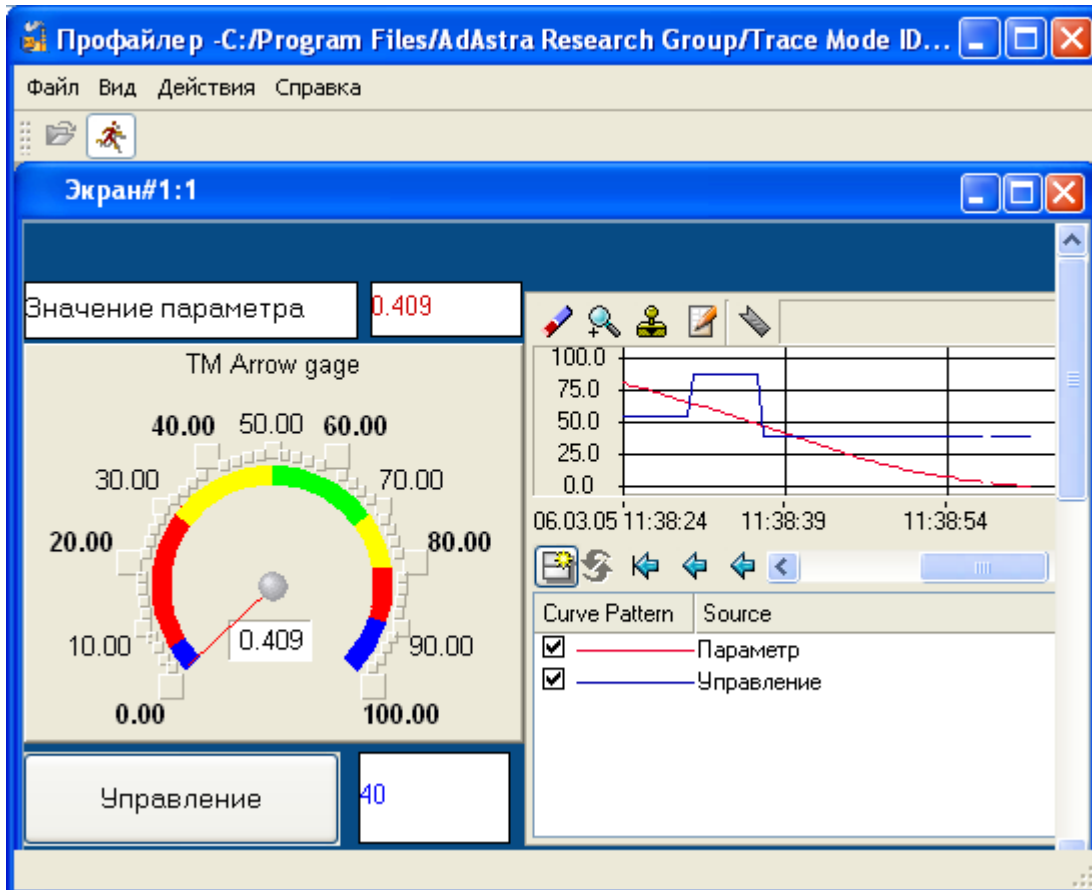
Осуществить сохранение проекта и запуск в соответствии с требованиями, изложенными выше:


- на инструментальной панели выбрать команду «**Сохранить для МРВ**»

(монитор реального времени)  и скомпилировать проект для запуска в реальном времени;

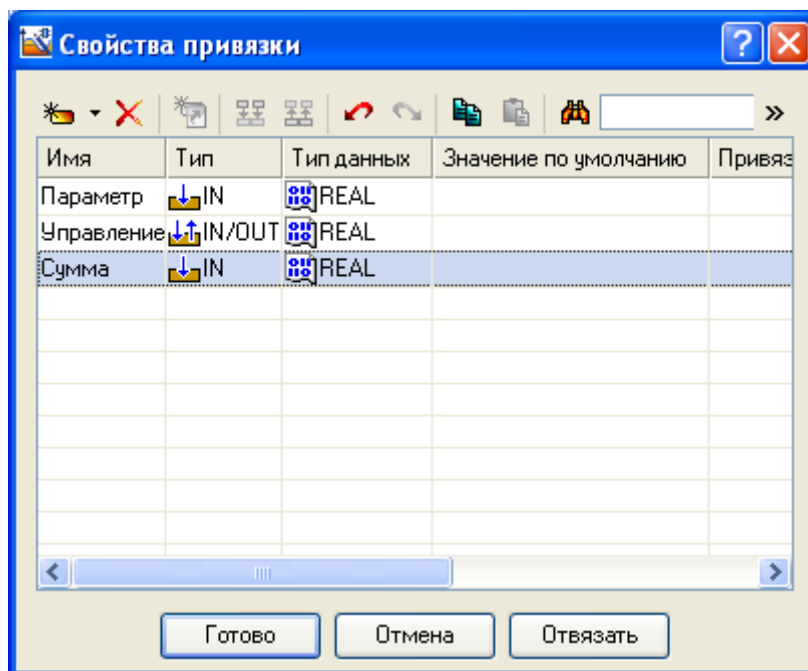
- ЛК выделить в слое система узел **RTM\_1**, выбрать иконку , соответствующую команде «Запустить профайлер» на инструментальной панели и активизировать режим исполнения.

С помощью кнопки **Управление** ввести величину управляющего воздействия и наблюдать результат на соседнем индикаторе и тренде.

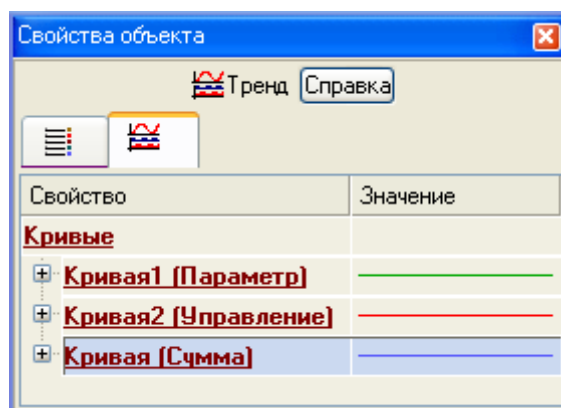


Доработаем графический экран. Скопировать два первых ГЭ  и разместить их ниже ГЭ «Кнопка». Изменить статический текст первого ГЭ на «Сумма», а динамику второго ГЭ привязать к третьему аргументу экрана типа *IN* с именем **Сумма**, который создать в процессе привязки.






Добавим еще одну кривую на тренд с привязкой к аргументу **Сумма**.



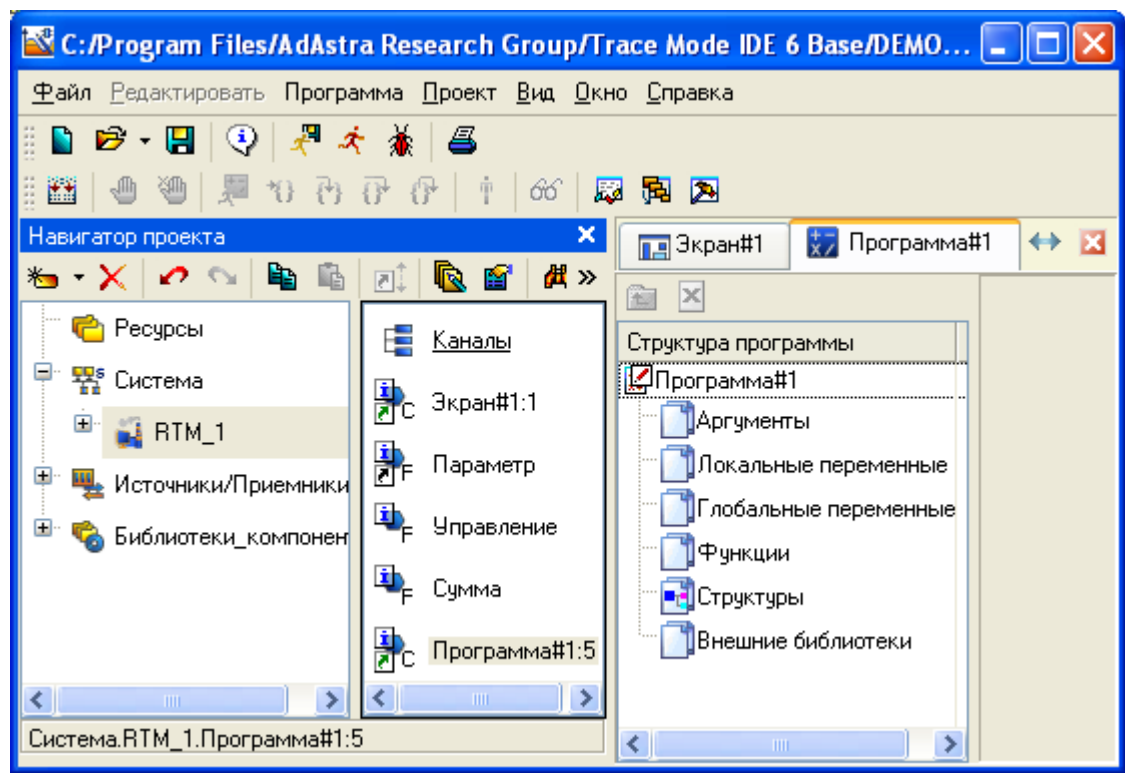
Создадим канал **Сумма** для третьего аргумента экрана:


- С помощью **ПК** вызвать контекстное меню для компонента «Экран#1:1» и открыть его свойства;
- выбрать вкладку **Аргументы**;
- выделить **ЛК** аргумент «Сумма» и с помощью иконки  создать новый канал:

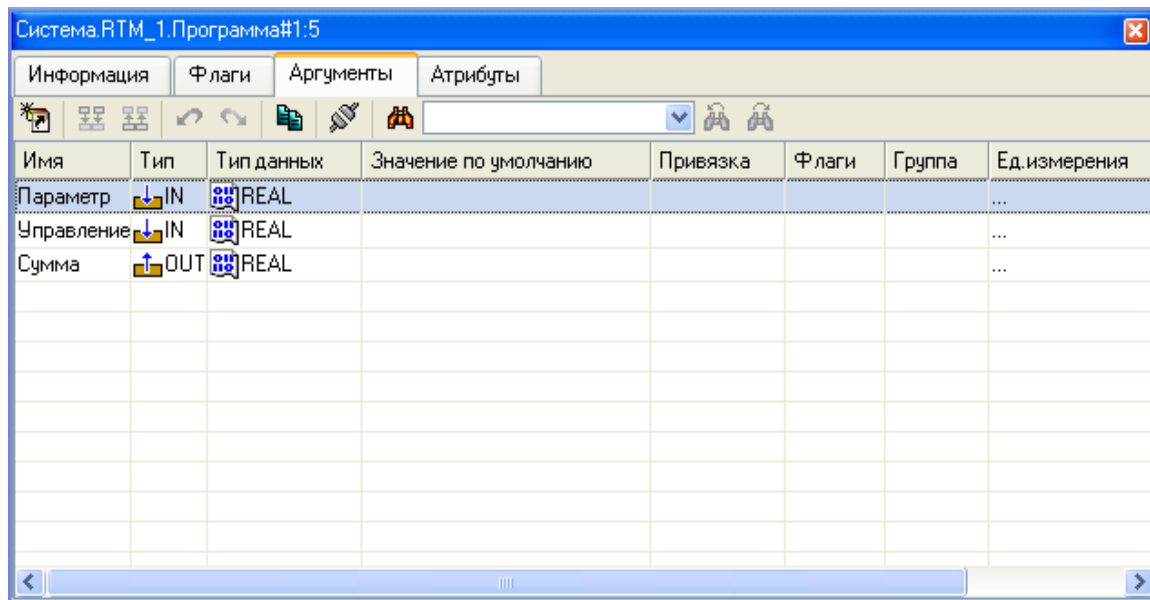
Система.RTM_1.Экран#1:1				
Информация    Флаги    Аргументы    Атрибуты				
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Параметр	↓ IN	REAL		Параметр:Реальное значение(Система)
Управление_In	↑ IN/OUT	REAL		Управление:Входное значение(Система)
Сумма	↓ IN	REAL		Сумма:Реальное значение(Система)

Создадим программу на языке **Техно ST (Structured text)**, в которой сумма двух аргументов с именами «Параметр» и «Управление» помещаются в третий с именем «Сумма»:

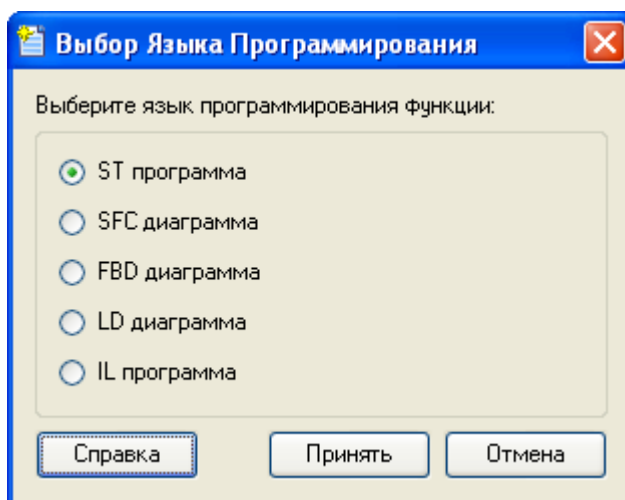
- ЛК открыть узел **RTM\_1**;
- создать в нем компонент «**Программа#1**» и двойным щелчком ЛК вызвать контекстное меню дерева шаблона компонента «**Программа#1**», выбрав в котором ЛК строку «**Аргументы**», вызвать табличный редактор;



- иконкой  с подсказкой **Создать аргумент** выполнить это действие, создав три аргумента с именами «Параметр», «Управление» и «Сумма». При этом первые 2 аргумента должны быть типа *IN*, а третий – *OUT*;





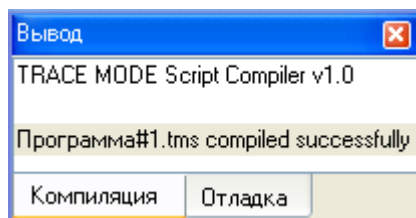
- выделить в дереве шаблона строку «Программа#1», и в открывшемся диалоге **Выбор языка программирования** выбрать язык *ST*:



- по нажатию экранной кнопки **Принять** в открывшемся окне редактора программ с объявленными переменными набрать следующую строку:

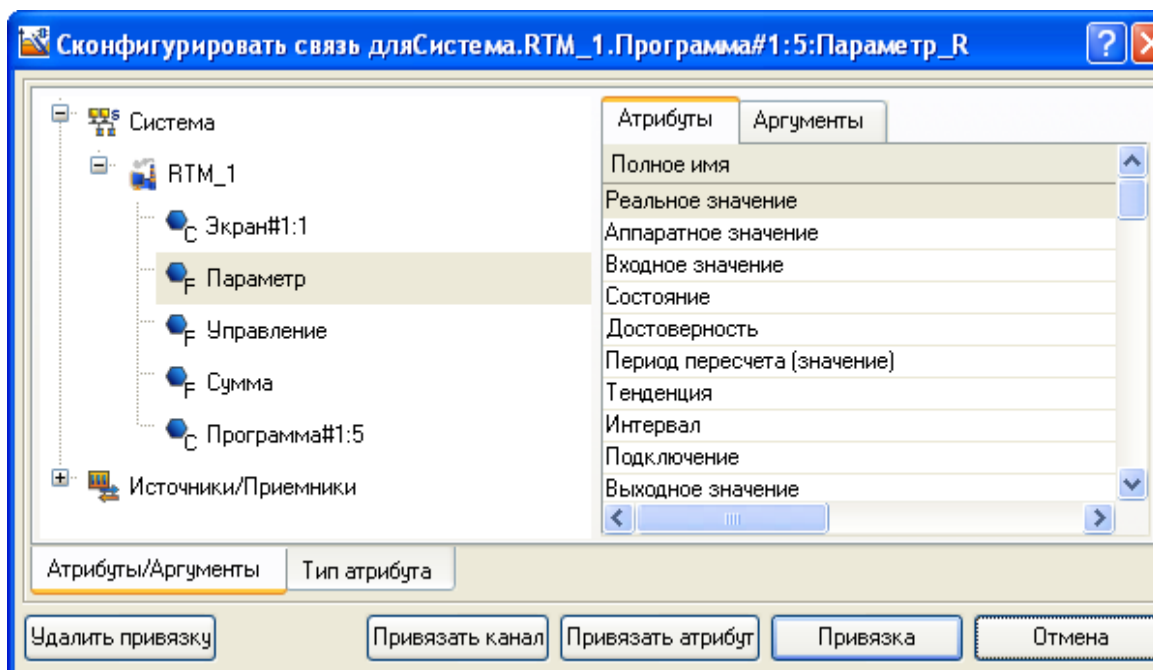
Сумма=Параметр+Управление;

- с помощью иконки , расположенной на инструментальной панели редактора, скомпилируем программу (или «горячей клавишей» **F7**) и убедимся в успешной компиляции в окне **Вывод**, вызываемого из инструментальной панели с помощью иконки :

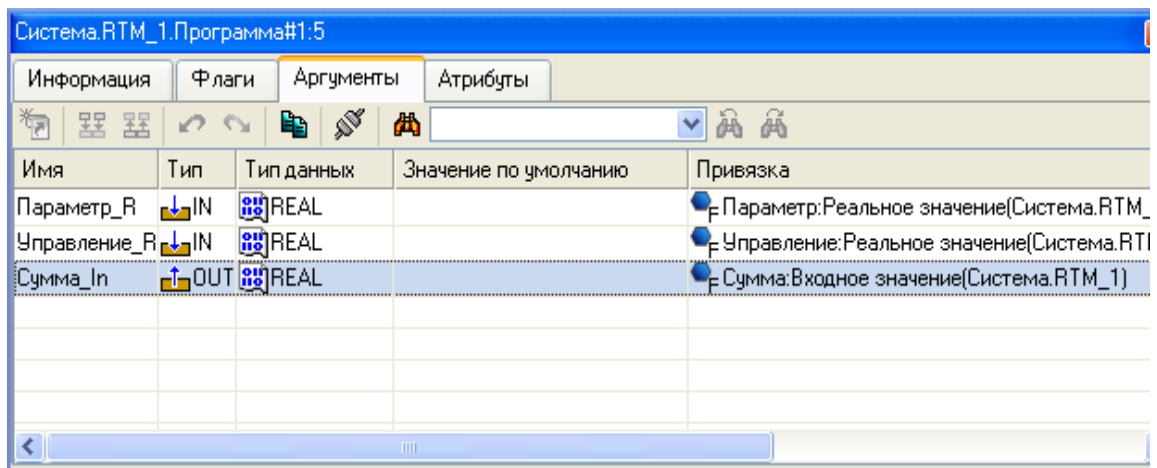


Выполним привязку аргументов программы к атрибутам каналов:

- вызвать свойства компонента «**Программа#1**» через контекстное меню;
- выбрать вкладку **Аргументы**;
- двойным нажатием в поле «**Привязка**» привязать аргументы программы к атрибутам каналов вручную – аргумент «**Параметр**» к реальному значению канала параметр, аргумент «**Управление**» к реальному значению канала управление;



- двойным щелчком в поле «**Привязка**» аргумента «**Сумма**» вызвать окно настройки связи, выбрать в нем атрибут «**Входное значение**» канала «**Сумма**» и кнопкой **Готово** подтвердить связь.



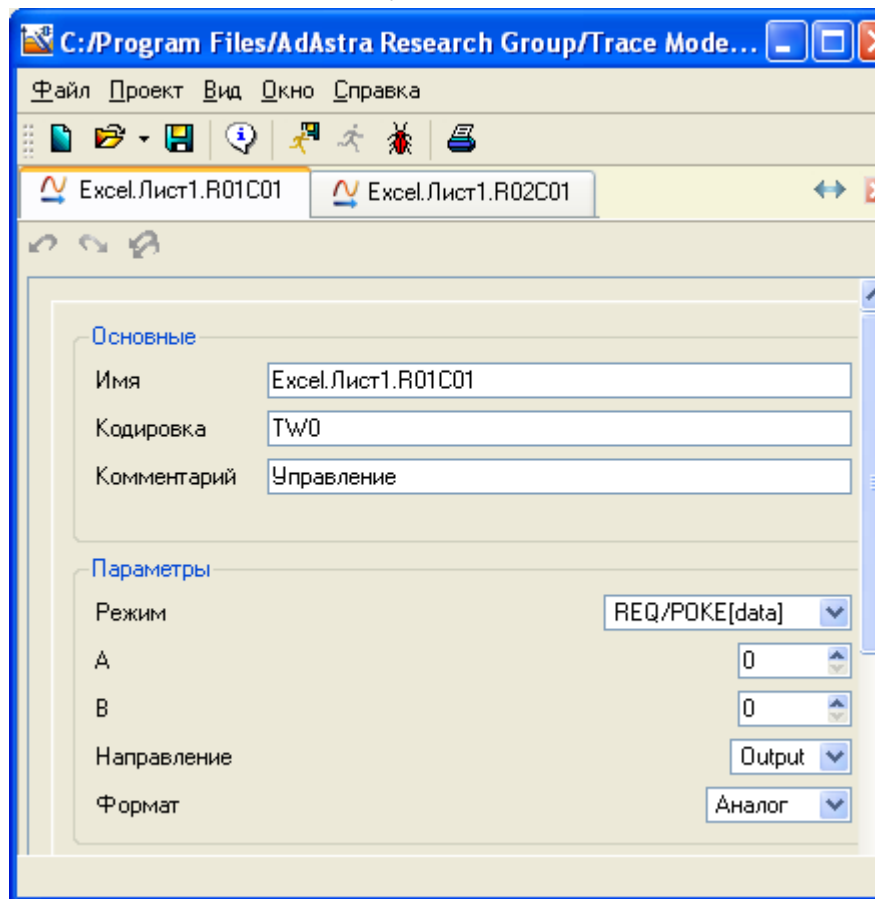
Сохранить и запустить проект. С помощью кнопки **Управление** ввести управляющее воздействие и наблюдать соответствующее изменение сигнала «Управление» и смещение сигнала «Сумма».

Осуществим связь протоколу *DDE* (Dynamic Data change) с *MS Windows* на примере *Excel*.

Организуем запрос данных из налов узла разработанного проекта с приложением *MS Windows*, в качестве которого выберем книгу *Excel*:

- Открыть «Источники/Получатели» и ПК создать в нем «DDE», переименовать группу как **Excel**;

- Открыть группу «Excel» и через ПК создать в ней 2 компонента «DDE»;



по

Excel-ни-

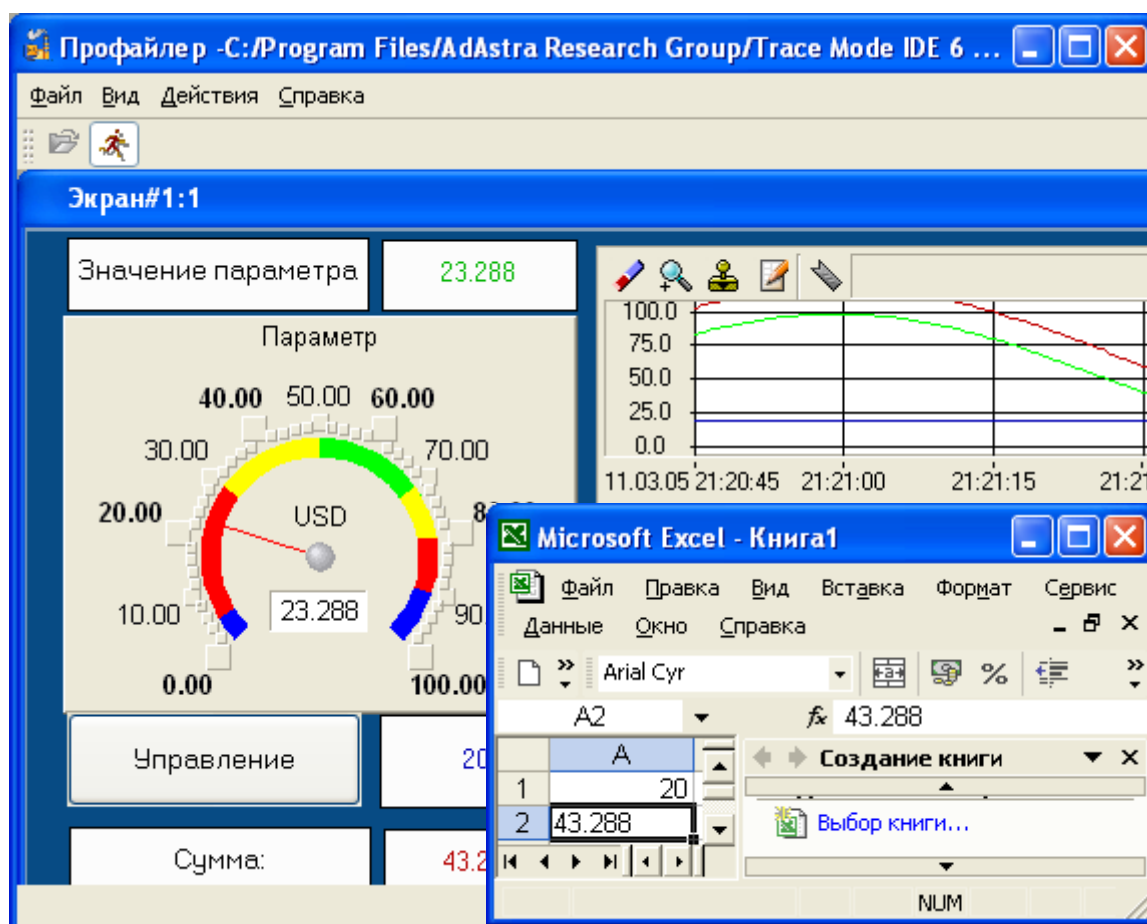
ка-

ко-

слой

через группу ем

- Двойным нажатием **ЛК** на компонент DDE#1 вызвать окно редактирования и изменить позиции экрана в соответствии с рисунком, представленном выше;
- Аналогичные действия осуществить над элементом DDE#2, изменив содержание поля «**Комментарий**» на «Сумма», а поля «**Имя**» на «Excel.Лист1.R02C01»;
- Захватить с помощью **ЛК** компонент «Excel.Лист1.R01C01», и, не отпуская **ЛК**, перетащить курсор на узел **RTM\_1**, откроется палитра компонентов узла, а курсор установить на канал «Управление». Отпустить левую клавишу;
- Захватить с помощью **ЛК** компонент «Excel.Лист1.R02C01», и, не отпуская **ЛК**, перетащить курсор на узел **RTM\_1**, откроется палитра компонентов узла, а курсор установить на канал «Сумма». Отпустить **ЛК**;
- Запустить *MS Excel* и созданный проект, убедиться, что значения в ячейках книги *Excel* изменяются вместе с соответствующими реальными значениями каналов узла.



## Порядок проведения лабораторной работы

Создайте элементарное **АРМ**, как указано в разделе «Описание виртуальной лабораторной установки»;

Для задания, предложенного преподавателем, определить объем автоматизации объекта и выбрать комплекс технических средств:

- датчики и исполнительные устройства (фирмы *Omron*, *Pepperl+Fuchs*, *Scaime*, *Siemens*);
- средства кабельной разводки и коммутации (фирмы *Belden*, *Hirschmann*, *Rittal*, *RST*, *Schroff*, *WAGO*);
- устройства ввода-вывода и связи с объектом – предназначены для ввода сигналов с датчиков в устройство обработки и вывода сигналов для управления исполнительными механизмами (фирмы *ADDI-DATA*, *Advantech*, *Dataforth*, *Diamond Systems*, *Fastwel*, *Grayhill*, *LenPromAvtomatika*, *Lippert*, *Octagon Systems*, *Pepperl+Fuchs*, *Pepperl+Fuchs Elcon*, *Scaime*, *Siemens*, *VIPA*, *VMIC*, *WAGO*);
- программируемые контроллеры (фирмы *Advantech*, *Dataforth*, *Diamond Systems*, *Fastwel*, *Grayhill*, *Lippert*, *Octagon Systems*, *Siemens*, *VIPA*, *VMIC*, *WAGO*).

## Содержание отчета

1. Цель исследований.
2. Итоговый экран создания элементарного **АРМ** с пояснением функций каждого графического элемента.
3. Задание на автоматизацию элемента участка технологического процесса (по выбору преподавателя).
4. Структурная схема системы управления.
5. Комплекс технических средств реализации структурной схемы с указанием технико-экономических характеристик и функций каждого.
6. Выводы по работе.

## Контрольные вопросы и задания

1. Назначение и функции схемы нормализации сигнала.
2. Назначение и функции согласующего устройства.
3. Назначение и функции *SCADA* систем.
4. Назначение и функции устройства выборки-хранения.

## Лабораторная работа №2

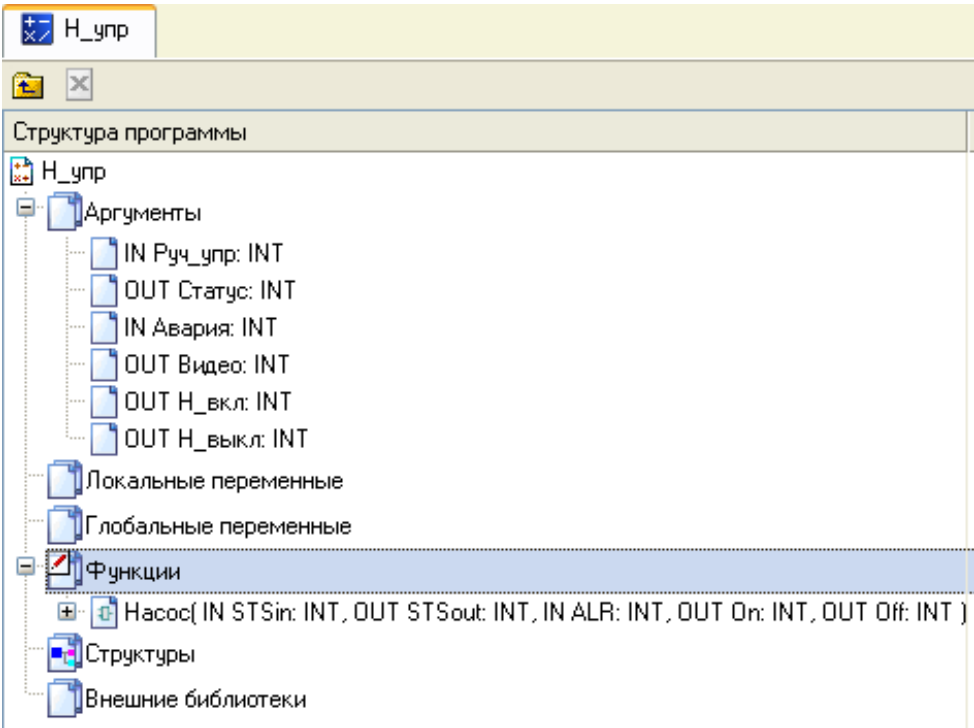
### ПРИМЕР ОБЪЕКТНОГО ПРОЕКТИРОВАНИЯ

Цель работы: создание элементарного АРМ насосной станции на примере объектного проектирования.

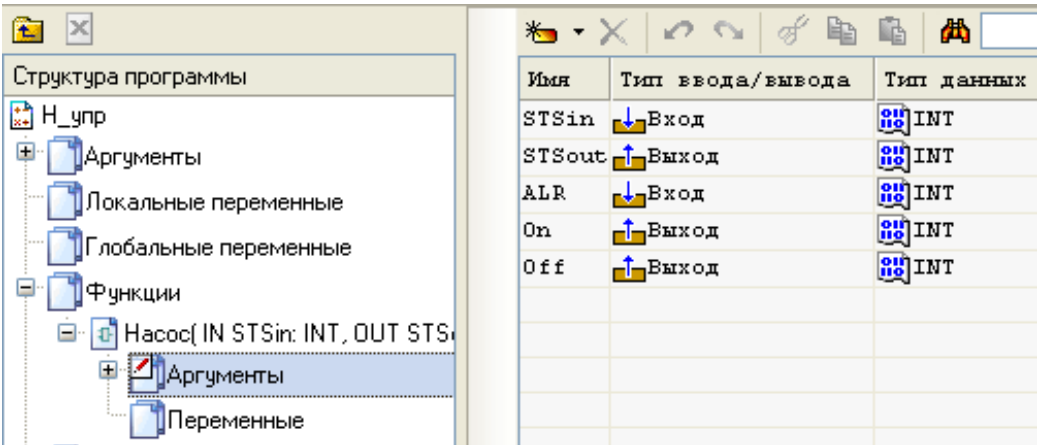
Указания к выполнению работы

Создадим шаблон программы **Н\_упр** управления насосом, предусмотрев аргумент для управления видеоклипом.

Структура программы:

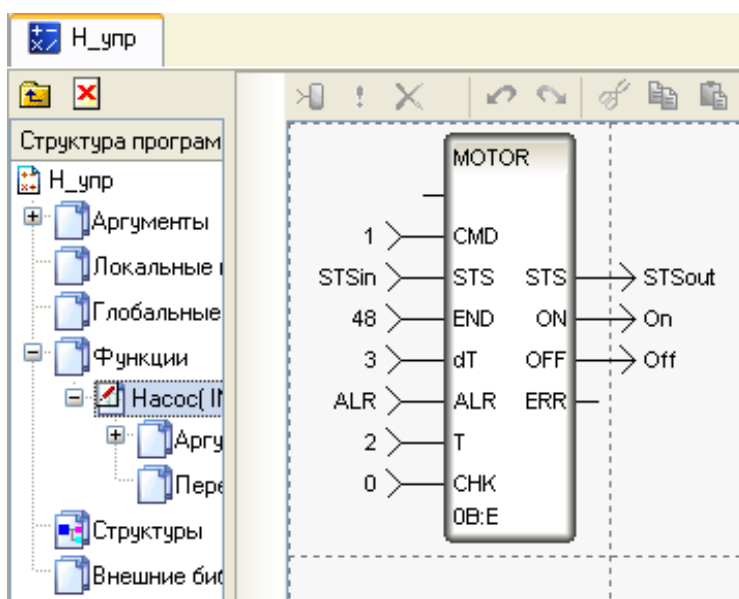


Аргументы функции управления насосом:



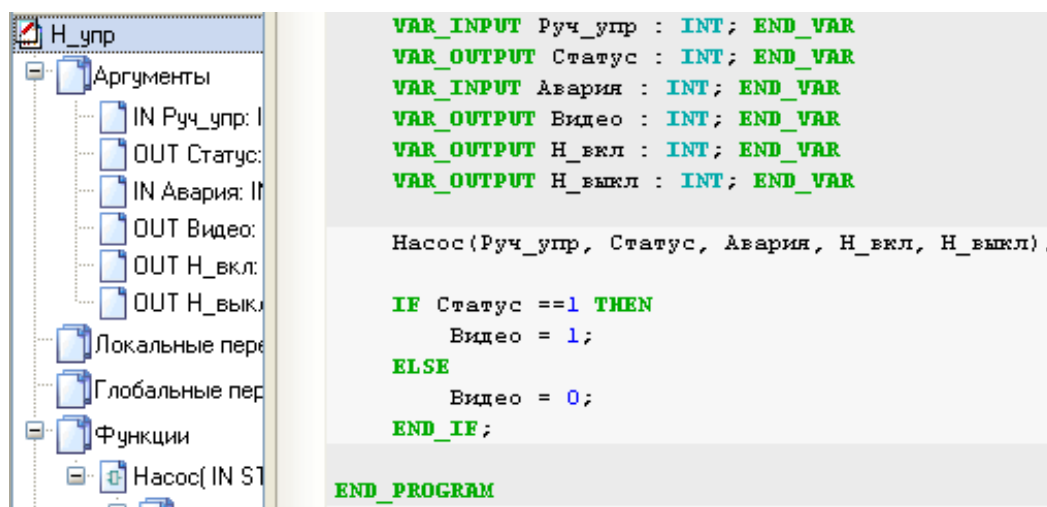


Функцию управления насосом удобнее всего разработать на языке **Техно FBD** – для этого потребуется один блок **MOTOR**:

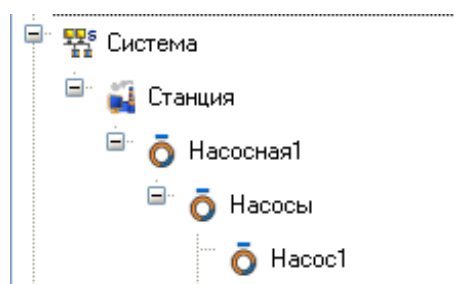


Аргументы основной ST-программы.

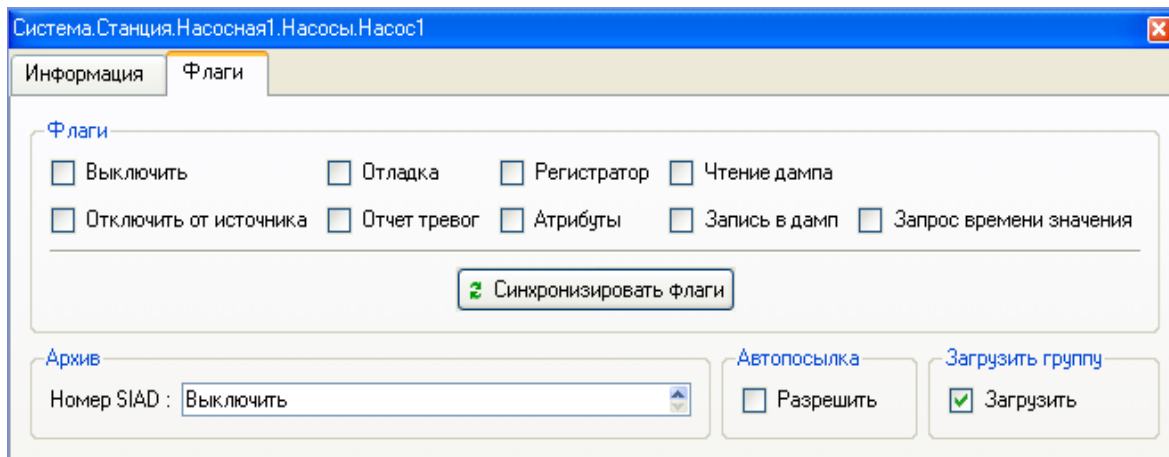
Листинг:



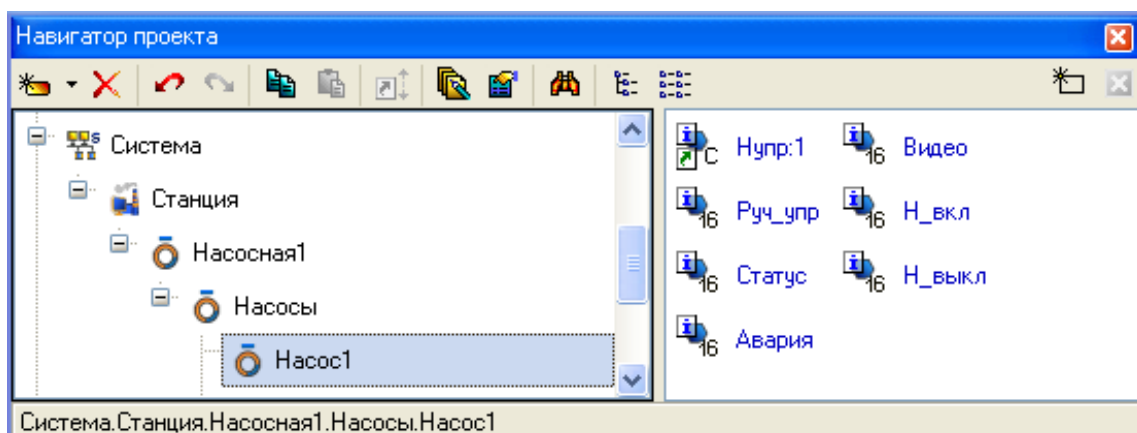
Рассмотрим процесс создания объекта **НАСОС1**. Для этого сначала создадим узел **Станция** со следующей структурой.



Установим для объекта **Насос1** флаг **Загрузить**.



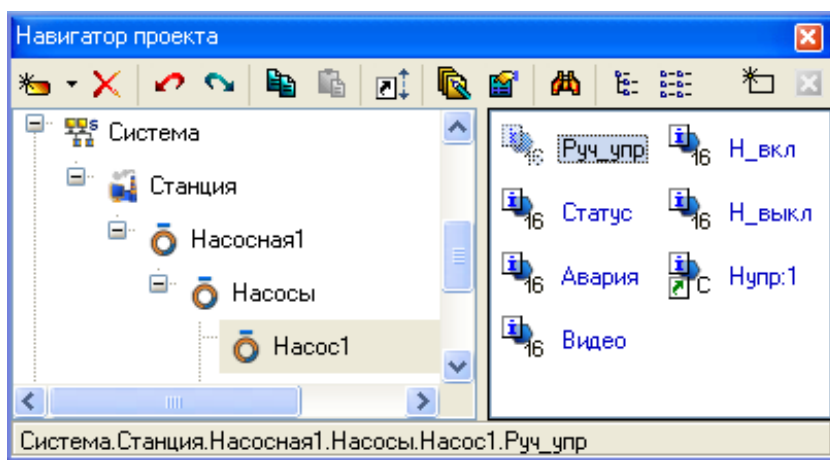
Перетащим с помощью мыши шаблон программы **Нупр** в группу **Насос1**, на вкладке **Аргументы** окна свойств созданного канала **CALL** выделим все аргументы и выполним команду **Создать по аргументам каналы с привязкой** – в объекте создадутся каналы, привязанные к соответствующим аргументам.



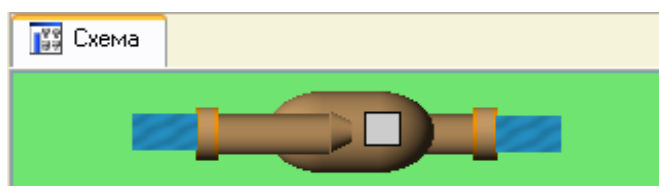
Скриншот окна 'Система.Станция.Насосная1.Насосы.Насос1.Нупр:1' с закладкой 'Аргументы'.

Имя	Тип	Тип данных	Значение	Привязка
Руч_упр	IN	INT	16	Руч_упр:Реальное значение (Система.Станция.Насосная1.Насосы.Насос1)
Статус	OUT	INT	16	Статус:Входное значение (Система.Станция.Насосная1.Насосы.Насос1)
Авария	IN	INT	16	Авария:Реальное значение (Система.Станция.Насосная1.Насосы.Насос1)
Видео	OUT	INT	16	Видео:Входное значение (Система.Станция.Насосная1.Насосы.Насос1)
Н_вкл	OUT	INT	16	Н_вкл:Входное значение (Система.Станция.Насосная1.Насосы.Насос1)
Н_выкл	OUT	INT	16	Н_выкл:Входное значение (Система.Станция.Насосная1.Насосы.Насос1)

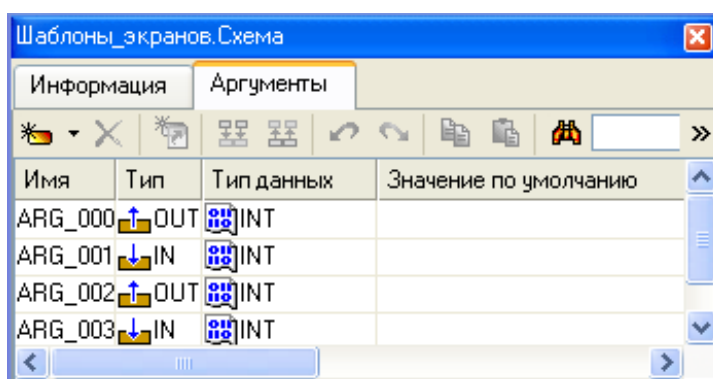
Изменим порядковые номера каналов в объекте путем выполнения последовательности действий Удаление Нупр:1 – Отмена удаления Нупр:1:



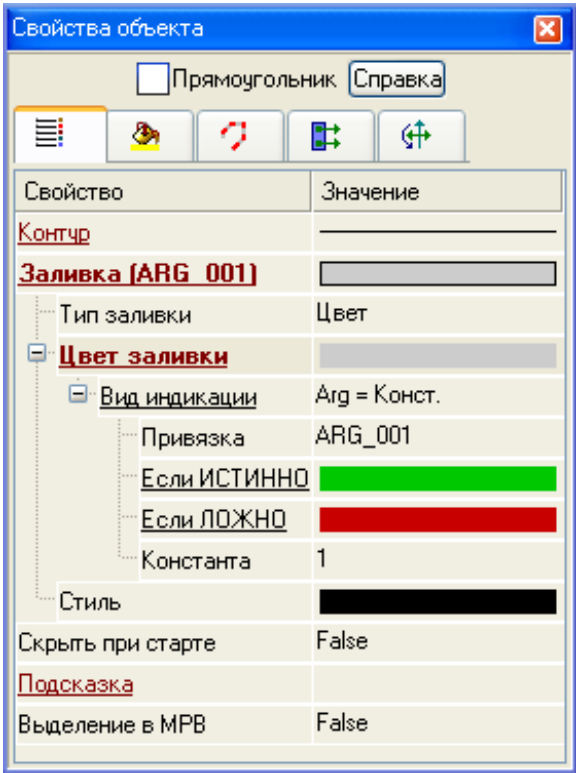
Создадим экран **Схема** с использованием библиотечных ресурсов *TRACE MODE* 6:



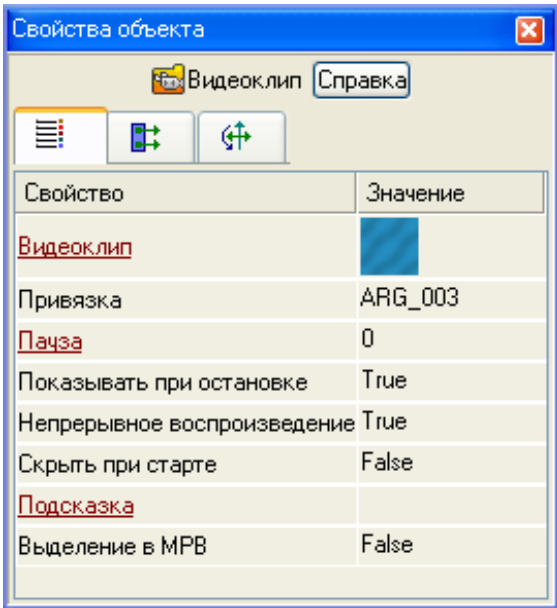
Аргументы экрана создадим с учетом их дальнейшей привязки к каналам объекта **Насос1**:



На экране размещен цветовой индикатор рабочего состояния насоса (динамизированный ГЭ Прямоугольник), настройки которого следующие. Цвет заливки соответствующий состоянию ИСТИННО – зеленый, а если ЛОЖНО – красный.



Настройки видеоклипа:



Перетащим с помощью мыши шаблон экрана **Схема** в группу **Насос1**, на вкладке **Аргументы** окна свойств созданного канала CALL выделим (Shift+ЛК) все аргументы и перетащим объект **Насос1** на вкладку **Аргументы** окна свойств канала **Схема** – аргументы канала CALL привяжутся к соответствующим каналам:

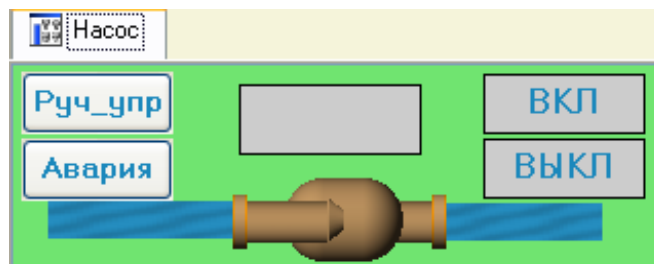
Система.Станция.Насосная1.Насосы.Насос1.Схема:2

Информация    Флаги    Аргументы    Атрибуты

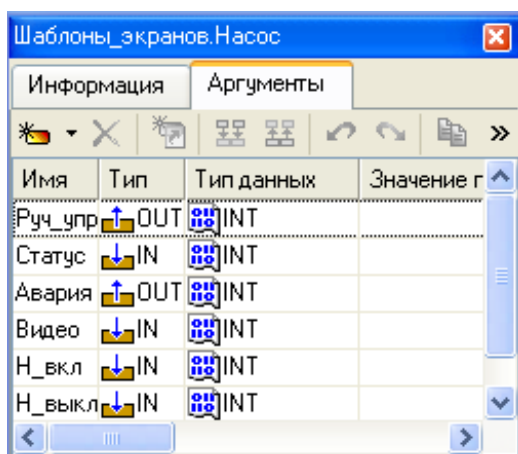
Имя    Тип    Тип дан    Привязка

Руч_упр_In	↑ OUT	INT	Руч_упр:Входное значение (Система.Станция.Насосная1.Насосы.Насос1)
Статус_R	↓ IN	INT	Статус:Реальное значение (Система.Станция.Насосная1.Насосы.Насос1)
Авария_In	↑ OUT	INT	Авария:Входное значение (Система.Станция.Насосная1.Насосы.Насос1)
Видео_R	↓ IN	INT	Видео:Реальное значение (Система.Станция.Насосная1.Насосы.Насос1)

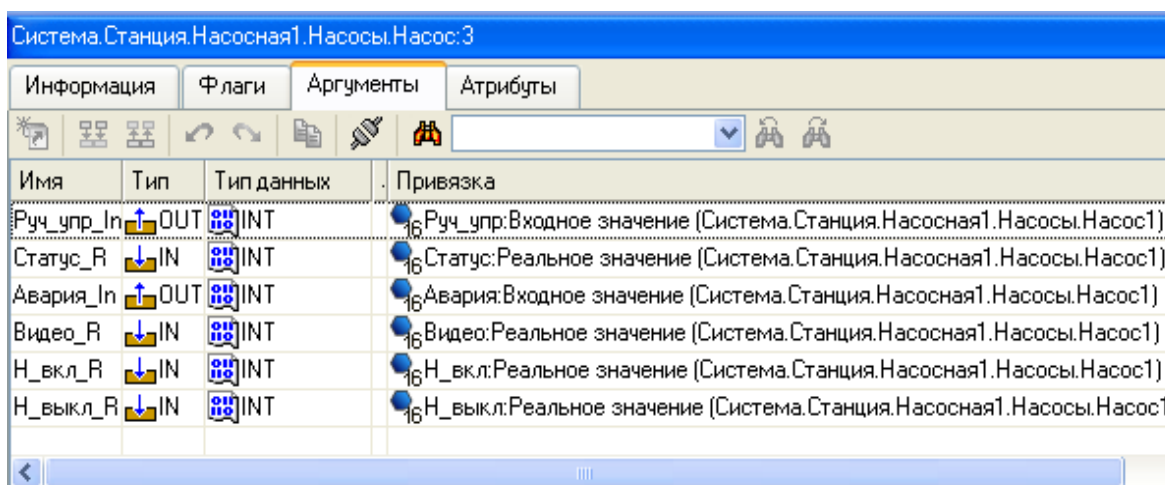
Создадим шаблон экрана **Насос**– пульт управления и отображения:



Аргументы экрана создадим с учетом их дальнейшей перепривязки к каналам:

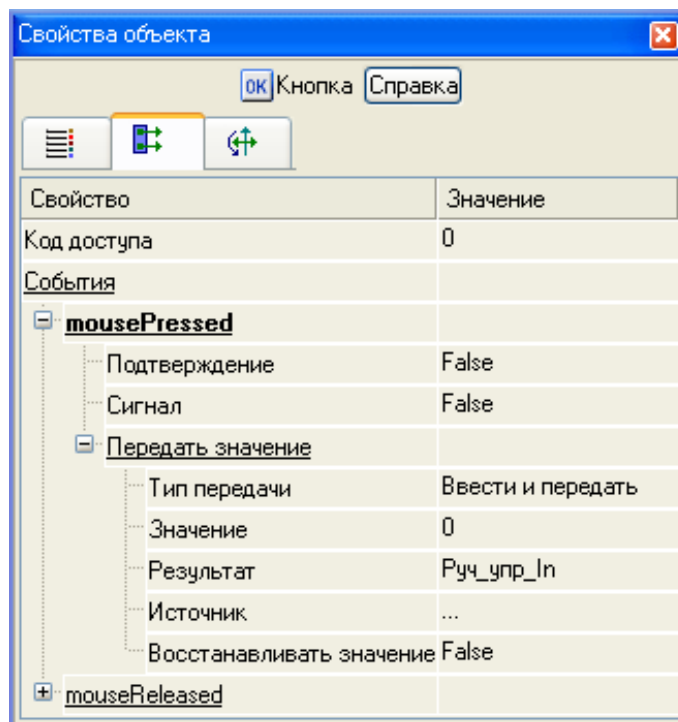


Перетащим с помощью мыши шаблон экрана **Насос** в группу **Насосы**, на вкладке **Аргументы** окна свойств созданного канала CALL выделим все аргументы и перетащим вкладку объект **Насос1** на выделенные аргументы – аргументы канала CALL привяжутся к соответствующим каналам:

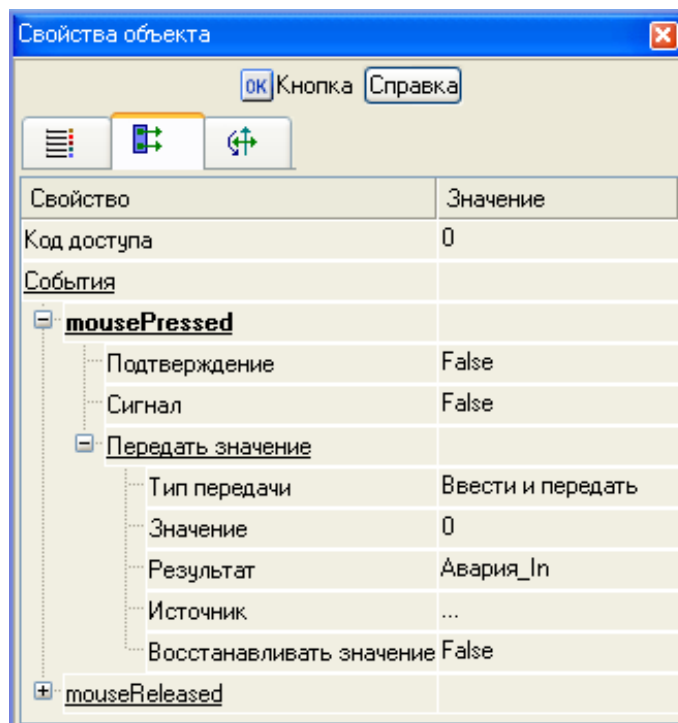


Настройки ГЭ шаблона экрана **Насос**:

- Кнопка **Руч\_упр** (ручное управление насосом):



- Кнопка **Авария** (моделирование сигнала аварии):



- Видеоклип привязан к аргументу **Видео**.

- Настройки центрального ГЭ **Текст** (отображение состояния насоса):

Свойства объекта

ABCТекст Справка

Свойство Значение

**Контур**

**Заливка (Статус\_R)**

Тип заливки Цвет

**Цвет заливки**

Вид индикации Arg в диапазоне

Привязка Статус\_R

Диапазоны

Диапазон	Значение	Цвет
1 3	1 3	Зеленый
3 4	3 4	Оранжевый
4 5	4 5	Синий
5 6	5 6	Красный
6 40	6 40	Черный

Стиль

Шрифт MS Shell Dlg.8

Выравнивание По центру

**Текст (Статус\_R)**

Вид индикации Arg в диапазоне

Привязка Статус\_R

Диапазоны

Использовать ресурсы False

Диапазон	Значение
1 3	ВКЛ
3 4	РЕЗЕРВ
4 5	РЕМОНТ
5 6	АВАРИЯ

**Цвет текста (Статус\_R)**

Вид индикации Arg = Констан.

Привязка Статус\_R

Если ИСТИННО

Если ЛОЖНО

Константа 1

Скрыть при старте False

**Подсказка**

Выделение в MPB False

Основная привязка Статус\_R



Во вкладке цвет заливки, использовать последовательно для диапазонов следующие цвета – зеленый; коричневый; синий; красный; черный. Цвет текста – синий, если ИСТИННО – черный, если ЛОЖНО – белый;

- Настройки текстового элемента **ВКЛ** (индикация генерации сигнала включения насоса):

Свойство	Значение
<b>Контур</b>	
<b>Заливка (H_vkl_R)</b>	
Тип заливки	Цвет
<b>Цвет заливки</b>	
Вид индикации	Arg = Констант.
Привязка	H_vkl_R
Если ИСТИННО	Зеленый
Если ЛОЖНО	Черный
Константа	1
Стиль	
Шрифт	MS Shell Dlg, 12, жирный
Выравнивание	По центру
<b>Текст</b>	ВКЛ
<b>Цвет текста (H_vkl_R)</b>	Голубой
Вид индикации	Arg = Констант.
Привязка	H_vkl_R
Если ИСТИННО	Белый
Если ЛОЖНО	Черный
Константа	1
Скрыть при старте	False
<b>Подсказка</b>	
Выделение в MPB	False
Основная привязка	H_vkl_R

Во вкладке цвет заливки использовать, если ИСТИННО – зеленый цвет, если ЛОЖНО – черный. Цвет текста – голубой, если ИСТИННО – белый, если ЛОЖНО – черный.

- Настройки текстового элемента **ВЫКЛ** (индикация генерации сигнала выключения насоса):

Свойства объекта

ABCТекст Справка

Свойство Значение

**Контур**

**Заливка (Н выкл R)**

Тип заливки Цвет

**Цвет заливки**

Вид индикации Arg = Констант.

Привязка Н\_выкл\_R

Если ИСТИННО

Если ЛОЖНО

Константа 1

Стиль

Шрифт MS Shell Dlg,12,жирный

Выравнивание По центру

**Текст** ВЫКЛ

**Цвет текста (Н выкл...)**

Вид индикации Arg = Констант.

Привязка Н\_выкл\_R

Если ИСТИННО

Если ЛОЖНО

Константа 1

Скрыть при старте False

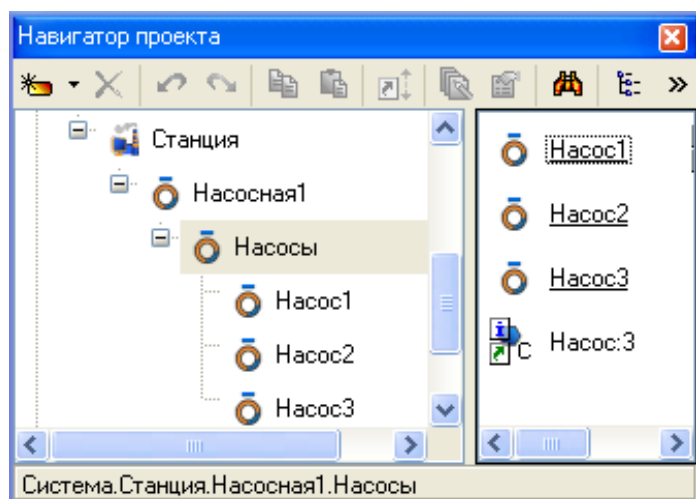
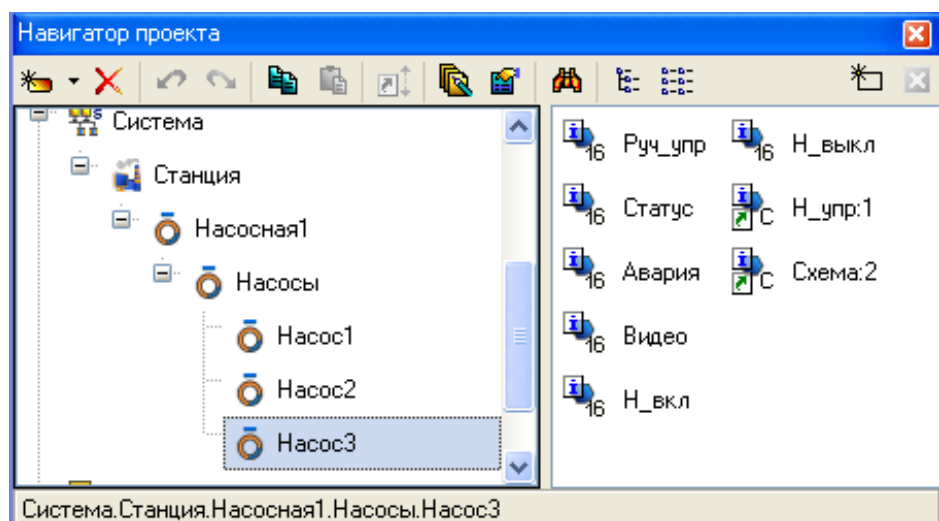
**Подсказка**

Выделение в MPB False

Основная привязка Н\_выкл\_R

Во вкладке цвет заливки использовать, если ИСТИННО – красный цвет, если ЛОЖНО – черный. Цвет текста – голубой, если ИСТИННО – белый, если ЛОЖНО – черный.

С помощью копирования/обычной вставки объекта **Насос1** создадим в группе **Насосы** объекты **Насос2** и **Насос3**:



Создадим шаблон главного экрана:

**УПРАВЛЕНИЕ НАСОСАМИ**

Насос:3

Выбрать
Сбросить

NAME

ОБЩАЯ СХЕМА

НАСОС1

НАСОС2

НАСОС3

Схема:2

Схема:2

Схема:2

Единственный аргумент этого шаблона предназначен для реализации перепри-  
вязки к объектам:

Имя	Тип	Тип данных	Значение п
Насос		IN/OUT	UDINT

## Настройки ГЭ Переключатель каналов:

Свойство	Значение
Привязка	Насос
Класс каналов	ALL
Условие выборки каналов	OBJ
Шрифт	MS Shell D
Период	1
Колонки	
Колонка1	NAME
Скрыть при старте	False
Подсказка	

Ссылка на экран **Насос**: привязана к каналу CALL группы **Насосы**, вызывающему шаблон экрана **Насос**. Ссылка создается путем использования механизма *drag-and-drop* +Ctrl.

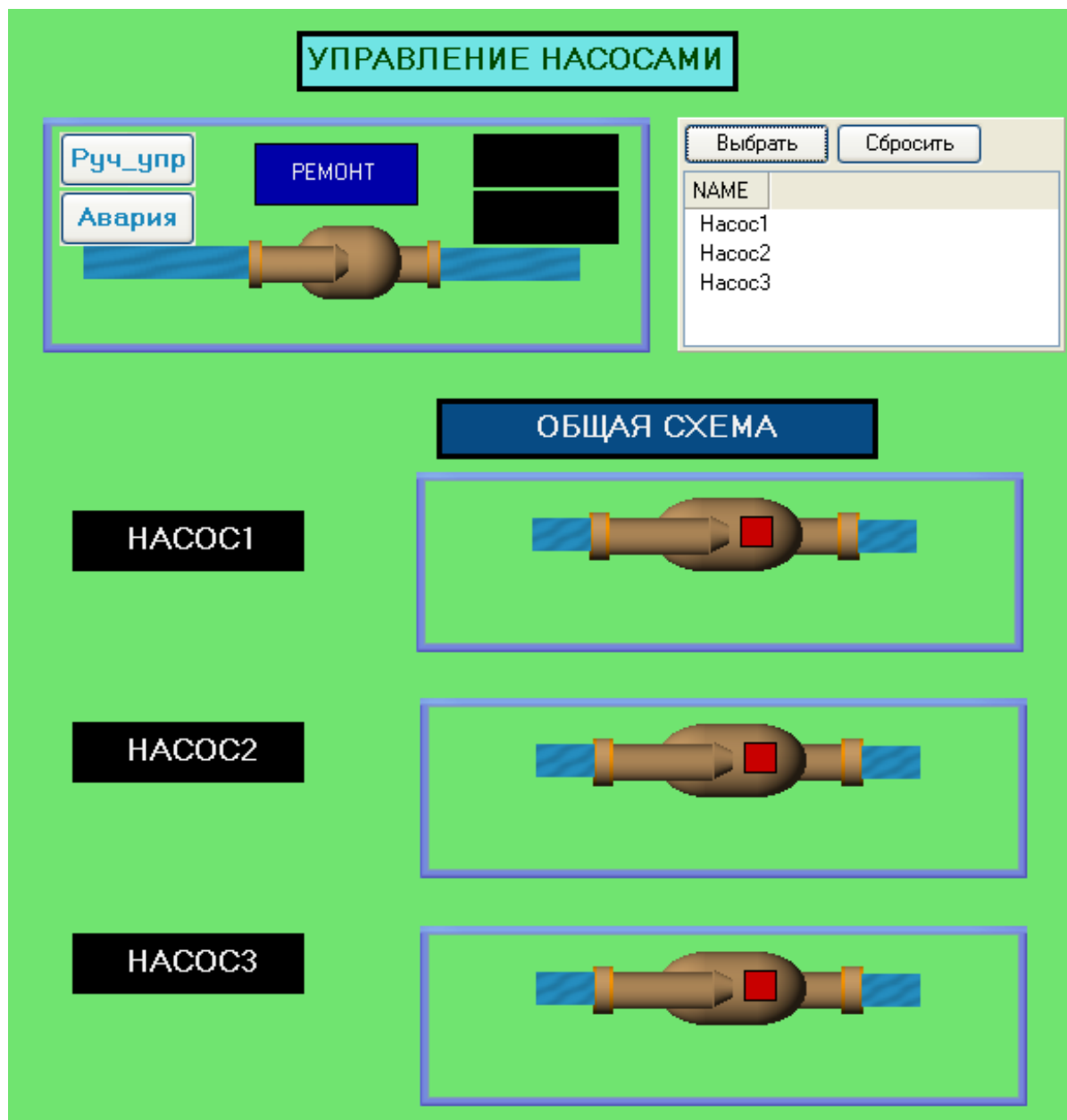
Ссылки на экраны **Схема** привязаны к вызовам шаблона экрана **Схема** соответствующих объектов.

Перетащим шаблон главного экрана в группу **Насосы** и привяжем аргумент созданного канала CALL:

Имя	Тип	Тип данных	Привязка	Ф
Насос_3_ID	IN/OUT	UDINT	Насос:3.ID (Система.Станция.Насосная1.Насосы)	

Так как при создании экранов каждому объекту присваивается уникальный идентификационный номер по мере создания, следовательно, главный экран не будет появляться первым на MPV. Чтобы этого избежать, необходимо вызвать на редактирование канал Гл\_экран и отметить поле **Отработать**, а в поле **На старте** проставить 2.

Запустим проект – пульт отобразит состояние насоса 1:



С помощью ГЭ **Переключатель каналов** привяжем пульт к объекту **Насос2** и с помощью кнопки **Руч. упр** включим насос. При подаче команды включения (1 после 8) вначале сработает индикатор генерации сигнала включения, затем насос включится.

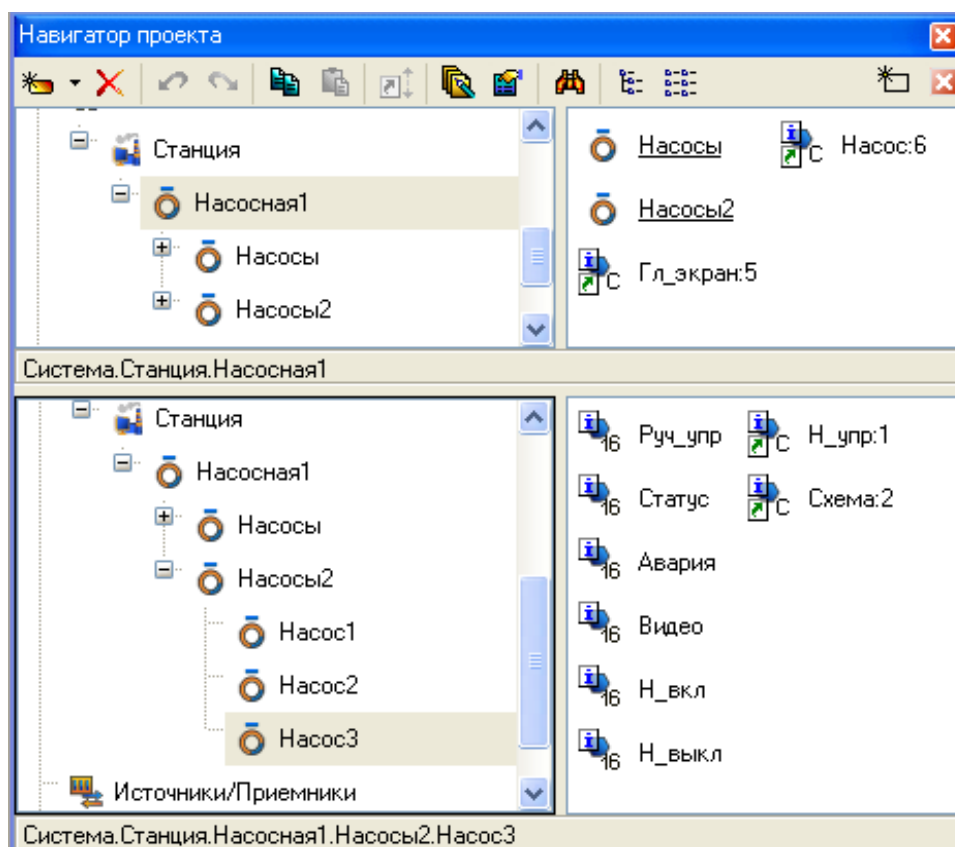
Привяжем пульт к насосу 3 и переведем его в резерв (команда 8). Таким образом, перепривязывая пульт к нужному насосу, можно управлять всеми насосами и отображать подробную информацию о их состоянии.

Объектный подход к проектированию дает возможность простой модификации проекта в дальнейшем. Пусть, например, насосная станция расширилась за счет добавления группы из трех насосов.

Переместим каналы **CALL Гл.экран** и **Насос** из группы **Насосы** в группу **Насосная1**.

Создадим в группе **Насосная1** группу **Насосы2** с помощью копирования и обычной вставки группы **Насосы**. Установим флаг **Загрузить** для группы **Насосы** и **Насосы2**.

Структура проекта примет следующий вид:



В шаблоне главного экрана скопируем/вставим группу трех ссылок на экраны **Схема**:

УПРАВЛЕНИЕ НАСОСАМИ

Насос:6

Выбрать

Сбросить

NAME

ОБЩАЯ СХЕМА

НАСОСЫ

НАСОСЫ2

Схема:2

Схема:2

Схема:2

Схема:2

Схема:2

Схема:2

Образованные ГЭ привяжем к соответствующим экранам **Схема** объектов группы **Насосы2**.

Запустим проект и убедимся, что в результате довольно простой модификации функции пульта расширились – теперь с помощью пульта можно управлять любым из шести насосов.

### Порядок проведения лабораторной работы

Создайте элементарное **АРМ**, как указано в разделе «Указания к выполнению лабораторной работы».



Учитывая выполнение курсового проекта по дисциплине «Электроснабжение промышленных предприятий» на основании объектного проектирования создать диспетчерский пульт схемы электроснабжения в соответствии с заданием на проектирование.

### **Содержание отчета**

1. Цель исследований.
2. Итоговый экран создания элементарного.
3. Описание программы Н\_упр с указанием входных и выходных аргументов.
4. Программное обеспечение, итоговый экран, описание работы синтезированного диспетчерского пульта.
5. Выводы по работе.

### **Контрольные вопросы и задания**

1. В чем преимущества объектного проектирования?
2. Назначение и свойства объекта Переключатель каналов.
3. Назначение и свойства объекта Ссылка на экран.

### **Лабораторная работа №3**

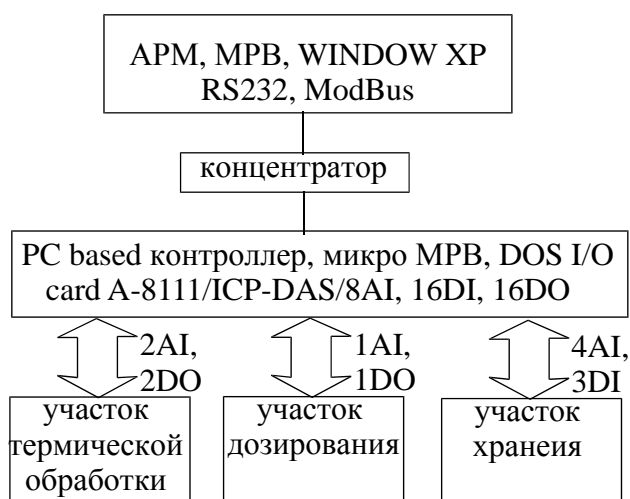
## ИССЛЕДОВАНИЕ ВЛИЯНИЯ ПРОПОРЦИОНАЛЬНО-ДИФФЕРЕНЦИАЛЬНОГО-ДИФФЕРЕНЦИАЛЬНОГО РЕГУЛЯТОРА НА КАЧЕСТВО ПРОЦЕССА РЕГУЛИРОВАНИЯ

Цель работы: построить систему контроля и управления технологического процесса (ТП) участка термической обработки и оценить влияние параметров пропорционально-дифференциально-дифференциального (ПДД) регулятора на качество процесса регулирования.

### Указания к выполнению работы

Рассматриваемый ТП ведется на трех участках: термической обработки, хранения и дозирования. Необходимо построить систему контроля и управления ТП с учетом имеющихся точек контроля, исполнительных механизмов и аппаратных средств автоматизации. Все участки управляются *PC – based* контроллером.

Участок термической обработки использует два входных аналоговых сигнала –



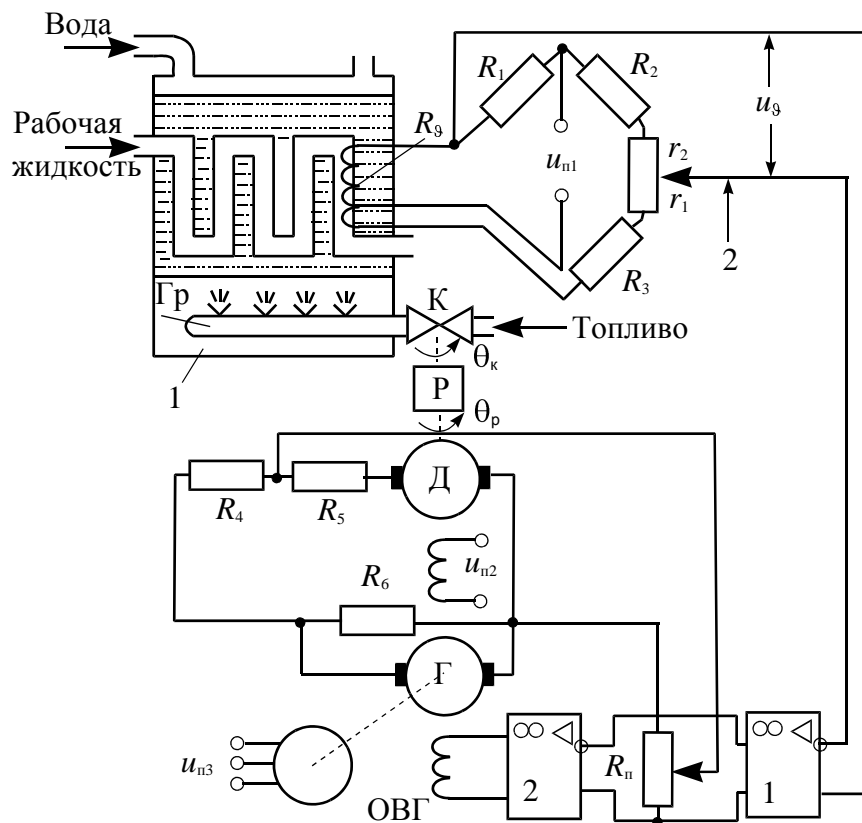
2AI и два выходных дискретных – 2DO. Технологическая задача – поддержание постоянной температуры в аппарате – технологической установке, осуществляющей длительный нагрев исходного сырья (функция регулирования). Регулирование температуры производится путем изменения расхода поступающего в аппарат теплоносителя. Алгоритм управления – пропорционально-дифференциальный-дифференциальный (ПДД). Способ управления исполнительным механизмом – широтно-импульсная модуляция (ШИМ). Аналоговые сигналы в *PC – based* контроллере обрабатываются 12 разрядным аналого-цифровым преобразователем (АЦП) и имеют представление в кодах (0 – 4095). Для измеряемой температуры коды соответствуют входному диапазону датчика температуры

(0 – 100) С, для расхода теплоносителя – (0 – 10) м<sup>3</sup>/ч. АРМ контролирует подключенные к *PC* – based контроллеру технологические параметры (функции мониторинга) и задает настройки регулятора (функция управления).

*PC-based* контроллер подключен к АРМ по сети через концентратор, используемый сетевой протокол – TCP/IP. В качестве контроллера выступает обычный PC-совместимый компьютер с установленной в системную шину ISA платой ввода/вывода А-8111 с 8AI, 16DI и 16DO /ICP DAS/, работающий под управлением ОС DOS (данное устройство не подлежит моделированию, а заменяется имитатором).

### Описание создания системы контроля и управления

Рассмотрим принцип действия системы автоматического регулирования температуры. Термометр  $R_g$  изменяет свое сопротивление в зависимости от температуры



рабочей жидкости, протекающей по трубопроводу через печь 1. Перемещением ползунка задатчика 2 нарушаем баланс моста.

При полном балансе моста имеем

$$R_g = \frac{R_2 + r_2}{R_1 + r_1} R_3$$

При перемещении ползунка задатчика влево  $r_1$  уменьшается, а  $r_2$  возрастает, что приводит к увеличению  $R_g$  и наоборот. Мост питается переменным током, поэтому амплитуда

напряжения небаланса зависит от величины отклонения температуры  $\vartheta_{\Pi}$  от заданной  $\vartheta_3$ , а фаза – от знака этого отклонения.

Напряжение небаланса  $u_g$ , пропорциональное температуре печи, подается на вход операционного усилителя ОУ1. Усиленное напряжение после второго операционного усилителя ОУ2 поступает на обмотку возбуждения ОВГ генерато-

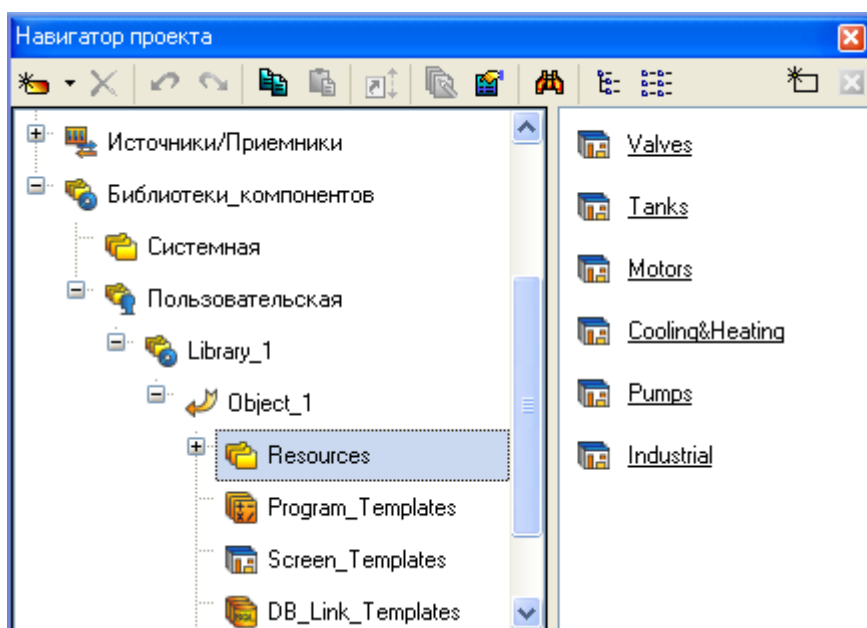
ра постоянного тока  $\Gamma$ , который питает электродвигатель Д. Якорь двигателя через редуктор Р поворачивает кран К, изменяя количество топлива, подаваемого в горелки  $\Gamma_r$ . При равновесии моста напряжение рассогласования  $u_d = 0$  и кран занимает вполне определенное положение, которое соответствует равенству температур датчика  $\vartheta_z$  и печи  $\vartheta_{\Pi}$ .

Использование варианта тахометрической обратной связи, как указано в примере, рассмотренном выше, представляется как альтернативный способ. На современном этапе применяют ПДД – регуляторы и способ управления исполнительным механизмом – ШИМ.

## Описание виртуальной лабораторной установки

Проиллюстрируем создание системы автоматизации путем проектирования «от шаблонов», т.е. будем создавать информационную базу проекта – каналы по аргументам разрабатываемых шаблонов экранов и программ, дополняя основной подход методами автопостроения и связывания каналов в узлах проекта.

Откроем интегрированную среду и создадим новый проект. В качестве стиля разработки выберем **Стандартный**.

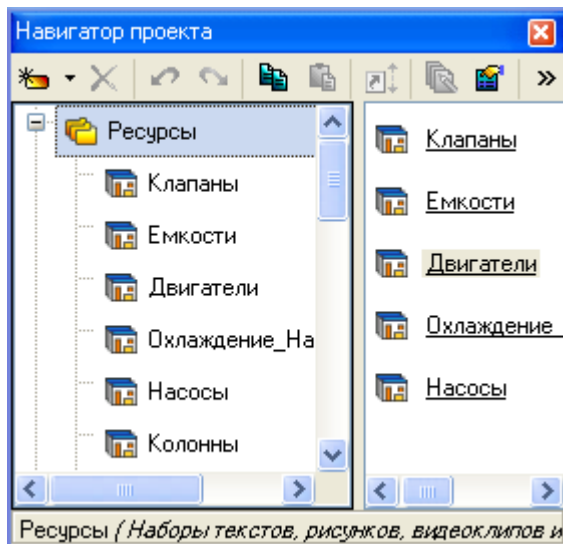



Перейдем в слой **Библиотеки\_компонентов**, где в разделе **Пользовательская** откроем библиотеку **Библиотека\_1**. Сохраненный в данной библиотеке объект **Объект\_1** содержит в своем слое **Ресурсы**, необходимый для дальнейшей разработки, набор графических объектов – изображения клапанов, емкостей, двигателей и т.д.

Перенесем группы в слой **Ресурсы** текущего проекта с помощью механизма **drag-and-drop** и переименуем их как показано ниже.

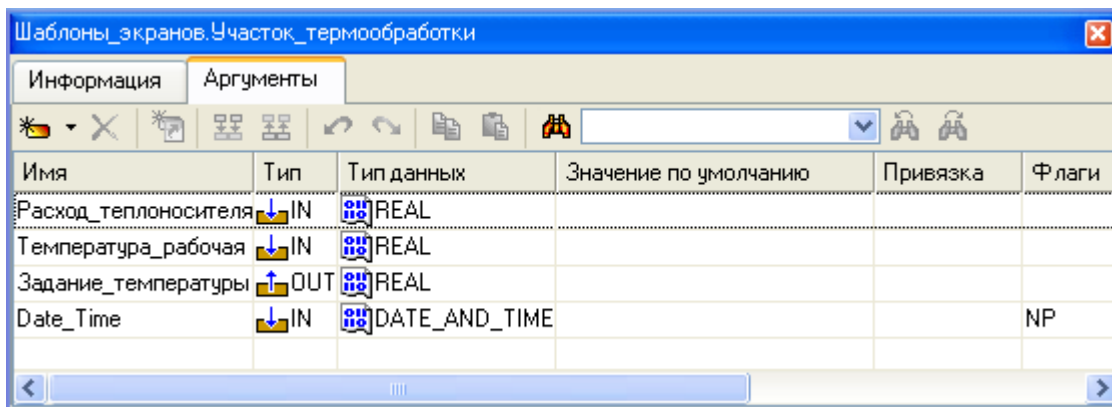
Здесь же в слое **Ресурсы** создадим группу Картинки для помещения в нее текстур, которые будут применены в оформлении создаваемых графических экранов. Создадим в группе Картинки новый компонент – **Библиотека\_Изображений#1**.


Откроем двойным щелчком ЛК вновь созданную библиотеку для редактирова-

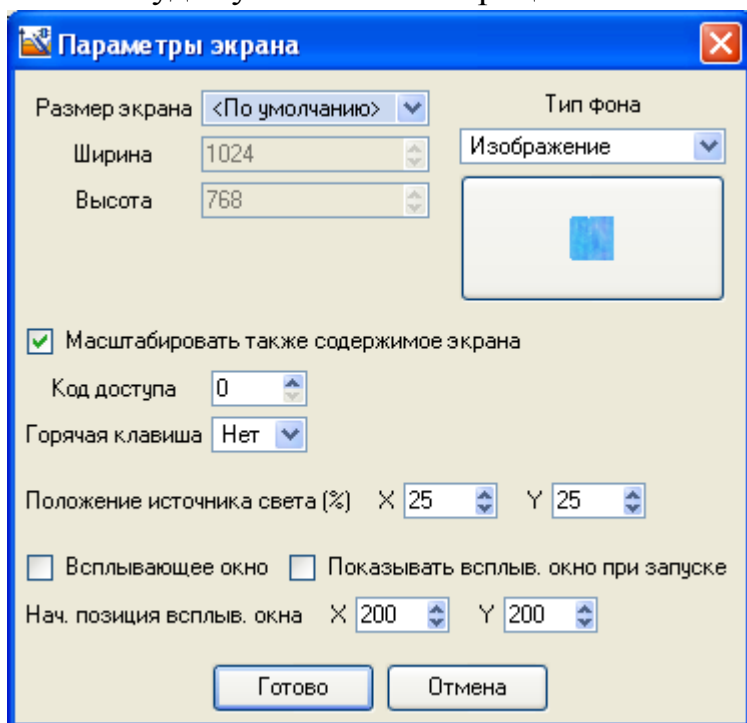


ния. Для ее наполнения воспользуемся иконкой  с подсказкой «Импортировать» на панели инструментов. В открывшемся диалоге выбора файлов для импорта укажем поддиректорию **...\Lib\Texture**. Выберем все файлы и нажмем экранную кнопку **Открыть**.

Подобным описанному выше способом создадим в слое **Ресурсы** группу Анимация, в ней – библиотеку **Библиотека\_Видеоклипов#1**. Наполним ее содержимым **...\Lib\Animation**. После проведения подготовительных мероприятий сохраним выполненную работу, нажав **ЛК** и указав имя (использовать латинский






шрифт). Перейдя в слой **Шаблоны экранов**, создадим в нем **Экран#1**. На созданном экране будут отображаться технологические параметры участка термической обработки, с него же будем осуществлять формирование задания на поддержание рабочей температуры. Переименуем его в **Участок\_термообработки**. Создадим также экран для задания параметров ПДД-регулятора. В соответствии с ТЗ на проектирование назначим аргументы экрану участка термообработки – щелчок **ПК** на вновь созданном шаблоне экрана и выбор из выпадающего списка пункта **Свойства**, там – вкладка **Аргументы**. Здесь и далее с помощью иконки  с подсказкой *Создать аргумент*, создаются необходимые аргументы, задаются их имена, тип данных, значения по умолчанию, привязки, флаги и т.д. Те аргументы, значения которых будут отображаться на экране, имеют тип **IN**, а те, что задаются с клавиатуры АРМ, отображаются на экране и пересылаются в конечном итоге в PC-based контроллер, имеют тип **OUT**. Флаг **NP** для аргумента **Date\_Time** означает то, что при автопостроении каналов по аргументам шаблона этот аргумент не будет участвовать в процессе автопостроения.

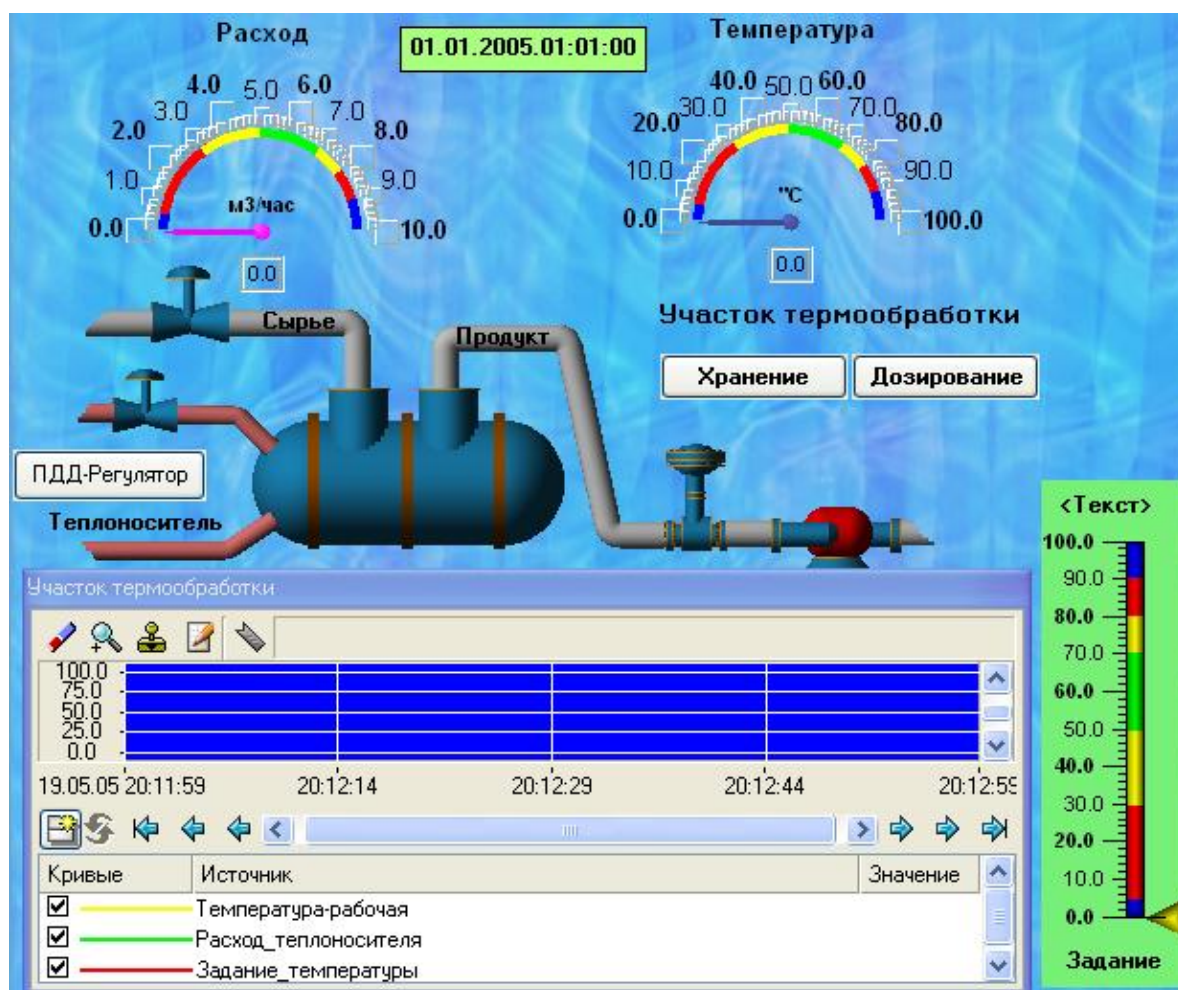


Закроем бланк свойств экрана.


Для перехода к непосредственному созданию и редактированию содержимого экрана участка термообработки дважды нажмем на нем **ЛК** мыши. Зададим в качестве фона экрана текстуру **ice\_0161**. Для этого выберем в основном меню пункт

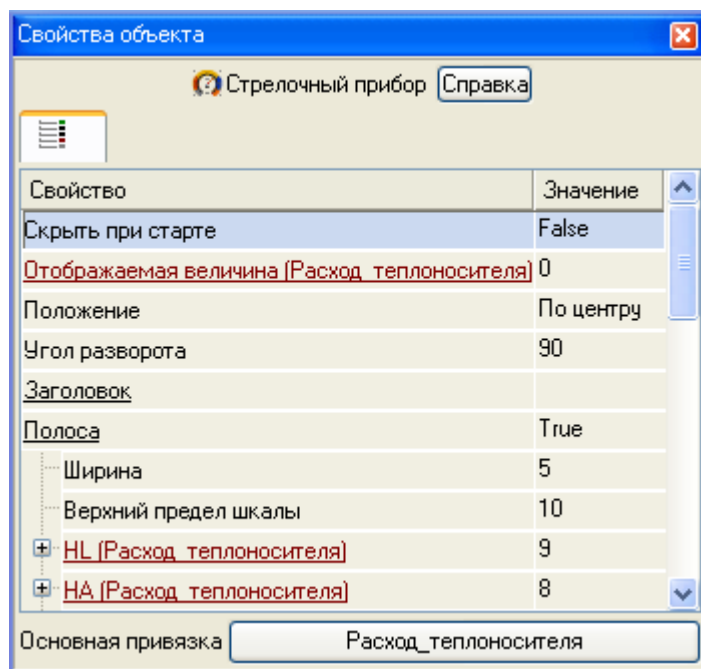
**Сервис**, в нем – **Параметры экрана**. В открывшемся диалоговом окне укажем в качестве типа фона изображение, а из имеющихся в библиотеке текстур – например ice\_0161. После нажатия экранной кнопки **Готово** фон графического экрана будет изменен на указанный.

С помощью графических объектов (ГО), сохраненных в ресурсных библиотеках и вызываемых с помощью иконки  **Объект** панели инструментов, а также графических элементов (ГЭ) объемных труб  и текста , создадим статическую часть экрана. Примерный вид экрана представлен ниже.

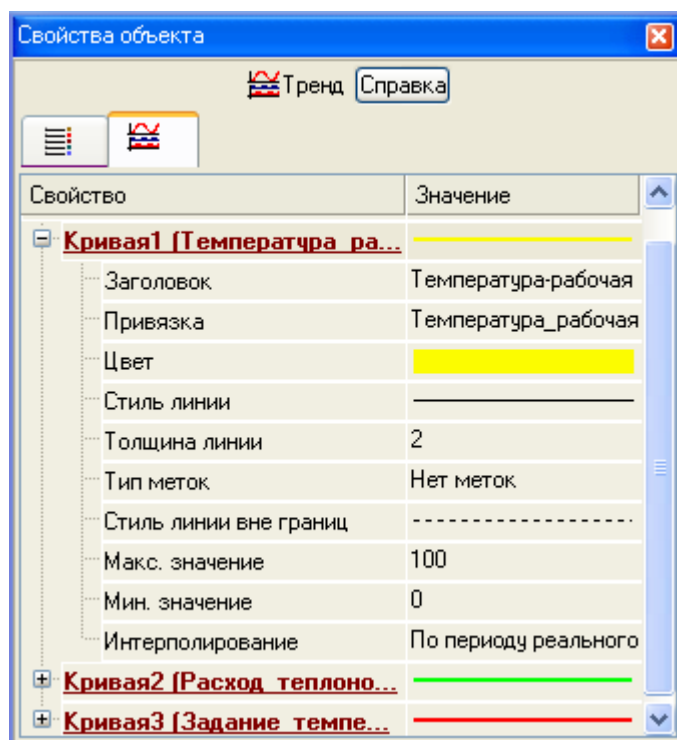



Графические объекты размещаются с использованием метода *drag-and-drop* и допускают масштабирование. Для изменения размера ГО необходимо выделить его **ЛК** и с помощью позиционирования указателя мыши в узловые точки выполнить необходимые корректирующие действия.

Значения расхода теплоносителя и рабочей температуры будем отображать с помощью ГЭ Стрелочный прибор . Разместив их на экране, двойным щелчком **ЛК** откроем свойства левого ГЭ и зададим ему свойства.






Аналогичным образом поступим с правым ГЭ, осуществив основную привязку к аргументу Температура\_рабочая.



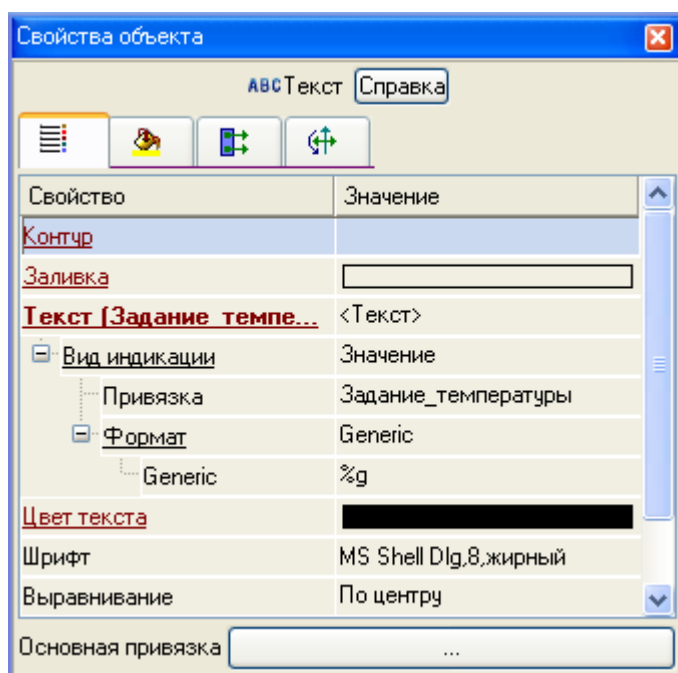
В нижней части экрана разместим ГЭ Тренд -  для вывода параметров Температура\_рабочая, Расход\_теплоносителя и Задание\_температуры. Основные свойства ГЭ оставим заданными по умолчанию, добавив заголовок "Участок термообработки".



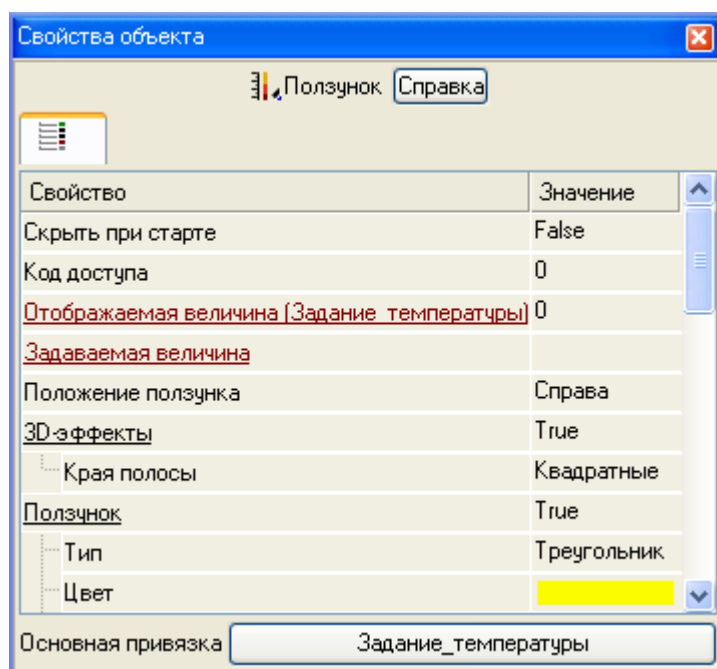
Добавим для отображения на тренде три кривые, связав их с соответствующими аргументами экрана, и зададим для них цвет и толщину линий, интервалы выводимых значений.

Для формирования задания регулятору разместим справа от ГЭ **Тренд** ГЭ **Прямоугольник** , он будет служить подложкой для ГЭ **Ползунок**, , с помощью которого будем задавать значение задания и отображать его же. Точную величину задания будем отображать в верхней части прямоугольника с помощью ГЭ **Текст** .

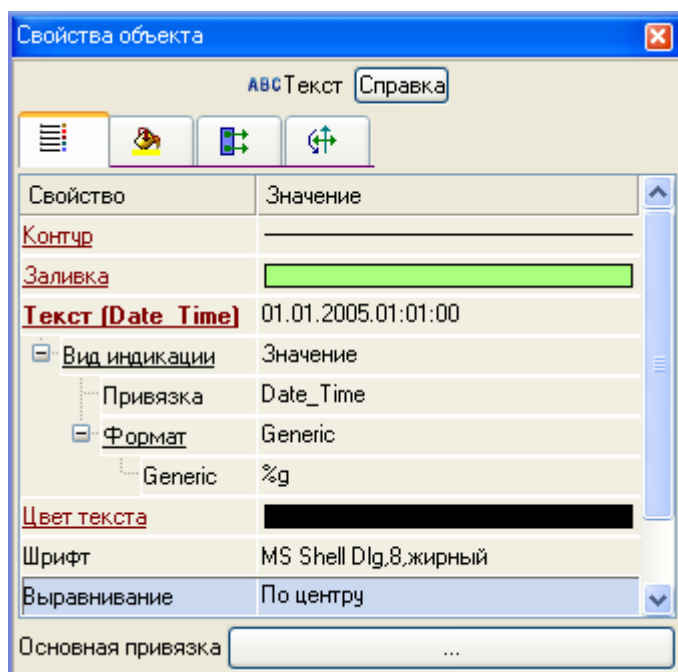
Откажемся от использования рамки и заливки для данного ГЭ. Свойства ГЭ **Текст** будут:



Для ГЭ **Ползунок**, откажемся от рамки и заливки, а свойства назначим такими, как изображены на рисунке, представленном ниже.





В верхней части графического экрана разместим ГЭ **Текст**, для отображения текущего времени. Свойства – цвет, шрифт, привязка и формат вывода будут такими:

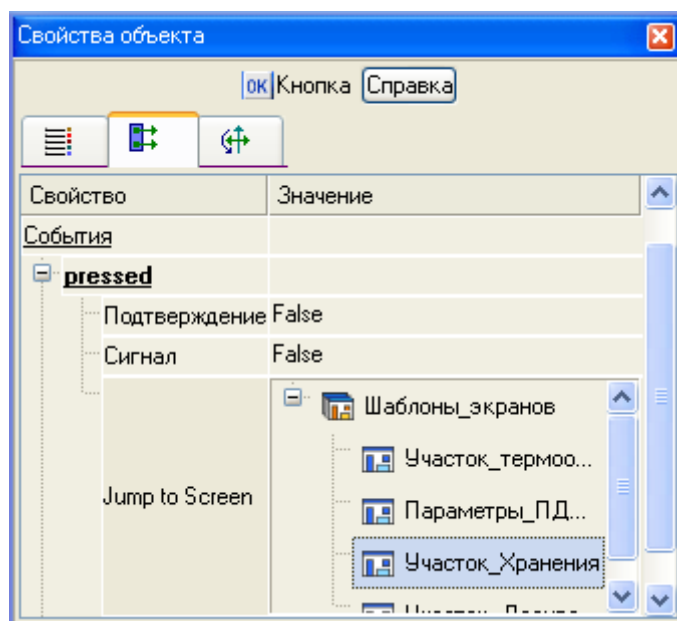


Также на экране разместим надпись – Участок термообработки.

Так как для АРМ будет разработано еще два экрана (создадим в **Шаблоне экранов** дополнительно экраны: Участок\_дозирования, Участок\_Хранения), то

для осуществления переходов между экранами необходимо предусмотреть соответствующие средства. В качестве них будем использовать ГЭ Кнопка .

Двойным щелчком ЛК на ГЭ Кнопка – Хранение откроем его свойства, в разделе События  выделим ЛК пункт **pressed** (событие по нажатию ЛК на ГЭ) и по нажатию ПК добавим переход на экран. В открывшемся пункте **Jump to Screen** по нажатию ЛК из списка выберем шаблон экрана Участок\_хранения:

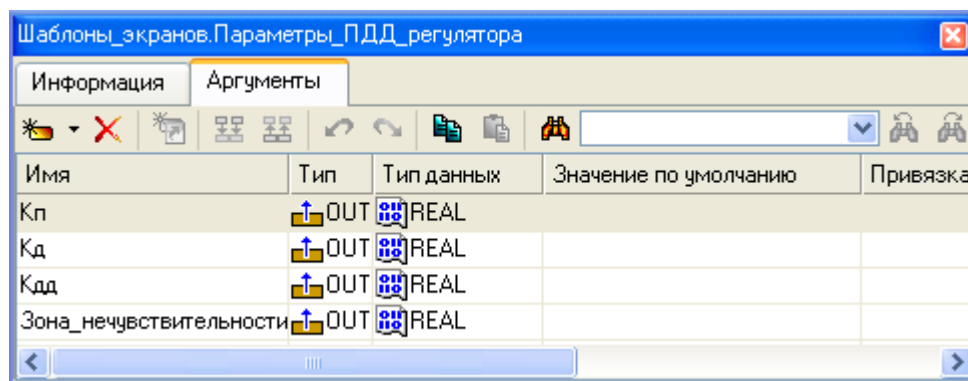


Подобным образом поступим для организации перехода на экран участка дозирования.

Параметры ПДД-регулятора – Кп, Кд, Кдд и Зону нечувствительности будем формировать с помощью всплывающего окна, открытие данного окна оформим также с помощью ГЭ Кнопка, который разместим в левой части экрана, привязку выполним как в описанных выше случаях.

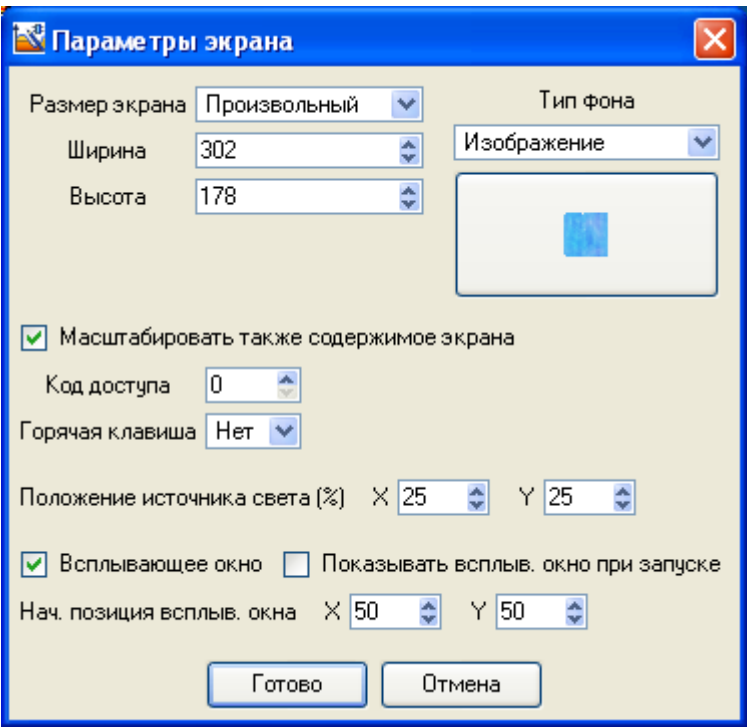
Аргументы экрана Параметры\_ПДД-регулятора зададим следующим образом:


Откроем экран на редактирование. Для задания экрану свойств выпадающего



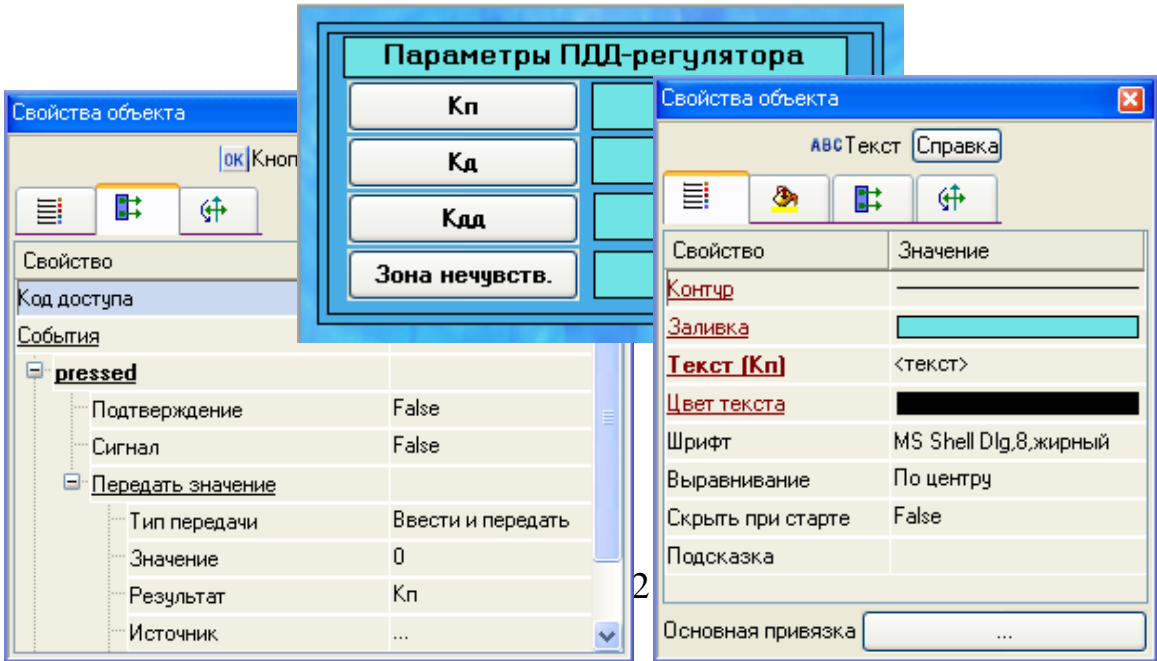
окна выберем в основном меню пункт **Сервис – Параметры экрана**. В открыв-

шемся диалоговом окне зададим размеры экрана, фон, определим экран как всплывающее окно и укажем начальную позицию при первом вызове:



На данном экране разместим ГЭ **Рамка** , переместим его вниз с помощью соответствующей позиции контекстного меню вызываемого нажатием **ПК** на ГЭ, затем в верхней части экрана с помощью ГЭ **Текст** зададим заголовок экрана – Параметры ПДД-регулятора. Далее разместим ГЭ **Кнопка** для посылки значений параметров и левее ее ГЭ **Текст**, для их отображения. Воспользовавшись инструментарием для размножения форм отображения **Меню Сервис – Тиражирование** создадим кнопки ввода и окна отображения для других аргументов экрана. В итоге получим:

Отредактируем созданные ГЭ, приведя их привязки в соответствие с



и  
ана.  
при-

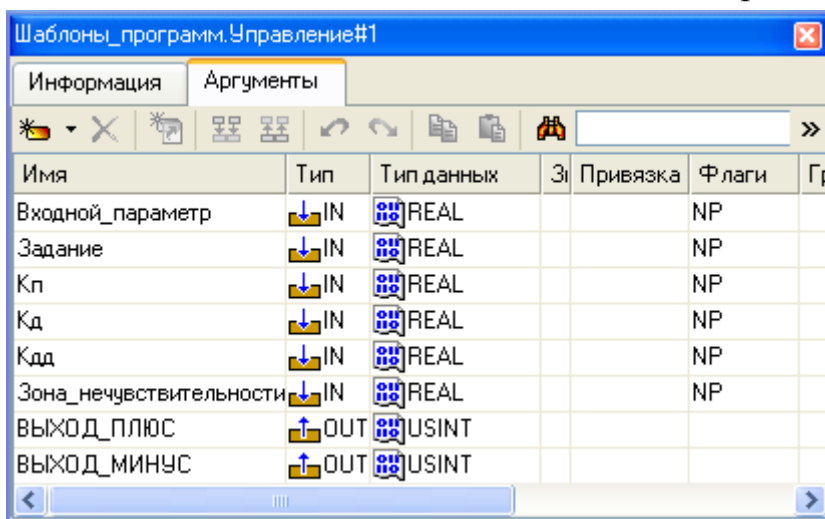
мер, для ГЭ осуществляющих ввод и отображение коэффициента пропорционального усиления (Кп) получим:

Сохраним выполненную работу. Таким образом, графическая часть проекта для АРМ выполнена и настал черед разработки математического обеспечения нашего проекта – шаблонов программ.

## Написание программ

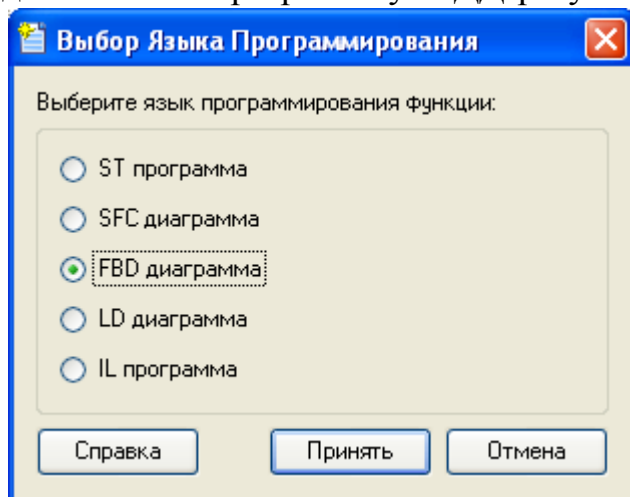
Продолжая разработку проекта принятым способом, создадим шаблон программы, реализующую управляющую функцию поддержания температуры. В левом окне навигатора проекта ЛК выберем слой **Шаблоны программ**, по щелчку ПК создадим компонент Программа#1.

Выделив созданный компонент ЛК, изменим его имя на Управление#1, так как




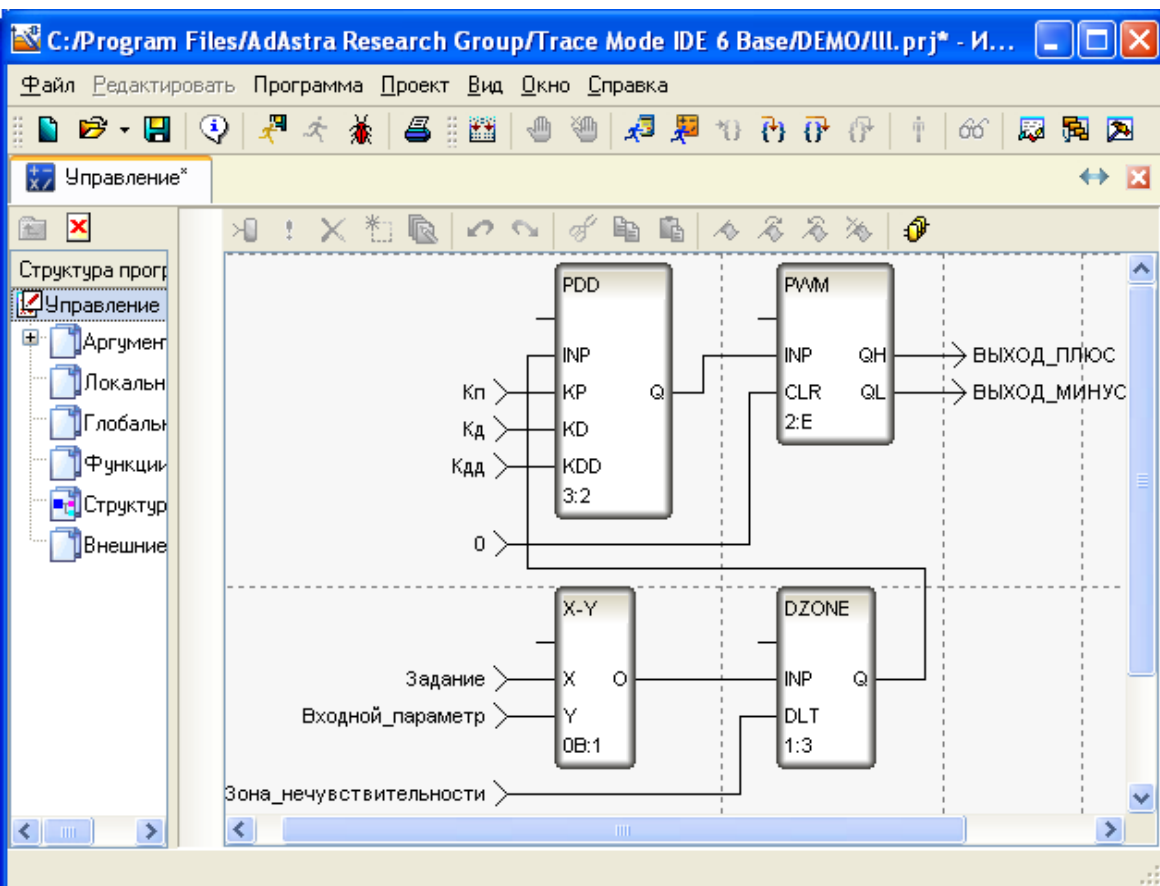
данная программа будет создана для загрузки в PC-based контроллер с целью поддержания заданной температуры в аппарате на участке термообработки.

Двойным щелчком ЛК на компоненте Управление#1 откроем окно редактора шаблонов программ и, выделив ЛК пункт **Аргументы**, перейдем в табличный редактор аргументов. Создадим аргументы для данного шаблона программы исходя из ТЗ на разработку ПДД-регулятора с управлением исполнительным механизмом методом широтно-импульсной модуляции (ШИМ).




После определения входных/выходных аргументов приступим непосредственно к разработке программы. Для этого выделим **ЛК** имя созданного шаблона и в появившемся диалоге выбора языка программирования выберем Техно FBD (*Function Block Diagram*).

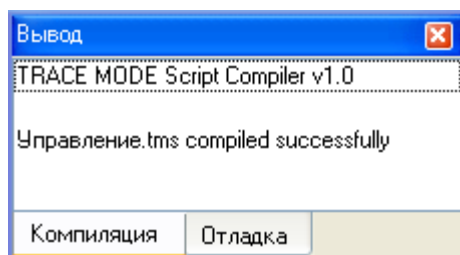
В открывшемся окне создания программ выберем **ЛК** иконку  с подсказкой «Показать/скрыть палитру FBD блоков» для доступа к библиотеке функциональных блоков, назначаем их в рабочее поле редактора, группируем, определяем внутренние связи между входами и выходами блоков, назначаем привязки к



аргументам. Готовая программа выглядит, так как показано на рисунке.

Перед отладкой разработанный шаблон программы необходимо скомпилировать. Для этого используем функциональную клавишу **F7**. Результат компиляции показывается в окне **Вывод** , которое может быть открыто из основного меню интегрированной среды разработки (меню **Вид** – пункт **Вывод**).

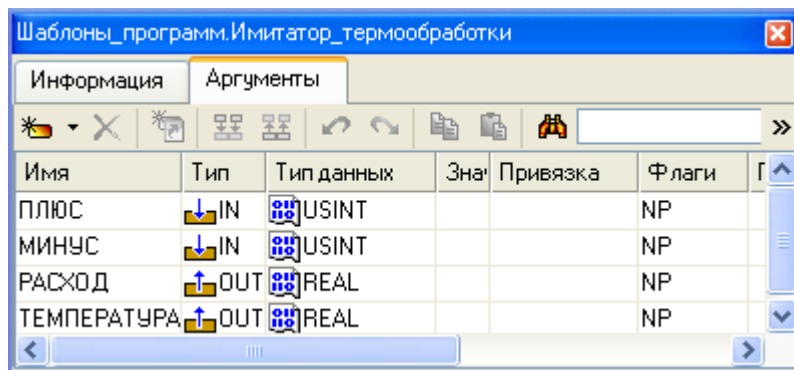
В нашем случае окно содержит сообщение об удачном окончании процесса



компиляции:

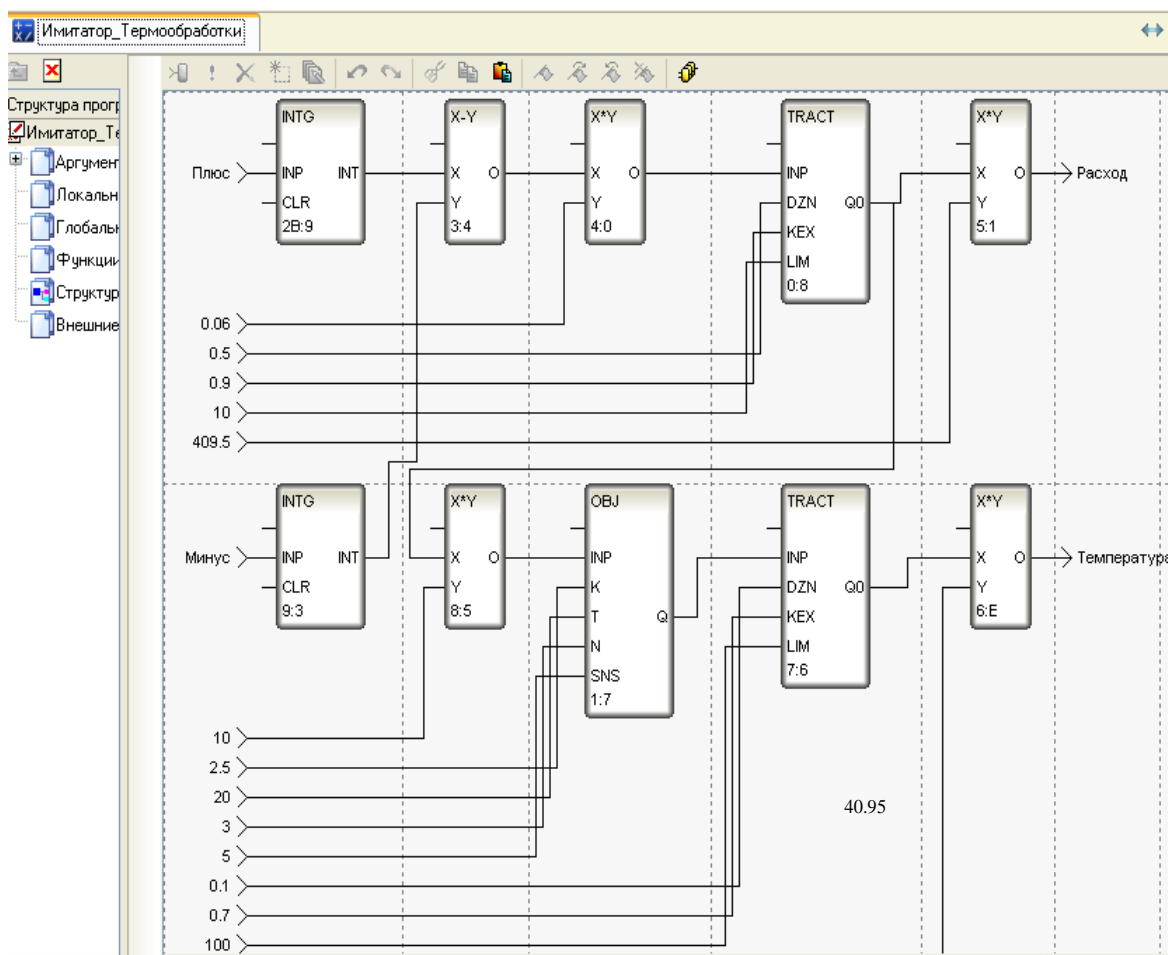
Для проверки работоспособности созданного проекта прибегнем к имитации технологического объекта на участке термообработки. В слое шаблонов программ создадим программу Имитатор\_термообработки и зададим ей аргументы:

Входные аргументы **Плюс** и **Минус** – сигналы от ПДД – регулятора с ШИМ, а



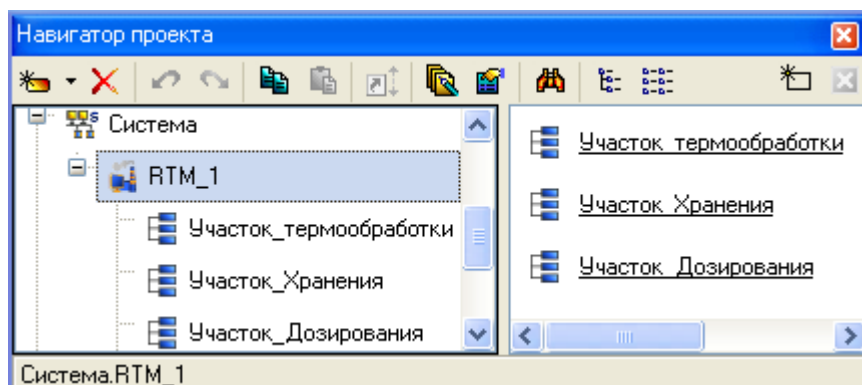
выходные аргументы соответственно расход теплоносителя и температура в аппарате. Построим математическую модель аппарата, основываясь на предположении, что его функционирование описывается классическим звеном – инерционным звеном первого порядка с запаздыванием. Исходя из постановки задачи, для создаваемой модели входным воздействием будет служить расход теплоносителя, выходным – температура в аппарате. Параметры объекта назначим следующими: коэффициент передачи (усиления) 2.5, постоянная времени 20 (в тактах пересчета), запаздывание 3 (в тактах пересчета), а также наложим на его выход гармонические и случайные помехи. В качестве языка программирования применим Техно FBD.

Функциональные блоки **TRACT** применены для сглаживания и ограничения формируемых выходных параметров, а функциональные блоки **X\*Y** 5 и 6 – для приведения значений к выходному диапазону 12-ти разрядного аналого-цифрового преобразователя. Откомпилируем созданную программу.




### Узел проекта и база каналов

Произведем выбор слоя **Система** щелчком ЛК мыши. Далее с помощью ПК создадим узел МРВ (монитора реального времени) для АРМ – **RTM\_1**. В узле **RTM\_1** изменим название группы Каналы на **Участок\_Термообработки** и создадим группы **Участок\_Хранения**, **Участок\_Дозирования**.



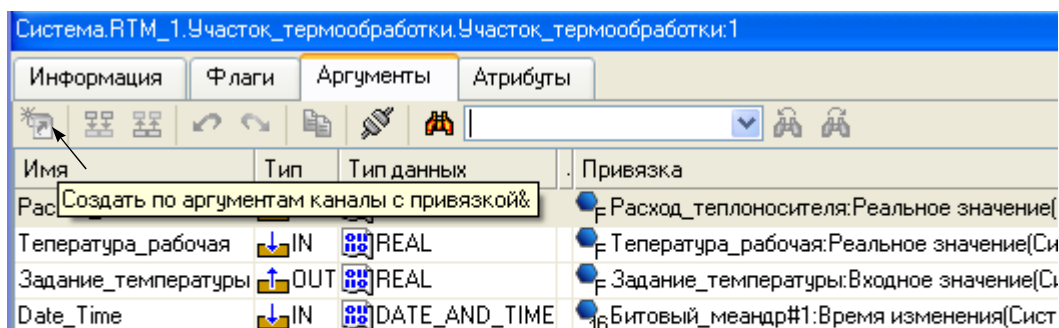


С помощью щелчка ЛК по иконке  с подсказкой «Открыть дополнительное окно навигатора» выполним это действие и откроем в верхнем окне **Шаблоны экранов**, а в нижнем – группу **Участок\_Термообработки** вновь созданного узла АРМ **RTM\_1**.

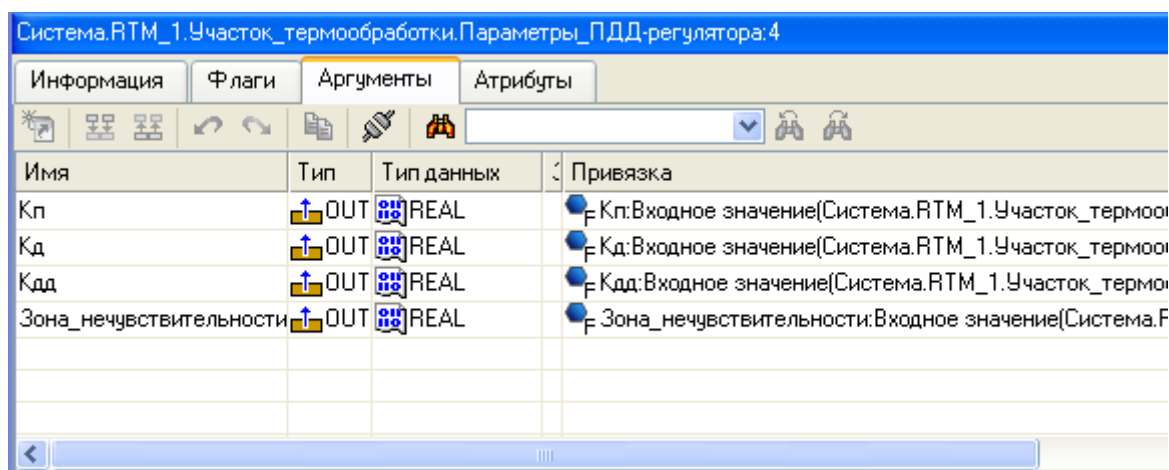
Выделим ЛК мышью экран **Параметры\_ПДД-регулятора** и, удерживая ЛК, «перетащим» его в группу **Участок\_Термообработки** узла **RTM\_1**, отпустим ЛК. Аналогичным образом поступим с экраном **Участок\_термообработки** и с программами **Имитатор\_Термообработки**, **Управление**.

Организуем вывод времени на графическом экране. Создадим в слое **Источники/приемники** группу **Генераторы**. В ней – генератор **Битовый меандр#1**. С помощью мыши перетащим созданный источник в группу **Участок\_Термообработки** узла **RTM\_1**. Открыв свойства канала класса **Вызов Участок\_термообработки** в узле **RTM\_1**, привяжем к аргументу **Date\_Time** атрибут **Время изменения** созданного канала **Битовый\_меандр#1**.

Следующим шагом создадим каналы по аргументам разработанного шаблона экрана. Для этого войдем в группу каналов АРМ – узла **RTM\_1** **Участок\_Термообработки** и вызовем свойства канала-вызова **Участок\_Термообработки**. Перейдем во вкладку **Аргументы**, выделим ЛК первый аргумент и с помощью щелчка ЛК мышью на иконке «Создать по аргументам каналы с привязкой» создадим каналы в выбранной группе и свяжем их атрибуты с аргументами шаблона экрана.



Также поступим при создании каналов по аргументам каналов-вызова **Параметры\_ПДД-регулятора**, **Управление** и **Имитатор\_Термообработки**:



Вызвав на редактирование узел **RTM\_1**, определим период пересчета каналов.

RTM\_1

Основные | Архивы | Отчет тревог / Дамп / Параметры | Таймауты | Дополнительно

Имя: RTM\_1

Атрибуты

Пароль:

Характеристика:

Пересчет

Период: 10

Разрешение: 0.01

Информация | Флаги | Аргументы | Атрибуты

Имя	Тип	Тип данн	Привязка	Фла
Входной_параметр	IN	REAL	Температура_рабочая:Реальное значение(Система.RTM...	NP
Задание	IN	REAL	Задание_температуры:Реальное значение(Система.RT...	NP
Кп	IN	REAL	Кп:Реальное значение(Система.RTM_1.Участок_термо...	NP
Кд	IN	REAL	Кд:Реальное значение(Система.RTM_1.Участок_термо...	NP
Кдд	IN	REAL	Кдд:Реальное значение(Система.RTM_1.Участок_терм...	NP
Зона_нечувствительности	IN	REAL	Зона_нечувствительности:Реальное значение(Система...	NP
ВЫХОД_ПЛЮС	OUT	USINT	16_ВЫХОД_ПЛЮС:Входное значение(Система.RTM_1.Уча...	
ВЫХОД_МИНУС	OUT	USINT	16_ВЫХОД_МИНУС:Входное значение(Система.RTM_1.Уча...	

Для каналов связанных с управлением оборудования, зададим им начальные значения и установим флаг **Отработать** в их основном бланке редактирования. Так для Кп установим 1.000, Кд – 0.1, Кдд – 0.01, Зона\_нечувствительности – 0.5, Задание\_температуры – 20.

Проект Вид Окно Справка

Кд Кп

Имя: Кп Кодировка: TC5

Комментарий:

Границы

☐ Использовать

ВП: 0 ВА: 0 ВГ: 0 НГ: 0 НА: 0 НП: 0

Гистерезис: 0

☐ Контроль границ

Обработка

☐ Использовать

Апертура: 0 Пик: 0 Сглаж.: 0 Множитель: 1 Смещение: 0

☒ Масштабирование

Масштабирование

In	Множитель	Смещение	A
Max	1	+	Max

Системные

Основные

Тип: Input

Размерность: ...

Период: 1 Единица измерения: цикл

Автопосылка

☐ Включить

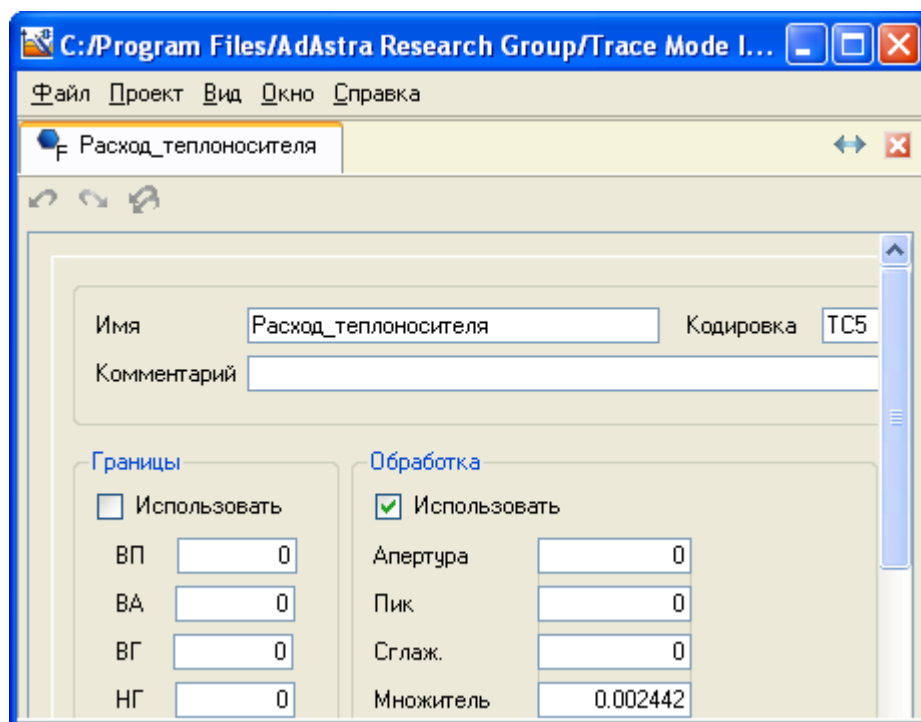
Индекс:

☒ Отработать

На старте: 1

Для аналоговых параметров установим границы и определим процедуру пересчета в физические величины, учитывая то обстоятельство, что все данные поступают от 12-разрядных АЦП, в кодах, лежащих в диапазоне (0 – 4095). Так, например, для параметра Расход\_теплоносителя множитель соответствует – 0.002442, а для Температура\_рабочая – 0.02442.

Запустим программу.



## Порядок проведения лабораторной работы

Выполнить действия, предписанные предыдущими разделами.

Исследовать влияние параметров ПДД – регулятора на длительность (Т) и качество переходного процесса.

Таблица 2.1

Кп		0.5	1	2	10	20	30	Примечание
								д=0.1 дд=0.01 она.нечув.=0.5

Таблица 2.2

Кд		0.02	0.1	1	5	7	10	Примечание
								п=1 дд=0.01 она.нечув.=0.5

Таблица 2.3

дд		002	005	008	01	02	04	примечание
								д=0.1 п=1 она.нечув.=0.5

Таблица 2.4

о- ечув.		05	1	3	5	8	2	примечание
								д=0.1 дд=0.01 п=1

### Содержание отчета

1. Цель исследований.
2. Итоговый экран участка термообработки с указанием функций всех графических элементов
3. Таблицы 2.1 – 2.4.
4. Характеристики  $T = f(K_{п})$ ,  $T = f(K_{д})$ ,  $T = f(K_{дд})$ ,  
 $T = f(K_{она.нечув.})$ .
5. Описание алгоритма программы Управление и Имитатор термообработки.
6. Выводы по работе.

### Контрольные вопросы и задания

1. Дайте описание FBD блоку «Управление устройством типа двигатель» (используйте help системы).
2. Дайте описание FBD блоку «Нечеткий регулятор» (используйте help системы).
3. Дайте описание FBD блоку «Настройка нечеткого регулятора» (используйте help системы).
4. Дайте описание FBD блоку «Блок адаптивного регулирования» (используйте help системы).
5. На какие показатели качества переходного процесса влияют параметры Кдд, Кд, Зона нечувствительности?

### Лабораторная работа №4

#### СОЗДАНИЕ АРХИВА И ОТЧЕТА ТРЕВОГ

Цель работы: построить систему контроля и управления технологического процесса (ТП) участков хранения и дозирования, реализовать функции архивиро-

вания и создания отчета тревог.

### **Указания к выполнению работы**

Участок хранения использует три дискретных входных сигнала – 3DI и четыре аналоговых входных – 4AI, выполняет только функцию мониторинга. На входе контроллера содержатся данные о сигналах состояния входной двери в хранилище (0 – закрыта, 1 – открыта), вентиляции (0 – не работает, 1 – работает) и пожарной сигнализации (0 – задымления нет, 1 – задымление). Контролируются показатели, характеризующие такие параметры, как уровень заполнения резервуара, температура в хранилище, давление и влажность воздуха. Аналоговые величины преобразуются в 12-разрядном аналого-цифровом преобразователе (кода 0 – 4095) и соответственно контролируемые величины имеют диапазоны (0 – 5) м, (0 – 100) С и (0 – 100) %.

Участок дозирования контролируется одним аналоговым входным сигналом – 1AI и одним выходным дискретным – 1 DO. По заданию, передаваемому с АРМ, производится отпуск готового продукта, контроль отпуска ведется по показаниям расходомера ((коды 0 – 4095), расход(0 – 100 л/мин)), в качестве управляемого оборудования выступает дозирующий насос (включение – подача от контроллера логической «1» на модуль гальванической развязки, имеющий в качестве нагрузки пускатель насоса, отключение – логический «0»).

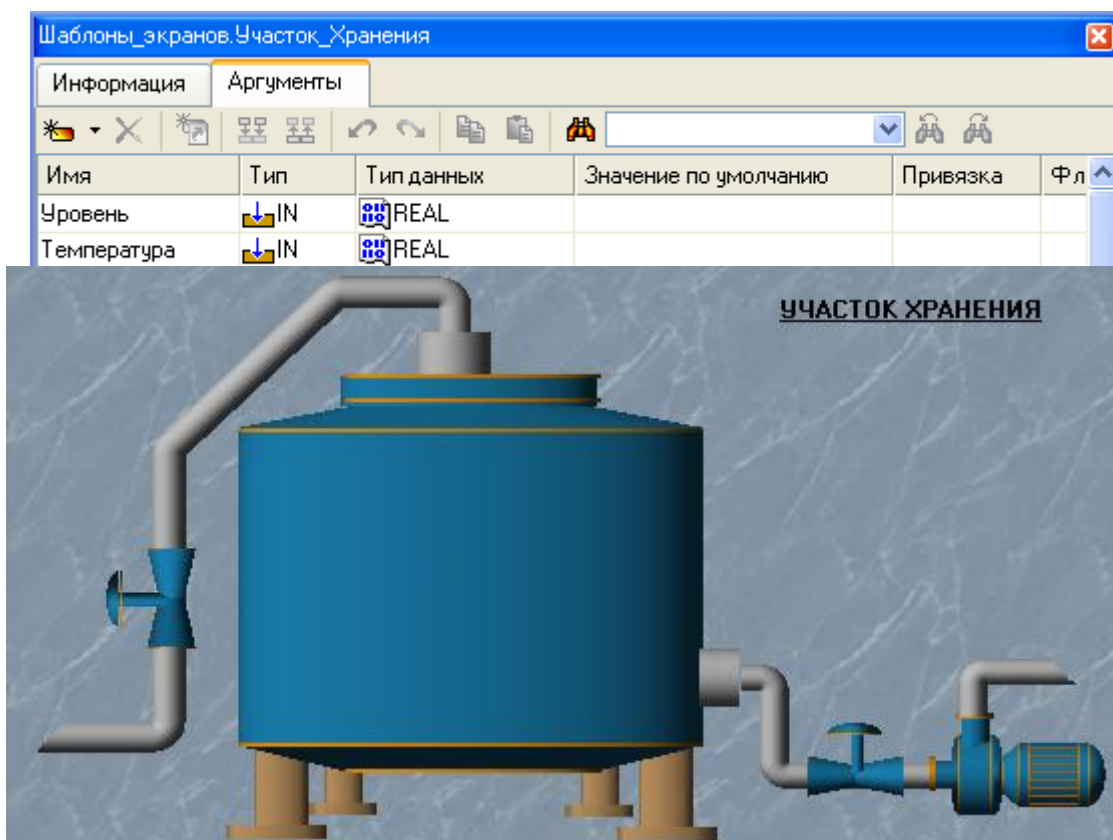
PC – based контроллер подключен к АРМ по сети через концентратор, используемый сетевой протокол – TCP/IP. В качестве контроллера (не моделируется) выступает PC – совместимый компьютер с установленной в системную шину ISA платой ввода/вывода А-8111, работающий под управлением ОС DOS.

Во время работы системы необходимо записывать в таблицу СУБД MS ACCESS данные по параметрам хранения (уровень, температура, давление и влажность) с меткой времени каждые пять минут.

В системе необходимо предусмотреть возможность работы двух пользователей – разработчика и оператора. Оператор в отличие от разработчика не должен иметь возможности вносить изменения в структуру системы.

### **Описание виртуальной лабораторной установки**

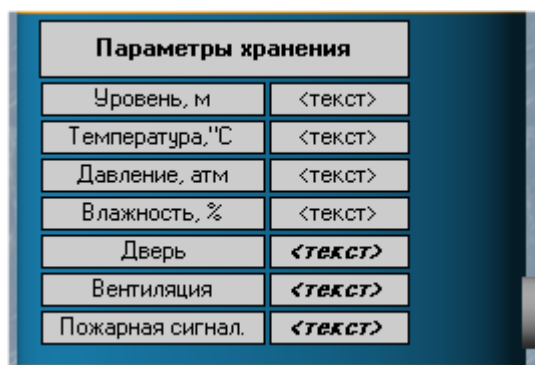
Аргументы для шаблона экрана **Участок\_хранения** будут следующие:



Аргументы **Событие\_Дверь**, **Событие\_Вент** и **Событие\_Пож** предназначены для отображения и квитирования событий с использованием ГЭ **События** 🚨. Аргументы **Слой\_основной** и **Слой\_тренд** – для управления видимостью слоев экрана. В первом слое будет отображаться мнемосхема участка хранения, во втором – тренд значений параметров хранения. Флаг **NP**, выставленный для аргументов не позволит создавать соответствующие каналы при операциях автопостроения.

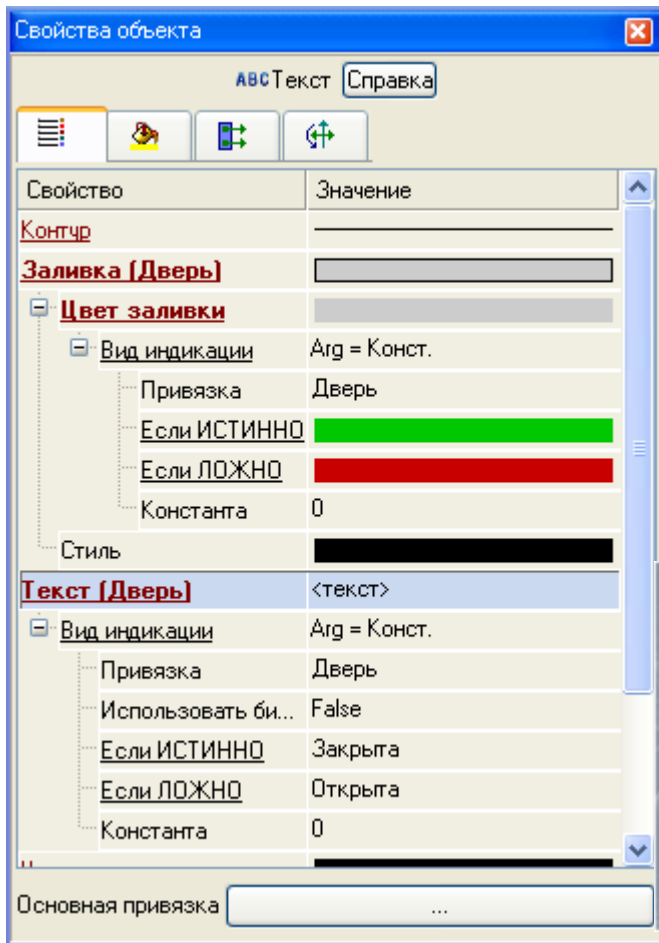
Зададим для экрана в качестве фона изображение – одну из текстур, имеющихся в библиотеке, с помощью ГО 🖼️ и ГЭ 📏, **ABC** выполним статическую часть рисунка так как показано выше.

Определим с использованием ГЭ **ABC** вывод значений параметров хранения.



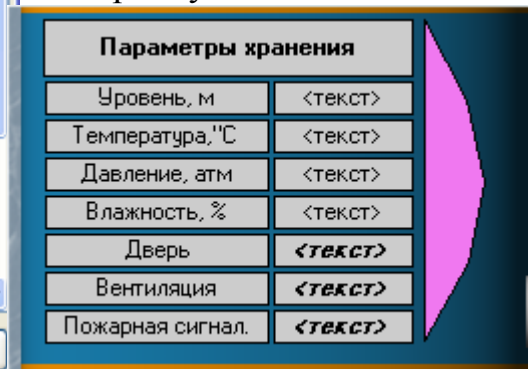
Выполним привязку ГЭ к аргументам шаблона экрана, установим формат вывода значений как, например, для уровня:

<b>Текст (Уровень)</b>	<текст>
<input checked="" type="checkbox"/> Вид индикации	Значение
Привязка	Уровень
<input checked="" type="checkbox"/> Формат	Float
Float	%.1f



Для отображения состояния дискретных сигналов открытия/закрытия двери в хранилище, включения/отключения вентиляции и срабатывания пожарной сигнализации применим совместно цветовую и текстовую индикацию, определяемую для ГЭ **ABC**. Так, для отображения состояния двери в хранилище назначим ГЭ следующие свойства.

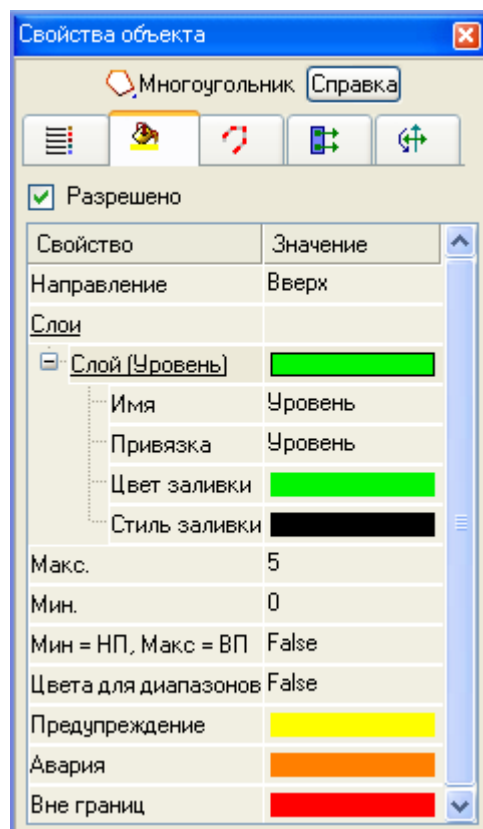
Подобным образом выполним настройку свойств для остальных



будем отображать с помощью гистограммы произвольной формы, которую создадим с помощью ГЭ **Многоугольник**.

В свойствах для данного ГЭ определим динамическую заливку, привязав ее к соответствующему аргументу шаблона экрана, задав цвета фона и заполнения и указав границы.

Последовательно скопируем из шаблона экрана **Участок\_термообработки** и перенесем шаблон **Участок\_хранения** ГЭ для отображения текущего времени и кнопки реходов по экранам. Для этого выделяем ЛК обходимый ГЭ, (для выделения нескольких можно воспользоваться нажатием и ем клавиши **Ctrl** либо выделить мышью ласть при нажатой ЛК) и используем




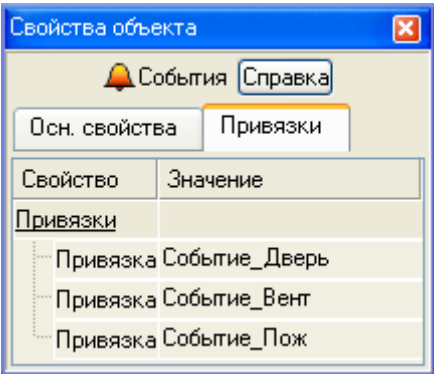
сем  
пе-  
нео  
ГЭ  
ни-  
обл  
па-



нели инструментов. Для вставки – . Можно применять и клавишные комбинации для работы с буфером обмена – **Ctrl-C** и **Ctrl-V**. После вставки ГЭ необходимо произвести их привязку к аргументам текущего шаблона экрана.

В нижней части экрана разместим ГЭ **События** , выделим его ЛК и отцентрируем его горизонтально с помощью соответствующего пункта меню.

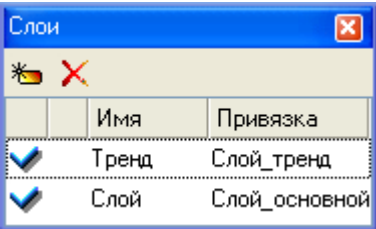
Основные свойства ГЭ  оставим без изменения, во вкладке **Привязки** определим три привязки к аргументам шаблона экрана.






Таким образом, текущий экран будет выглядеть как:

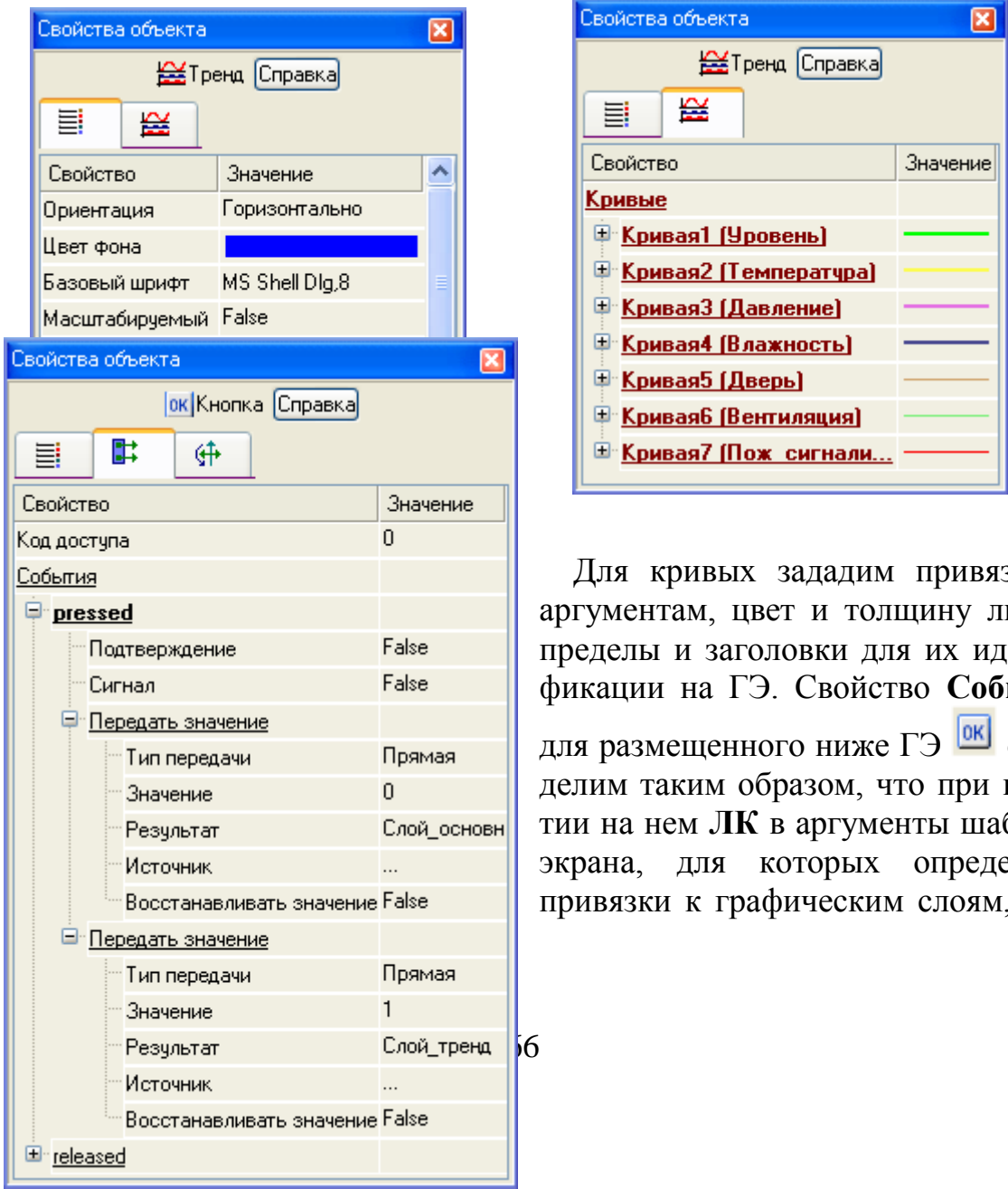



Для вывода на тренде текущих значений параметров хранения создадим дополнительный графический слой для шаблона экрана **Участок\_хранения**. Через пункт **Вид** основного меню откроем окно графических слоев.



В окне слоев переименуем существующий слой в **Слой** и привяжем его к аргументу шаблона **Слой\_основной**. С помощью  создадим новый слой, переименуем его в **Тренд** и привяжем к аргументу **Тренд**. Для вновь созданного слоя в крайней левой позиции окна щелчком **ЛК** сделаем его доступным для редактирования и отображения.

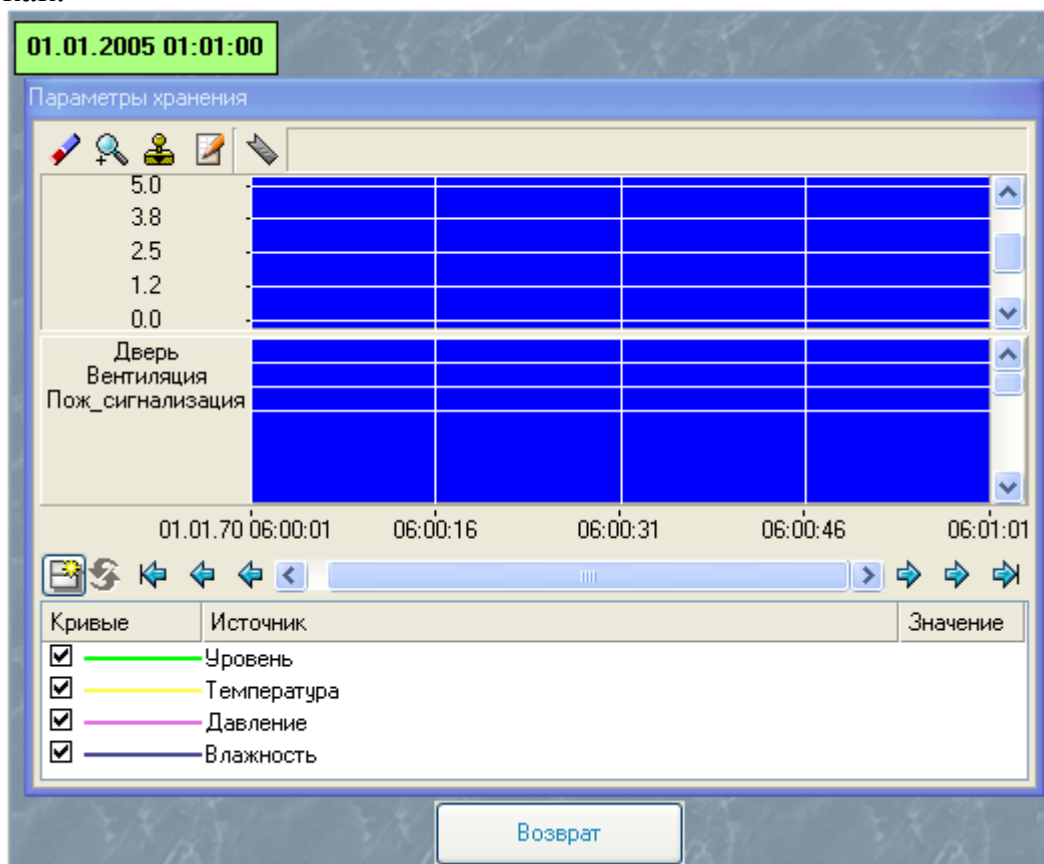
Разместим в данном слое (при этом в окне слоев он должен быть выделен **ЛК**) ГЭ **Тренд**  и **Кнопка** . Для тренда зададим семь новых кривых и определим основные свойства как:



Для кривых зададим привязки к аргументам, цвет и толщину линий, пределы и заголовки для их идентификации на ГЭ. Свойство **События** для размещенного ниже ГЭ  определим таким образом, что при нажатии на нем **ЛК** в аргументы шаблона экрана, для которых определены привязки к графическим слоям, осу-

ществлялись прямые посылки. Значения, посылаемые в данные аргументы, управляют видимостью слоев, **0** – слой отображается, **1** (любое значение, отличное от 0) – нет.

ГЭ **Текст** ABC для вывода текущего времени и названия участка копируются в слой **Тренд** из слоя **Слой**. Слой **Тренд** шаблона экрана **Участок\_хранения** будет выглядеть как:



Для перехода к просмотру значений параметров хранения на тренде, размещенном в слое **Тренд**, в слое **Слой** создадим и определим свойства ГЭ

OK **Тренд** следующим образом.

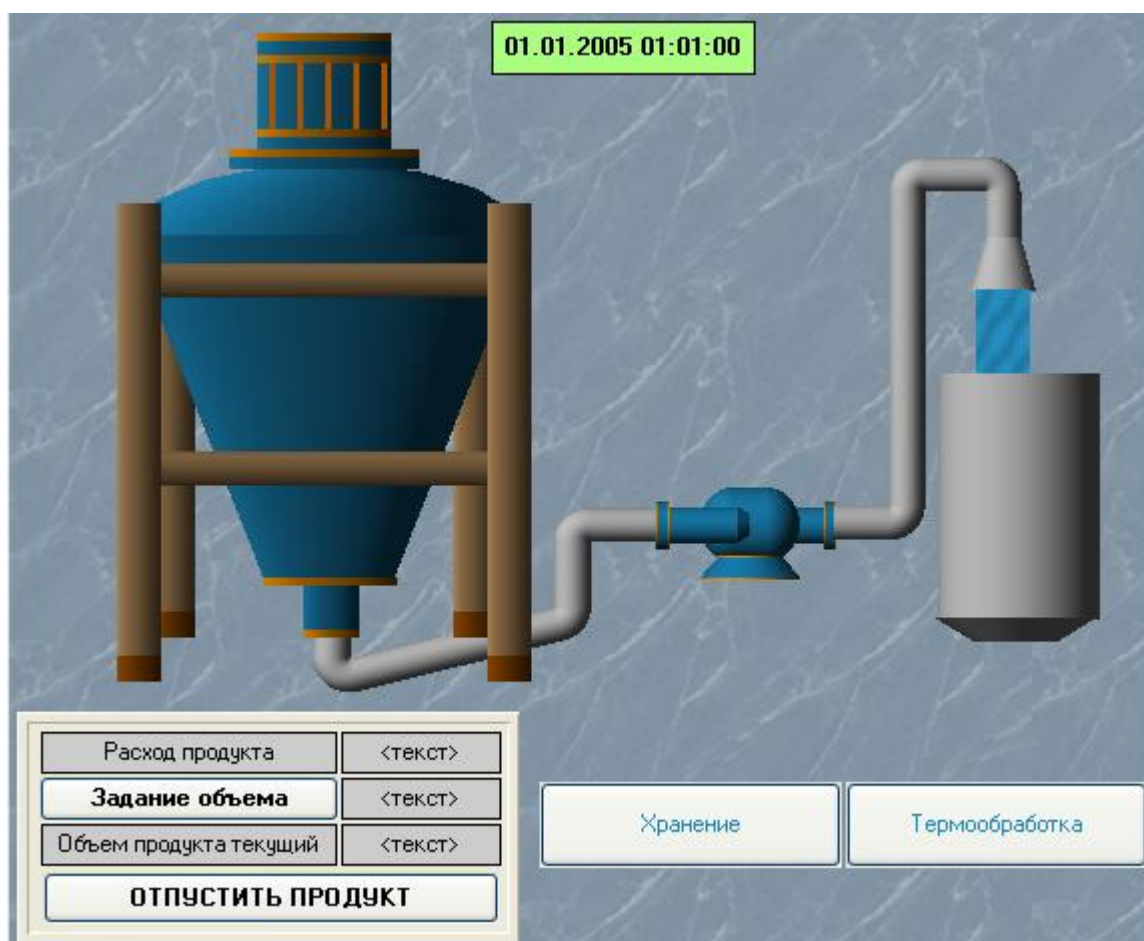
Для того чтобы при запуске проекта в реальном времени были возможны операции со слоями, перед сохранением проекта на диск все слои должны быть отмечены галочками в крайней левой позиции окна слоев экрана.

Свойство	Значение
<input checked="" type="checkbox"/> <b>pressed</b>	
Подтверждение	False
Сигнал	False
<input checked="" type="checkbox"/> <b>Передать значение</b>	
Тип передачи	Прямая
Значение	1
Результат	Слой_основной
Источник	...
Восстанавливать значение	False
<input checked="" type="checkbox"/> <b>Передать значение</b>	
Тип передачи	Прямая
Значение	0
Результат	Слой_тренд

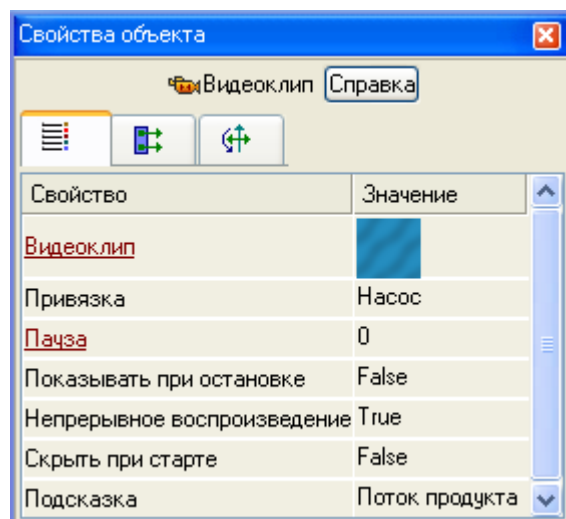
Для шаблона экрана **Участок\_дозирования** аргументы зададим как:


Шаблоны_экранов.Участок_дозирования					
Информация		Аргументы			
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Ф...
Расход_продукта	↓ IN	REAL			
Задание_объема	↑ OUT	REAL			
Разрешение	↑ OUT	USINT			
Объем_продукта_текущий	↓ IN	REAL			
Насос	↓ IN	USINT			
Сброс_объема	↑ OUT	USINT			NP
Date_Time	↓ IN	DATE_AND_TIME			NP

Используя описанные выше приемы, разместим ГЭ и свяжем их с аргументами шаблона экрана таким образом, чтобы получить следующее:

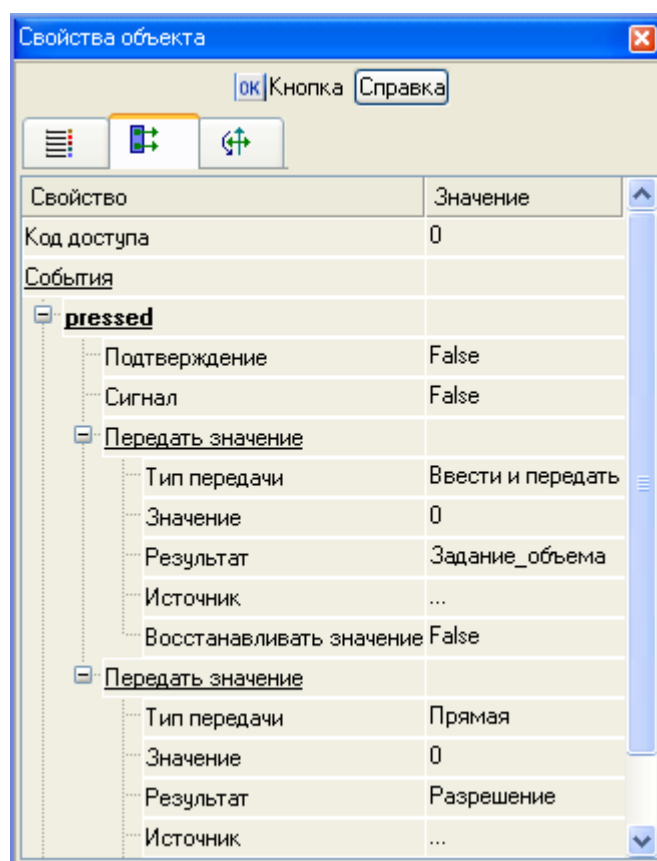


Видеоклип, изображающий поток продукта привязан к аргументу экрана следующим образом:



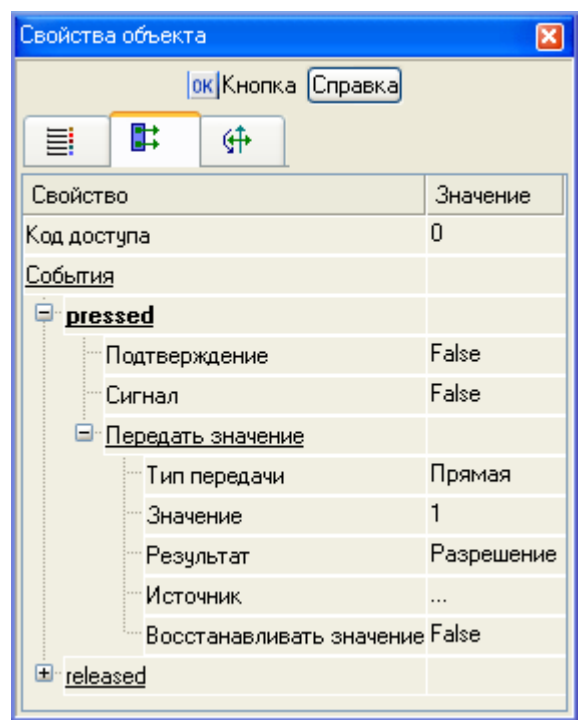
Пояснения требует и ГЭ  **Задание объема**. По нажатию на нем ЛК будет осуществляться посылка значений в два аргумента экрана.

Первое – собственно для задания величины отпускаемого объема продукта,



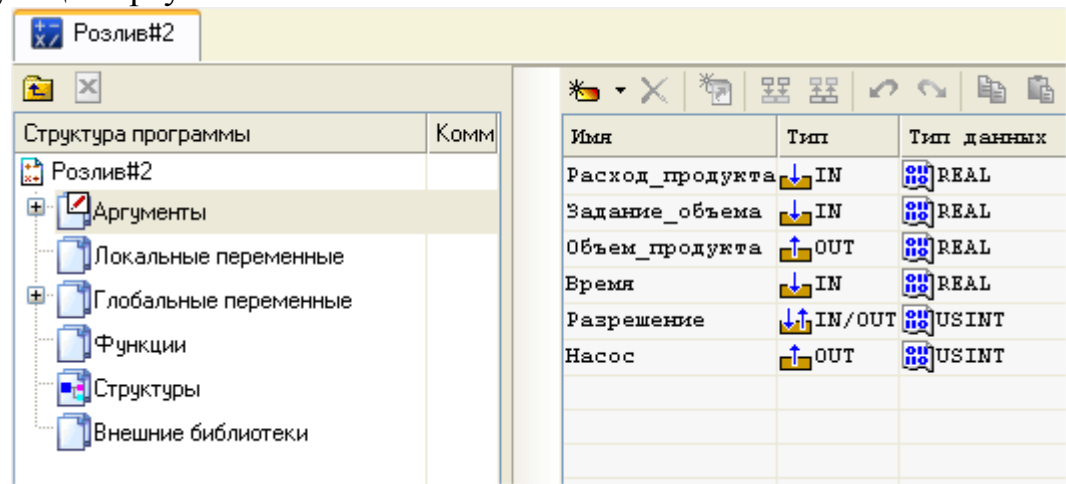
второе – для сброса накопленной в контроллере величины объема по предыдущему циклу разлива. По нажатию ЛК на ГЭ ОТПУСТИТЬ ПРОДУКТ будет посылаться "1" в соответствующий аргумент экрана, в момент, когда нужный объем

продукта будет отпущен, контроллер сбросит атрибут **Входное значение** канала в АРМ, связанный с аргументом экрана **Разрешение** в "0".



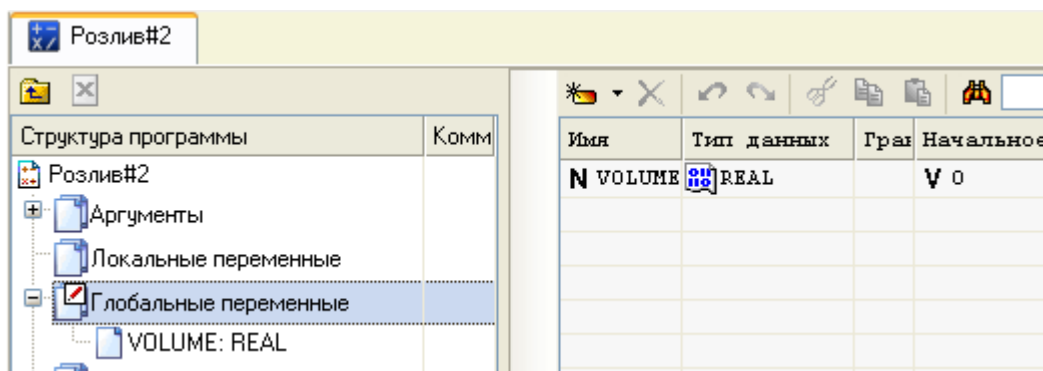
Сохраним выполненную работу, нажав **ЛК** . Таким образом, графическая часть проекта для АРМ выполнена и настал черед разработки математического обеспечения нашего проекта – шаблонов программ.

Для процесса дозирования продукта создадим программу **Розлив#2**, задав для нее следующие аргументы:



Кроме того, определим одну глобальную переменную для накопления в ней величины объема отпускаемого продукта:





Для разработки программы воспользуемся языком Техно ST, для чего в окне выбора языка программирования укажем **ST программа**.

Текст программы может выглядеть следующим образом (показано ниже).

```


PROGRAM
    VAR_INPUT Расход_продукта : REAL; END_VAR
    VAR_INPUT Задание_объема : REAL; END_VAR
    VAR_OUTPUT Объем_продукта : REAL; END_VAR
    VAR_INPUT Время : REAL; END_VAR
    VAR_INOUT Разрешение : USINT; END_VAR
    VAR_OUTPUT Насос : USINT; END_VAR











































    //Подготовка к процессу
    if Разрешение==0 then
        //Удержание насоса выключенным
        Насос=0;
    end_if;
    //Проверка разрешения
    if Разрешение==1 then
        //Включение насоса
        Насос=1;
        //В каждом такте считаем накапливаемый объем
        //(время переводим из мс в минуты)
        //коды АЦП от сигнала расходомера - в физическую величину (л/мин);
        //накапливаемый объем заносим в глобальную переменную - аккумулятор
        VOLUME=VOLUME+Расход_продукта*(Время/1000.0/60.0);
        Объем_продукта=VOLUME;
    end_if;
    //Проверка условия заполнения, погрешность не учитываем
    if Разрешение==1 && Объем_продукта>=Задание_объема then
        //Отключение насоса и сброс в ноль аргумента Разрешение
        Насос=0;
        //Запрет дальнейшего расчета объема
        Разрешение=0;
        //Сброс аккумулятора
        VOLUME=0;
    end_if;

END_PROGRAM

```

Откомпилируем программу для включения в проект нажатием **F7**.

Сохраним выполненную работу, нажав ЛК . После того как графическая часть АРМ и математическое обеспечение для PC-based контроллера выполнены, необходимо приступить к созданию узлов проекта и сформировать их базы каналов.

Информация	Флаги	Аргументы	Атрибуты		
<div><div><div><div></div><div></div></div></div><div><div></div><div></div></div><div></div></div>					
Имя	Тип	Тип данных	Значение	Привязка	Флаги
Уровень	 IN	 REAL		 Уровень:Реальное значение(Систе...	
Температура	 IN	 REAL		 Температура:Реальное значение(Си...	
Давление	 IN	 REAL		 Давление:Реальное значение(Систе...	
Влажность	 IN	 REAL		 Влажность:Реальное значение(Сист...	
Дверь	 IN	 USINT		 Дверь:Реальное значение(Система...	
Вентиляция	 IN	 USINT		 Вентиляция:Реальное значение(Сис...	
Пож_сигнализация	 IN	 USINT		 Пож_сигнализация:Реальное значе...	
Событие_Дверь	 IN/OUT	 REAL			NP
Событие_Вент	 IN/OUT	 REAL			NP
Событие_Пож	 IN/OUT	 REAL			NP
Date_Time	 IN	 DATE_AND_TIME			NP
Слой_основной	 IN	 USINT	0		NP
Слой_тренд	 IN	 USINT	1		NP









В узле RTM\_1 создадим группы каналов **Участок\_Хранения** и **Участок\_Дозирования**. Выделяя ЛК соответствующие шаблоны экранов и удерживая ЛК, перетащим их в эти группы узла RTM\_1.



Предварительно для **Участка\_Хранения** зададим для аргументов **Слой\_основной** и **Слой\_тренд** значения по умолчанию – соответственно **0** и **1**. После выполнения процедуры автопостроения получим результат представленный выше.






















Выполним автопостроение каналов для **Участка\_Дозирования**.

Система.RTM\_1.Участок\_Дозирования.Участок\_дозирования:3

ИнформацияФлагиАргументыАтрибуты






Имя	Тип	Тип данных	Привязка	Флаги
Расход_продукта	 IN	 REAL	 Расход_продукта:Реальное значение(Сист...	
Задание_объема	 OUT	 REAL	 Задание_объема:Входное значение(Сист...	
Разрешение	 OUT	 USINT	 16 Разрешение:Входное значение(Система...	
Объем_продукта_те...	 IN	 REAL	 Объем_продукта_текущий:Реальное зна...	
Насос	 IN	 USINT	 16 Насос:Реальное значение(Система.RTM...	
Сброс_объема	 OUT	 USINT	 Объем_продукта_текущий:Входное знач...	NP
Date_Time	 IN	 DATE_AND_TIME	 16 Битовый_меандр#1:Время изменения(С...	NP








Также в слое **Источники/приемники** создадим группу компонентов **Диагностика\_и\_сервис\_3**. В ней – подгруппу **Диагностика\_1**, а в ней – компонент **@Calc\_Loop\_1**, контролирующий в миллисекундах время пересчета базы каналов узла. Данный источник (с атрибутом **значение**) свяжем с каналом **Время** (пункт **Свойства – Привязка**), расположенным в группе **Участок\_Дозирования**.

Каналы **Date\_Time** участков хранения и дозирования свяжем с каналом **Битовый Меандр Участка\_Термообработки** (атрибут – **время изменения**). В результате для **Участка\_Дозирования** получим базу каналов, представленную выше.

Для аналогового параметра установим границы и определим процедуру пересчета в физические величины, учтя то обстоятельство, что все данные поступают от 12-ти разрядного АЦП, т.е. в кодах, лежащих в диапазоне чисел (0 – 4095). Для параметра **Расход\_продукта** (пункт – **редактировать**) множитель соответствует – 0.02442.

Создадим дополнительное окно **Навигатора проекта** и откроем в нем слой **Шаблонов программ**. Перетащим шаблон программы **Розлив#2** в группу **Участок\_Дозирования**. В результате будет создан вызов программы **Розлив#2**. Для создания соответствующих каналов в группе **Участок\_Дозирования** и связывания их атрибутов с аргументами программы перейдем во вкладку **Аргументы** канала класса **Вызов Розлив#2** и с помощью щелчка ЛК мыши на иконке  получим требуемый результат.

Система.RTM_1.Участок_Дозирования.Розлив#2:10				
Информация	Флаги	Аргументы	Атрибуты	
				
Имя	Тип	Тип данных	Привязка	Флаг
Расход_продукта	↓ IN	REAL	Расход_продукта:Реальное значение(Система.RTM_1...	NP
Задание_объема	↓ IN	REAL	Задание_объема:Реальное значение(Система.RTM_1...	NP
Объем_продукта	↑ OUT	REAL	Объем_продукта_текущий:Реальное значение(Систе...	NP
Время	↓ IN	REAL	Время:Реальное значение(Система.RTM_1.Участок_...	
Разрешение	↓ IN/OUT	USINT	Разрешение:Входное значение(Система.RTM_1.Участ...	NP
Насос	↑ OUT	USINT	Насос:Входное значение(Система.RTM_1.Участок_Д...	NP

Выполним настройку архива и отчета тревог в APM.

Для помещения реальных значений каналов в архив **SIAD/SQL 6** и ведения отчета тревог по таким событиям, как пересечение заданных для канала уставок и границ, необходимо для узла проекта предварительно задать ряд общих параметров, в дальнейшем индивидуально или групповым способом установить для каналов соответствующие атрибуты **СПАД** и/или **Отчет тревог**. Так, вызвав на редактирование узел **RTM\_1**, определим файл архива – **СПАД 1**.

Основные | **Архивы** | Отчет тревог / Дамп / Параметры | Таймауты | Дополнительно

СПАД 1

Имя файла: archive.rep

Состояние архива: TRUE

Размер файла, Мб: 128

Кэш

Период сохранения: 300

Размер, Мб: 48

Максимум очереди записей: 450000

Во вкладке **Отчет тревог/Дамп/Параметры** определим следующие параметры:

Основные | Архивы | **Отчет тревог / Дамп / Параметры** | Таймауты | Дополнительно

Отчет тревог

Имя файла: alarms.txt

Максимум записей: 1000

Состояние: TRUE

Формат даты: %x %X

В бланке **Основные** задают IP-адрес АРМ (в нашем случае нет необходимости) и направления обмена данными по сети.

Основные | Архивы | Отчет тревог / Дамп / Параметры | Таймауты | Дополнительно

Имя: RTM\_1

Атрибуты

Пароль:

Характеристика:

Пересчет

Период: 10

Разрешение: 0.01

Сеть

Имя компьютера / IP адрес:

Адаптеры

Системный

Первый

Второй

Третий

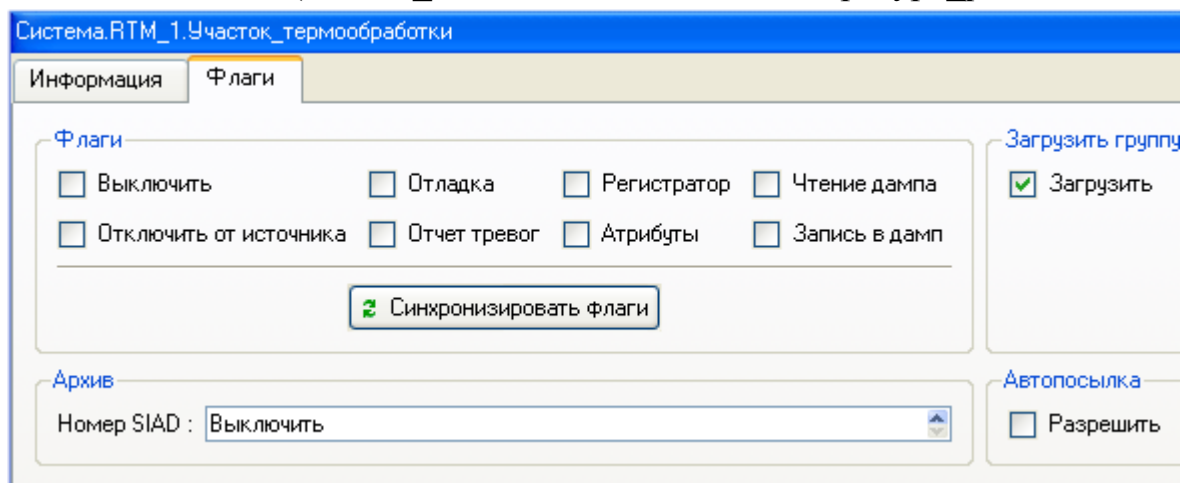
Мост

Назначить: ☐

Номер: 0

Закроем бланк редактирования узла RTM\_1. Для получения возможности просмотра в отладчике сведенных в группы каналов, откроем свойства группы Участок\_Термообработки и установим во вкладке **Флаги** флаг **Загрузить группу**. Подобным образом поступим в отношении других групп.

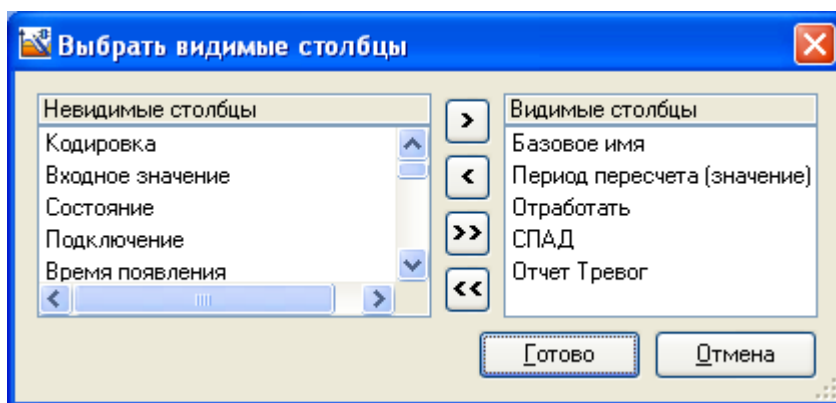
Для каналов (Расход\_теплоносителя, Температура\_рабочая, Зада-



ние\_температуры, Кп, Кд, Кдд, Зона\_нечувствительности) группы **Участок\_Термообработки**, при их редактировании, в бланке **Архивация** установить СПАД – 1.

Для помещения каналов группы **Участок\_Хранения** в архив и отчет тревог вызовем на редактирование данную группу.

В открывшемся окне щелчком **ПК** на строке атрибутов каналов вызовем настроечное меню отображения атрибутов и в правом окне сформируем список отображаемых в таблице атрибутов.



Для выделенных каналов будет определена запись их реальных значений в архив **Тип1**. Для группы **Участок\_Хранения** атрибуты будут установлены следующим образом:

Содержимое группы					
Y= LN					
Базовое имя	Период пересчета (зн)	Отработать	СПАД	Отчет Тревог	
Участок_хранения:2	1	0	No	0	
Уровень	1	0	1	1	
Температура	1	0	1	1	
Давление	1	0	1	1	
Влажность	1	0	1	1	
Дверь	1	0	1	1	
Вентиляция	1	1	1	1	
Пож_сигнализация	1	0	1	1	

Для Участка\_Дозирования:

Содержимое группы					
Y= LN					
Базовое имя	Период пересчета (зн)	Отработать	СПАД	Отчет Тревог	
часток_дозирования:3	1	0	No	0	
Расход_продукта	1	0	1	0	
Задание_объема	1	1	1	0	
эм_продукта_текущий	1	0	1	0	
Насос	1	0	1	0	
Разрешение	1	1	No	0	

Для аналоговых параметров группы каналов **Участок\_Хранения** зададим сообщения в отчет тревог. С этой целью создадим для узла RTM\_1 новую группу **Словари сообщений**.

В данной группе создадим новый компонент – **Словарь сообщений для канала класса Float**.

Двойным щелчком **ЛК** откроем его для редактирования и создадим сообщения для помещения в отчет тревог.

Основные

Имя


Тревоги

Текст	Категория	Направление передачи
NORMA	<> Без категории	AR+G
Переход через ВП границу	<W> Предупреждение	AR+G
Переход через НП границу	<W> Предупреждение	AR+G
Переход через Верхнюю АВАРИЙНУЮ границу	<A> Тревога	AR+G
Переход через Нижнюю АВАРИЙНУЮ границу	<A> Тревога	AR+G
Выход показаний за верхний предел!!!	<E> Ошибка	AR+G
Выход показаний за нижний предел!!!	<E> Ошибка	AR+G

Для подключения созданного словаря к каналам класса **Float** группы **Участок\_Хранения** либо откроем последовательно каналы на редактирование двойным щелчком ЛК и на вкладке **Архивация** укажем индекс аварийного словаря, или используем редактор групповой правки:

Содержимое группы

Y= LN

Базовое имя	Период пересчета (зн)	Отработано	СПАД	Отчет Тревог	Индекс аварийного сл
Участок_хранения:2	1	0	No	0	не назначен
Уровень	1	0	1	1	Словарь соо...  ...
Температура	1	0	1	1	Словарь сообщений д
Давление	1	0	1	1	Словарь сообщений д
Влажность	1	0	1	1	Словарь сообщений д
Дверь					
Вентиляция					
Пож_сигнализация					

Задать привязку

- Участок\_термообработки
- Участок\_Хранения
- Участок\_Дозирования
- Словари\_сообщений
  - Словарь сообщений для Участка\_Хранения

Для каналов группы **Участок\_Хранения**, принимающих от **PLC** дискретные сигналы, также создадим словари для помещения сообщений в отчет тревог, но уже выбрав за основу словарь для каналов класса **HEX16**. В нем определим сообщения для нулевого бита:

**Основные**

Имя:

**Тревоги**

Текст	Категория	Направление передачи
Дверь в хранилище закрылась	<M> Сообщение	AR+G
Дверь в хранилище открылась	<M> Сообщение	AR+G
1bit_Off	<> Без категории	AR+G

Для вентиляции:

**Основные**

Имя:

**Тревоги**

Текст	Категория	Направление передачи
Вентиляция в хранилище отключилась	<M> Сообщение	AR+G
Вентиляция в хранилище включена	<M> Сообщение	AR+G
1bit_Off	<> Без категории	AR+G

Для пожарной сигнализации:

**Основные**

Имя:

**Тревоги**


Текст	Категория	Направление передачи
Пожарная сигнализация - все в порядке	<M> Сообщение	AR+G
ПОЖАР в хранилище!	<M> Сообщение	AR+G
1bit_Off	<> Без категории	AR+G

Подключим созданные словари к каналам таким же образом, как и в рассмотренном выше случае. Сохраним выполненную работу.

Для фиксации событий по изменению состояния дискретных сигналов участка хранения – открытия/закрытия двери в хранилище, состояния системы вентиляции и пожарной сигнализации, а также для их отображения на экране и квити-рования создадим в группе **Участок\_хранения** узла АРМ три канала класса **Событие**. Переименуем их (**Событие\_Дверь**, **Событие\_Вентиляция**, **Событие\_Пожарная\_Сиг-нализация**) и, открыв дополнительное окно **Навигатора**

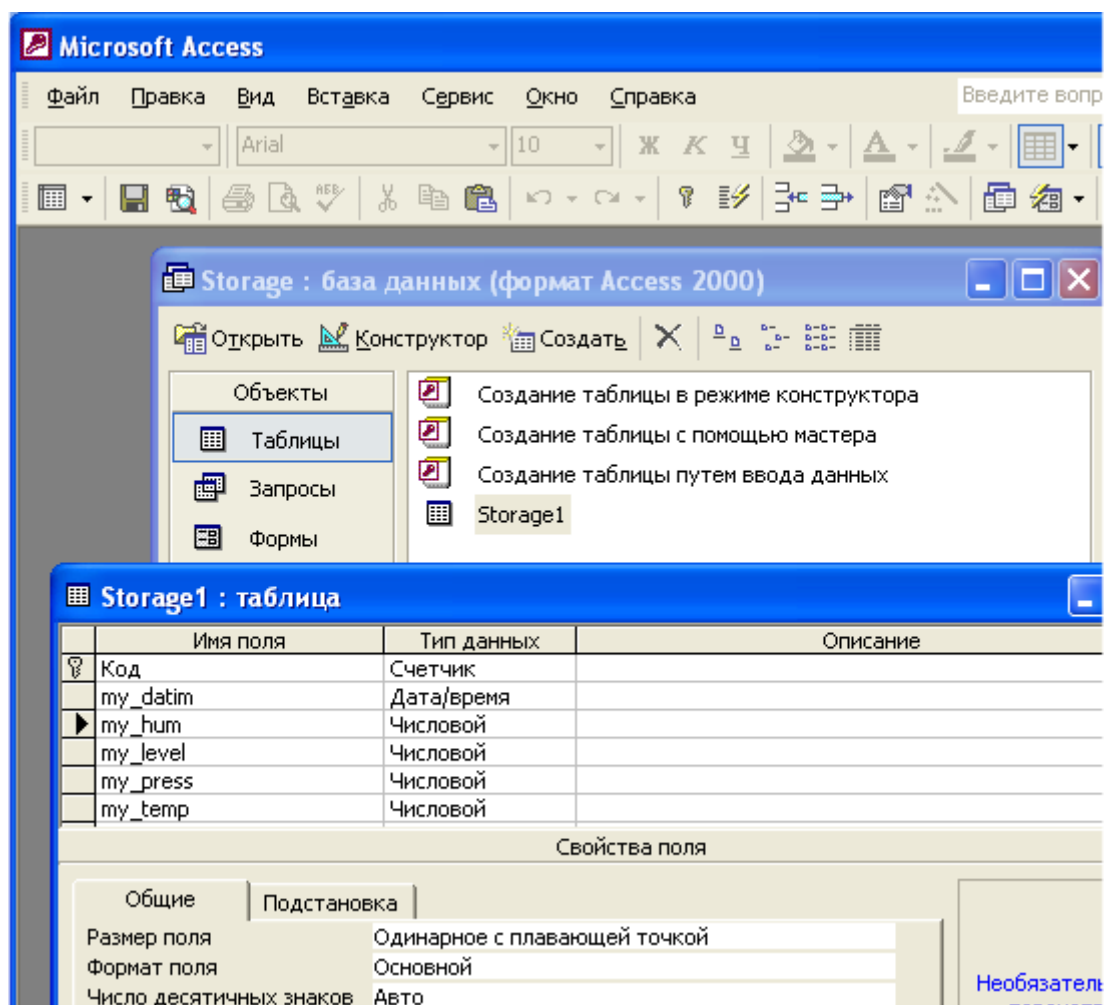
проекта на той же группе, свяжем каналы класса Событие (пункт **Свойства – Привязка**) с помощью мыши с соответствующими каналами класса **HEX16 Дверь, Вентиляция и Пожарная\_сигнализация** (атрибут – **реальное значение**).

Двойным щелчком ЛК мыши откроем на редактирование канал **Событие\_Дверь** и установим атрибут **Размер стека аварий** равным **20**. Таким же образом поступим в отношении других двух каналов.

Так как предварительно нами уже был подготовлен ГЭ  **События** на шаблоне экрана **Участок\_хранения**, выполним привязку атрибутов (**реальное значение**) вновь созданных каналов класса **Событие** к аргументам канала класса **Вызов Участок\_хранения**.

Сохраним выполненную работу, и осуществим связь с СУБД *MS Access*.

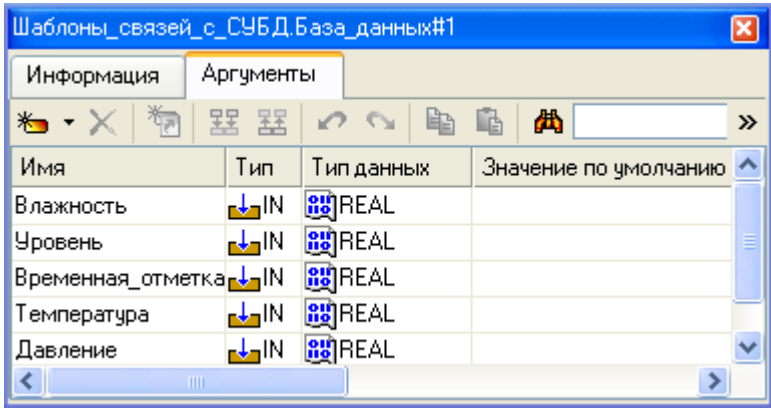
В панели управления *MS Windows* в разделе **Язык и региональные стандарты** во вкладке **Региональные параметры** необходимо настроить в качестве разделителя целой и дробной части числа **точку**.



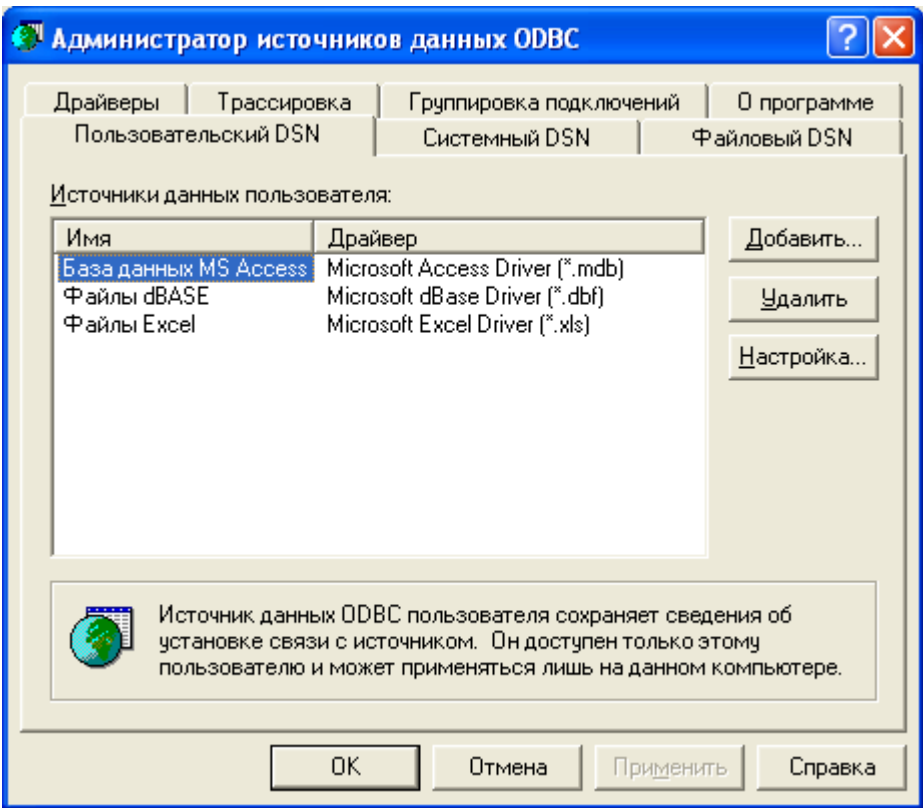
По сформулированному выше ТЗ необходимо данные, получаемые по участку хранения, записывать в таблицу реляционной базы данных. Структура таблицы

**Storage1** базы данных *MS Access Storage.mdb* для накопления значений технологических параметров, поступающих в реальном времени от АРМ будет такой как показано выше (поле **код** – ключевое, длинное целое).

В навигаторе проекта откроем слой шаблонов связей с СУБД и создадим новый компонент **База данных#1**. Откроем свойства шаблона и зададим ему аргументы: Двойным щелчком **ЛК** откроем шаблон на редактирование. По щелчку **ЛК** на



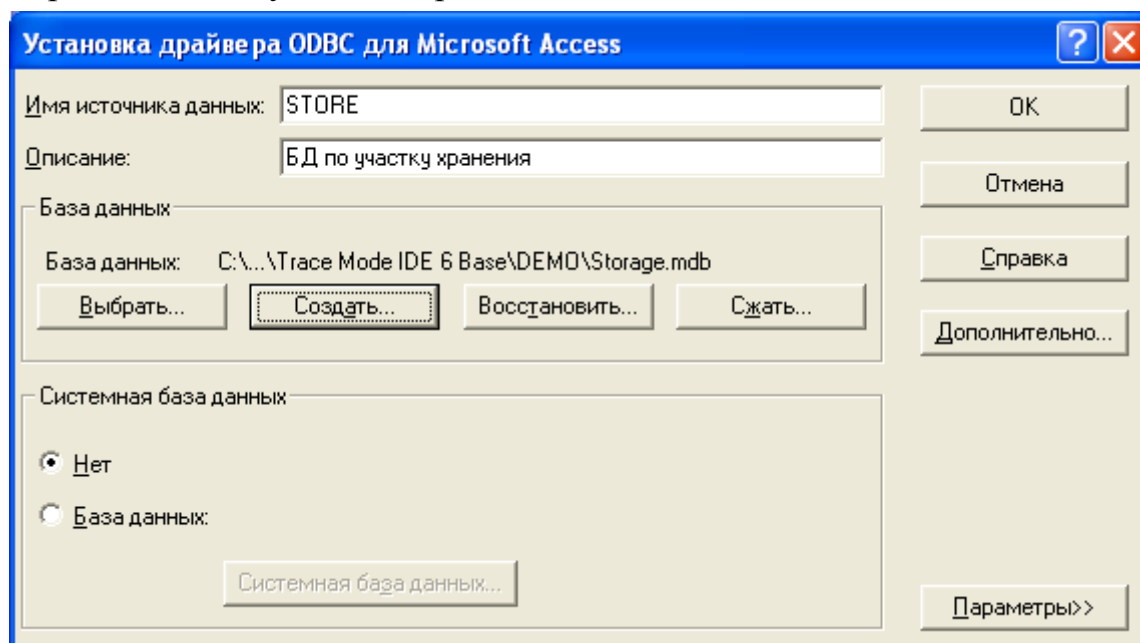
экранной кнопке «Администратор ODBC» во вкладке **Пользовательский DSN** с помощью клавиши «Добавить» создадим новый источник данных – **STORE**,



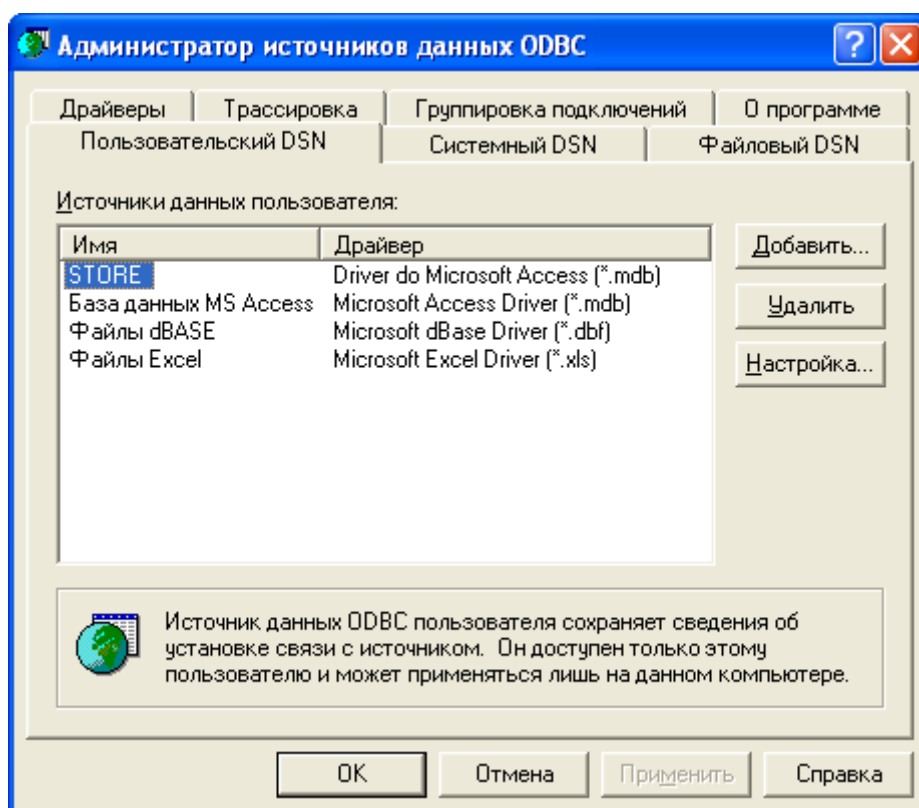
выбрав *Microsoft Access Driver*.



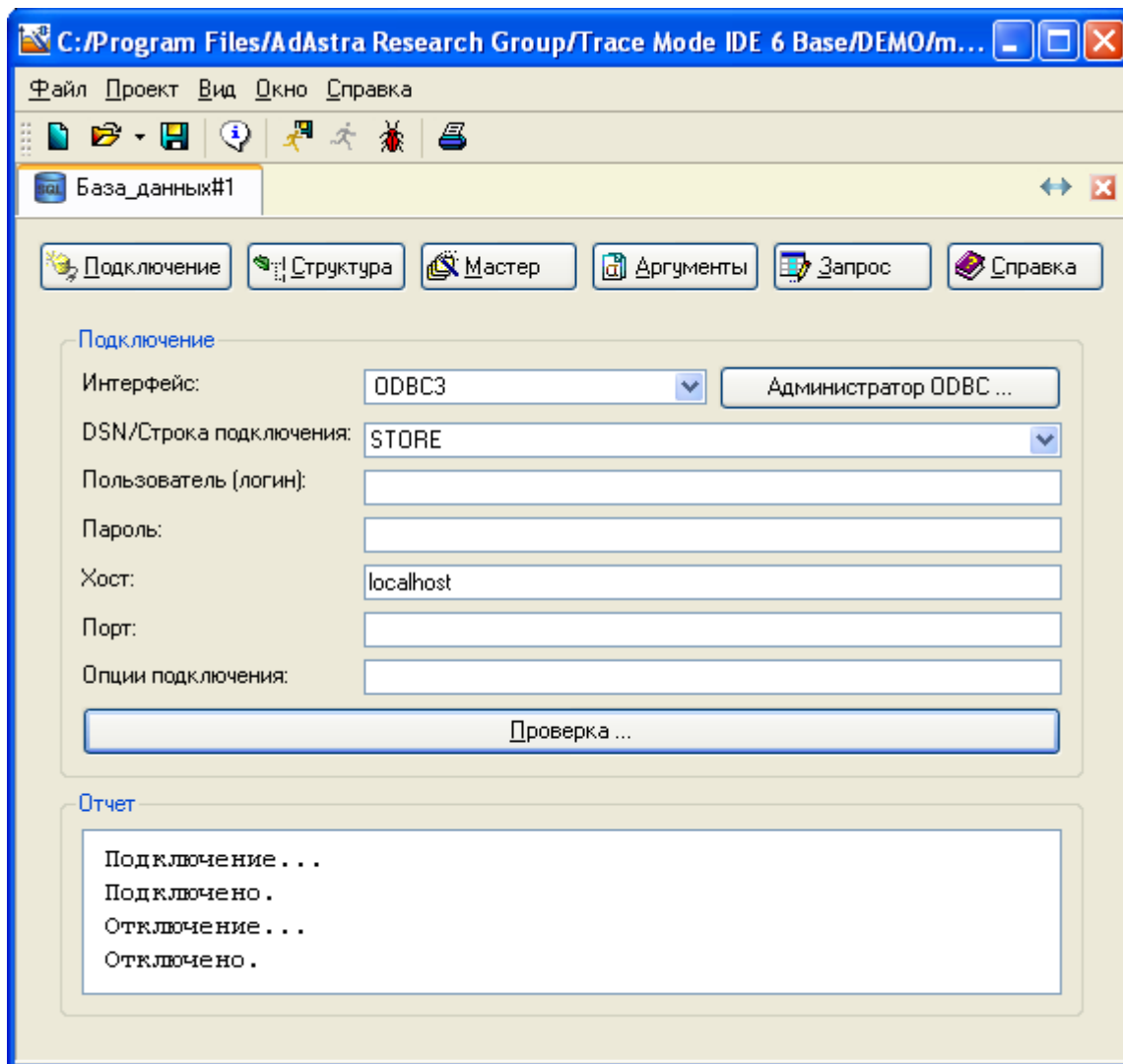
Настроим его следующим образом,



выбрав в этом бланке файл БД *MS Access* – *Storage.mdb* данных по участку хранения и нажмем «**ОК**». В числе пользовательских **DSN** появится новый – **STORE**.



В строке «**DSN/Строка подключений**» укажем – **STORE** и нажмем клавишу «**Проверка**», осуществим таким образом правильность подключения источника данных в окне «**Отчет**»:




Далее или с помощью мастера запросов, либо непосредственно создадим на языке **SQL** запрос, помещающий связанные с аргументами шаблона атрибуты каналов в таблицу **Storage1** файла **Storage.mdb** с созданием новых записей:

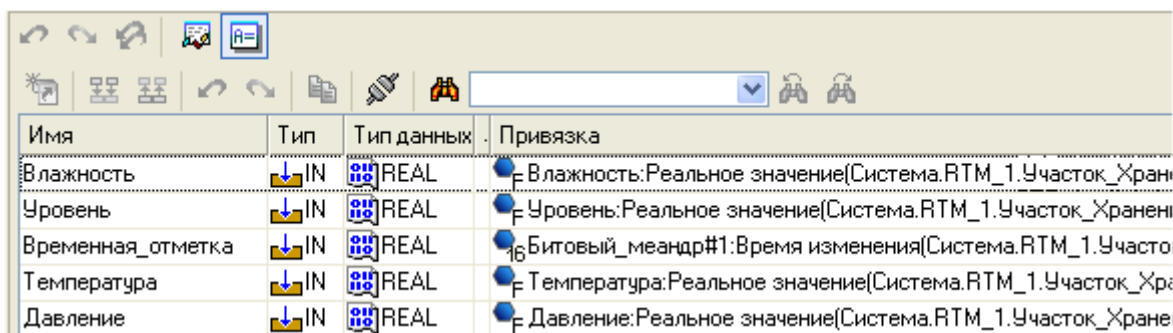
```

INSERT INTO Storage1
(
    my_datim, my_hum, my_level, my_press, my_temp
)
VALUES
(
    '#Временная_отметка#', '#Влажность#', '#Уровень#', '#Давление#',
    '#Температура#'
)

```

Откроем дополнительное окно **Навигатора проекта**, настроив его на отображение группы **Участок\_Хранения** узла АРМ. Выделим ЛК шаблон **База данных#1** и, удерживая ЛК нажатой, перетащим шаблон в группу **Участок\_Хранения**. В группе **Участок\_Хранения** будет создан канал класса **Вызов База данных#1**, щелчком ПК откроем его для редактирования.

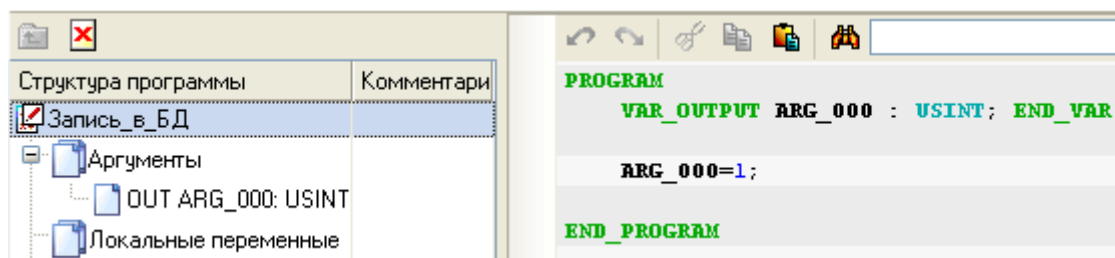
Для связывания аргументов созданного запроса с атрибутами каналов перейдем с помощью иконки  в табличный редактор аргументов и выполним привязку (вручную) к каналам группы **Участок\_Хранения**:




Имя	Тип	Тип данных	Привязка
Влажность	IN	REAL	Влажность:Реальное значение(Система.RTM_1.Участок_Храни
Уровень	IN	REAL	Уровень:Реальное значение(Система.RTM_1.Участок_Храни
Временная_отметка	IN	REAL	16-Битовый_меандр#1:Время изменения(Система.RTM_1.Участок_Храни
Температура	IN	REAL	Температура:Реальное значение(Система.RTM_1.Участок_Храни
Давление	IN	REAL	Давление:Реальное значение(Система.RTM_1.Участок_Храни

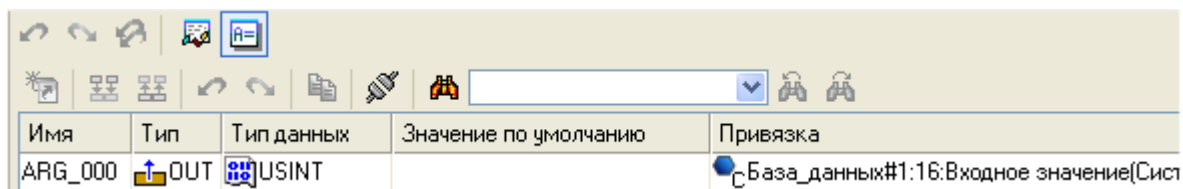
Для того чтобы при запуске на АРМ МРВ каждые пять минут в файл БД Storage.mdb производилась запись данных по участку хранения, необходимо во входное значение канала класса **Вызов База данных#1** подавать номер выполняемого запроса, в нашем случае **1**. Сделать это можно, например, с помощью программы или ГЭ "**Кнопка**". Создадим в слое **Шаблонов программ** новую программу **Запись\_в\_БД**. Реализуем ее следующим образом:




Откомпилируем ее и перетащим в группу **Участок\_хранения**, принадлежащую




узлу **RTM\_1**. Откроем на редактирование канал класса **Вызов Запись\_в\_БД** и установим в основном бланке период, например, **5** циклов.

Перейдем с помощью иконки  к аргументам канала и произведем привязку следующим образом:

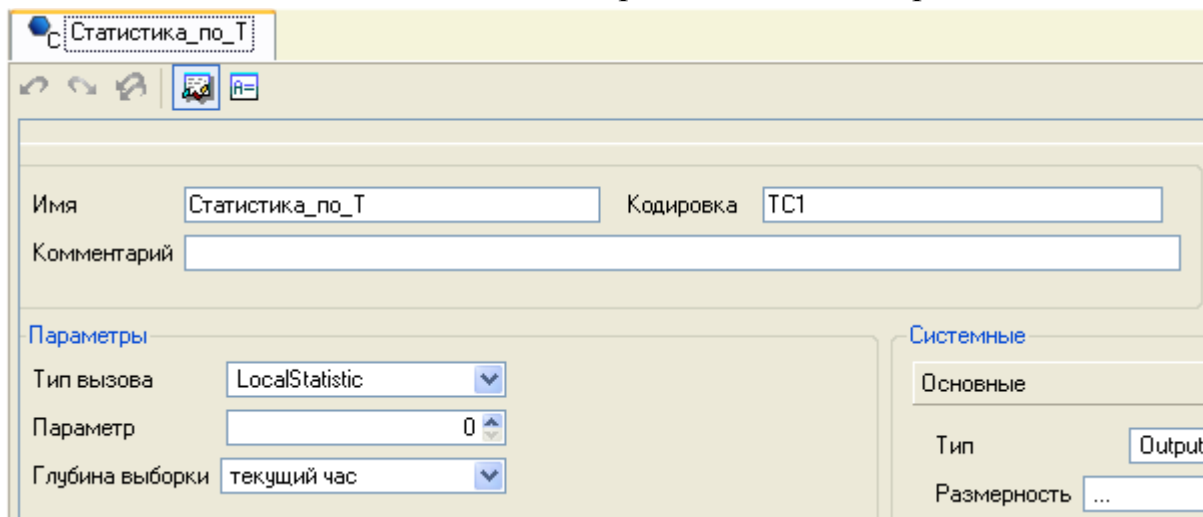


Имя	Тип	Тип данных	Значение по умолчанию	Привязка
ARG_000	 OUT	 USINT		 База_данных#1:16:Входное значение(Сист

Изменяя период пересчета данного канала, можно регулировать темп записи значений параметров участка хранения в файл БД *MS Access*. Сохраним выполненную работу, нажав ЛК .


Поскольку все данные о значениях технологических параметров записываются в архив **SIAD/SQL 6**, то, например, для получения статистических характеристик такого параметра как **Температура\_рабочая** в аппарате участка термообработки воспользуемся реализованным в TRACE MODE механизмом обработки данных, накапливаемых в локальном архиве. Создадим в группе **Участок Термообработки** узла **RTM\_1** канал класса **Вызов** и двойным щелчком ЛК откроем его основной бланк редактирования свойств. Изменим его атрибуты – **Имя**, **Тип** и **Тип вызова**:

Заданный **Тип вызова LocalStatistic** предназначен для проведения статистиче-

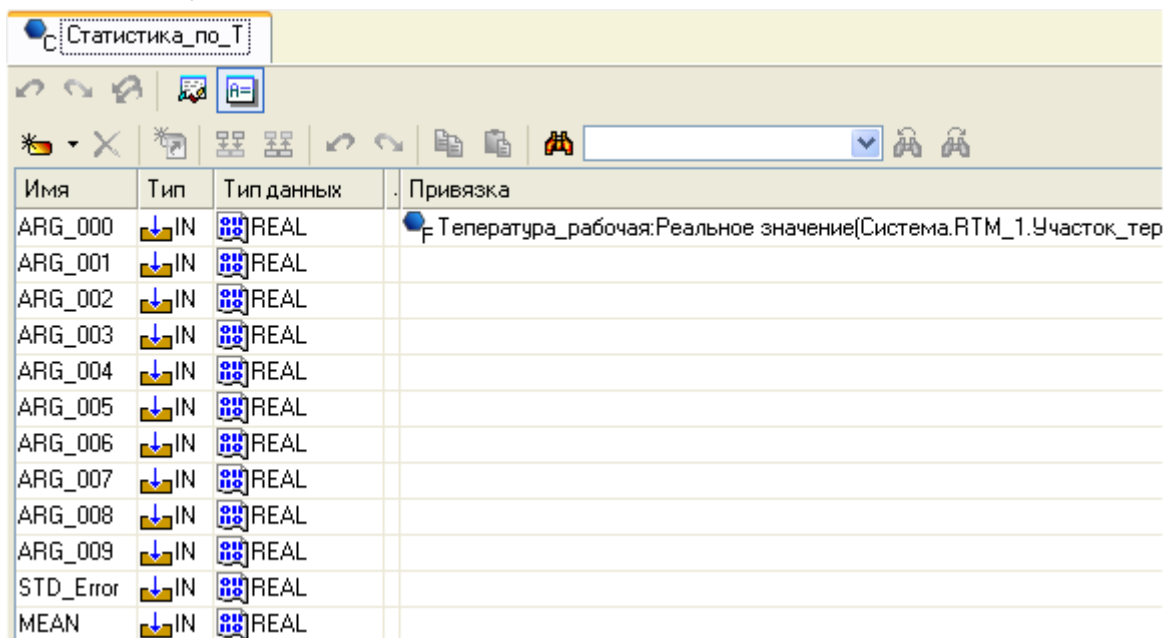


Имя		Кодировка
Статистика_по_T		TC1
Комментарий		
<b>Параметры</b>		
Тип вызова	LocalStatistic	
Параметр	0	
Глубина выборки	текущий час	
<b>Системные</b>		
Основные		
Тип	Output	
Размерность	...	



ской обработки данных локального архива по каналу. Так как процесс термообработки осуществляется в течение нескольких часов, атрибут **Глубина выборки** оставляем по умолчанию для оценки изменения температуры в аппарате за текущий час. Тип канала **Статистика\_по\_T** зададим **Output**. Перейдем с помо-

щью иконки  в табличный редактор аргументов канала. Создадим 12 аргументов, переименуем два последних, а к **ARG\_000** выполним привязку к каналу **Температура\_рабочая**.

В аргумент **STD\_Error** будет записываться среднеквадратическое отклонение, а в **MEAN** – среднее арифметическое значение по указанному в аргументе **ARG\_000** каналу.



Имя	Тип	Тип данных	Привязка
ARG_000	IN	REAL	Температура_рабочая:Реальное значение(Система.RTM_1.Участок_тер
ARG_001	IN	REAL	
ARG_002	IN	REAL	
ARG_003	IN	REAL	
ARG_004	IN	REAL	
ARG_005	IN	REAL	
ARG_006	IN	REAL	
ARG_007	IN	REAL	
ARG_008	IN	REAL	
ARG_009	IN	REAL	
STD_Error	IN	REAL	
MEAN	IN	REAL	

Для вывода среднего значения и среднеквадратического отклонения на графическом экране откроем в слое **Шаблонов экрана** шаблон **Участок\_термообработки** и разместим на нем три ГЭ Текст  и один ГЭ Кнопка  – для инициации обработки канала **Статистика\_по\_T**.



Для ГЭ **Текст** откажемся от рамок и от заливок. Создадим три новых аргумента шаблона экрана и привяжем их к ГЭ:

Шаблоны экранов.Участок\_термообработки

Информация Аргументы

Имя Тип Тип данных Привязка Флаги

Расход_теплоносителя	IN	REAL		
Тепература_рабочая	IN	REAL		
Задание_температуры	OUT	REAL		
Date_Time	IN	DATE_AND_TIME		NP
STD_Err	IN	REAL		
Mean	IN	REAL		
STATISTICS	IN/OUT	REAL		

Для ГЭ **Кнопка** определим прямую посылку значения **1** по нажатию на нем **ЛК** мыши.

OK Кнопка Справка

Свойство Значение

События

**pressed**

Подтверждение	False
Сигнал	False
<b>Передать значение</b>	
Тип передачи	Прямая
Значение	1
Результат	STATISTICS
Источник	...
Восстанавливать значение	False

Затем откроем свойства канала класса **Вызов Участок\_термо-обработки** узла **RTM\_1** и произведем автоматическую привязку вновь созданных аргументов.

Система.RTM\_1.Участок\_термообработки.Участок\_термообработки:1

Информация Флаги Аргументы Атрибуты

Имя Тип Тип данных Привязка

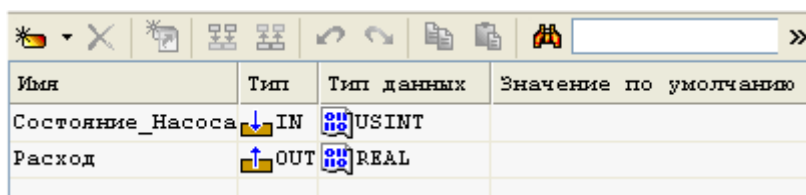
Расход_теплоносителя	IN	REAL	Расход_теплоносителя:Реальное значение(Система.F
Тепература_рабочая	IN	REAL	Тепература_рабочая:Реальное значение(Система.RTf
Задание_температуры	OUT	REAL	Задание_температуры:Входное значение(Система.RT
Date_Time	IN	DATE_AND_TIME	Битовый_меандр#1:Время изменения(Система.RTM_
STD_Err	IN	REAL	Статистика_по_T:STD_Error(Система.RTM_1.Участок_
Mean	IN	REAL	Статистика_по_T:MEAN(Система.RTM_1.Участок_тер
STATISTICS	IN/OUT	REAL	Статистика_по_T:Входное значение(Система.RTM_1.!





Аргументы шаблона экрана **STD\_Err** и **Mean** будут привязаны к соответствующим аргументам канала класса **Вызов Статистика\_по\_T**, а аргумент шаблона экрана **STATISTICS** к атрибуту **Входное значение** этого же канала. Таким образом, при работе в реальном времени при нажатии ЛК мыши на ГЭ **Кнопка** будет производиться статистическая обработка накапливаемых в локальном архиве за текущий час реальных значений канала **Температура\_рабочая**, и на графическом экране выводиться среднее значение и его среднеквадратическое отклонение.

Сохраним созданный проект, используя иконку .

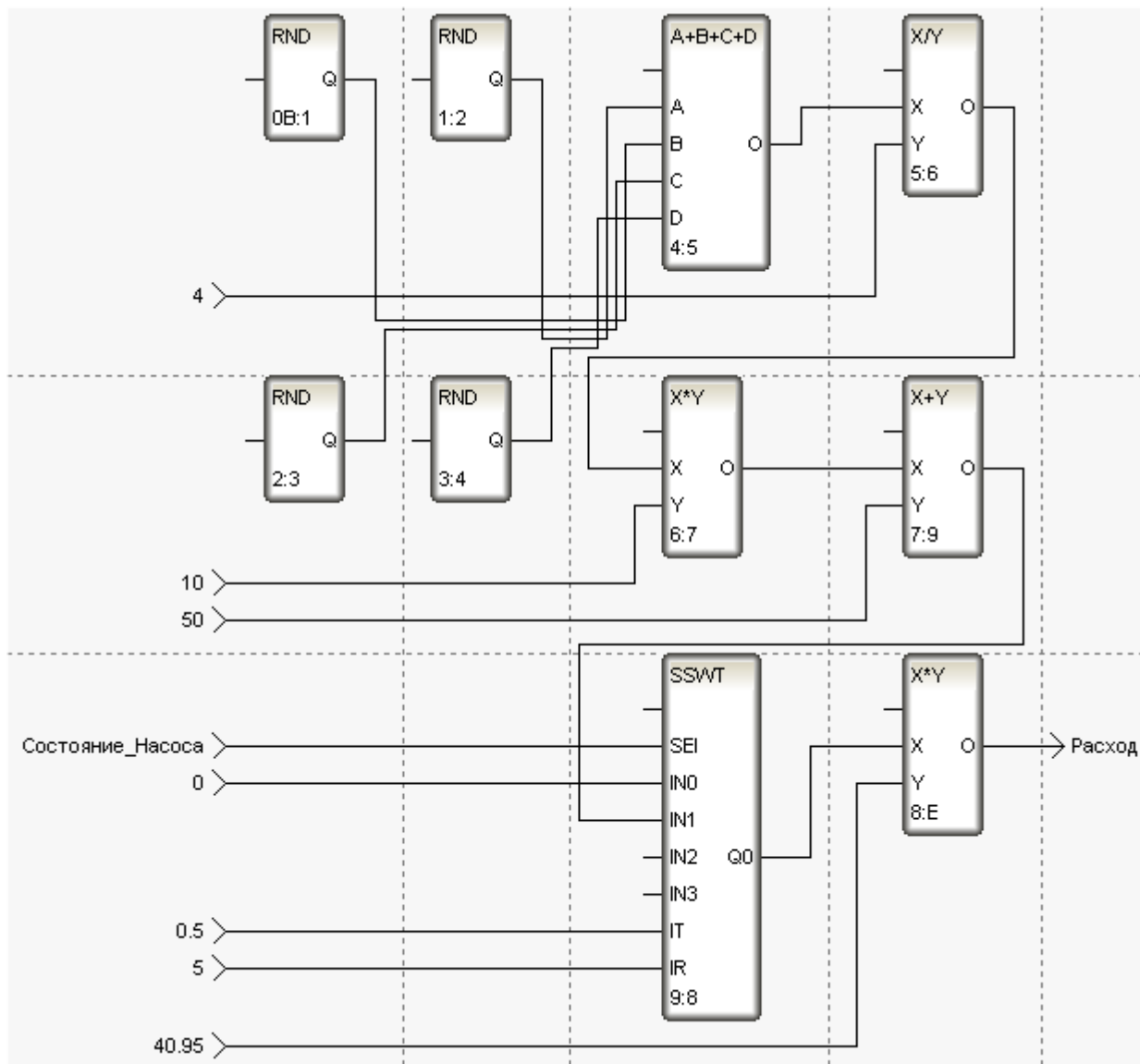
Для проверки работоспособности созданного проекта прибегнем к имитации сигналов, передаваемых от участка хранения, и расходомером готового продукта участка дозирования.

Для имитации расхода продукта создадим шаблон программы **Имитатор\_Расход\_Продукта** и также воспользуемся языком Техно FBD. Аргументы программы определим как показано на рисунке.



Имя	Тип	Тип данных	Значение по умолчанию
Состояние_Насоса	 IN	 USINT	
Расход	 OUT	 REAL	

Случайную составляющую расхода будем формировать, используя центральную предельную теорему из комбинации четырех генераторов случайных чисел в диапазоне (0,1) с равномерным распределением. Номинальное значение расхода зададим равным 50, а для плавного изменения в моменты включения/выключения дозирующего насоса применим функциональный блок переключения с динамической балансировкой – SSWT. Не забудем с помощью блока X\*Y привести выходное значение программы к диапазону 12-ти разрядного АЦП.



Откомпилируем программу. В дополнительном окне навигатора проекта выделим группу **Участок\_Дозирования** и перетащим в него шаблон программы. Для созданного таким образом канала класса **Вызов Имитатор\_Расход\_Продукта** выполним привязку атрибутов каналов к аргументам.

Система.RTM_1.Участок_Дозирования.Имитатор_Расход_Продукта.15				
Информация    Флаги    Аргументы    Атрибуты				
Имя	Тип	Тип данных	Значение	Привязка
Состояние_Насоса	IN	USINT		Насос:Реальное значение(Система.RTM_1.Участок_Дозирования)
Расход	OUT	REAL		Расход_продукта:Входное значение(Система.RTM_1.Участок_Дозирования)

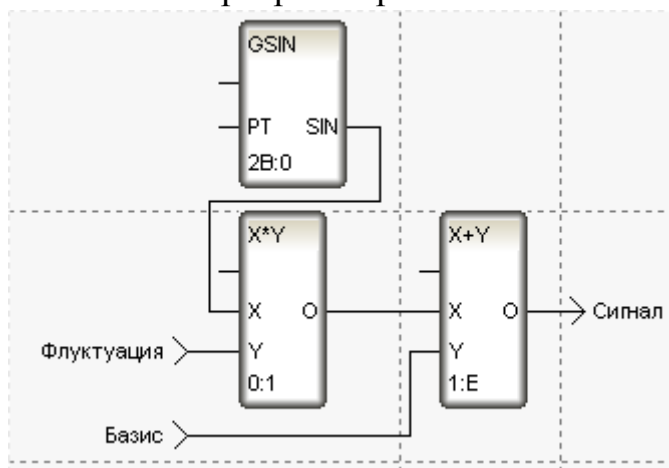


Так как от датчиков поступают данные как аналоговые, так и дискретные, то для их имитации на АРМ создадим две соответствующие программы. Причем для каждого типа параметров создадим по одной программе, а варьируя значения аргументов, задаваемых как константы – в поле **Значение по умолчанию** табличного редактора аргументов, будем устанавливать величины параметров и их "случайные" флуктуации.

Создадим в слое **Шаблоны программ** новую программу – **Имитатор\_UX\_Аналог**. Объявим для нее аргументы:

Имя	Тип	Тип данных	Значение по
Сигнал	OUT	REAL	
Флуктуация	IN	REAL	
Базис	IN	REAL	

Программу разработаем на языке программирования Техно FBD.



Изменяющийся по синусоидальному закону "случайный" компонент в диапазоне  $(-1,1)$  будем масштабировать, задавая параметр **Флуктуация**, и добавлять к параметру **Базис**, определяющему установившееся значение. Откомпилируем программу нажатием функциональной клавиши F7.

Для привязки разработанной программы выделим в окне навигатора проекта группу **Участок\_Хранения** узла **RTM\_1**. Выделим канал **Уровень** и откроем его свойства. Во вкладке **Информация** определим свойство канала **Вызов** путем указания в диалоге шаблона программы **Имитатор\_UX\_Аналог**.


Система.RTM\_1.Участок\_Хранения.Уровень

Информация    Флаги    Аргументы    Атрибуты

Имя:  Узел:  Тип:

Привязка:

Вызов:

Иконка: 

Перейдя в появившуюся после этого действия вкладку **Аргументы**, привяжем аргументы программы к атрибуту канала и зададим константы.

Система.RTM\_1.Участок\_Хранения.Уровень

Информация    Флаги    Аргументы    Атрибуты

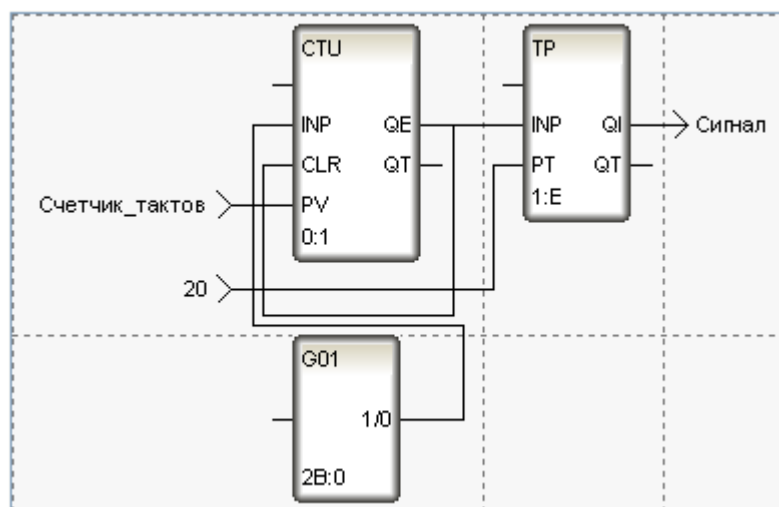
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Сигнал	↑ OUT	REAL		Ф Уровень:Реальное значение(Система.RTM_1.
Флуктуация	↓ IN	REAL	0.25	
Базис	↓ IN	REAL	3.5	

Подобным образом поступим в отношении каналов **Температура**, **Давление** и **Влажность**, указав для них соответственно параметры **Флуктуация** и **Базис** как (0.25,24), (0.025,0.98) и (0.2,70).

Для имитации дискретных сигналов создадим шаблон программы **Имитатор\_УХ\_Дискрет** со следующими аргументами:

Имя	Тип	Тип данных	Значение по
Сигнал	↑ OUT	USINT	
Счетчик_тактов	↓ IN	REAL	

Разработанная на Техно FBD программа будет выглядеть так:



Данная программа будет периодически имитировать срабатывание дискретного датчика. Задавая входным аргументом **Счетчик\_тактов** количество тактов, поступающих от генератора G01, до момента срабатывания датчика (установки триггера TP), при его достижении выходной аргумент Сигнал будет удерживаться в состоянии "1" в течение последующих 20 тактов, затем произойдет сброс его в состояние "0", а далее картина будет повторяться. После компиляции программы выделим канал Дверь в группе **Участок\_Хранения** узла **RTM\_1** и в его свойствах определим свойство **Вызов** следующим образом:

Система.RTM\_1.Участок\_Хранения.Дверь

Информация	Флаги	Аргументы	Атрибуты
Имя:	Дверь		Узел: Система
Кодировка:	TC2		Тип: Канал
Комментарий:			
Привязка:			
Вызов:	Имитатор_УХ_Дискрет(Шаблоны_программ)		
Иконка:			

Во вкладке **Аргументы** свяжем выходной аргумент программы с атрибутом **Реальное значение** текущего канала и зададим период "срабатывания" датчика:

Система.RTM\_1.Участок\_Хранения.Дверь

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Сигнал	OUT	USINT		16 Дверь:Реальное значение(Система.RTM_1
Счетчик_тактов	IN	REAL	250	

Для канала **Пожарная\_сигнализация** поступим аналогичным образом, определив для него периодичность срабатывания как 1000. В отношении канала **Вентиляция** поступим несколько иначе – зададим ему постоянное значение, соответствующее включенному состоянию системы вентиляции (**Вентиляция** – **Редактировать**, и в бланке **Основные** установить флаг в окне **Отработать**, и 1 – в окне **На старте**).



Таким образом, разработанный в предыдущих главах проект системы автоматизации, доступен для запуска и отладки, причем все внешние сигналы заменены имитаторами.

С целью пресечения несанкционированного доступа к разрабатываемой системе автоматизации необходимо определить круг пользователей системы и назначить им права доступа, назначить идентификаторы для входа в систему и пароли. С этой целью в узле АРМ создадим новую группу **Пользователи ТМ**.

Войдем в нее и создадим два новых компонента – **Пользователя**. Откроем их последовательно для редактирования, изменяя названия и назначая права, идентификаторы и пароли. Цель, преследуемая при этом, заключается в наделении пользователя **admin** (разработчика) максимальными правами по доступу к системе с возможностью внесения изменений, а пользователя **oper** (оператора) – лишь необходимыми для текущей работы с системой.

Для пользователя **admin** в бланке установим все флаги в позициях страниц Система, Пользователи, SCADA, T-Factory, а также введем логин – **admin** без указания пароля.

Для пользователя **oper** установим флаги в позициях: **Запуск, Останов, Изменение, Квитирование, Запрет одновременного входа и Авто-logout**. Укажем логин – **oper** и пароль.

Сохраним созданный проект на диск, используя иконку , а затем, с помощью щелчка **ЛК** мыши на иконке , создадим файлы для загрузки на целевые платформы для проведения отладки выполненного проекта.

### Содержание отчета

1. Цель исследований.
2. Текст программы **Розлив#2** с пояснениями.
3. Текст программы **Имитатор\_Расход\_Продукта** с пояснениями.
4. Выводы по работе.

### Контрольные вопросы и задания

1. С какой целью вводят функцию архивирования в SCADA системы?

2. С какой целью вводят журнал тревог в SCADA системы?
3. Какие показатели оценивают качественно и количественно потребляемую электрическую мощность?

## **Литература**

1. <http://www.adastra.ru>
2. Краткий каталог продукции ПРОСОФТ 2011/2019 (10.0)
3. Петрова А.М. Автоматическое управление. – М.: Форум, 2019.–240 с.
4. Гайдук А.Р. Теория автоматического управления в примерах и задачах с решениями в MATLAB. – М., Лань, 2011. – 464 с.

## Содержание

<b>ВВЕДЕНИЕ</b> .....	3
<b>ЛАБОРАТОРНЫЕ РАБОТЫ</b> .....	4
<b>Лабораторная работа №1</b>	
СОЗДАНИЕ ЭЛЕМЕНТАРНОГО АРМ .....	4
<b>Лабораторная работа №2</b>	
ПРИМЕР ОБЪЕКТНОГО ПРОЕКТИРОВАНИЯ .....	24
<b>Лабораторная работа №3</b>	
ИССЛЕДОВАНИЕ ВЛИЯНИЯ ПРОПОРЦИО- НАЛЬНО-ДИФФЕРЕНЦИАЛЬНОГО-ДИФФЕРЕНЦИ- АЛЬНОГО РЕГУЛЯТОРА НА КАЧЕСТВО ПРОЦЕССА РЕГУЛИРОВАНИЯ.....	42
<b>Лабораторная работа №4</b>	
СОЗДАНИЕ АРХИВА И ОТЧЕТА ТРЕВОГ .....	62
<b>Список литературы</b> .....	94