

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

В.И. Устич

ПРОЕКТИРОВАНИЕ SCADA - СИСТЕМ

Учебно–методическое пособие по выполнению лабораторных работ по
дисциплине «Системы реального времени» для студентов, обучающихся в
бакалавриате по направлению подготовки
09.03.01 – Информатика и вычислительная техника

Калининград
Издательство ФГБОУ ВО «КГТУ»
2019

УДК 681.5

РЕЦЕНЗЕНТ

кандидат технических наук, доцент, заведующий кафедрой систем управления и вычислительной техники ФГБОУ ВО «Калининградский государственный технический университет» В.А. Петрикин

Устич, В.И.

Проектирование SCADA - систем: учебно–методическое пособие по выполнению лабораторных работ по дисциплине «Системы реального времени»/ В.И. Устич. – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2018.-115 с.

В настоящем пособии приведены задания по лабораторным работам, методические указания по их выполнению, а также теоретические сведения по теме лабораторных работ.

Ил. 206, список лит. – 3 наименования

Учебно–методическое пособие рассмотрено и одобрено кафедрой систем управления и вычислительной техники ФГБОУ ВО "Калининградский государственный технический университет» 10 апреля 2019 г., протокол № 04

Учебно–методическое пособие рекомендовано к печати методической комиссией факультета автоматизации производства и управления ФГБОУ ВО "Калининградский государственный технический университет» 17 апреля 2019 г., протокол № 03

УДК 681.5

© Федеральное государственное
бюджетное образовательное
учреждение высшего образования
"Калининградский государственный
технический университет", 2019 г.
© Устич В.И., 2019 г.

ОБЩИЕ ОРГАНИЗАЦИОННО – МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Целью освоения дисциплины «Системы реального времени» является формирование знаний и навыков по анализу, синтезу и применению систем реального времени (СРВ).

Задачи изучения дисциплины:

- формирование базовых понятий об области использования, преимуществах и принципах построения СРВ;
- приобретение теоретических знаний и практических навыков по анализу и синтезу СРВ;
- приобретение практических навыков эксплуатации СРВ, реализованных на базе микроконтроллеров (программируемых логических контроллеров).

Дисциплина «Системы реального времени» входит в состав профессионального модуля по выбору студента «Разработка автоматизированных систем обработки информации и управления (АСОИУ)» (МВ1) вариативной части образовательной программы бакалавриата.

Дисциплина опирается на компетенции, полученные при изучении таких дисциплин, как «Теоретические основы автоматизированного управления», «Архитектура АСОИУ».

Результаты освоения дисциплины могут быть использованы при выполнении выпускной квалификационной работы, а также в дальнейшей профессиональной деятельности.

В результате изучения дисциплины студент должен

знать:

- принципы программного и аппаратного построения систем реального времени на основе микропроцессорной техники (программируемых логических контроллеров) и особенности их применения;

уметь:

- выбирать программные и аппаратные средства при проектировании систем реального времени, программировать и отлаживать системы на базе программируемых логических контроллеров;

владеть:

- навыками программирования, наладки, настройки и обслуживания систем реального времени на базе программируемых логических контроллеров.

Современные системы реального времени в аппаратной части в значительной мере базируются на микропроцессорных вычислительных устройствах и, в частности, регулирующих контроллерах, в программной – на операционных системах реального времени (ОСРВ). Микропроцессорные контроллеры с ОСРВ представляют собой программно-технические средства автоматизации коллективного пользования, что позволяет применять их как для создания локальных систем управления технологическим объектом с небольшим количеством регулируемых координат, так и для построения

распределенных АСУ ТП большой информационной мощности. Построение уровневых СРВ, в частности, уровня SCADA, реализуется, как правило, с использованием программных средств, таких как, например, пакет TRACE MODE.

Данная часть лабораторного практикума по дисциплине выполняется в компьютерном классе кафедры автоматизации производственных процессов с использованием пакета TRACE MODE. В представленном лабораторном практикуме использованы материалы, входящие в комплект описания работы пакета TRACE MODE компании AdAstra Research Group, Ltd (руководство пользователя – быстрый старт, по материалам, размещенным на сайте <http://www.adastra.ru>).

ПЕРЕЧЕНЬ РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Пономарев, О.П. Интегрированные системы проектирования и управления. Ч.1. Общие сведения о SCADA - системах, программируемых логических контроллерах: учеб. пособие / О. П. Пономарев. - Калининград: КГТУ, 2008. - 83 с.

2. Герасимов, А.В. SCADA система Trace Mode 6 [Электронный ресурс]: учеб. пособие / А.В. Герасимов, А.С. Титовцев; Министерство образования и науки Российской Федерации, Государственное образовательное учреждение высшего профессионального образования «Казанский государственный технологический университет». - Казань: КГТУ, 2011. - 128 с. (ЭБС «Университетская библиотека онлайн»).

3. Руководство пользователя TRACE MODE 6 & T-FACTORY SOFTLOGIC SCADA/HMI MES EAM HRM. Интегрированная платформа для управления производством. БЫСТРЫЙ СТАРТ (<http://www.adastra.ru/products/rukovod/>).

ЛАБОРАТОРНАЯ РАБОТА №1

Создание простейшего проекта в пакете TRACE MODE

1. Цель работы – ознакомление с процессом создания операторского (человеко-машинного) интерфейса системы мониторинга, содержащего один узел АРМ (автоматизированное рабочее место), с использованием механизма автопостроения каналов TRACE MODE методом "от шаблонов".

2. Задание по лабораторной работе: создать операторский интерфейс, произвести действия по созданию узла АРМ: графического экрана; стрелочного прибора, привязке к аргументу; автопостроению канала; заданию границ и уставок.

3. Методические указания по выполнению лабораторной работы.

3.1. Ознакомиться с созданием узла АРМ (п. 4.1).

3.2. Создать графический экран по описанию, приведенному в п. 4.2.

3.3. Привязать к аргументу стрелочный прибор (п. 4.3).

3.4. Произвести автопостроение канала (п. 4.4).

3.5. Задать границы и уставки (п.4.5).



3.6. Создать генератор синуса и привязать его к каналу (п. 4.6).

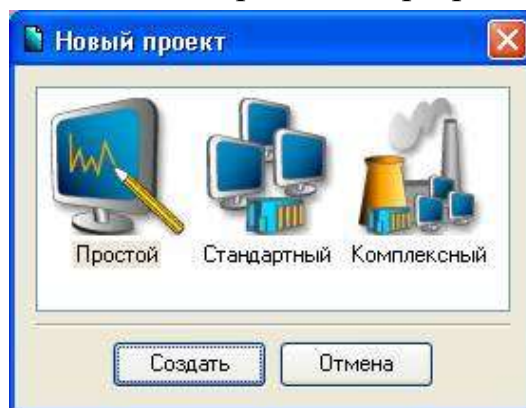
3.7. Запустить проект, отладить его работу (при необходимости) (п. 4.7).

4. Процедура создания проекта.

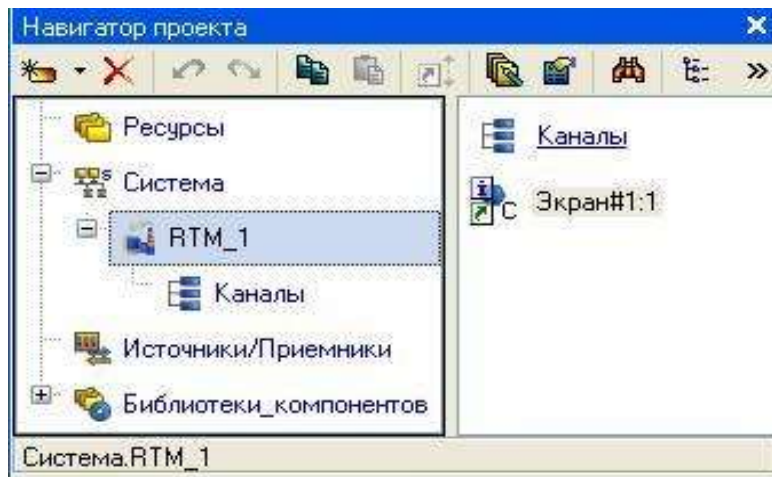
4.1. Создание узла АРМ.

4.1.1. Загрузить инструментальную систему двойным щелчком левой

клавиши (ЛК) мыши по иконке  рабочего стола Windows и с помощью иконки  инструментальной панели создать новый проект. При этом в открывшемся на экране диалоге выберем стиль разработки **Простой**.



4.1.2. После нажатия ЛК мыши на экранной кнопке **Создать** в левом окне Навигатора проекта появится дерево проекта с созданным узлом АРМ **RTM_1**. В правом окне Навигатора проекта отобразится содержимое узла – пустая группа **Каналы** и один канал класса **Вызов Экран#1**, предназначенный для отображения на узле АРМ графического экрана.



4.2. Создание графического экрана.

Двойным щелчком ЛК на компоненте **Экран#1** открыть окно графического редактора.

4.2.1. Создание статического текста.

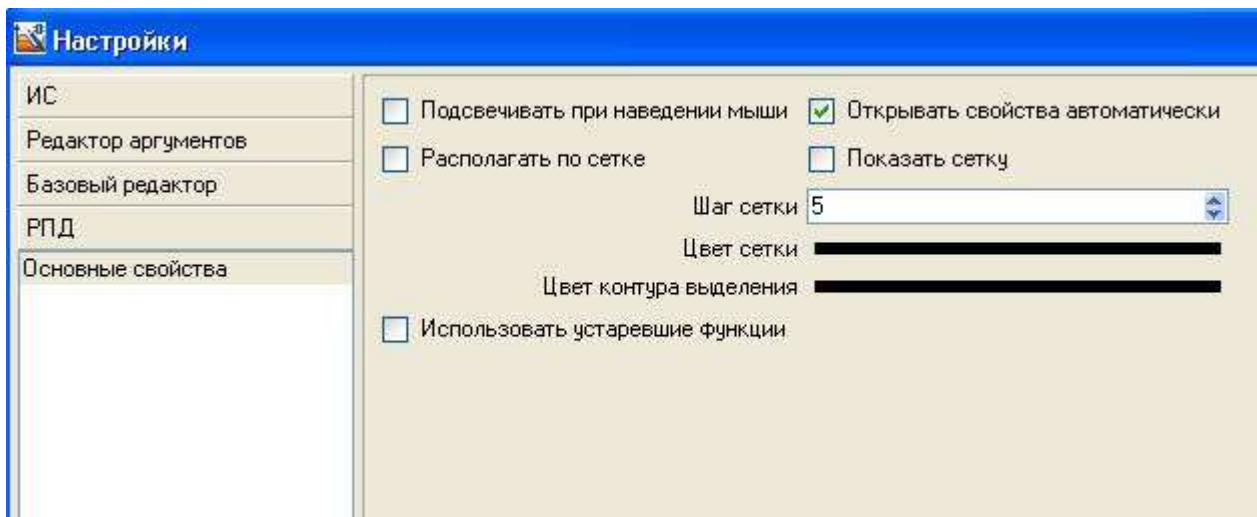
Разместить в левом верхнем углу экрана статический текст - надпись **Значение параметра.**

Для этого необходимо выполнить следующие действия: на панели инструментов графического редактора ЛК мыши выделить иконку графического элемента (ГЭ) Текст **ABC** и в поле графического редактора установить прямоугольник ГЭ, для чего:

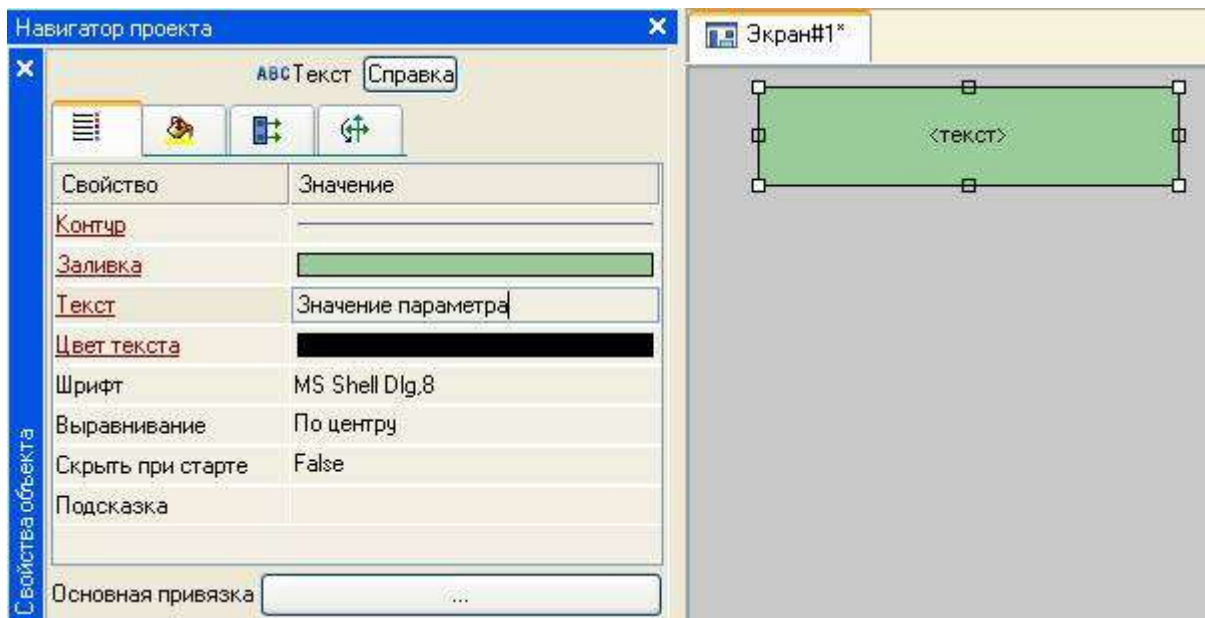
- зафиксировать ЛК точку привязки - левый верхний угол;
- развернуть прямоугольник движением курсора до необходимого размера;
- зафиксировать ЛК выбранный ГЭ.



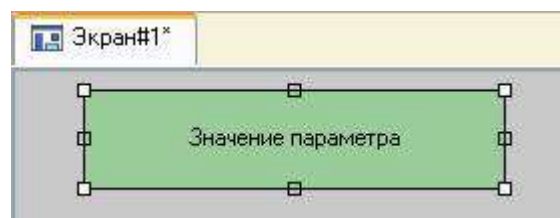
Для перехода в режим редактирования атрибутов размещенного ГЭ выделить ЛК иконку **☞** на панели инструментов, двойным щелчком ЛК по ГЭ открыть окно его свойств.



В правом поле строки **Текст** набрать **Значение параметра**.



Закрывать окно свойств щелчком ЛК по иконке . В результате ГЭ примет следующий вид.



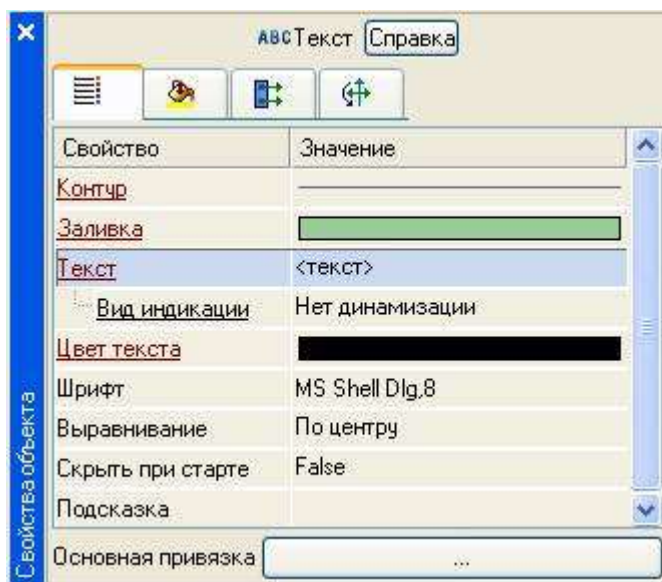
Если введенный текст не уместился в прямоугольнике ГЭ, то необходимо выделить его и растянуть до нужного размера с помощью мыши.

4.2.2. Создание динамического текста, создание аргумента экрана в процессе настройки динамического текста.

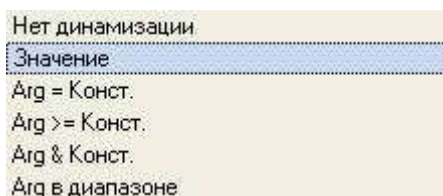
Необходимо подготовить на экране вывод динамического текста для отображения численного значения какого-либо источника сигнала – внешнего

или внутреннего - путем указания динамизации атрибута **Текст** ГЭ и определить назначение аргумента шаблона экрана. Для этого необходимо проделать следующие действия:

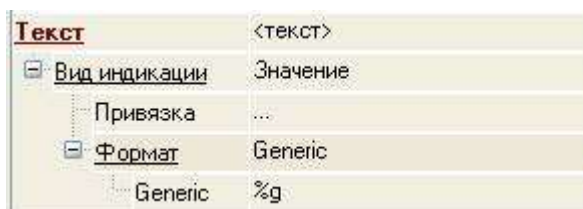
- создать и разместить новый ГЭ **ABC** справа от ГЭ с надписью **Значение параметра**;
- открыть свойства вновь размещенного ГЭ;
- двойным щелчком ЛК на строке **Текст** вызвать меню **Вид индикации**.





В правом поле строки щелчком ЛК вызвать список доступных типов динамизации атрибута и из всех предлагаемых типов выбрать ЛК **Значение**.

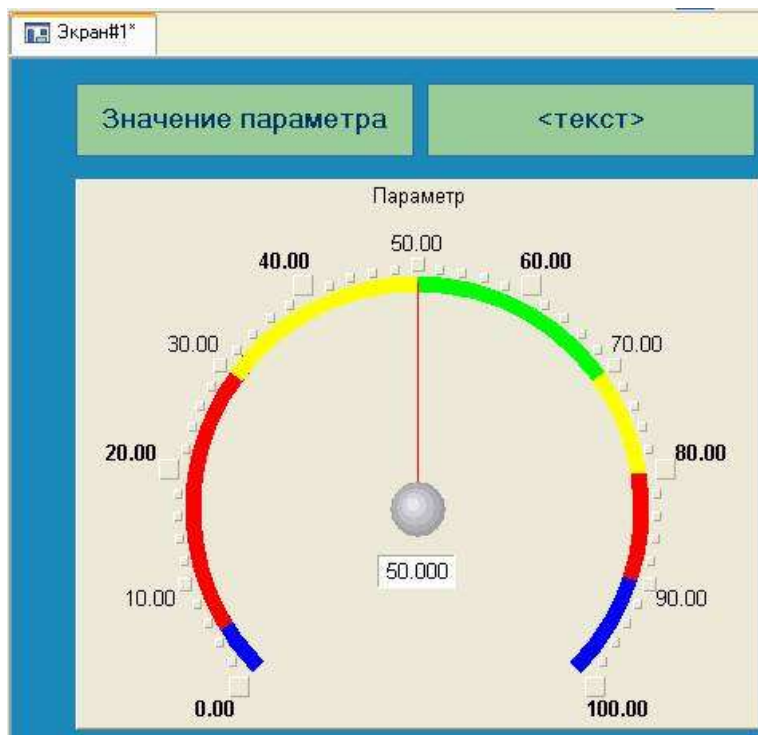



В открывшемся меню настройки параметров динамизации:



- выполнить щелчок ЛК в правом поле строки **Привязка**;
- в открывшемся окне **Свойства привязки**, нажать ЛК по иконке  на панели инструментов и тем самым создать аргумент шаблона экрана;

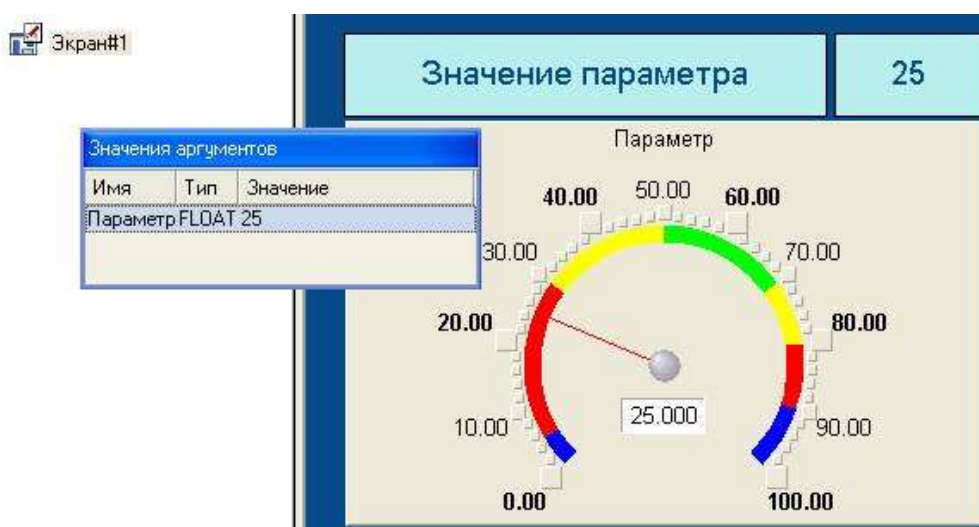
- закрыть окно свойств ГЭ .



Для проверки правильности привязок ГЭ к аргументам экрана можно воспользоваться режимом эмуляции. Переход в режим эмуляции осуществляется с помощью иконки  на панели инструментов. При ее нажатии на экран графического редактора выводится окно задания значения аргумента в соответствующем поле:

Значения аргументов		
Имя	Тип	Значение
Параметр	FLOAT	0

Например, введя значение 25, наблюдаем:

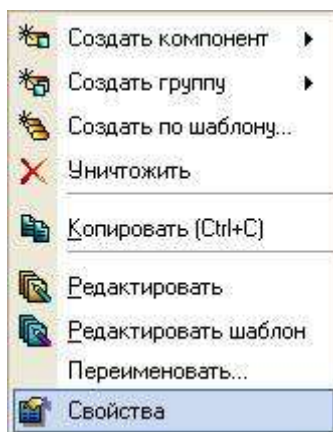


Выход из режима эмуляции – повторное нажатие ЛК по иконке .

4.4. Автопостроение канала.

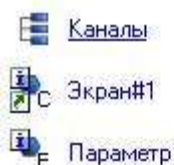
Для создания канала в узле проекта по аргументу шаблона экрана можно воспользоваться процедурой автопостроения. Для этого:

- в слое **Система** выберем ЛК узел **RTM_1**;
- в поле компонентов узла ЛК выберем **Экран#1**;
- щелчком правой кнопки мыши (ПК) вызовем контекстное меню;
- в нем щелчком ЛК откроем свойства компонента **Экран#1**:



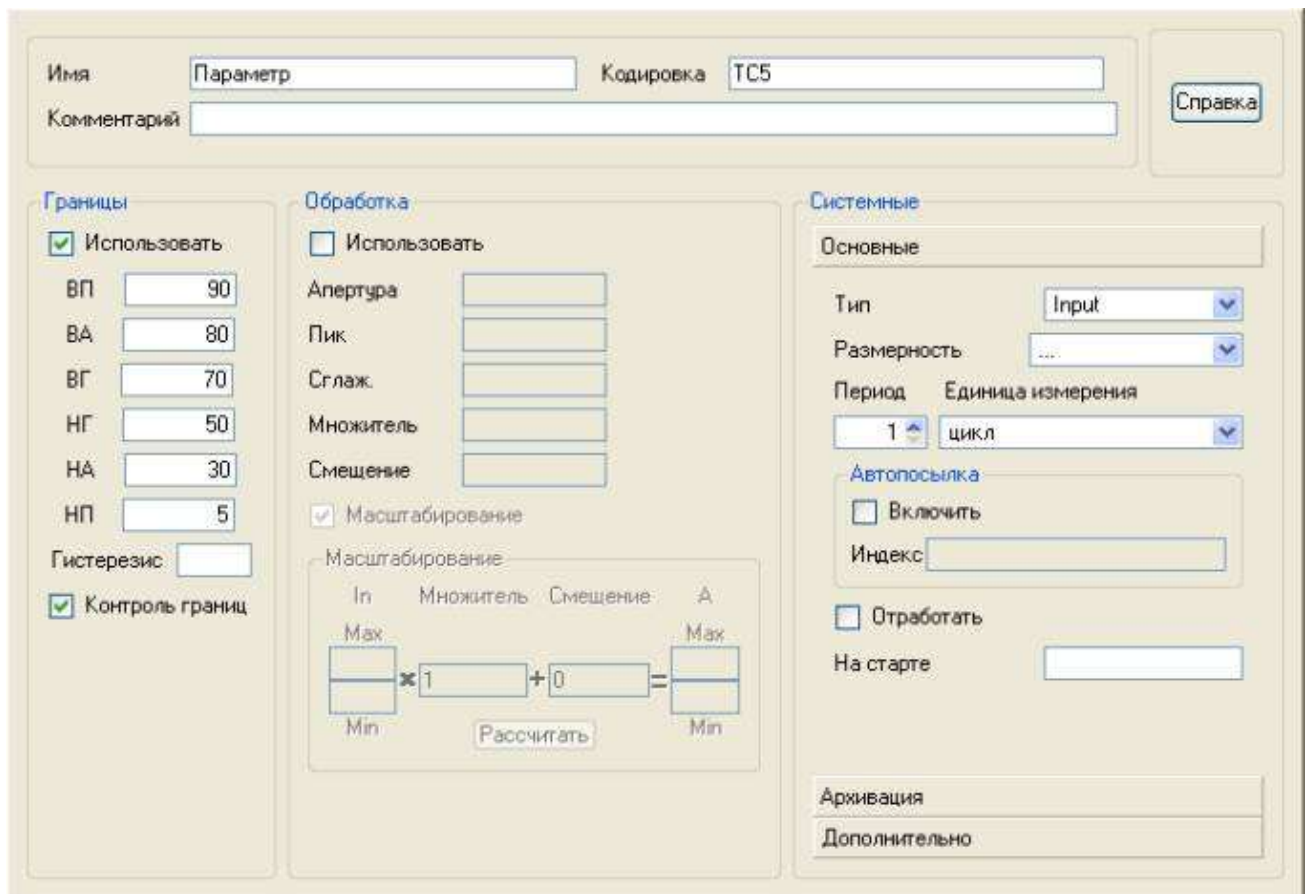
- выберем ЛК вкладку **Аргументы**;

- выделим ЛК аргумент **Параметр** и с помощью иконки  создадим канал класса **Float** типа **Input** с именем **Параметр**:



4.5. Задание границ и уставок.

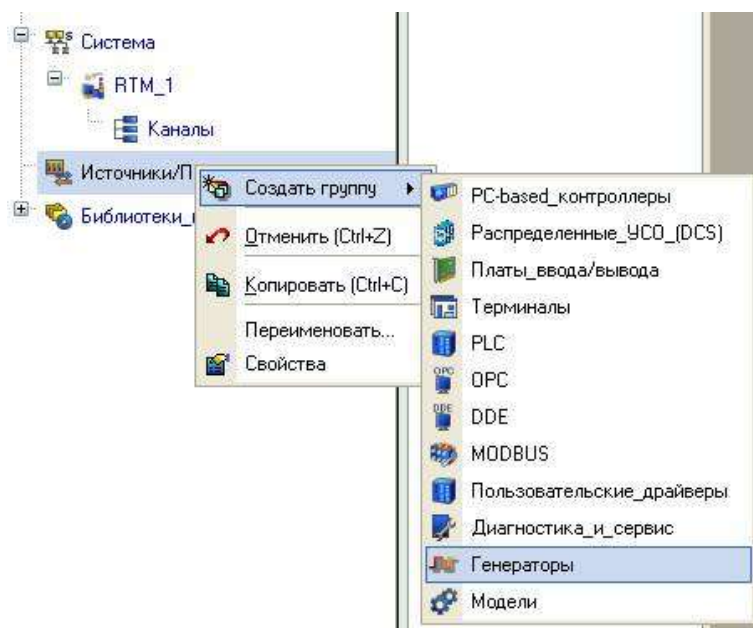
Двойным щелчком ЛК по каналу **Параметр** открыть бланк редактирования его атрибутов и заполнить раздел **Границы** следующим образом:



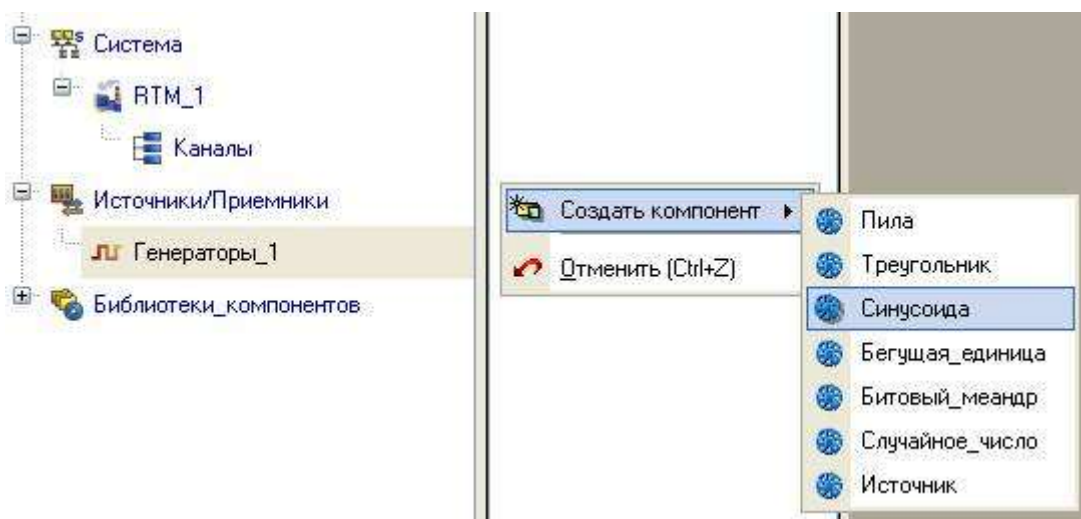
4.6. Создание генератора синуса и привязка его к каналу.

Для проверки в работе выполненных средств изображения необходимо ввести в состав проекта источник сигнала – внутренний генератор синусоиды и связать его с созданным каналом. Для этого продедем следующие действия:

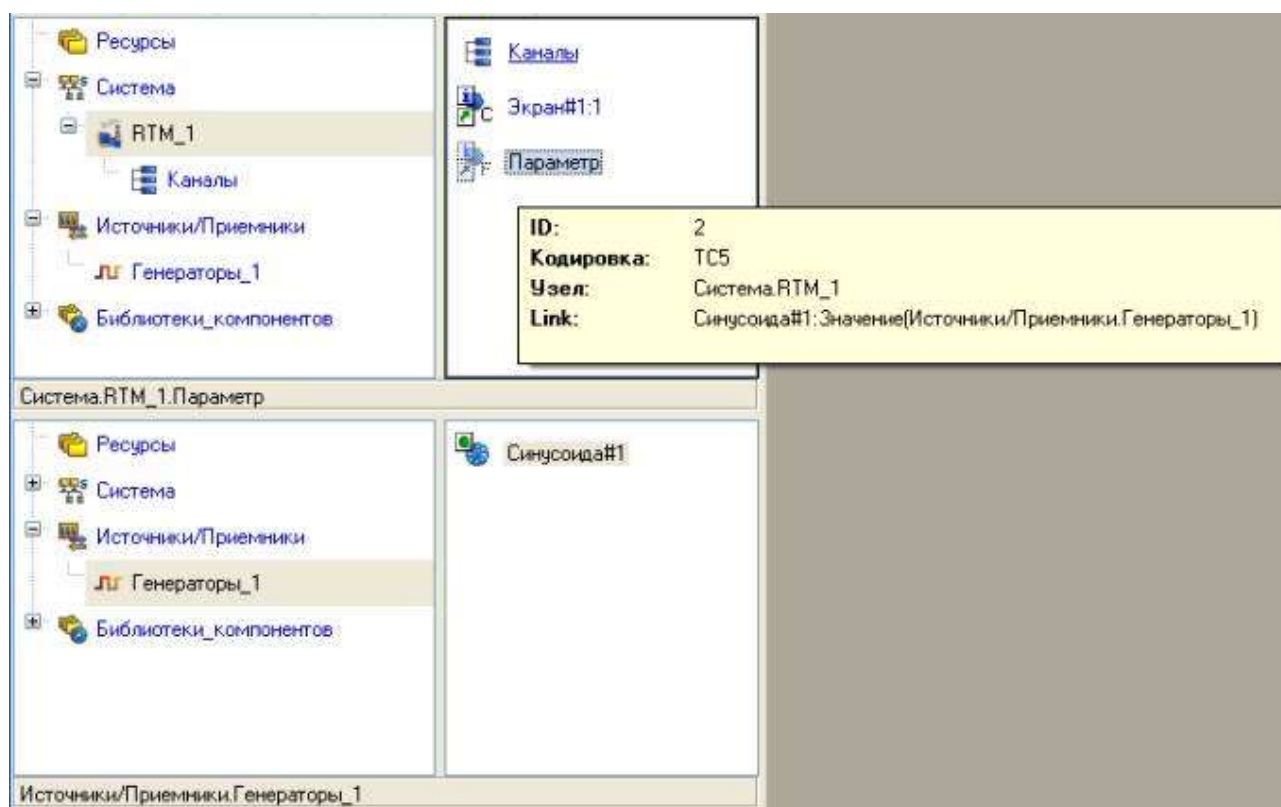
- открыть слой **Источники/Приемники** и через ПК создать в нем группу компонентов **Генераторы**:



- двойным щелчком ЛК открыть группу **Генераторы_1** и через ПК создать в ней компонент **Синусоида**;





- захватить с помощью ЛК созданный источник и, не отпуская ЛК, перетащить курсор на узел **RTM_1** в слое **Система**, а затем, в открывшемся окне компонентов, на канал **Параметр**. Отпустить ЛК.



4.7. Запуск проекта.

Предварительно необходимо сохранить проект с помощью иконки , задав в открывшемся окне имя **QS_Lesson_1.prj**.

На инструментальной панели выбрать ЛК иконку  и подготовить проект для запуска в реальном времени;

- ЛК выделить в слое Система узел RTM_1, а после, нажав ЛК иконку  на инструментальной панели, запустить режим исполнения. В открывшемся окне ГЭ справа от надписи "Значение параметра" должно показываться изменение синусоидального сигнала. То же значение должен отображать и стрелочный прибор:



5. Контрольные вопросы.
- 5.1. Порядок создания узла АРМ.
- 5.2. Порядок создания статического текста.
- 5.3. Порядок создания динамического текста.
- 5.4. Перечислите функции окна Свойства привязки.
- 5.5. Перечислите действия по созданию стрелочного прибора.

ЛАБОРАТОРНАЯ РАБОТА №2

Добавление в рабочий проект функции управления

1. Цель работы – введение в состав графического экрана (лабораторная работа №1) ГЭ, позволяющего реализовать ввод числовых значений с клавиатуры, и создание нового аргумента шаблона экрана для их приема.

2. Задание по лабораторной работе: провести действия по простейшей обработке данных; созданию программы на языке ТехноST.

3. Методические указания по выполнению лабораторной работы.

3.1. Провести редактирование графического экрана (п. 4.1).

3.2. Привязать аргументы экрана к каналу (п.4.2).

3.3. Разместить ГЭ Тренд (п. 4.3).

3.4. Запустить проект на исполнение (п. 4.4).

3.5. Произвести программирование простейшей обработки данных (п.4.5).

3.6. Создать программу на языке Техно ST (п. 4.6).


3.7. Привязать аргументы программы (п. 4.7).

3.8. Запустить проект на исполнение (п. 4.8).

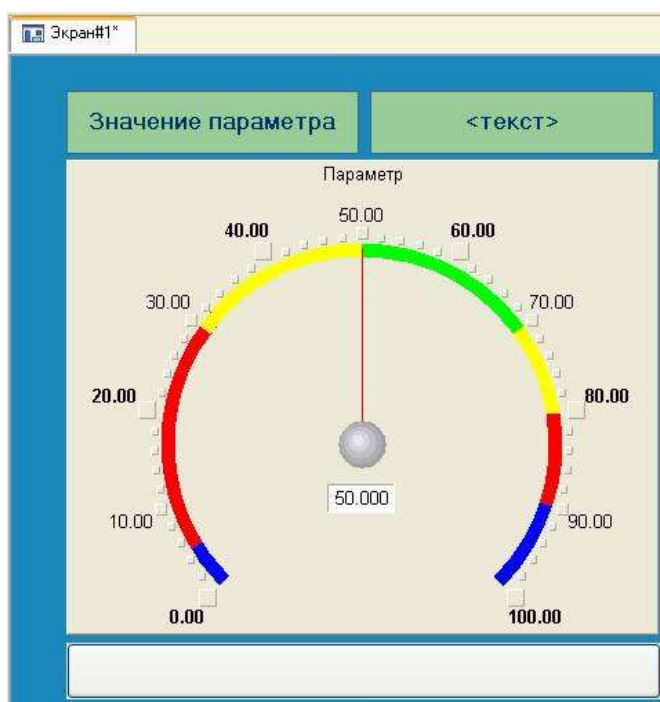
4. Процедура изменения проекта.



4.1. Редактирование графического экрана.

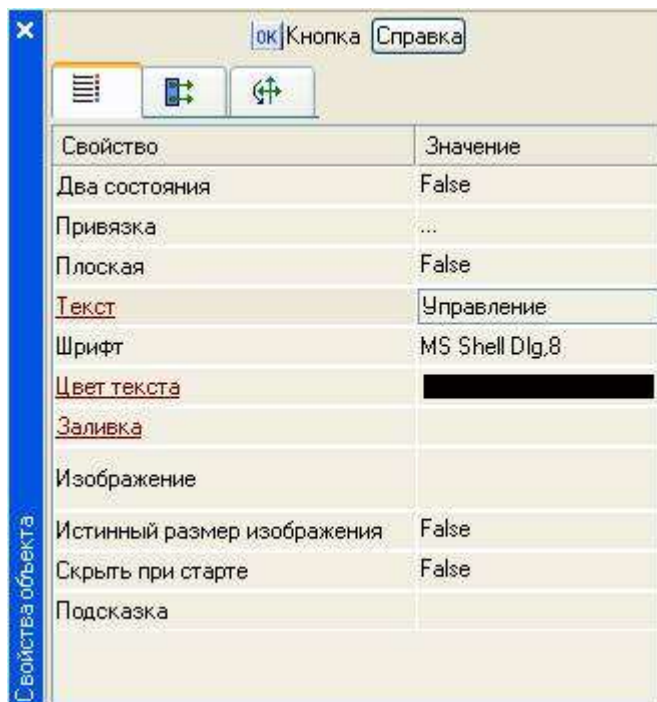
Для редактирования графического экрана необходимо:


- на инструментальной панели графического редактора выбрать ЛК иконку ГЭ Кнопка  ;

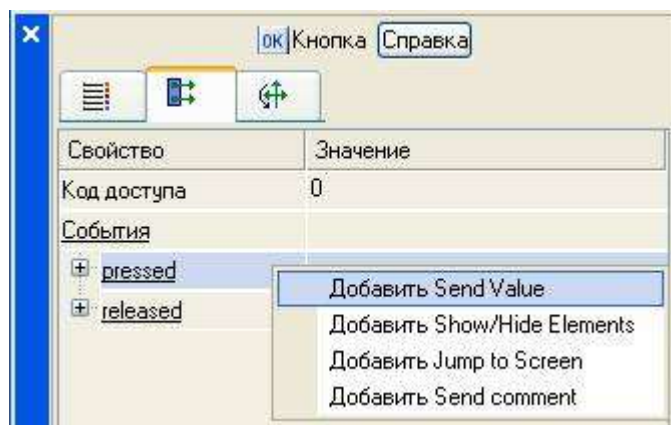
- с помощью мыши разместить ГЭ в поле экрана под ГЭ  ;



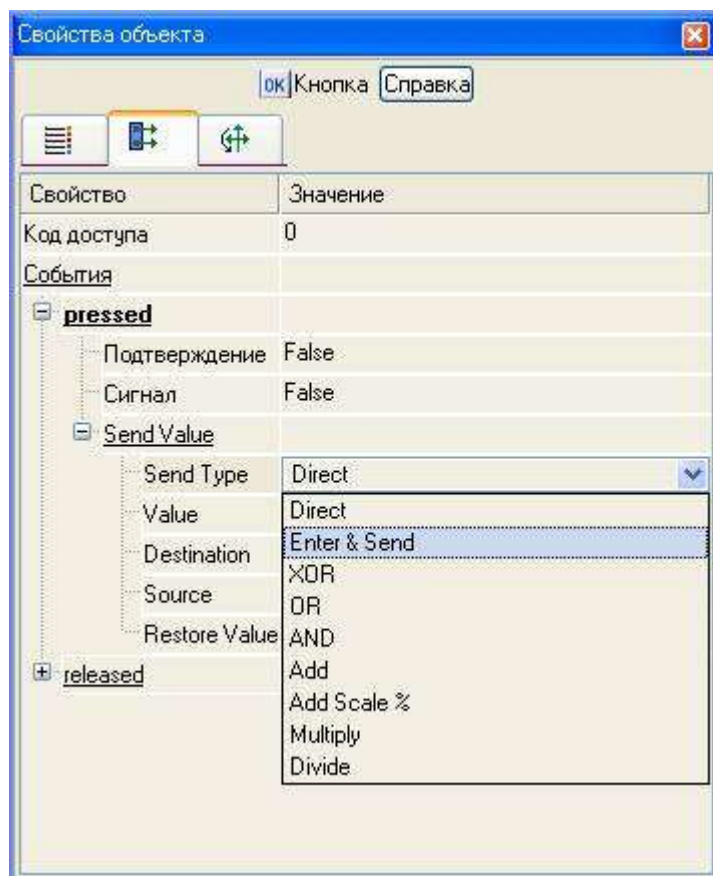
- перейти в режим редактирования , выделить ГЭ  ЛК и вызвать окно его свойств:



- в поле **Текст** ввести надпись **Управление**;
- открыть бланк **События**  и ПК раскрыть меню **По нажатию (pressed)**;
- выбрать из списка команду **Добавить Send Value**;

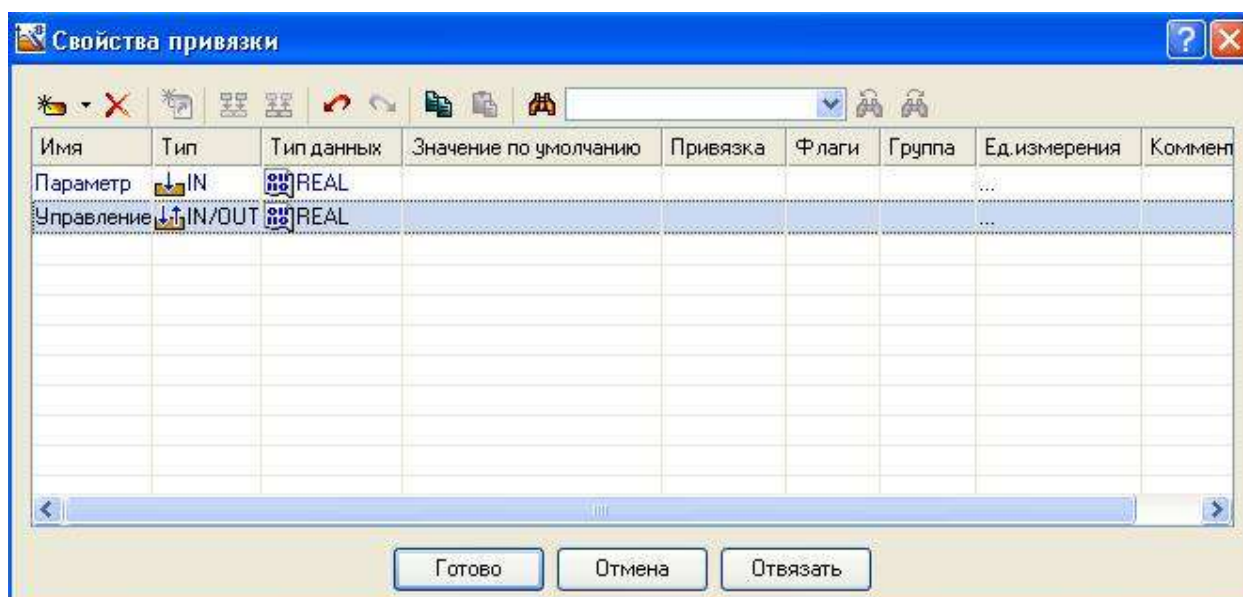


- в раскрывшемся меню настроек выбранной команды в поле **Тип передачи (Send Type)** выбрать из списка **Ввести и передать (Enter & Send)**.



Щелчком ЛК в поле **Результат (Destination)** вызвать табличный редактор аргументов:

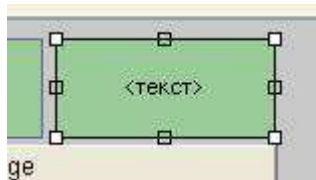
- создать еще один аргумент и задать ему имя **Управление**;
- изменить тип аргумента на **IN/OUT**, кнопкой **Готово** подтвердить привязку атрибута ГЭ к этому аргументу:





Закреть окно свойств ГЭ с помощью щелчка ЛК по иконке

Для отображения вводимого с клавиатуры значения необходимо выполнить размещение ГЭ **Текст**, воспользовавшись, например, уже имеющимся на графическом экране ГЭ путем его копирования/вставки и перепривязки. Для этого необходимо:

- выделить ЛК ГЭ **Текст**, служащий для отображения аргумента **Параметр**:



- с помощью иконки  на панели инструментов или комбинацией клавиш **Ctrl+C** скопировать выделенный ГЭ Текст в буфер обмена;

- далее с помощью иконки  или комбинацией клавиш **Ctrl+V** извлечь копию ГЭ из буфера обмена и поместить ее на графический экран;

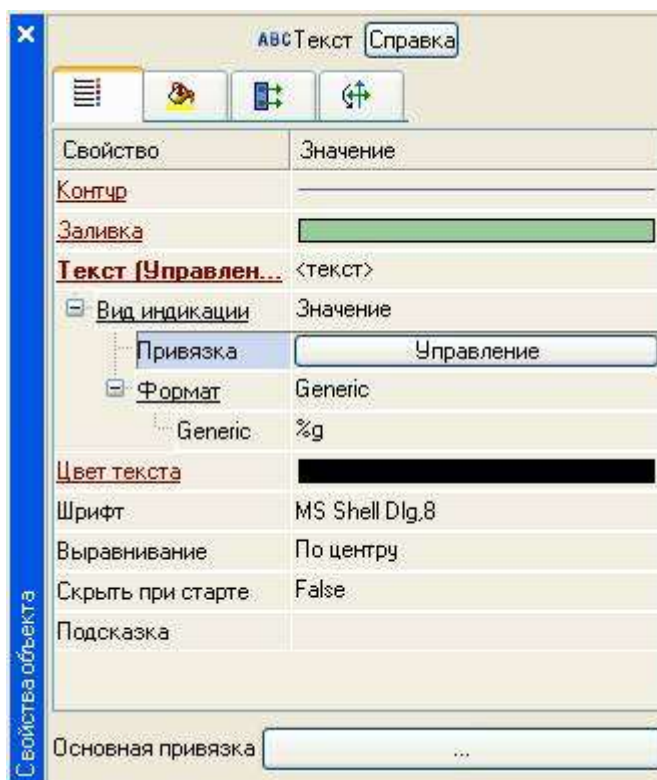
- переместить, удерживая нажатой ЛК, копию ГЭ Текст справа от размещенного на экране ГЭ Кнопка;

- двойным щелчком ЛК на перемещенном ГЭ Текст открыть окно его свойств;

- двойным щелчком ЛК на строке **Текст** вкладки основных свойств  перейти к настройке динамизации данного атрибута ГЭ;

- в правом поле строки **Привязка** щелчком ЛК открыть табличный редактор аргументов шаблона экрана;

- выделить ЛК в списке аргумент **Управление** и щелчком ЛК по экранной кнопке **Готово** подтвердить привязку атрибута ГЭ Текст к данному аргументу шаблона экрана;



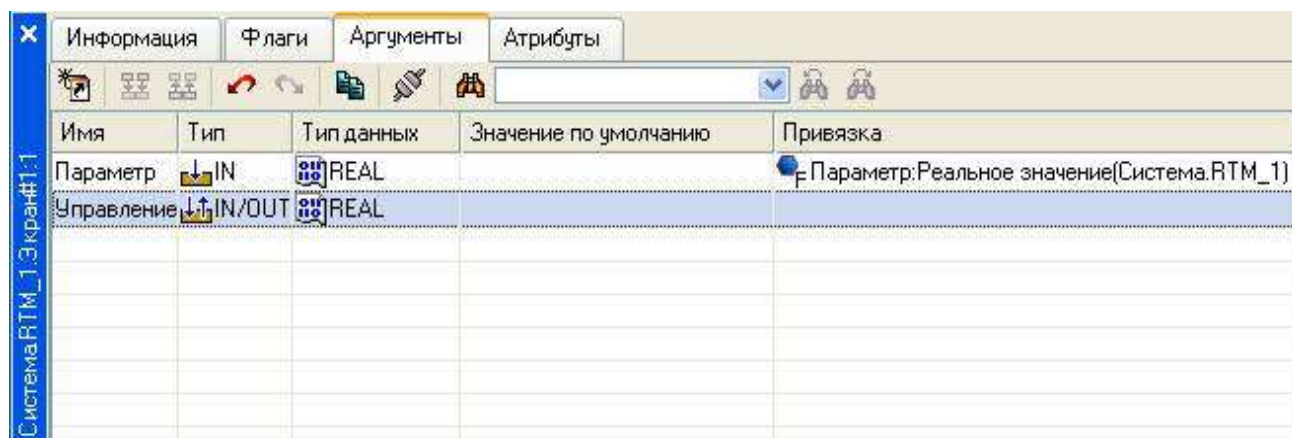
- закрыть окно свойств ГЭ Текст.

4.2. Привязка аргумента экрана к каналу.


Для создания по аргументу **Управление** шаблона экрана нового канала необходимо отредактировать привязку атрибута канала к аргументу шаблона экрана:

- в слое **Система** открыть узел **RTM_1**;

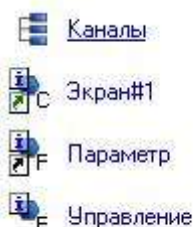
- по щелчку ПК вызвать через контекстное меню свойства компонента **Экран#1**;



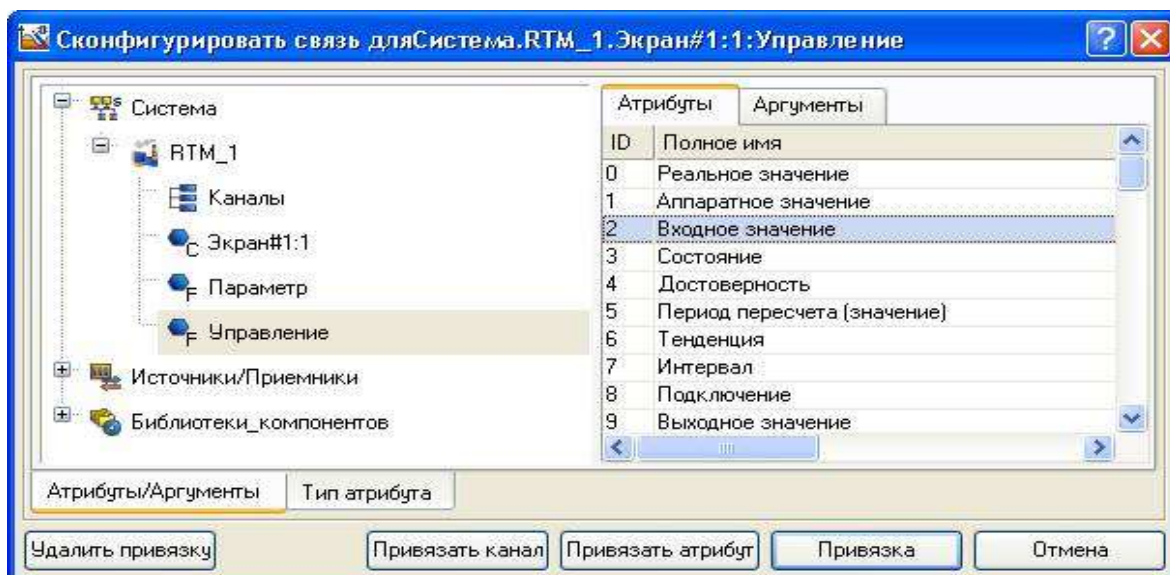
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Параметр	IN	REAL		Параметр:Реальное значение(Система.RTM_1)
Управление	IN/OUT	REAL		

- выбрать вкладку **Аргументы**, ЛК выделить аргумент **Управление** и с помощью иконки  выполнить автопостроение канала;




- в результате в узле **RTM_1** будет создан канал с именем **Управление**.



Двойным щелчком ЛК в поле **Привязка** аргумента **Управление** необходимо вызвать окно настройки связи, выбрать в нем атрибут **Входное значение** канала **Управление** и кнопкой **Привязка** подтвердить связь аргумента экрана **Управление** с атрибутом **Входное значение** канала **Управление**. Далее закрыть окно свойств компонента **Экран#1**.

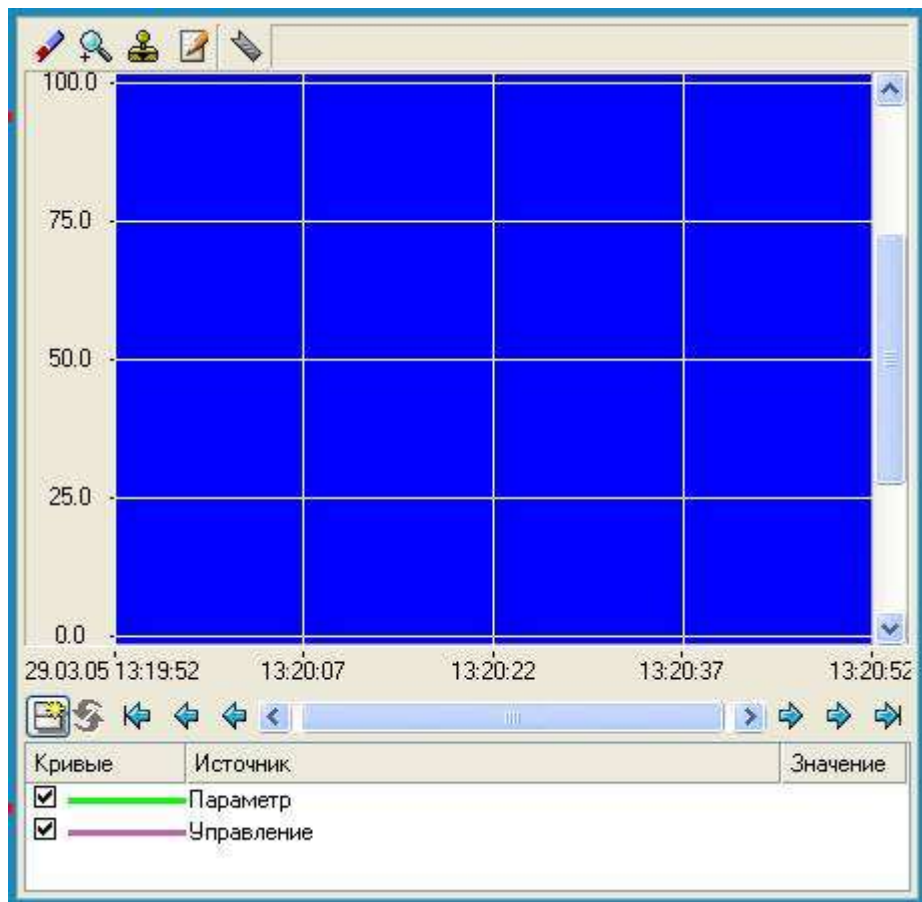


4.3. Размещение ГЭ Тренд.




Дополним созданный экран новым ГЭ для совместного просмотра изменений значений каналов узла во времени и отслеживании предыстории – трендом. Для этого в правой части графического экрана необходимо разместить ГЭ Тренд  для вывода значений **Параметр** и **Управление**. Основные свойства ГЭ  можно оставить заданными по умолчанию. Перейдя во вкладку  и выделив ЛК строку **Кривые**, с помощью ПК необходимо создать еще одну новую кривую, настроив для кривых привязки к существующим аргументам, толщину и цвет линий.



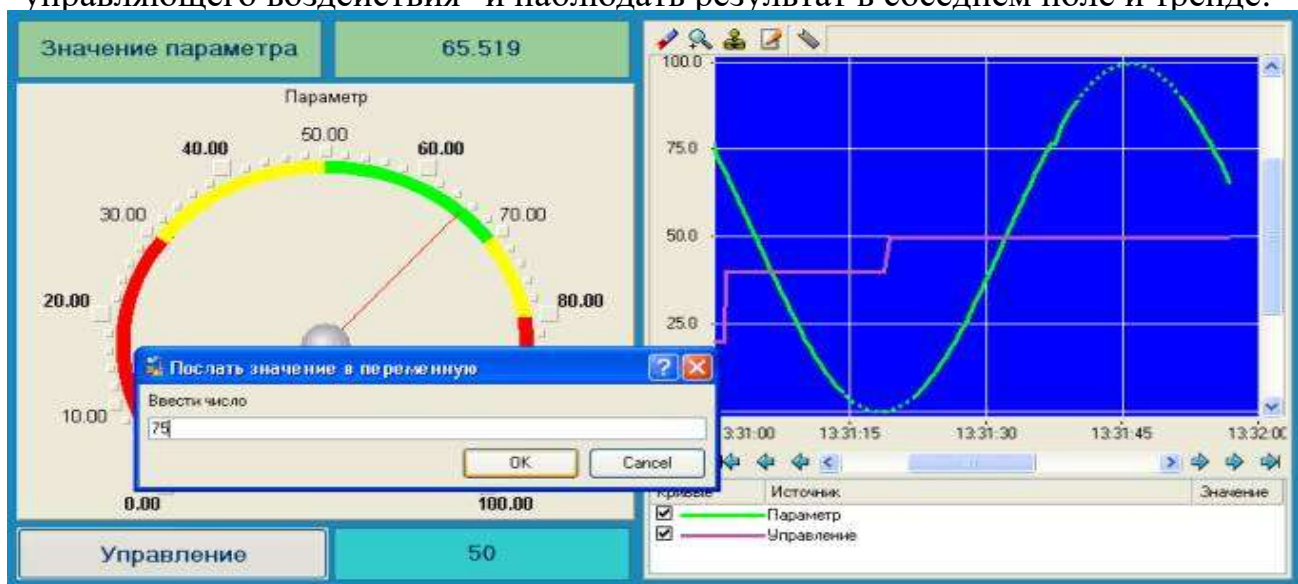
ГЭ примет следующий вид.



4.4. Запуск проекта.

Сохранить проект с помощью иконки ; на инструментальной панели выбрать ЛК  и подготовить тем самым проект для запуска в реальном времени; с помощью иконки  на инструментальной панели запустить проект на исполнение.

С помощью кнопки **Управление** необходимо ввести величину "управляющего воздействия" и наблюдать результат в соседнем поле и тренде.

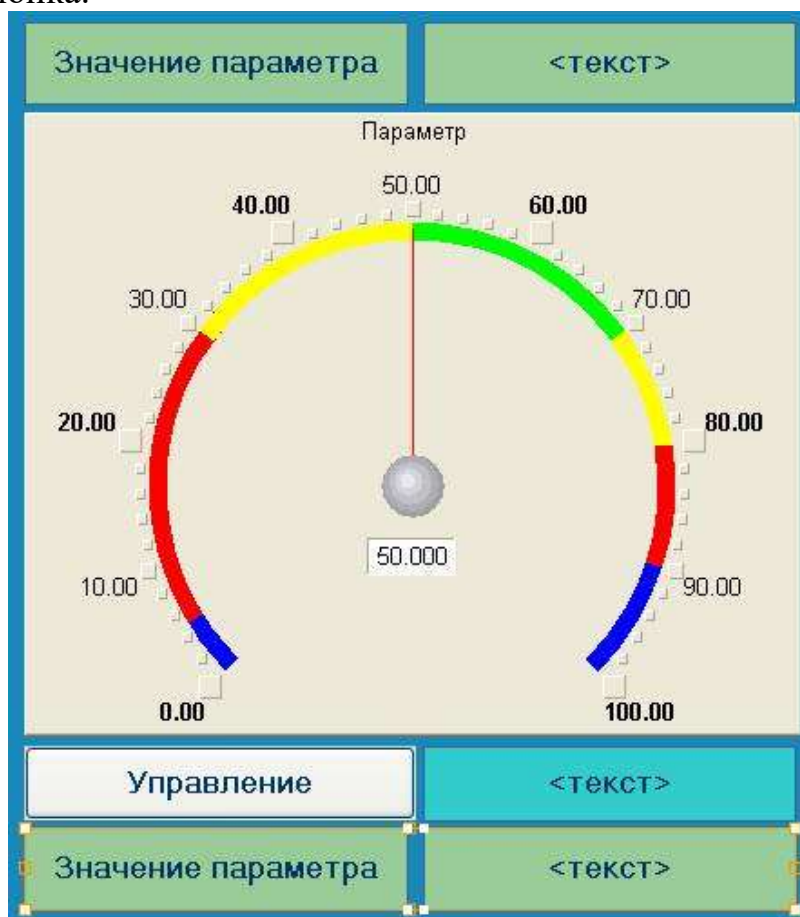


4.5. Простейшая обработка данных.

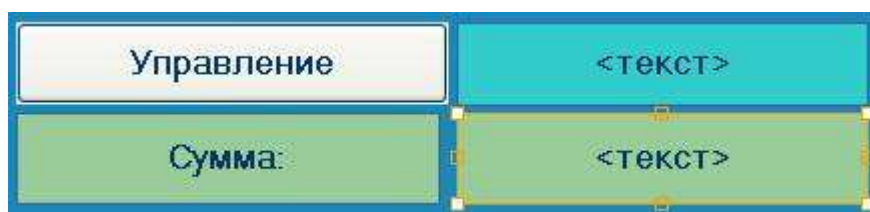
С помощью нового компонента проекта – шаблона программы необходимо связать два имеющихся канала операцией сложения. Пусть требуется суммировать реальные значения каналов **Параметр** и **Управление**, а результат помещать во вновь созданный аргумент экрана **Сумма** (с отображением на ГЭ Текст и Тренд) без создания дополнительного канала в узле проекта.

4.5.1. Доработка графического экрана.

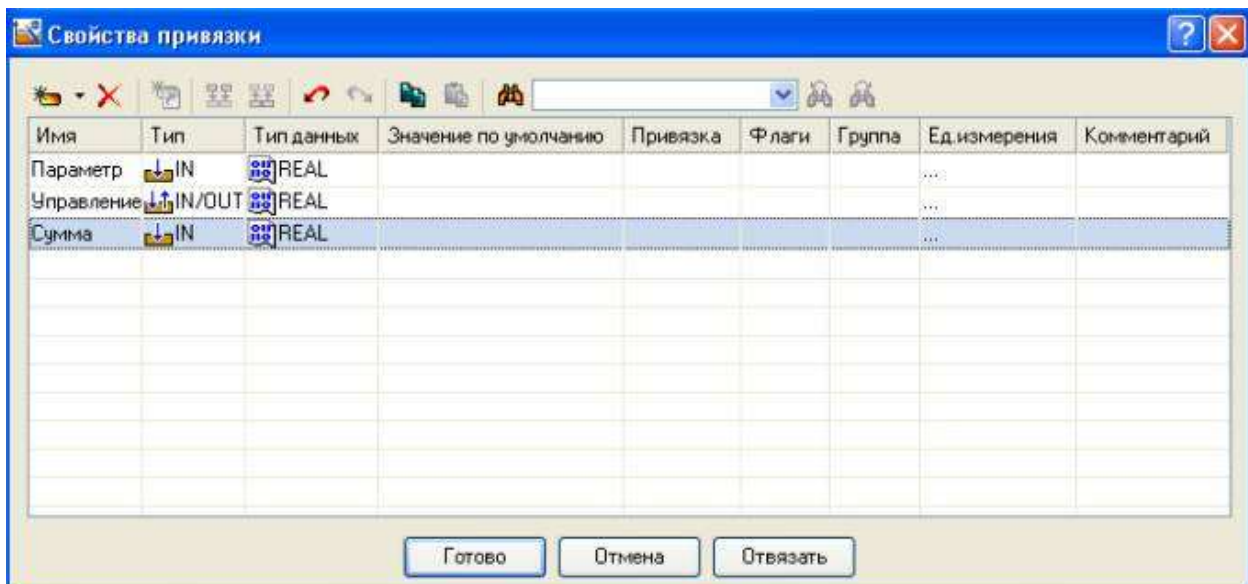
Скопируем два первых ГЭ – "Значение параметра" и "text" и разместим их ниже ГЭ Кнопка.



Далее необходимо изменить статический текст первого ГЭ на **Сумма:**.



Динамика второго ГЭ привязывается к новому - третьему аргументу шаблона экрана типа **IN** с именем **Сумма**, который создается в процессе привязки.



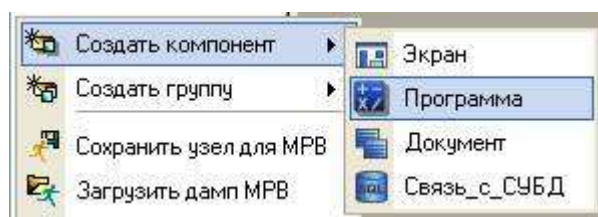
Далее добавить еще одну кривую на тренд с привязкой к аргументу **Сумма**.




4.6. Создание программы на языке Техно ST.

Требуется создать программу, в которой сумма двух аргументов, связанных с атрибутами **Реальное значение** каналов **Параметр** и **Управление**, будет помещаться в третий аргумент с именем **Сумма**. В дальнейшем воспользоваться возможностью связывания аргументов шаблонов для вывода на экран результата работы программы без создания дополнительного канала.

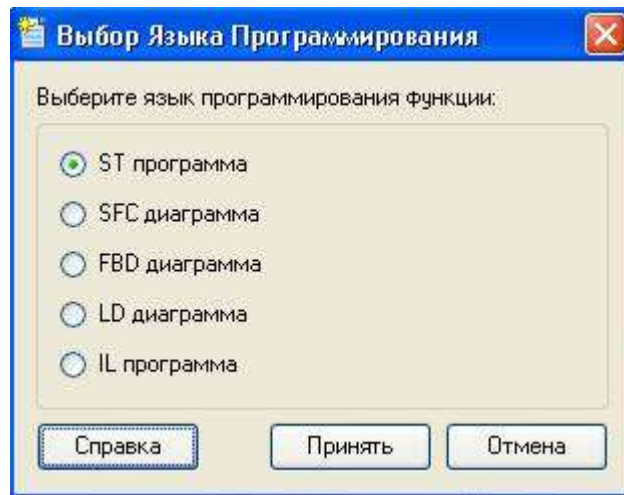
Двойным щелчком ЛК открыть узел **RTM_1**.



Выделением ЛК в дереве шаблона **Программа#1** строки **Аргументы** вызвать табличный редактор аргументов; с помощью иконки  создать в редакторе аргументов три аргумента с именами **Параметр**, **Управление** и **Сумма**.

Имя	Тип	Тип данных	Значение по умолчанию
Параметр	↓ IN	REAL	
Управление	↓ IN	REAL	
Сумма	↑ OUT	REAL	

Выделить ЛК в дереве шаблона строку **Программа#1** и в открывшемся диалоге **Выбор языка** выбрать язык **ST**.





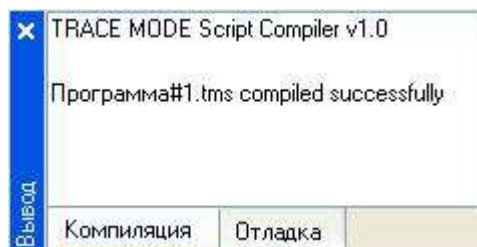
По нажатию экранной кнопки **Принять** в открывшемся окне редактора программ с объявленными переменными набрать следующую строку:

```
PROGRAM
VAR_INPUT Параметр : REAL; END_VAR
VAR_INPUT Управление : REAL; END_VAR
VAR_OUTPUT Сумма : REAL; END_VAR

Сумма=Параметр+Управление;

END_PROGRAM
```

С помощью иконки  на инструментальной панели редактора или нажатием "горячей клавиши" **F7** скомпилировать программу и убедиться в успешной компиляции в окне **Выход** (Output), вызываемом из инструментальной панели с помощью иконки .

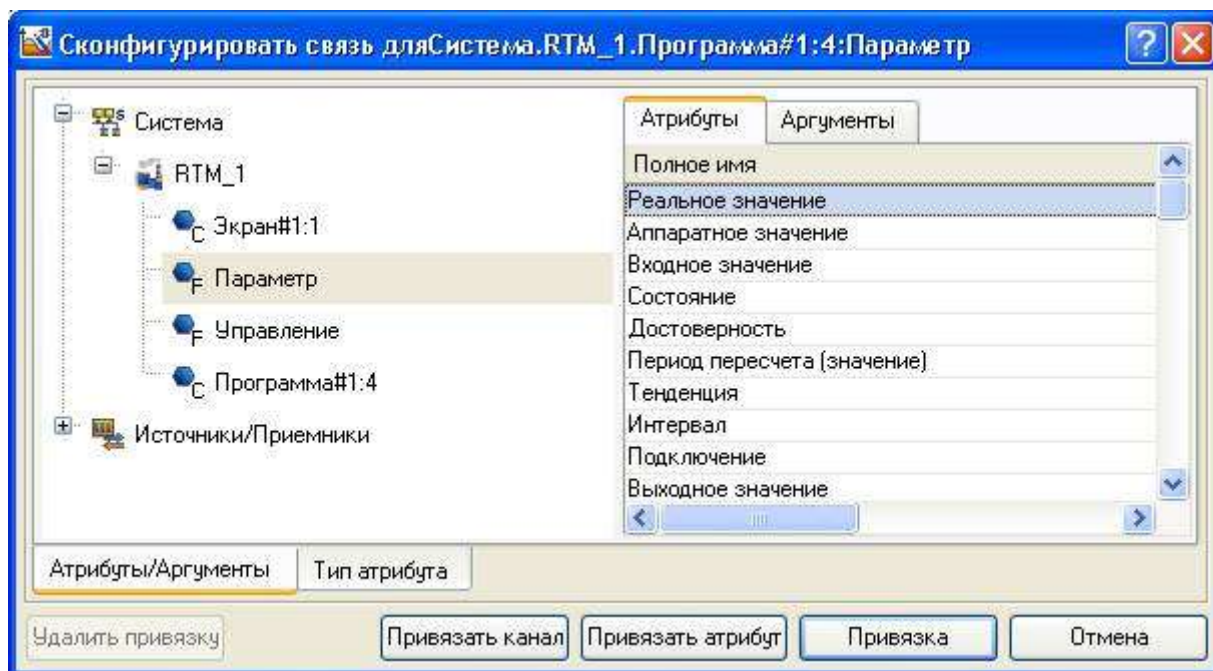


4.7. Привязка аргументов программы.

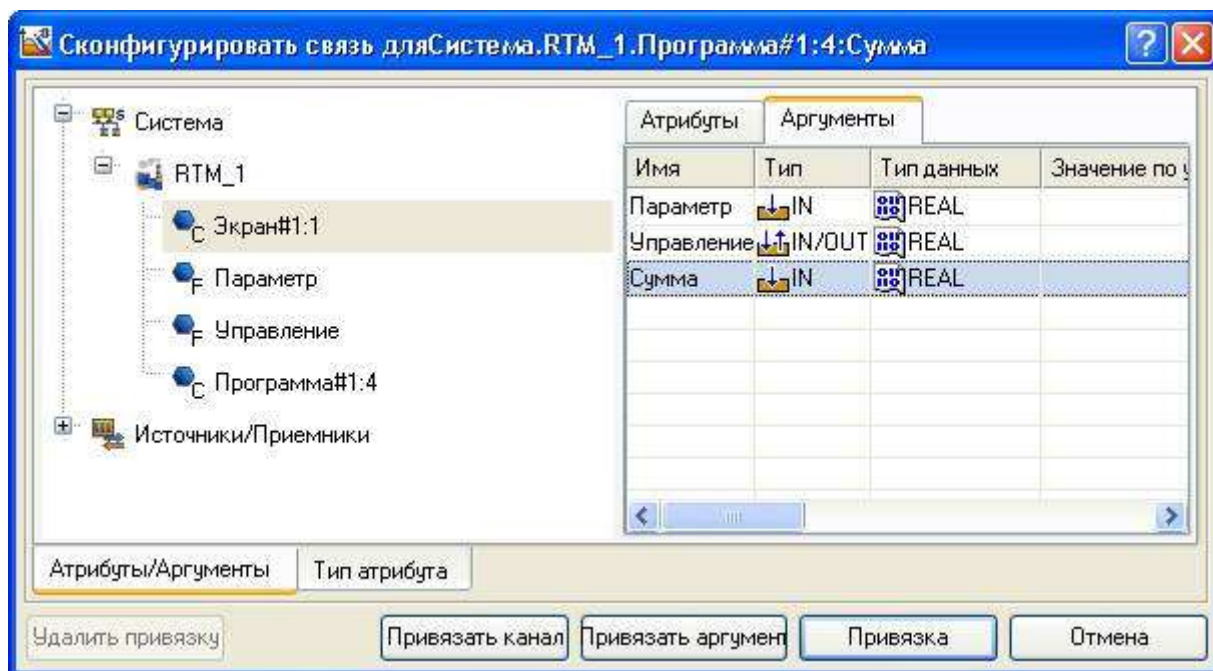
Выполнить привязку аргументов программы к атрибутам каналов:

- вызвать свойства компонента **Программа#1** через контекстное меню;
- выбрать вкладку **Аргументы**;

- двойным нажатием в поле **Привязка** связать аргументы программы с атрибутами каналов – аргумент **Параметр** к реальному значению канала **Параметр**, аргумент **Управление** к реальному значению канала **Управление**;



- двойным щелчком в поле **Привязка** аргумента программы **Сумма** вызвать окно настройки связи, выбрать в левом окне канал класса **Вызов Экран#1**, а в правом открыть вкладку **Аргументы** и указать в ней аргумент **Сумма**, затем щелчком ЛК по экранной кнопке **Привязка** подтвердить связь.






В результате будем иметь

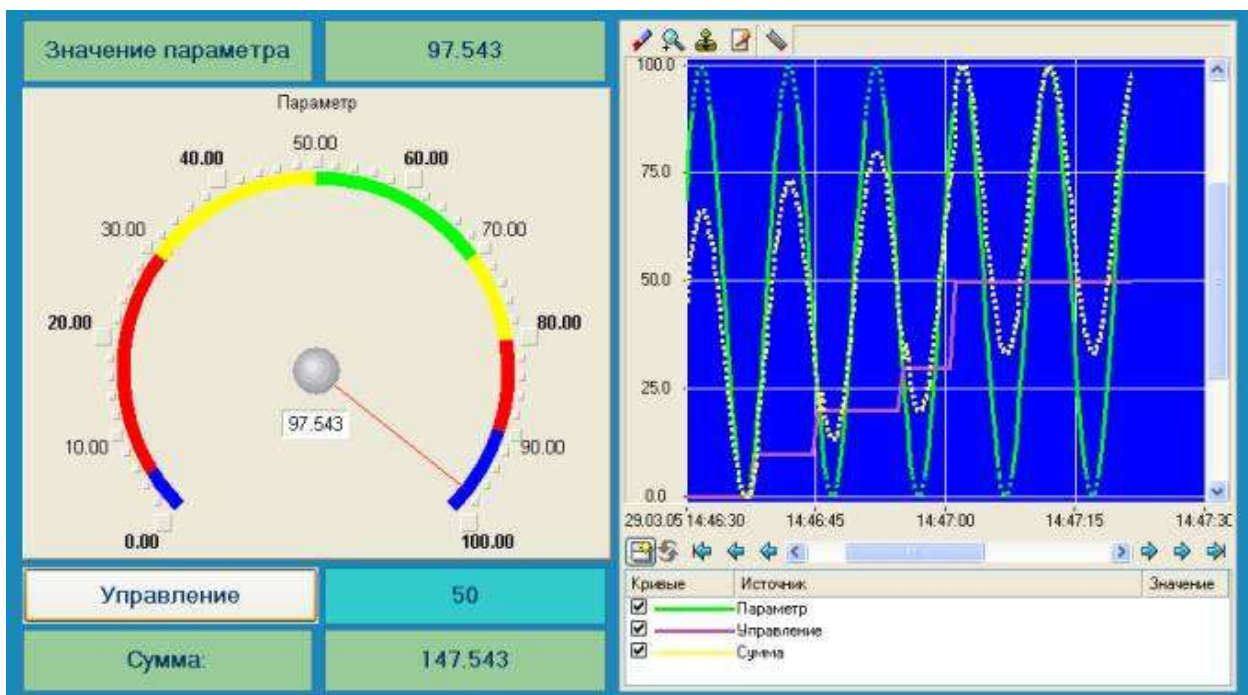
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Параметр	IN	REAL		Параметр:Реальное значение(Система.RTM_1)
Управление	IN	REAL		Управление:Реальное значение(Система.RTM_1)
Сумма	OUT	REAL		Экран#1:1:Сумма(Система.RTM_1)

Закрывать окно свойств компонента **Программа#1**.

4.8. Запуск проекта.

Сохранить проект с помощью иконки . На инструментальной панели выбрать ЛК иконку  и подготовить тем самым проект для запуска в реальном времени. С помощью иконки  на инструментальной панели запустить режим исполнения.

Вводя теперь с помощью кнопки "Управление" "управляющие воздействия", можно наблюдать соответствующее изменение реального значения канала **Управление** и результата работы программы суммирования.



5. Контрольные вопросы.

- 5.1. Как разместить экран ТРЕНД?
- 5.2. Как нужно доработать главный экран для простейшей обработки данных?
- 5.3. Этапы создания программы на языке ТехноST.

ЛАБОРАТОРНАЯ РАБОТА №3

Связь по протоколу DDE с приложением MS Windows на примере Excel

1. Цель работы – изучение применения протокола DDE для связи с приложением MS Windows на примере Excel.
2. Задание по лабораторной работе: использовать МРВ как DDE – сервер (клиент).
3. Методические указания по выполнению лабораторной работы.
 - 3.1. Изучить использование МРВ в качестве DDE – сервера (п. 4.1).
 - 3.2. Применить МРВ как DDE – клиент (п. 4.2).
4. Процедура связи по протоколу DDE с приложением MS Windows.
 - 4.1 Использование МРВ в качестве DDE – сервера.

Необходимо организовать запрос реальных значений каналов узла разработанного проекта приложением MS Windows, в качестве которого выбрана книга MS Excel. Для этого:

- выполнить запуск приложения MS Excel;
- записать в двух ячейках первого столбца запросы на получение данных:

=RTM0|GET!Параметр

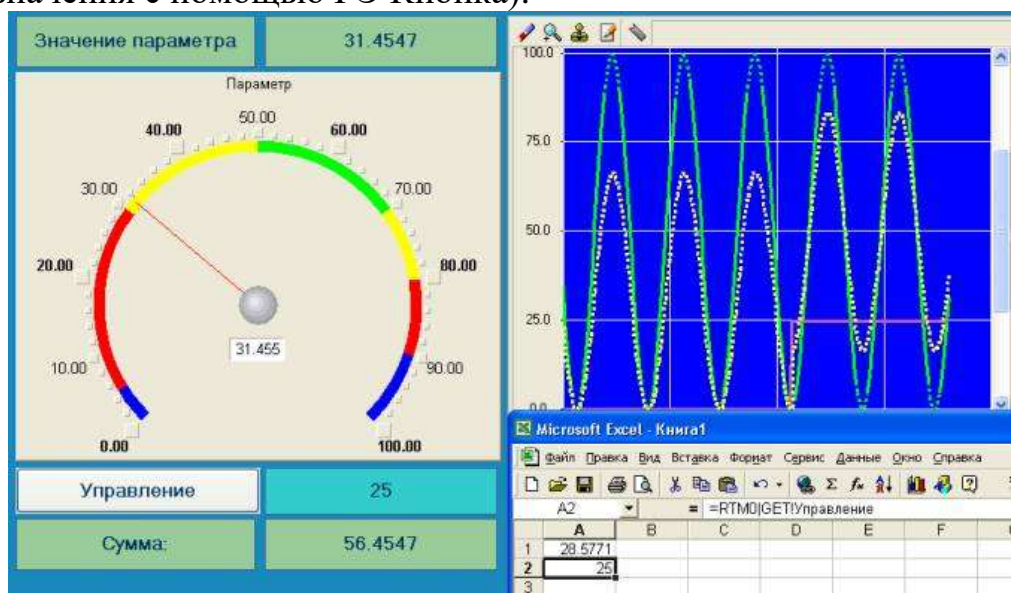
=RTM0|GET!Управление

где **0** – индивидуальный номер узла в проекте;

- запустить на исполнение узел АРМ **RTM_1**;

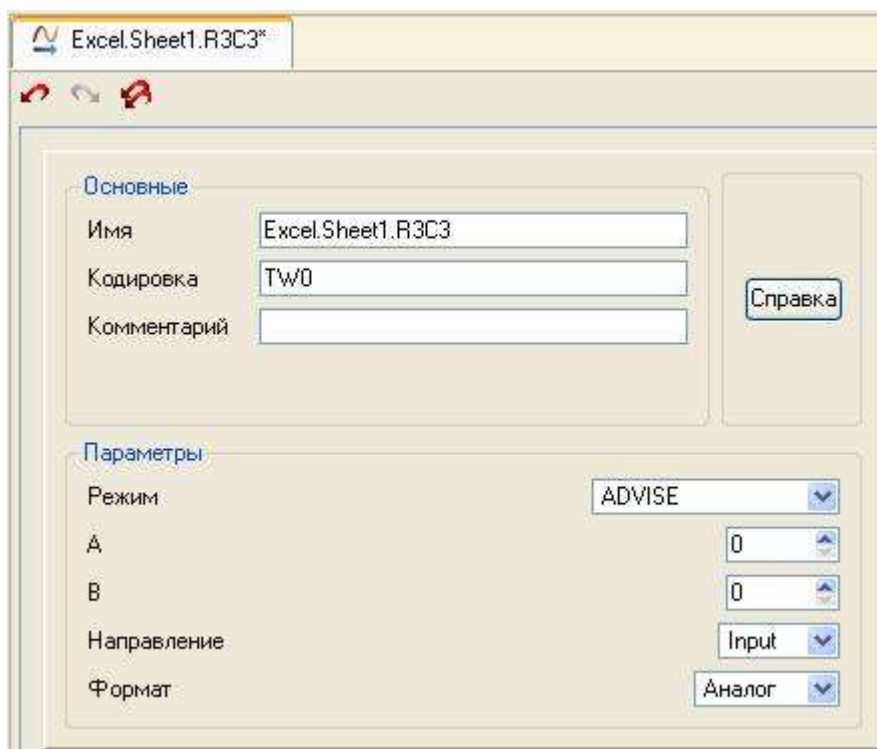
- в меню таблицы MS Excel **Правка** выбрать команду **Связи**, выделить оба параметра и нажать кнопку **Обновить**, после чего закрыть окно кнопкой **ОК**.

Необходимо убедиться, что значения в ячейках книги Excel изменяются вместе с соответствующими реальными значениями каналов узла (значения канала **Параметр** меняется постоянно, а канала **Управление** – после введения нового значения с помощью ГЭ Кнопка).



4.2. MPB как DDE – клиент.

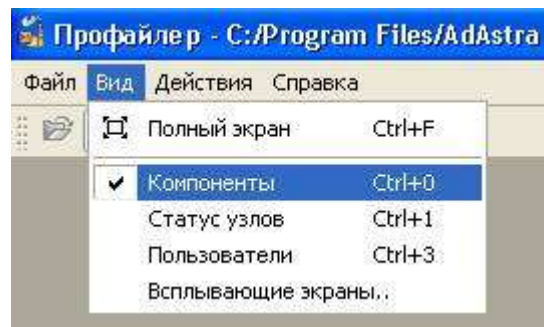
В том случае, когда требуется получать данные от внешнего приложения по протоколу **DDE**, MPB TRACE MODE 6 должен выступать в роли **DDE-клиента**. Например, если необходимо вводить во вновь создаваемый канал Из_таблицы узла **RTM_1** (в его атрибут **Входное значение**) данные из ячейки **R3C3** книги MS Excel, необходимо в слое **Источники/Приемники** создать новую группу **DDE**, а в ней – компонент **DDE#1** и отредактировать его следующим образом:



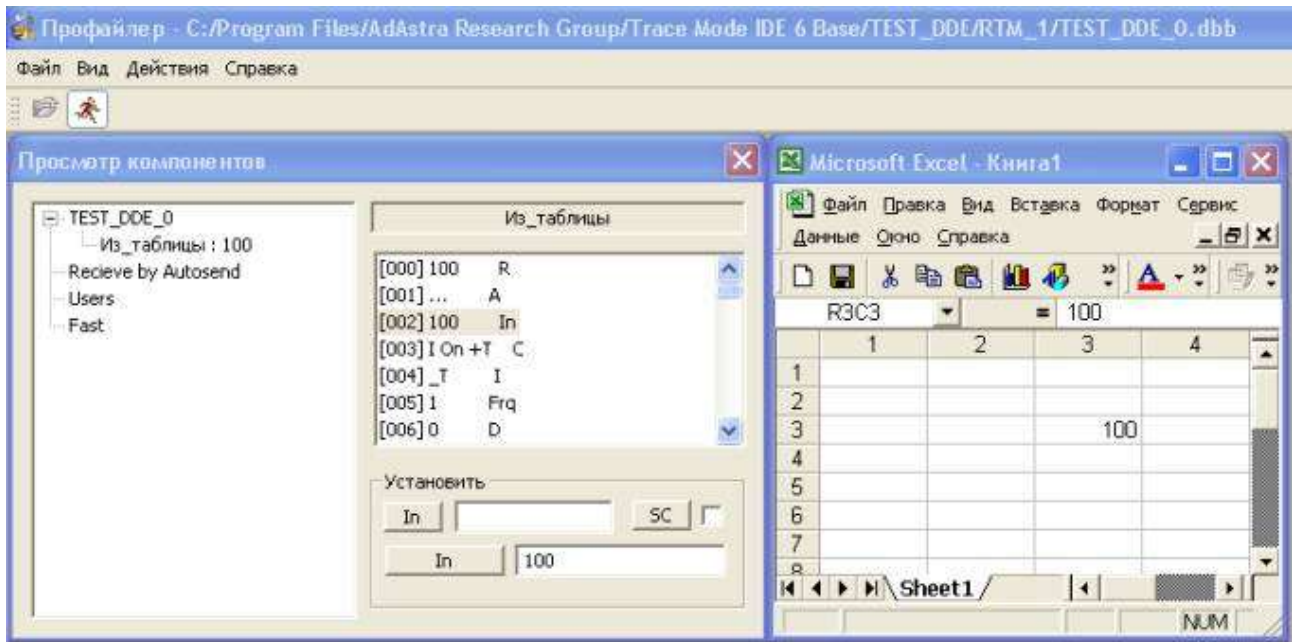
где в атрибуте **Имя**:

- Excel – имя приложения;
- Sheet1 – имя листа книги MS Excel;
- R3C3 – адрес ячейки;
- **ADVISE** – режим отправки клиенту значения при каждом его изменении.

После этого необходимо создать канал класса **Float** типа **Input** с именем Из_таблицы и привязать к нему с помощью механизма **drag-and-drop** источник **Excel.Sheet1.R3C3**. После процедур сохранения проекта и подготовки его к запуску в реальном времени запустить MS Excel, а затем узел АРМ **RTM_1**. Вводя в ячейку **R3C3** произвольные значения, их можно наблюдать в атрибутах канала Из_таблицы с помощью окна просмотра компонентов, открываемого через основное меню отладчика:



Таким образом, в результате будем наблюдать следующее.



5. Контрольные вопросы.

5.1. Как организовать запрос реальных значений каналов узла приложением?

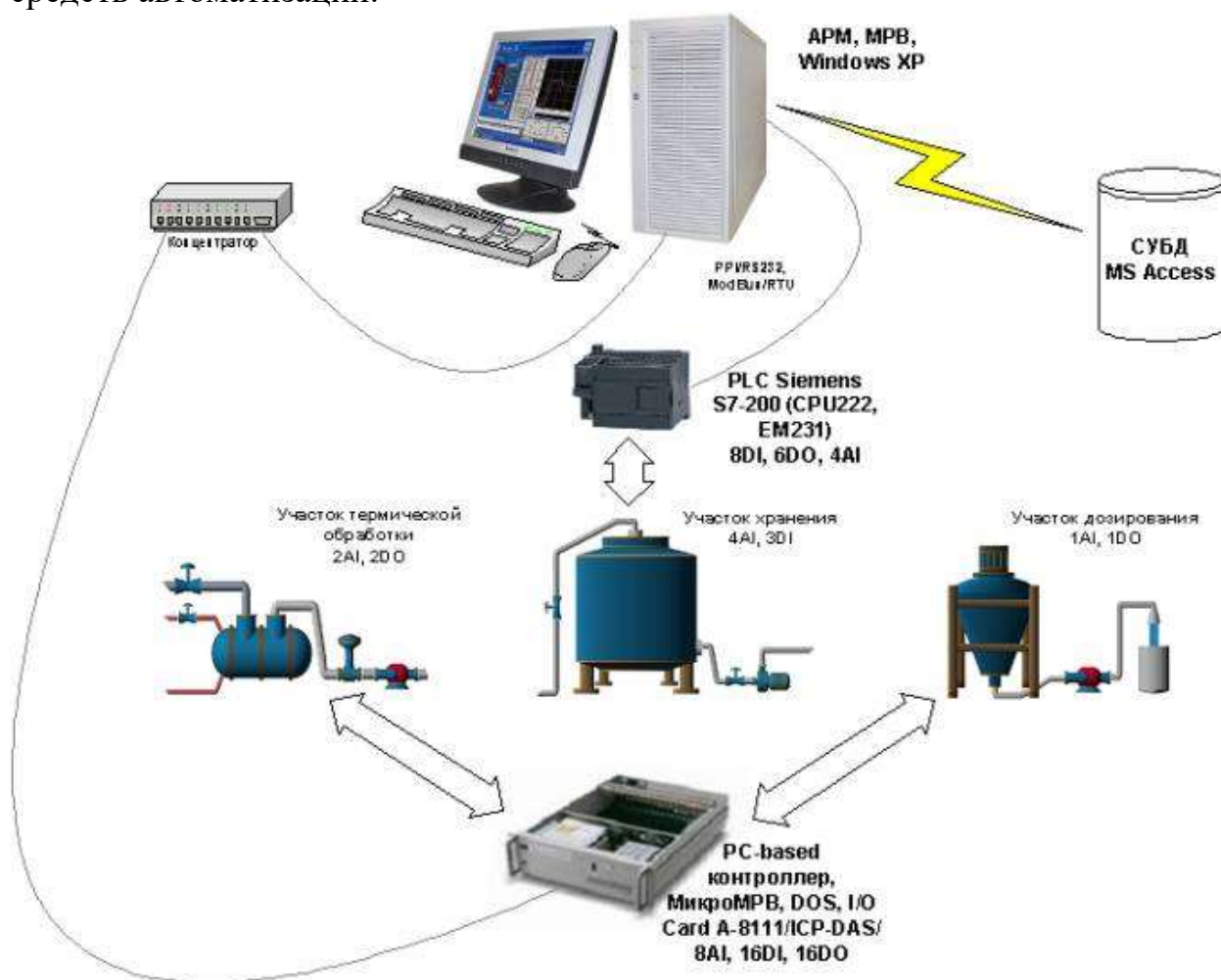
5.2. Как организовать получение данных от внешнего приложения?

ЛАБОРАТОРНАЯ РАБОТА №4

Построение операторского интерфейса: мониторинг, управление, регулирование


1. Цель работы – построение операторского интерфейса, реализующего функции мониторинга, управления и регулирования.
2. Задание по лабораторной работе: создать экраны АРМ.
3. Методические указания по выполнению лабораторной работы.
 - 3.1. Создать экраны АРМ, реализующие функции мониторинга, управления и регулирования (п. 4).
4. Процедура создания экранов АРМ, реализующих функции мониторинга, управления и регулирования.

В качестве примера рассматривается технологический процесс (ТП), который 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.

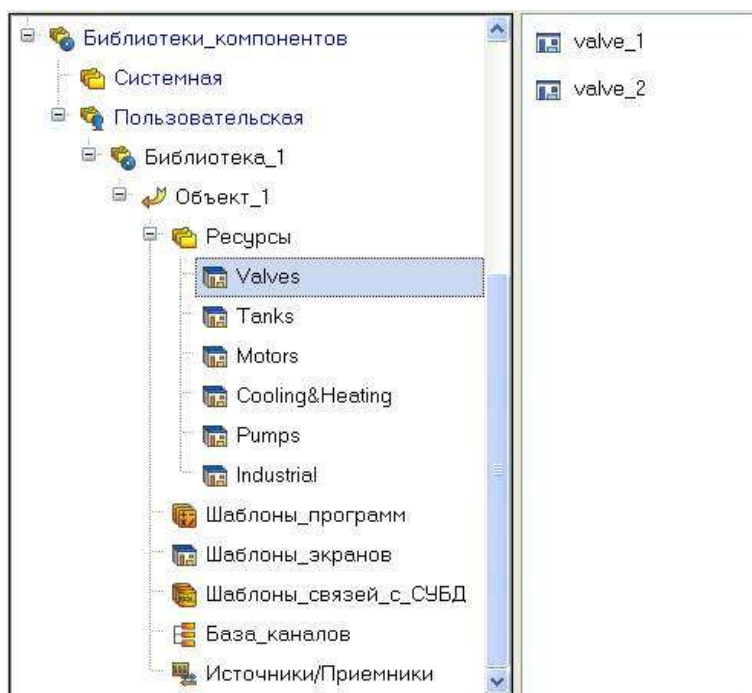


Информационная база проекта – каналы по аргументам разрабатываемых шаблонов экранов и программ, будет создаваться путем дополнения основного подхода методами автопостроения и связывания каналов в узлах проекта.

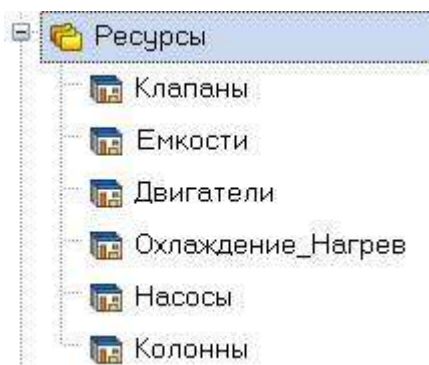
Для этого можно воспользоваться пользовательской библиотекой компонентов, скопировав файл **tmdevenv.tmul** из поддиректории **%TRACE MODE%\Lib** в директорию **%TRACE MODE%**.

Открыть интегрированную систему разработки и с помощью щелчка ЛК по иконке  создать новый проект. В качестве стиля разработки выбирается **Стандартный**.

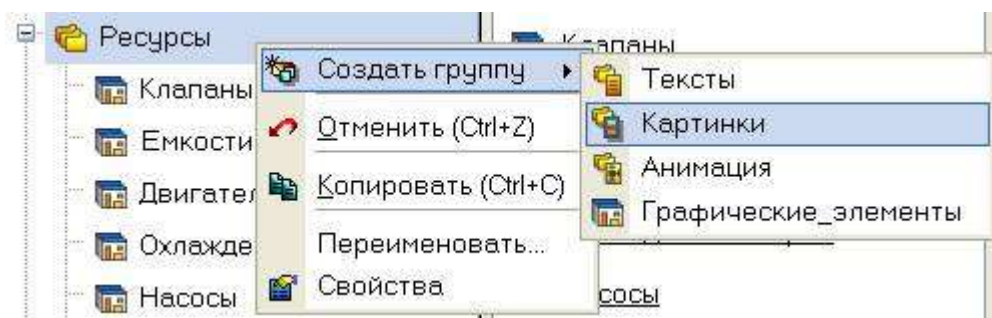
Перейти в слой **Библиотеки_компонентов**, где в разделе **Пользовательская** открыть библиотеку **Библиотека_1**. Сохраненный в данной библиотеке объект **Объект_1** содержит в своем слое **Ресурсы** необходимый для дальнейшей разработки набор графических объектов – изображения клапанов, емкостей, двигателей и т.д.



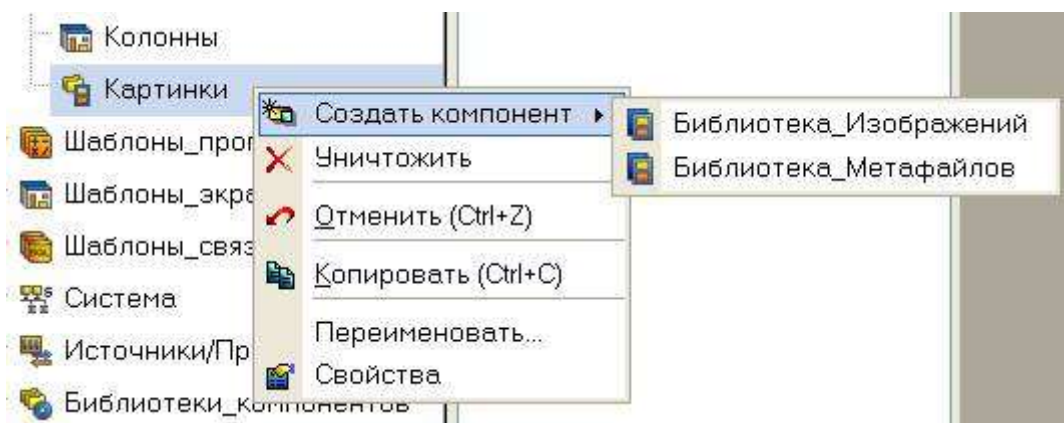
Перенести группы в слой **Ресурсы** текущего проекта с помощью механизма **drag-and-drop** и переименовать их, как показано ниже.




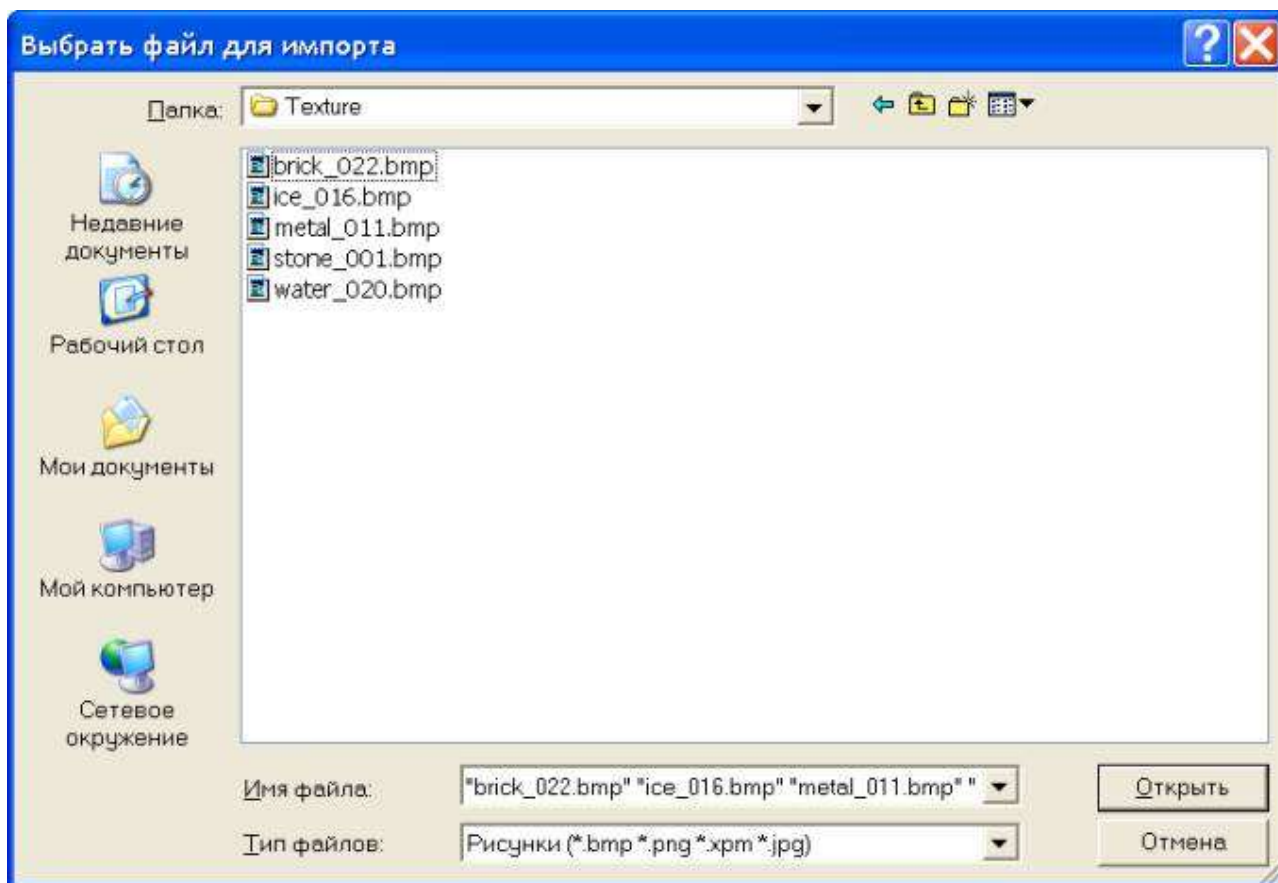
Здесь же в слое **Ресурсы** создать группу **Картинки** для помещения в нее текстур, которые будут применены в оформлении создаваемых графических экранов.



Создать в группе **Картинки** новый компонент — **Библиотека_Изображений#1**.



Открыть двойным щелчком ЛК вновь созданную библиотеку для редактирования. Для ее наполнения воспользоваться иконкой  на панели инструментов. В открывшемся диалоге выбора файлов для импорта указать поддиректорию `...\Lib\Texture`. Выбрать все файлы и нажать экранную кнопку **Открыть**.




Содержимое библиотеки **Библиотека_Изображений#1** станет следующим.




Подобным описанному выше способу создать в слое **Ресурсы** группу **Анимация**, в ней - библиотеку **Библиотека_Видеоклипов#1**. Наполнить ее содержимым ...\\Lib\\Animation.

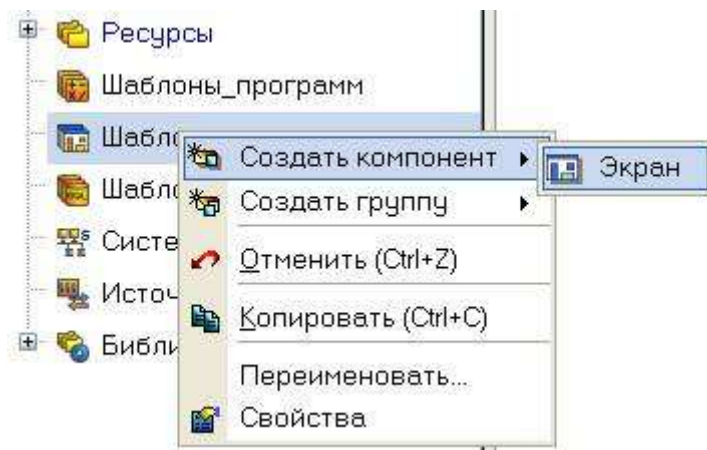


Из всех представленных в библиотеке видеоклипов будет использоваться только **fluid_blue**, остальные можно убрать с помощью иконки  на панели инструментов, предварительно выбрав видеоклип ЛК.

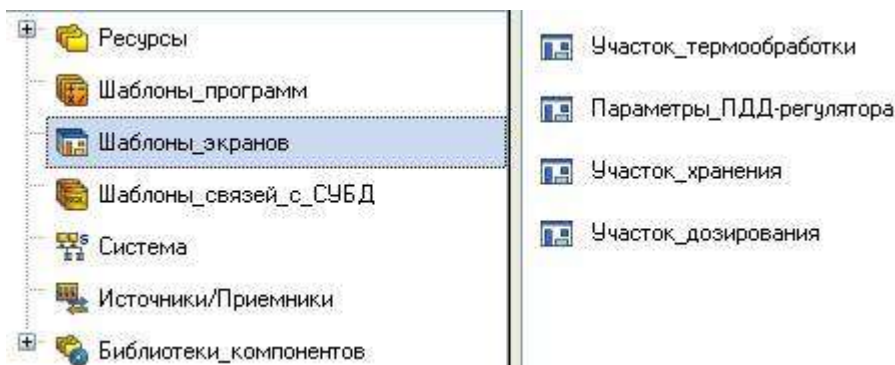
В качестве видеоклипов могут быть использованы практически любые имеющиеся файлы форматов **avi** или **mng**.


После проведения подготовительных мероприятий сохранить выполненную работу, нажав ЛК  и указав имя **QS_Lesson_2.prj**.

Перейдя в слой **Шаблоны_экранов**, создать в нем компонент **Экран#1**.




На созданном экране будут отображаться технологические параметры участка термической обработки, с него же будет осуществляться формирование задания на поддержание рабочей температуры. Далее переименовать его в **Участок_термообработки**. Создать также экраны для задания параметров ПДД-регулятора, участков хранения и дозирования. Переименовать их соответственно.

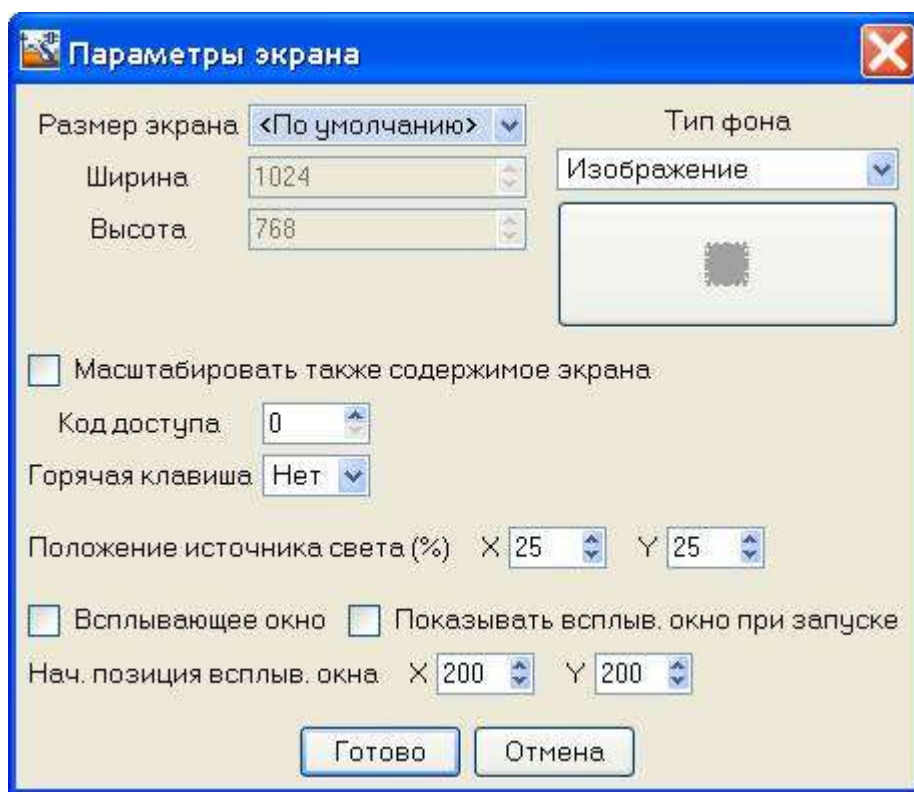





Назначить аргументы шаблону экрана участка термообработки – щелчок ПК на созданном шаблоне экрана и выбор из выпадающего списка пункта **Свойства**, далее переход во вкладку **Аргументы**. Здесь и далее с помощью иконки  создаются необходимые аргументы, задаются их имена, тип, тип данных, значения по умолчанию, привязки, флаги и т.д.

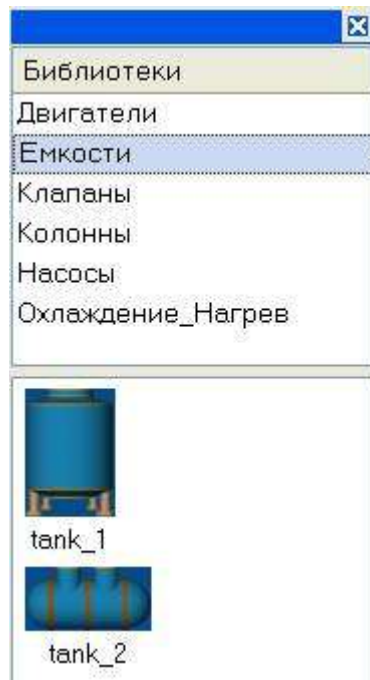
Информация		Аргументы			
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Расход_теплоносителя	IN	REAL			
Температура_рабочая	IN	REAL			
Задание_температуры	OUT	REAL			

Те аргументы, значения которых будут отображаться на экране, имеют тип **IN**, а те, что задаются с клавиатуры АРМ, отображаются на экране и должны пересылаться в конечном итоге, например, в PC-based контроллер, имеют тип **OUT**. В процедуре автопостроения каналов от шаблонов автопривязка аргументов будет осуществляться соответственно к атрибутам **Реальное** и **входное значение** каналов.

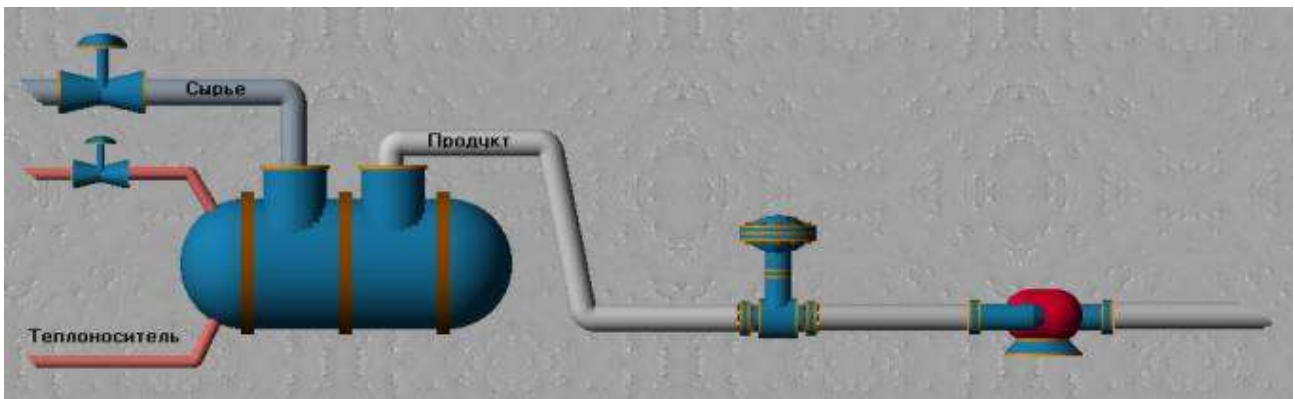
Закрывать бланк свойств экрана щелчком ЛК на . Для перехода к непосредственному созданию и редактированию содержимого экрана дважды нажать на нем ЛК мыши. Задать в качестве фона экрана текстуру **metal_011**. Для этого выбрать в основном меню пункт **Сервис**, в нем – **Параметры экрана**. В открывшемся диалоговом окне указать в качестве типа фона изображение, а из имеющихся в библиотеке текстур - **metal_011**.



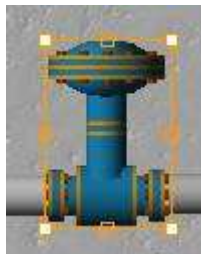
После нажатия экранной кнопки **Готово** фон графического экрана будет изменен на указанный. С помощью графических объектов (ГО), сохраненных в ресурсных библиотеках и вызываемых с помощью иконки  панели инструментов, а также графических элементов (ГЭ) объемных труб  и текста , создать статическую часть экрана.




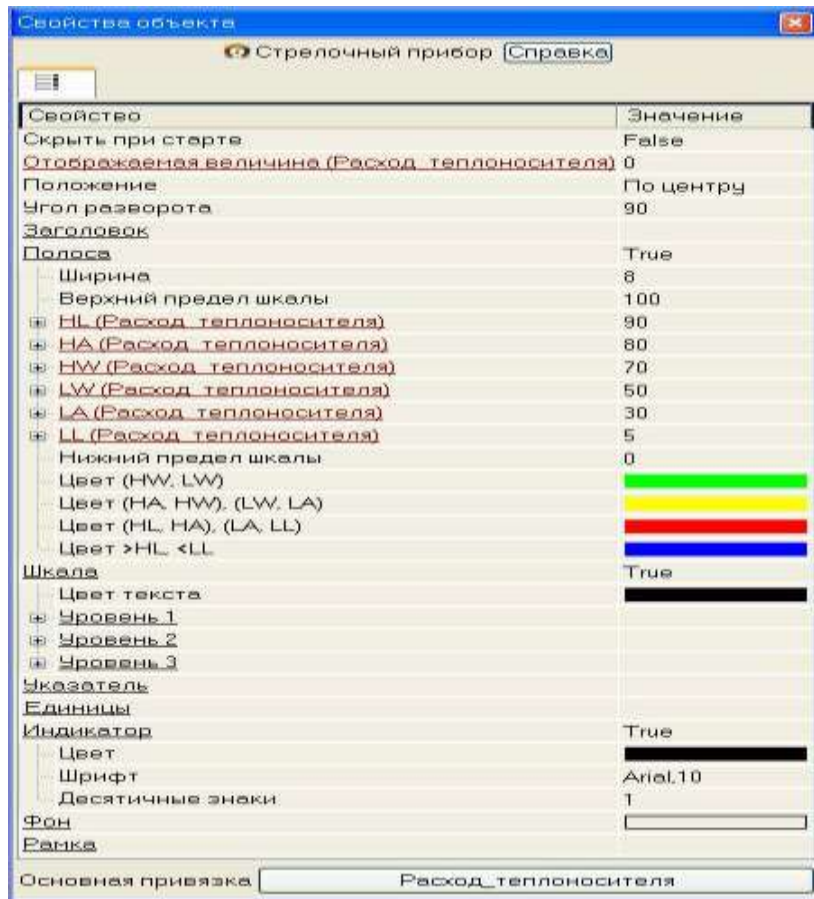
Примерный вид представлен ниже.



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 будут отображаться с помощью ГЭ Показывающий прибор . Разместив их на экране, двойным щелчком ЛК открыть свойства левого ГЭ и задать ему свойства, указанные ниже.



Аналогичным образом поступить с правым ГЭ.






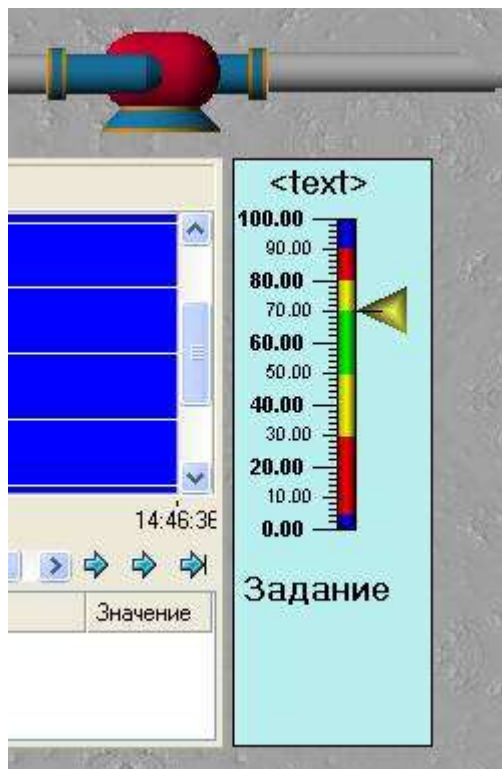
В нижней части экрана с помощью иконки  разместить ГЭ Тренд для вывода значений аргументов **Температура_рабочая**, **Расход_теплоносителя** и **Задание_температуры**. Основные свойства ГЭ оставить заданными по умолчанию, добавив заголовок **Участок термообработки**.



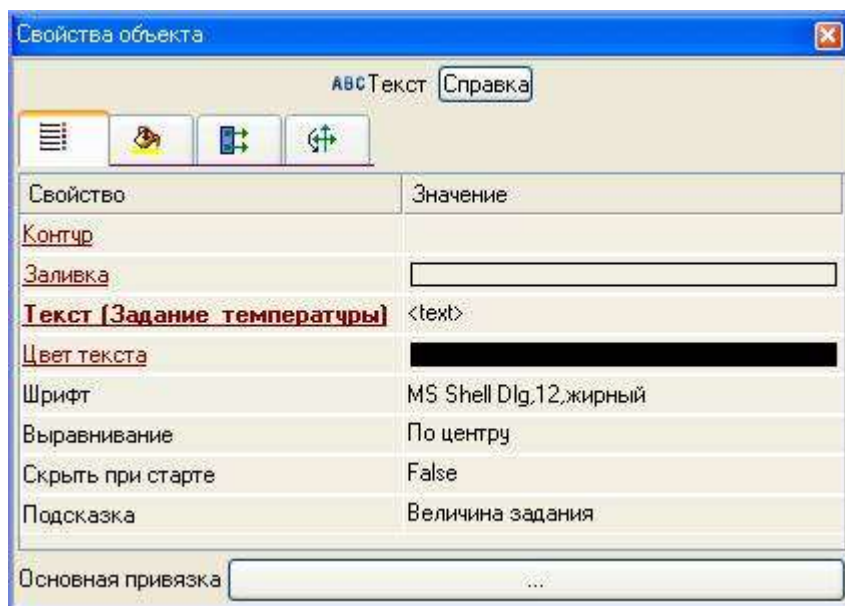
Определить для отображения на тренде три кривые, связать их с соответствующими аргументами экрана и задать для них цвет и толщину линий, интервалы выводимых значений.

Свойство	Значение
Кривые	
Кривая1 (Температура ра...)	
Заголовок	Температура_рабочая
Привязка	Температура_рабочая
Цвет	Yellow
Стиль линии	Solid
Толщина линии	3
Тип меток	Нет меток
Стиль линии вне границ	Dashed
Макс. значение	100
Мин. значение	0
Интерполирование	По периоду реального времени
Кривая2 (Расход теплоно...)	
Заголовок	Расход_теплоносителя
Привязка	Расход_теплоносителя
Цвет	Green
Стиль линии	Solid
Толщина линии	3
Тип меток	Нет меток
Стиль линии вне границ	Dashed
Макс. значение	100
Мин. значение	0
Интерполирование	По периоду реального времени
Кривая3 (Задание темпе...)	
Заголовок	Задание_температуры
Привязка	Задание_температуры
Цвет	Red
Стиль линии	Solid
Толщина линии	3
Тип меток	Нет меток
Стиль линии вне границ	Dashed
Макс. значение	100
Мин. значение	0
Интерполирование	По периоду реального времени

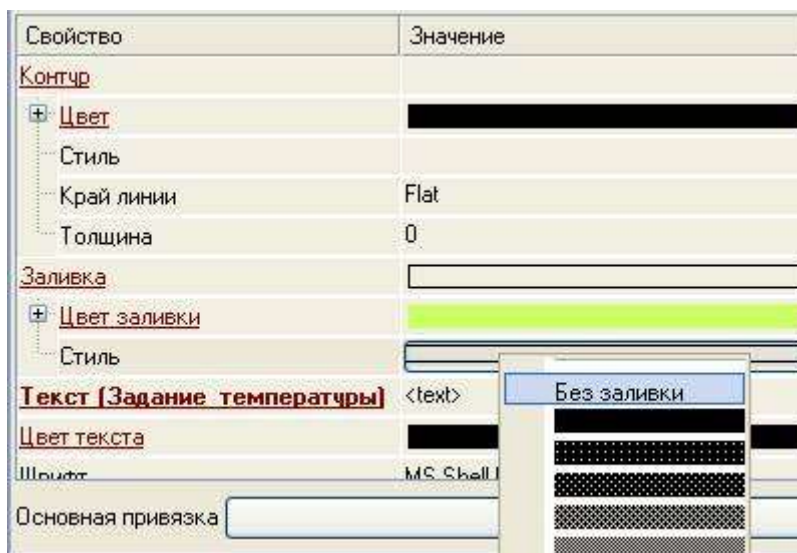
Для формирования задания регулятору разместить справа от ГЭ Тренд ГЭ Прямоугольник , он будет служить подложкой для ГЭ Ползунок , с помощью которого будет задаваться и отображаться величина задания. Точную величину задания будем отображать в верхней части прямоугольника с помощью ГЭ Текст .



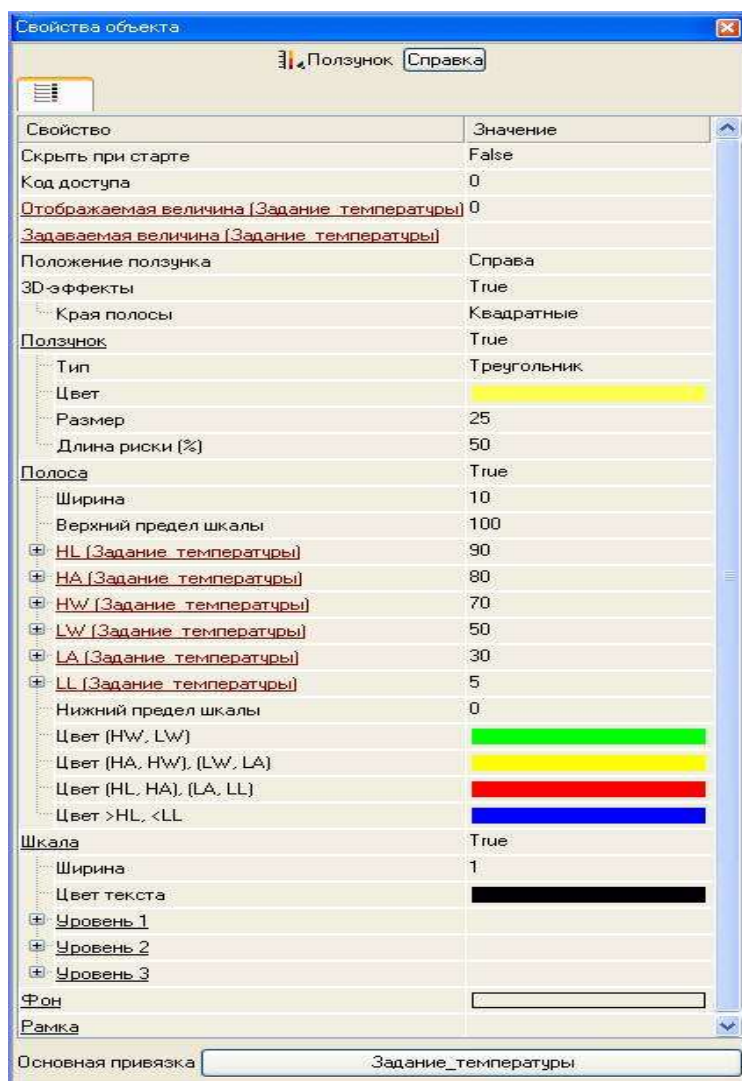
Свойства ГЭ **ABC** будут следующие.




Можно отказаться от использования рамки и заливки для данного ГЭ, задав ему свойства, указанные ниже.





Свойства ГЭ назначить следующим образом, также отказавшись от рамки и заливки.

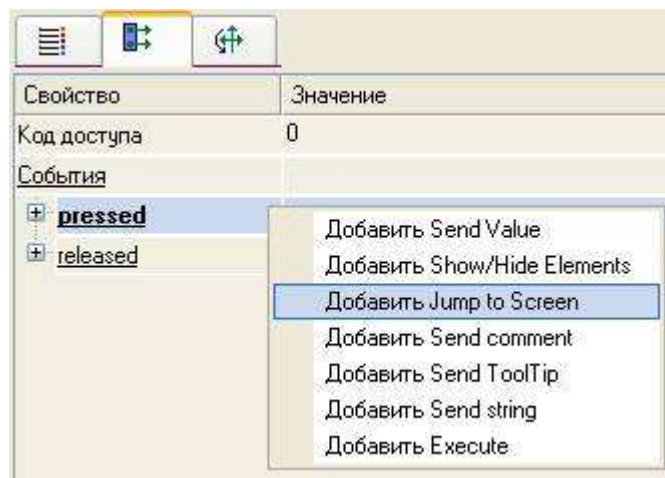


В правом верхнем углу экрана разместить надпись – **Участок термообработки.**

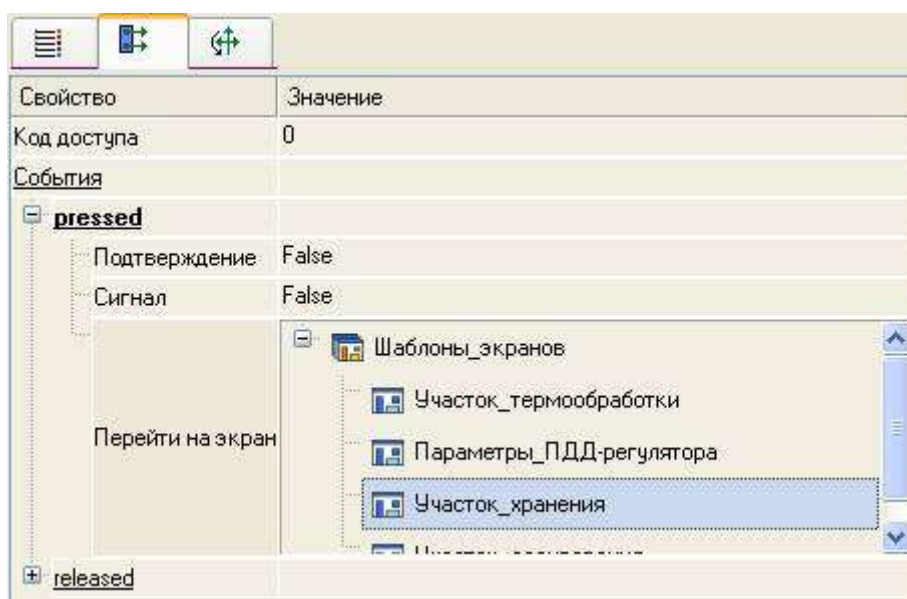
Так как для АРМ будет разработано еще два экрана (Хранение и Дозирование), то для осуществления переходов между экранами необходимо предусмотреть соответствующие средства. В качестве них будет использован ГЭ .




Двойным щелчком ЛК на ГЭ  **Хранение** открыть его свойства, в разделе **События**  выделить ЛК пункт **pressed** (событие по нажатию ЛК на ГЭ) и по нажатию ПК добавить переход на экран.




В открывшемся пункте **Jump to Screen** по нажатию ЛК из списка выбрать шаблон экрана **Участок_Хранения**.



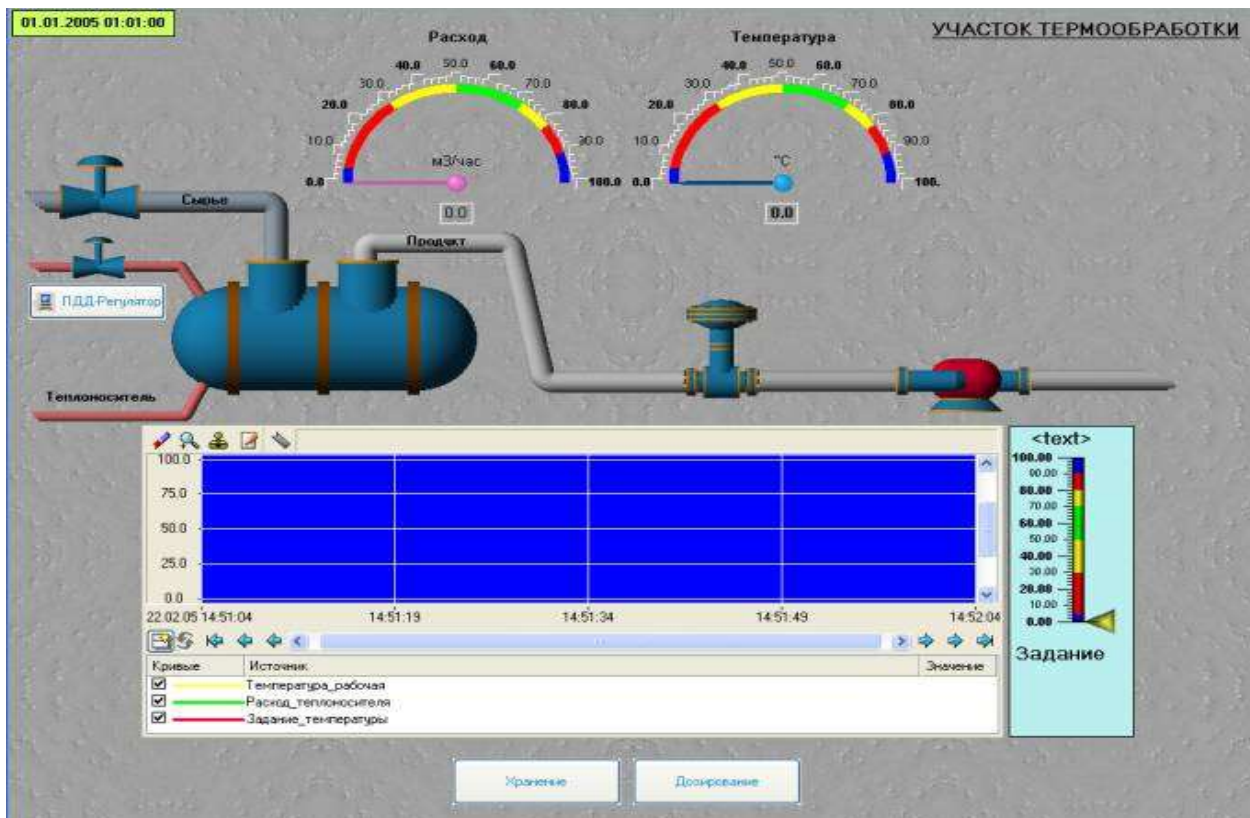
Подобным образом поступить для организации перехода на экран участка дозирования.

Параметры ПДД - регулятора – Кп, Кд, Кдд и зона нечувствительности будут формироваться с помощью всплывающего окна. Открытие данного окна оформить также с помощью ГЭ , который необходимо разместить в левой части экрана, привязку выполнить как в описанных выше случаях.

Для отображения в левом верхнем углу графического экрана текущей даты и времени можно воспользоваться ГЭ Календарь . Настройку ГЭ выполнить следующим образом.



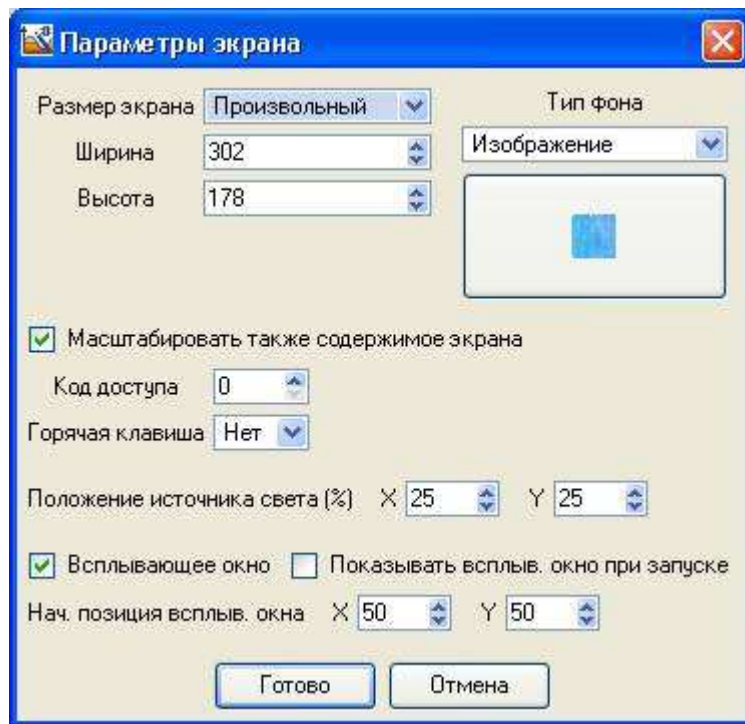
Таким образом, экран, представляющий на АРМ участок термической подготовки, подготовлен.








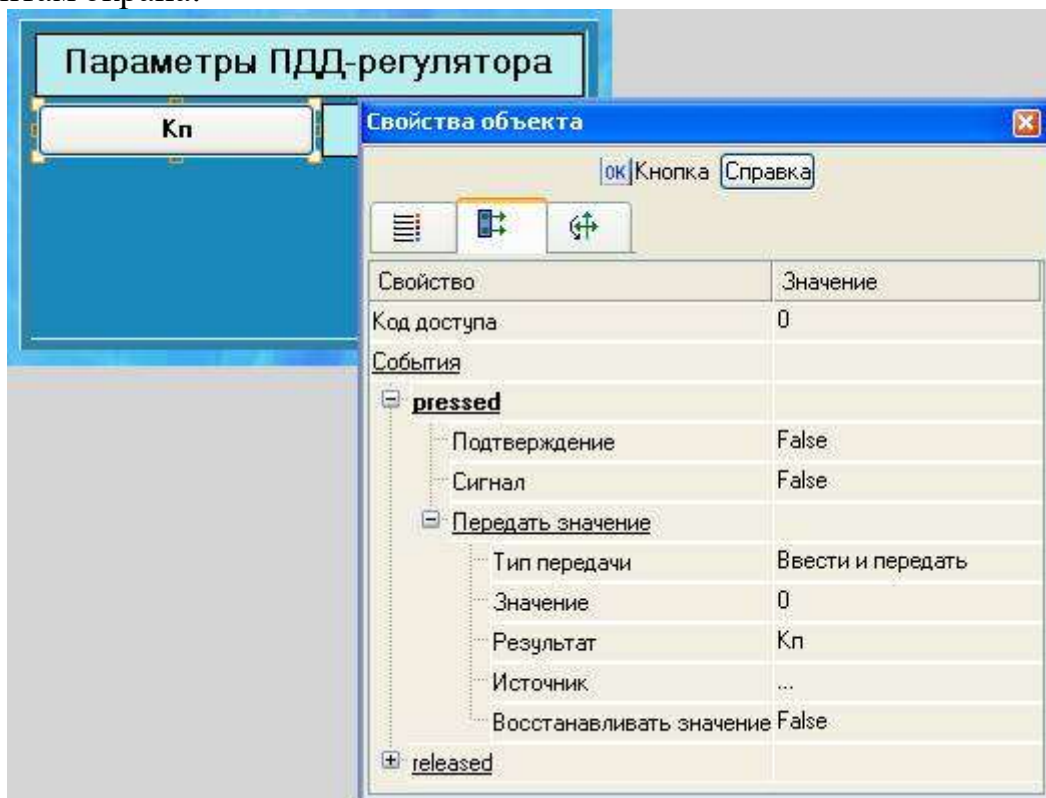
Аргументы экрана **Параметры_ПДД-регулятора** задать следующим образом.

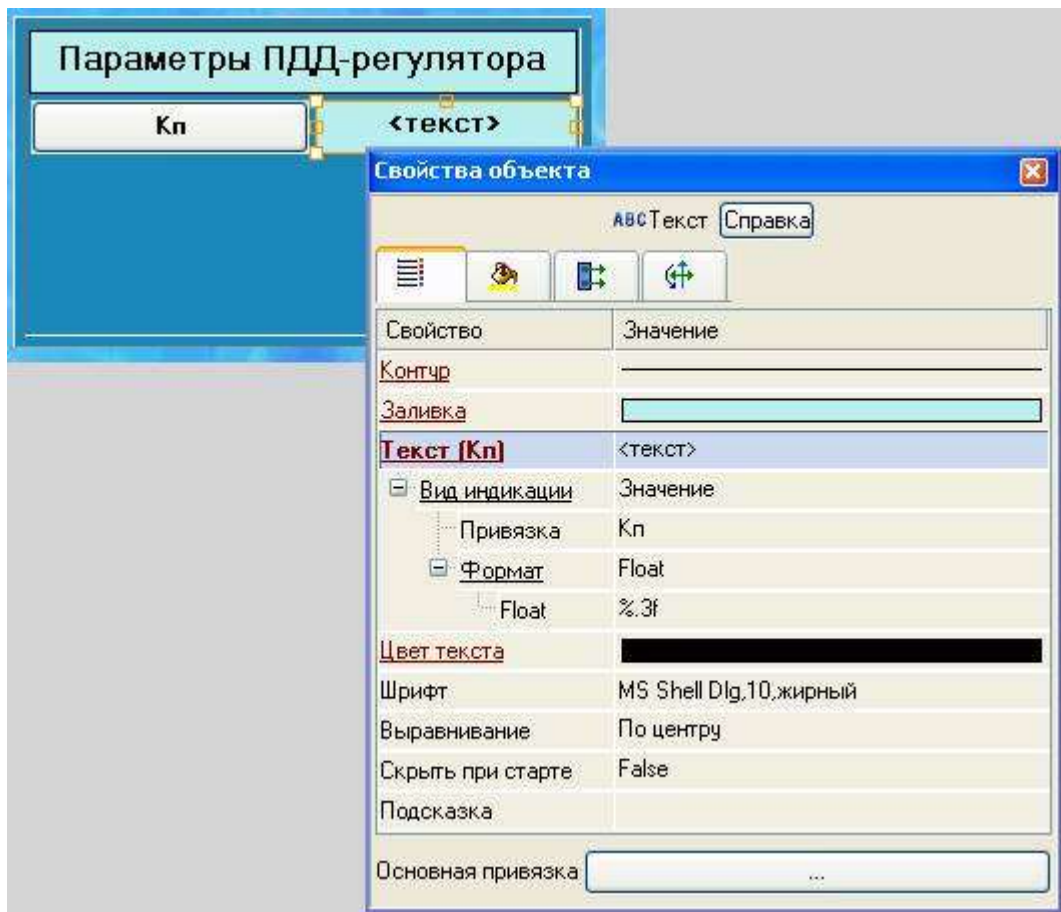
Информация		Аргументы			
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Кп	↑	OUT	REAL		
Кд	↑	OUT	REAL		
Кдд	↑	OUT	REAL		
Зона_нечувствительности	↑	OUT	REAL		


Открыть экран на редактирование. Для задания экрану свойств выпадающего окна выбрать в основном меню пункт **Сервис**, в нем – **Параметры экрана**. В открывшемся диалоговом окне задать размеры экрана, фон, определить экран как всплывающее окно и указать начальную позицию при первом вызове.

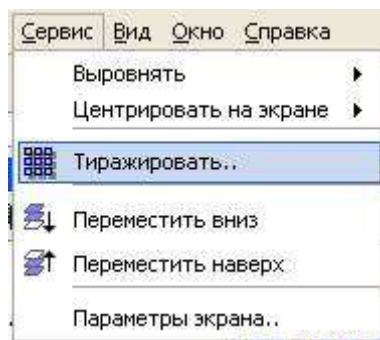


На данном экране разместить ГЭ Рамка , переместить его на задний план с помощью иконки  на панели инструментов, затем в верхней части экрана с помощью ГЭ Текст  задать заголовок экрана – Параметры ПДД-регулятора. Далее разместить ГЭ Кнопка  для посылки значений параметров и левее ее ГЭ Текст  для их отображения. Осуществить привязки ГЭ к аргументам экрана.

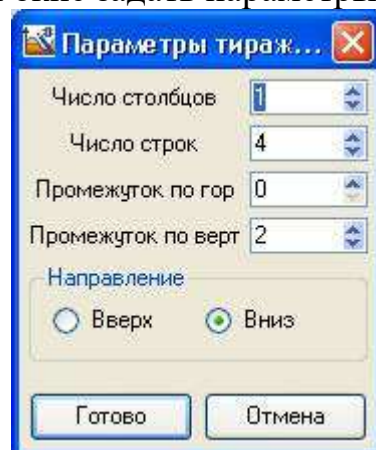





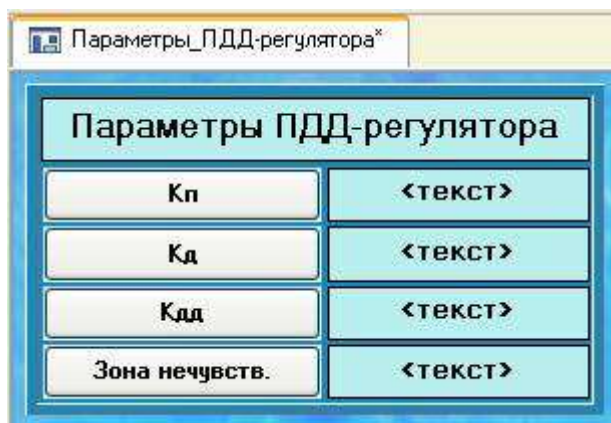
Затем выделить ЛК ГЭ  и воспользоваться инструментарием для тиражирования ГЭ.



В открывшемся диалоговом окне задать параметры.







Отредактировать надписи и привязки созданных ГЭ, подобным образом поступить в отношении ГЭ Текст .

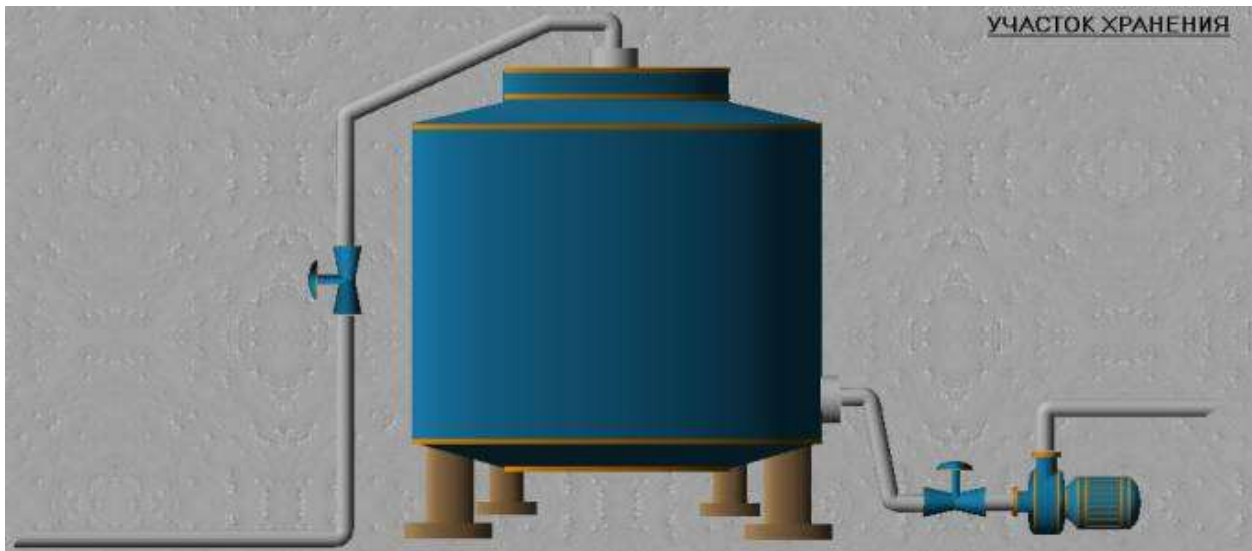


Аргументы для шаблона экрана **Участок_Хранения** будут следующие.

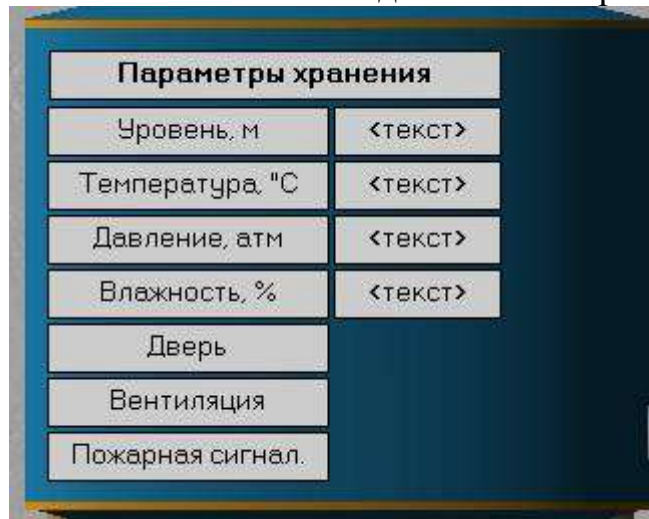
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Уровень	IN	REAL			
Температура	IN	REAL			
Давление	IN	REAL			
Влажность	IN	REAL			
Дверь	IN	USINT			
Вентиляция	IN	USINT			
Пож_сигнализация	IN	USINT			
Событие_Дверь	IN/OUT	REAL			NP
Событие_Вент	IN/OUT	REAL			NP
Событие_Пож	IN/OUT	REAL			NP
Слой_основной	IN	USINT	0		NP
Слой_тренд	IN	USINT	1		NP

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.

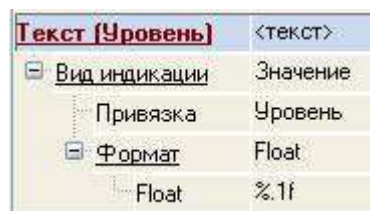
Задать для экрана в качестве фона изображение (одну из текстур), имеющихся в библиотеке, с помощью ГО  и ГЭ ,  выполнить статическую часть рисунка.




Определить с использованием ГЭ  вывод значений параметров хранения.




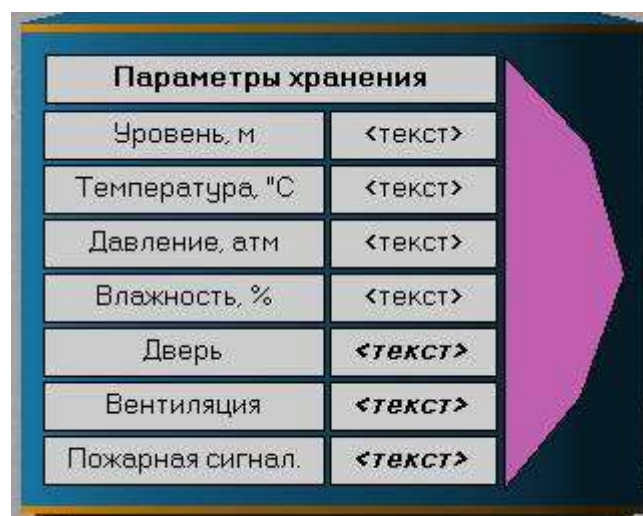
Выполнить привязку ГЭ к аргументам шаблона экрана, установить формат вывода значений, как, например, для аргумента **Уровень**.



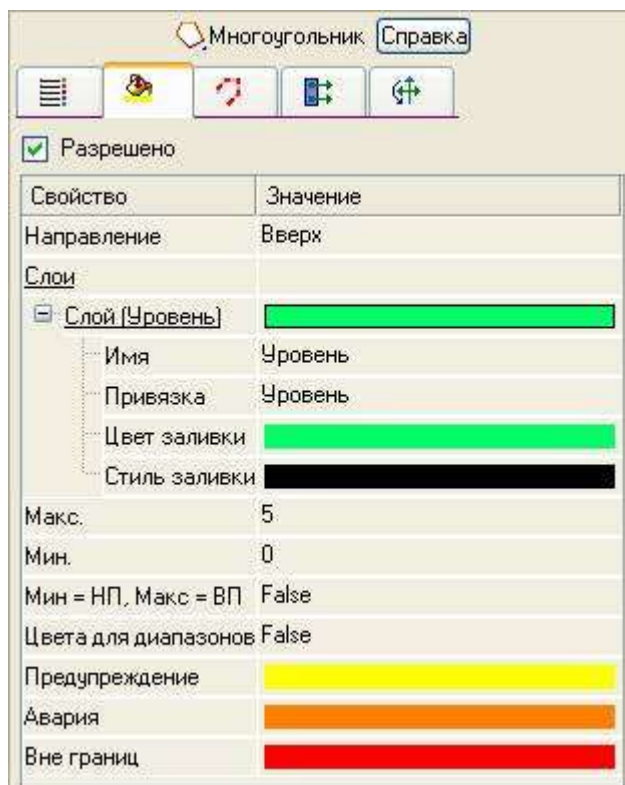
Для отображения состояния дискретных сигналов открытия/закрытия двери в хранилище, включения/отключения вентиляции и срабатывания пожарной сигнализации применить совместно цветовую и текстовую индикацию, определяемую для ГЭ . Так, для отображения текущего состояния двери в хранилище назначить ГЭ следующие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






Подобным образом выполнить настройку свойств для остальных ГЭ. Уровень продукта в емкости будет отображаться с помощью гистограммы произвольной формы, которую создадим с помощью ГЭ Многоугольник .

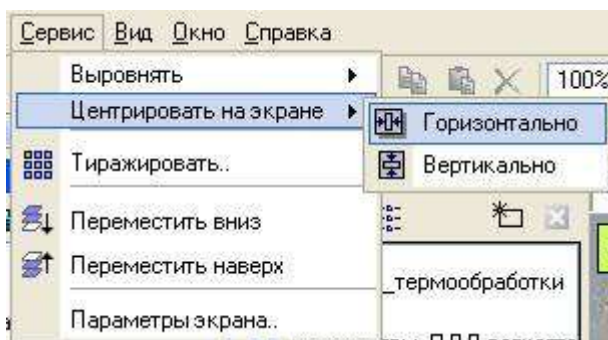



В свойствах для данного ГЭ определить динамическую заливку, привязав ее к соответствующему аргументу шаблона экрана, задав цвета фона и заполнения и указав границы.



Последовательно скопировать из шаблона экрана **Участок_термообработки** и перенести на шаблон **Участок_Хранения** ГЭ для отображения текущей даты/времени и кнопки переходов по экранам. Для этого выделить ЛК необходимый ГЭ (для выделения нескольких ГЭ можно воспользоваться нажатием и удержанием клавиши **Ctrl** либо выделить мышью область при нажатой ЛК) и использовать иконку  на панели инструментов, а для вставки - . Можно применять и стандартные комбинации клавиш для работы с буфером обмена: **Ctrl+C** и **Ctrl+V**. После вставки ГЭ необходимо произвести их перепривязку к аргументам текущего шаблона экрана.

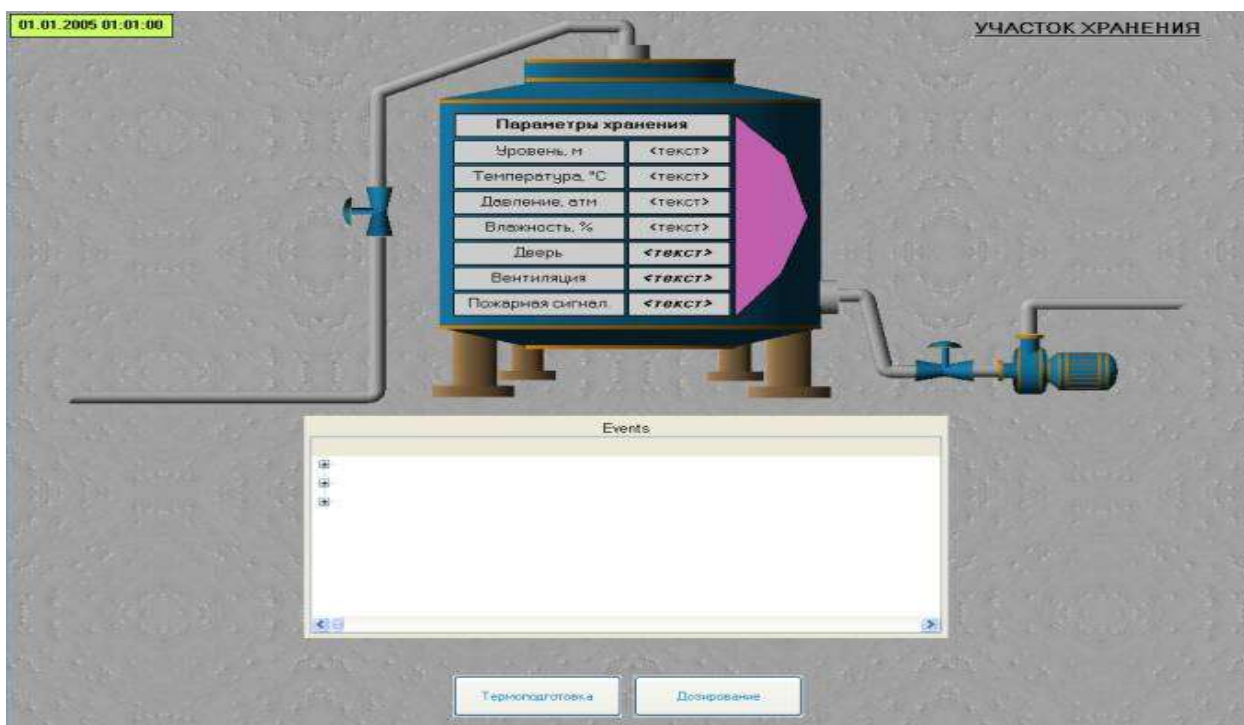
В нижней части экрана разместить ГЭ События , выделив его ЛК и отцентрировав горизонтально с помощью соответствующего пункта меню.



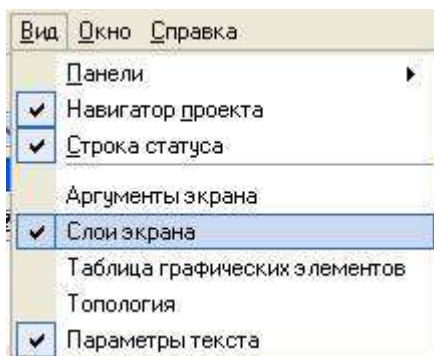
Основные свойства ГЭ  оставить без изменения, во вкладке **Привязки** определить следующие привязки к аргументам шаблона экрана.




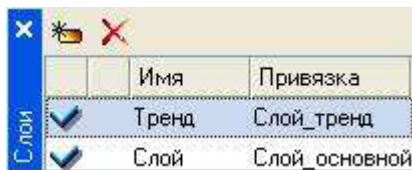
Таким образом, текущий экран будет иметь следующий вид.





Для вывода на тренде текущих значений параметров хранения создать дополнительный графический слой для шаблона экрана **Участок_Хранения**. Через пункт **Вид** основного меню открыть окно графических слоев.



В окне слоев экрана существующий слой с именем **Слой** привязать к аргументу шаблона **Слой_основной**. С помощью  создать новый слой, переименовать его в **Тренд** и привязать к аргументу **Слой_тренд**.




Разместить в данном слое (при этом в окне слоев он должен быть выделен ЛК) ГЭ Тренд  и Кнопка . Для тренда определим основные свойства, как показано ниже.



Далее задать семь кривых, указав для них привязки к соответствующим аргументам, цвет и толщину линий, пределы и заголовки для их идентификации на ГЭ.

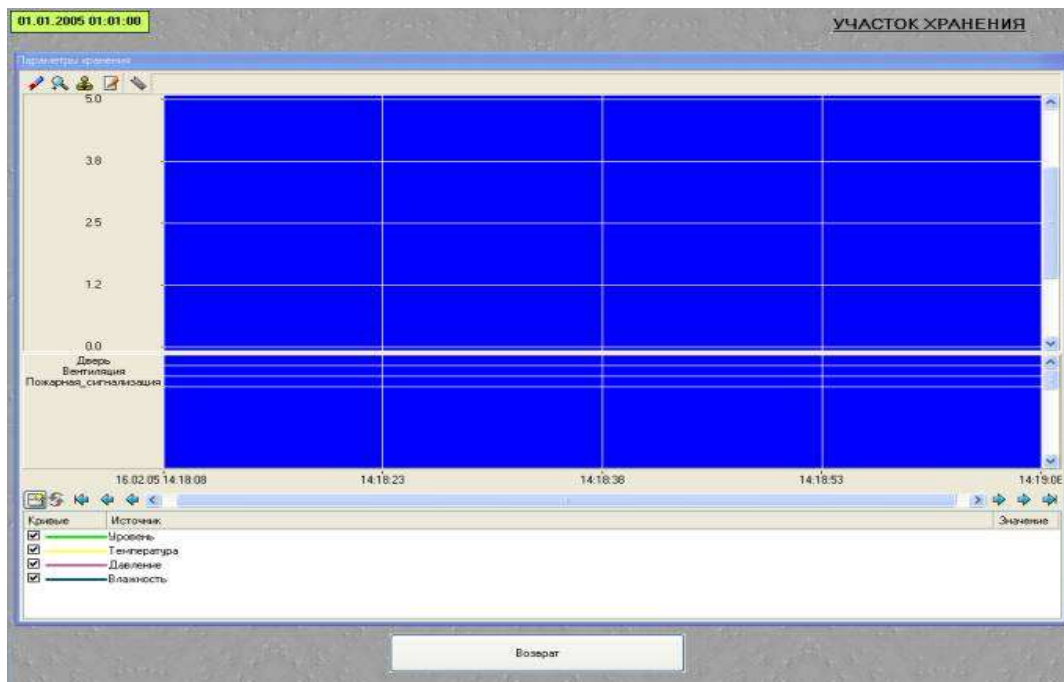



Свойство **События** для размещенного ниже ГЭ  **Возврат** определить таким образом, чтобы при нажатии на нем ЛК в аргументы шаблона экрана, для которых определены привязки к графическим слоям, осуществлялись прямые послылки. Значения, посылаемые в данные аргументы, управляют видимостью слоев, 0 – слой отображается, 1 (любое значение, отличное от 0) – нет.



Свойство	Значение
Код доступа	0
События	
pressed	
Подтверждение	False
Сигнал	False
Передать значение	
Тип передачи	Прямая
Значение	0
Результат	Слой_основной
Источник	...
Восстанавливать значение	False
Передать значение	
Тип передачи	Прямая
Значение	1
Результат	Слой_тренд
Источник	...
Восстанавливать значение	False
released	

ГЭ Календарь для вывода текущей даты/времени и ГЭ Текст с названием участка копируются в слой **Тренд** из слоя **Слой**. После выполнения указанных операций слой **Тренд** шаблона экрана **Участок_Хранения** будет выглядеть следующим образом.



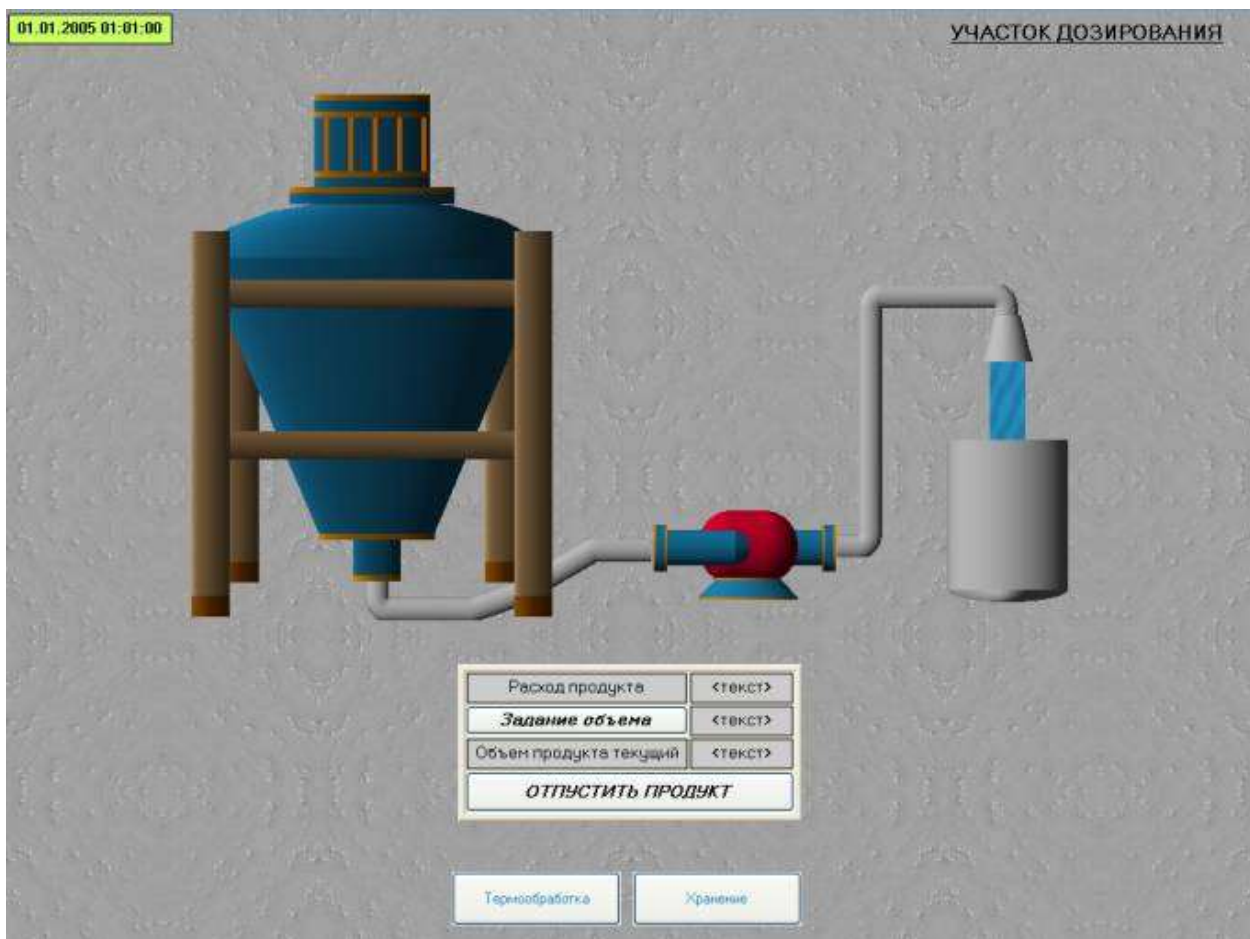
Для перехода к просмотру значений параметров хранения на тренде, размещенном в слое **Тренд**, в слое **Слой** определить свойства ГЭ  со статической надписью **Параметры хранения** следующим образом.

Свойство	Значение
Код доступа	0
События	
- mousePressed	
Подтверждение	False
Сигнал	False
- Передать значение	
Тип передачи	Прямая
Значение	1
Результат	Слой_основной
Источник	...
Восстанавливать значение	False
- Передать значение	
Тип передачи	Прямая
Значение	0
Результат	Слой_тренд
Источник	...
Восстанавливать значение	False
+ mouseReleased	
Основная привязка	...


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. Для шаблона экрана **Участок Дозирования** аргументы должны иметь вид.


Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Расход_продукта	↓ IN	REAL			
Задание_объема	↑ OUT	REAL			
Старт_Стоп	↑ OUT	USINT			
Объем_продукта_текущий	↓ IN	REAL			
Насос	↓ IN	USINT			
Сброс_объема	↑ OUT	USINT			NP

Используя описанные выше приемы, разместить ГЭ и связать их с аргументами шаблона экрана таким образом, чтобы получить следующее.



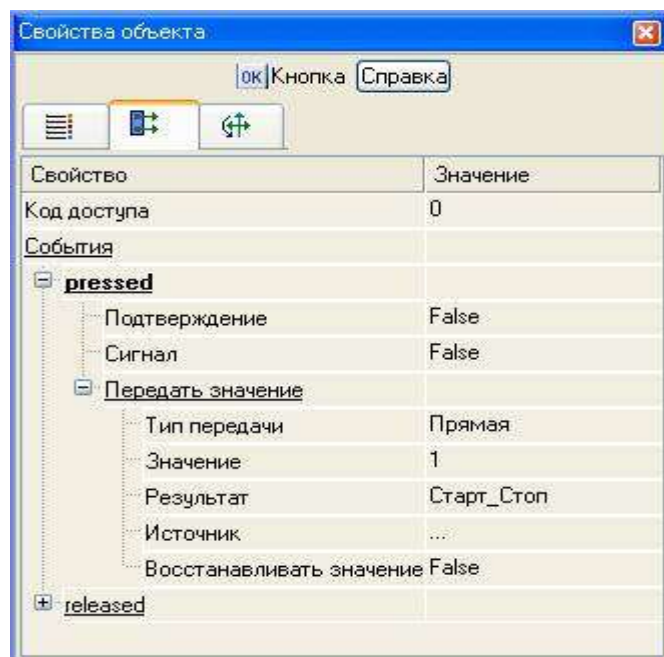
Видеоклип, изображающий поток продукта, привязан к аргументу экрана.


Свойство	Значение
<u>Видеоклип</u>	
Привязка	Насос
<u>Пауза</u>	0
Показывать при остановке	False
Непрерывное воспроизведение	True
Скрыть при старте	False
Подсказка	Поток продукта...

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

Свойство	Значение
Код доступа	0
События	
pressed	
Подтверждение	False
Сигнал	False
Передать значение	
Тип передачи	Ввести и передать
Значение	0
Результат	Задание_объема
Источник	...
Восстанавливать значение	False
Передать значение	
Тип передачи	Прямая
Значение	0
Результат	Сброс_объема
Источник	...
Восстанавливать значение	False
released	

Первое – собственно для задания величины отпускаемого объема продукта, второе – для сброса накопленной в контроллере величины объема по предыдущему циклу розлива. По нажатию ЛК на ГЭ **ОТПУСТИТЬ ПРОДУКТ** будет посылаться **1** в соответствующий аргумент экрана, в момент, когда заданный объем продукта будет отпущен, контроллер сбросит атрибут **Входное значение** канала в АРМ, связанный с аргументом экрана **Старт_Стоп** в **0**.



Сохранить выполненную работу, нажав ЛК . Таким образом, графическая часть проекта для АРМ выполнена.

5. Контрольные вопросы.

- 5.1. Как создать систему автоматизации путем проектирования «от шаблонов»?
- 5.2. Как использовать библиотеку изображений?
- 5.3. Как задать с помощью всплывающего окна параметры регулятора?

ЛАБОРАТОРНАЯ РАБОТА № 5

Написание программ на языках программирования ПЛК

1. Цель работы – освоение написания программ на языках программирования ПЛК.

2. Задание по лабораторной работе: создать шаблоны программ, реализующие управляющие функции – поддержания температуры и розлива продукта, а также вспомогательные, предназначенные для работы с дискретными сигналами.

3. Методические указания по выполнению лабораторной работы.

3.1. Провести программирование с использованием языка FBD (п. 4.1).

3.2. Откомпилировать созданную на языке FBD программу (п. 4.2).

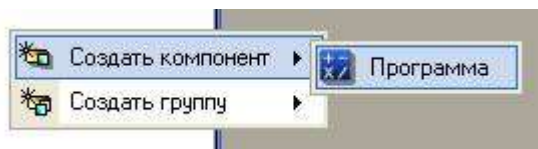
3.3. Провести программирование на языке ST (п. 4.3).

3.4. Написать программу для работы с дискретными сигналами (п. 4.4).

4. Процедура программирования ПЛК.

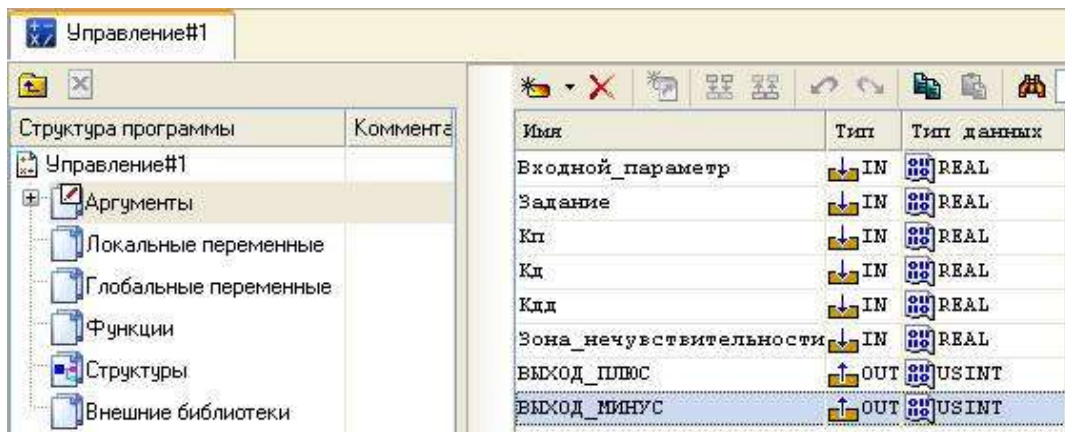
4.1. Написание программы с использованием языка FBD.

В левом окне навигатора проекта ЛК выбрать слой **Шаблоны_программ**, по щелчку ПК создать компонент Программа#1.




Выделить созданный компонент ЛК, изменить его имя на **Управление#1**, так как данная программа может быть создана для загрузки в PC-based контроллер с целью поддержания заданной температуры в аппарате на участке термообработки.

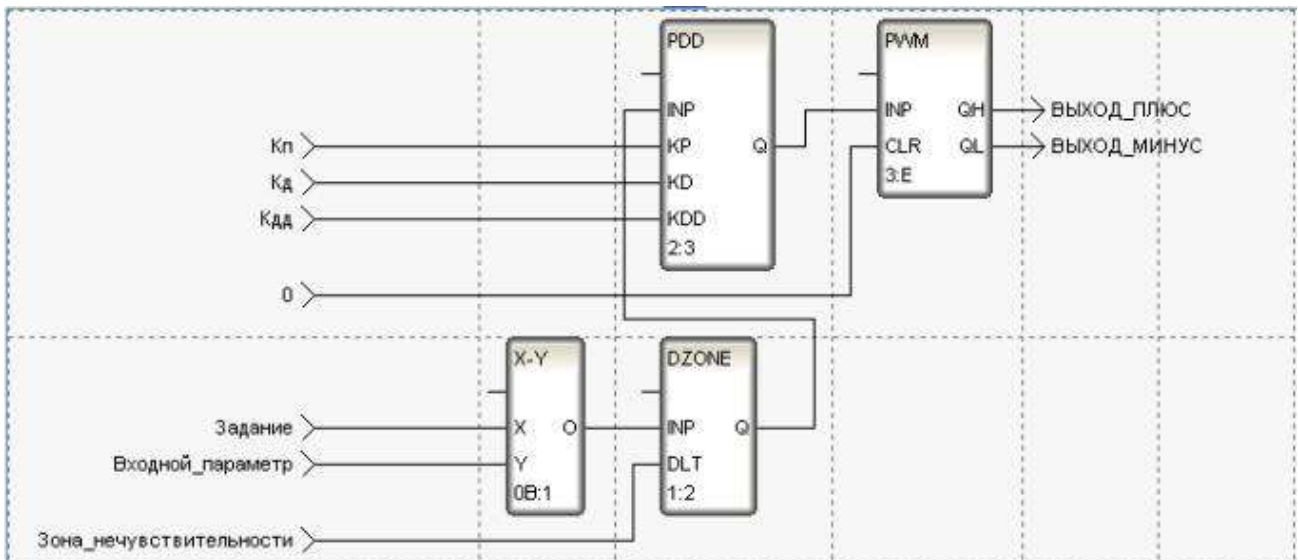
Двойным щелчком ЛК на компоненте **Управление#1** открыть окно редактора шаблонов программ и, выделив ЛК пункт **Аргументы**, перейти в табличный редактор аргументов. Создать аргументы для данного шаблона программы ПДД-регулятора с управлением исполнительным механизмом методом широтно-импульсной модуляции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





После определения входных и выходных аргументов приступить непосредственно к разработке программы. Для этого выделить ЛК имя созданного шаблона программы и в появившемся диалоге выбора языка программирования указать **FBD** диаграмму.

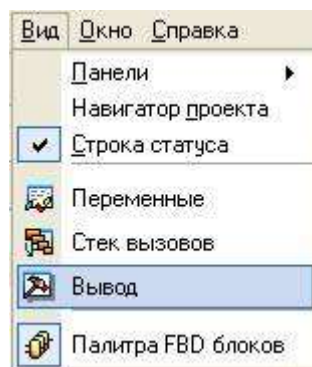


В открывшемся окне редактора программ выбрать ЛК иконку  для доступа к библиотекам функциональных блоков, далее выбирая ЛК необходимые блоки, перетащить их в рабочее поле редактора, сгруппировать, определить внутренние связи между входами и выходами блоков, назначить привязки к аргументам. Готовая программа выглядит следующим образом.

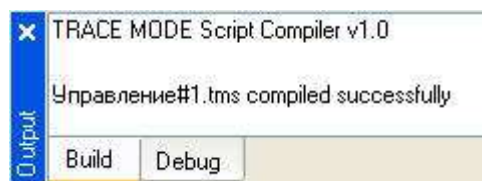


4.2. Компиляция программы на языке FBD.

Перед отладкой и включением в состав проекта разработанный шаблон программы необходимо скомпилировать. Для этого используется иконка  на панели инструментов или нажатие функциональной клавиши **F7**. Результат компиляции показывается в окне **Вывод**, которое может быть открыто с помощью иконки  на панели инструментов либо из основного меню интегрированной среды разработки.

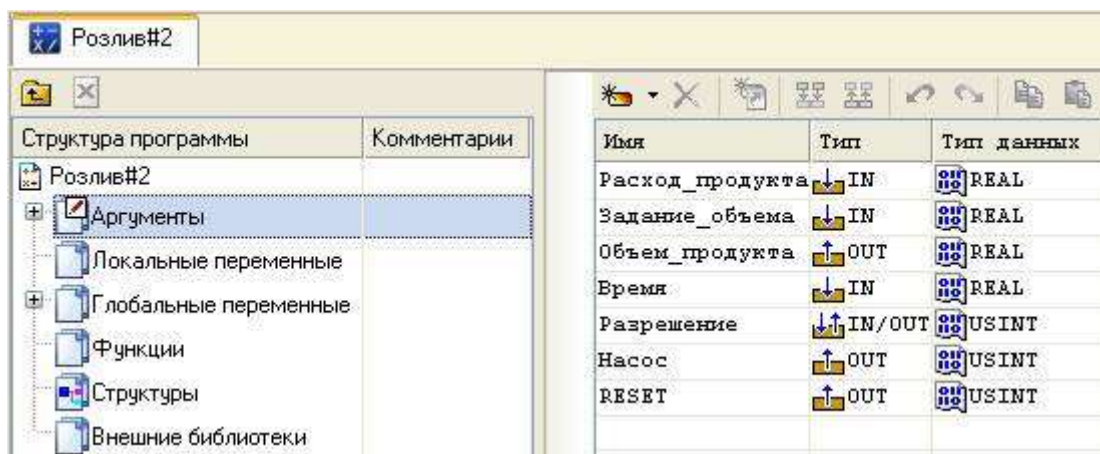


В нашем случае данное окно содержит сообщение об успешном окончании процесса компиляции.

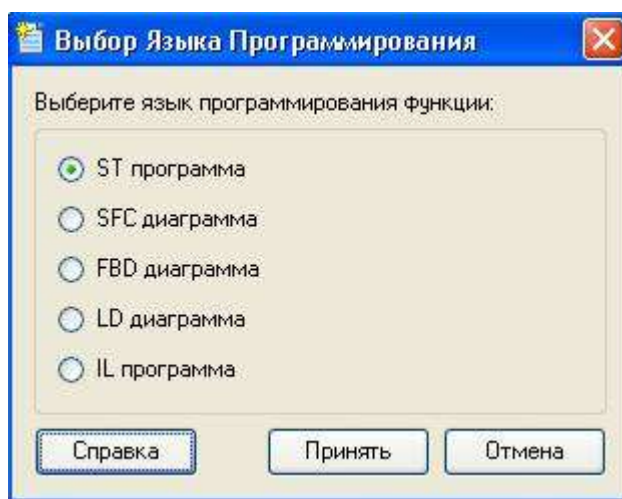


4.3. Программирование на языке ST.

Для процесса дозирования продукта создать программу **Розлив#2**, задав для нее следующие аргументы.



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



Текст программы может выглядеть следующим образом.

```

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
VAR_OUTPUT RESET : USINT; END_VAR

//Подготовка к процессу
if Разрешение==0 then
    //Удержание насоса выключенным
    Насос=0;
end_if;
//Проверка разрешения от АРМ ( Старт_Стоп = 1 ? )
if Разрешение==1 then
    //Включение насоса и удержание Старт_Стоп на АРМ
    Насос=1;
    RESET=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 * 0.02442 * (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 Старт_Стоп на экране АРМ
    Насос=0;
    RESET=0;
    //Запрет дальнейшего расчета объема
    Разрешение=0;
    //Сброс аккумулятора
    VOLUME=0;
end_if;

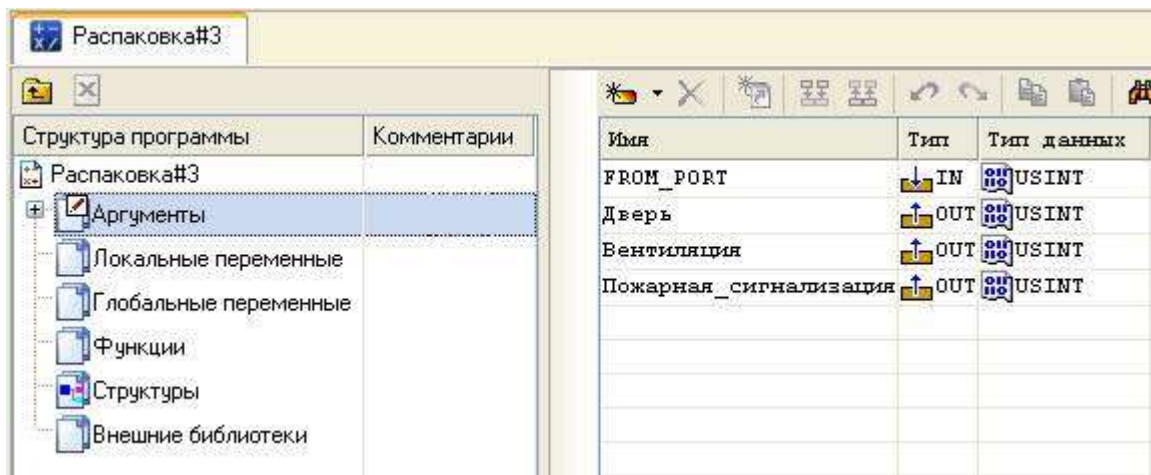
END_PROGRAM

```

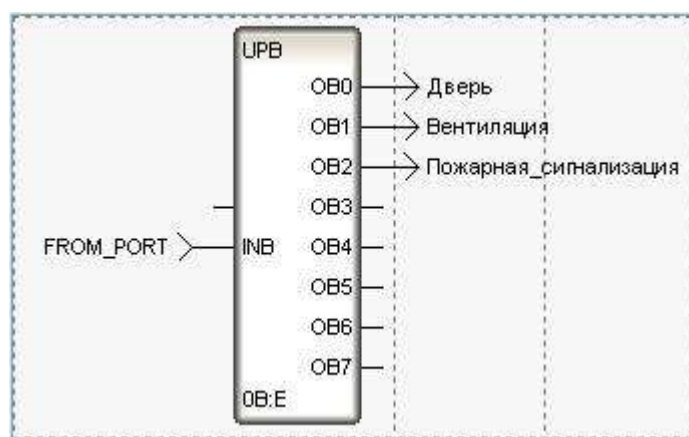
Откомпилируем программу для включения в проект нажатием **F7**.

4.4. Программа для работы с дискретными сигналами.

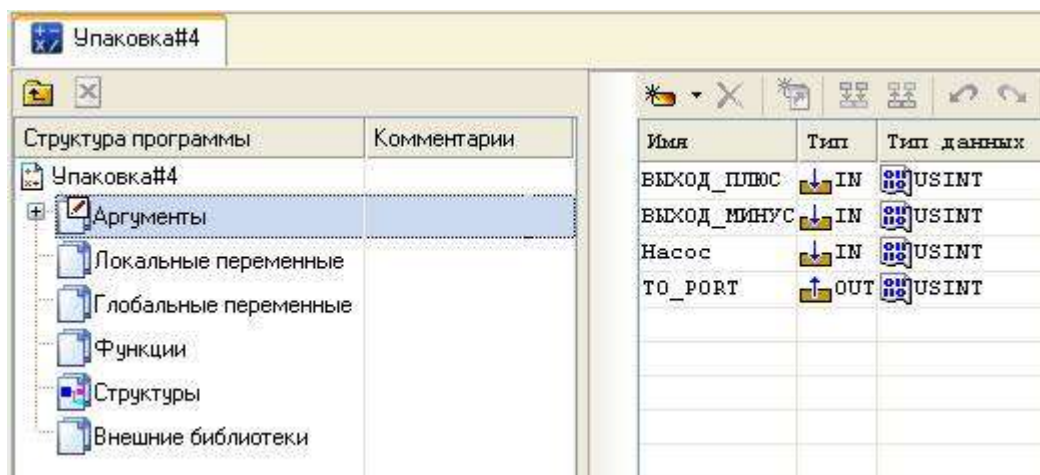
Для работы с входными и выходными дискретными сигналами необходимо разработать на **Техно FBD** программы **Распаковка#3** распаковки получаемого от контроллера PLC байта данных и **Упаковка#4** упаковки битов управления для PC-based контроллера в байт. Для шаблона программы **Распаковка#3** аргументы зададим, как указано ниже.

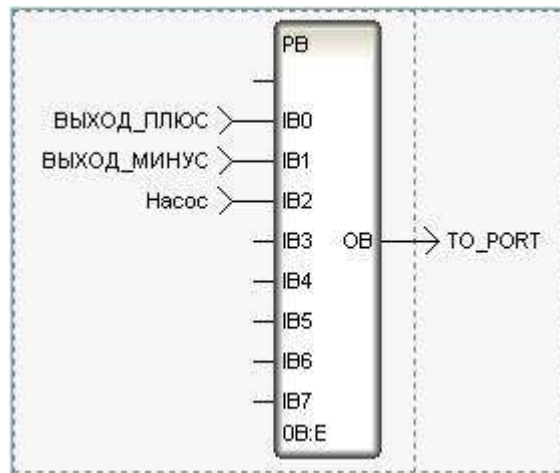


Программа будет иметь вид.



Соответственно для шаблона программы **Упаковка#4** аргументы и программа будут следующими.





Компиляция производится по нажатию функциональной клавиши **F7**.

Сохранить выполненную работу, нажав ЛК .

5. Контрольные вопросы.

5.1. Как использовать окно редактора шаблонов программ?

5.2. Как использовать функциональные блоки библиотеки?

5.3. Какова последовательность компиляции программы?

ЛАБОРАТОРНАЯ РАБОТА № 6

Создание узлов проекта и баз каналов

1. Цель работы – освоение принципов создания узлов проекта с использованием механизма автопостроения.

2. Задание по лабораторной работе: создать узлы проекта – АРМ и РС-based контроллера, для которых в дальнейшем будут формироваться базы каналов, используя механизм автопостроения.

3. Методические указания по выполнению лабораторной работы.

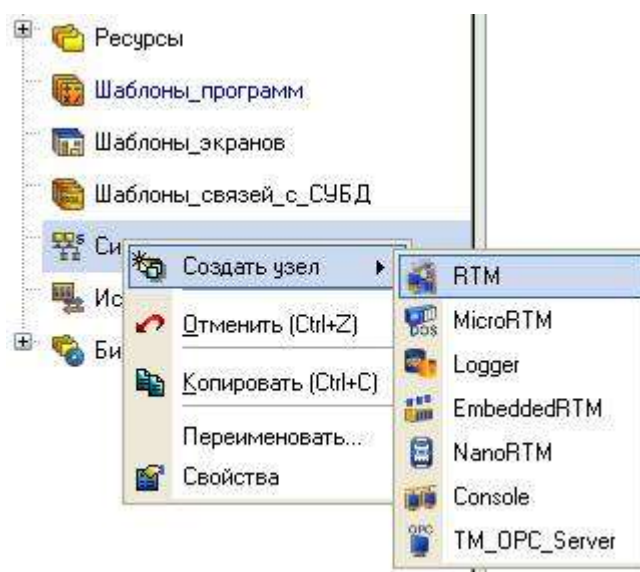
3.1. Разработать шаблоны экранов и программ (п. 4).

3.2. Создать узлы проекта – АРМ и РС-based контроллера (п. 4).

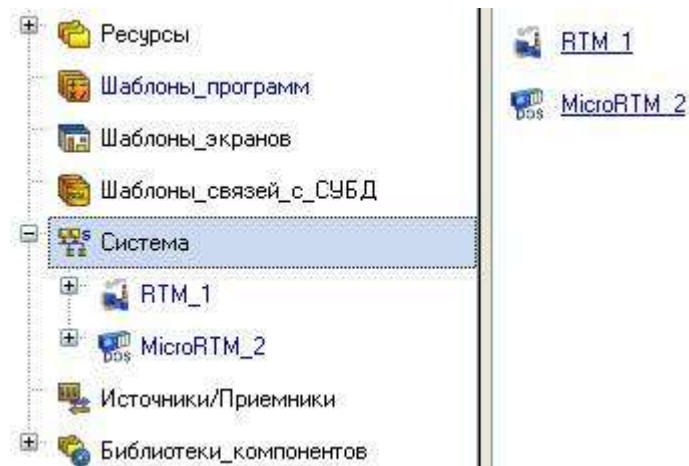
3.3. Сформировать базы каналов, используя механизм автопостроения (п. 4).


4. Процедура построения узлов проекта и баз каналов.

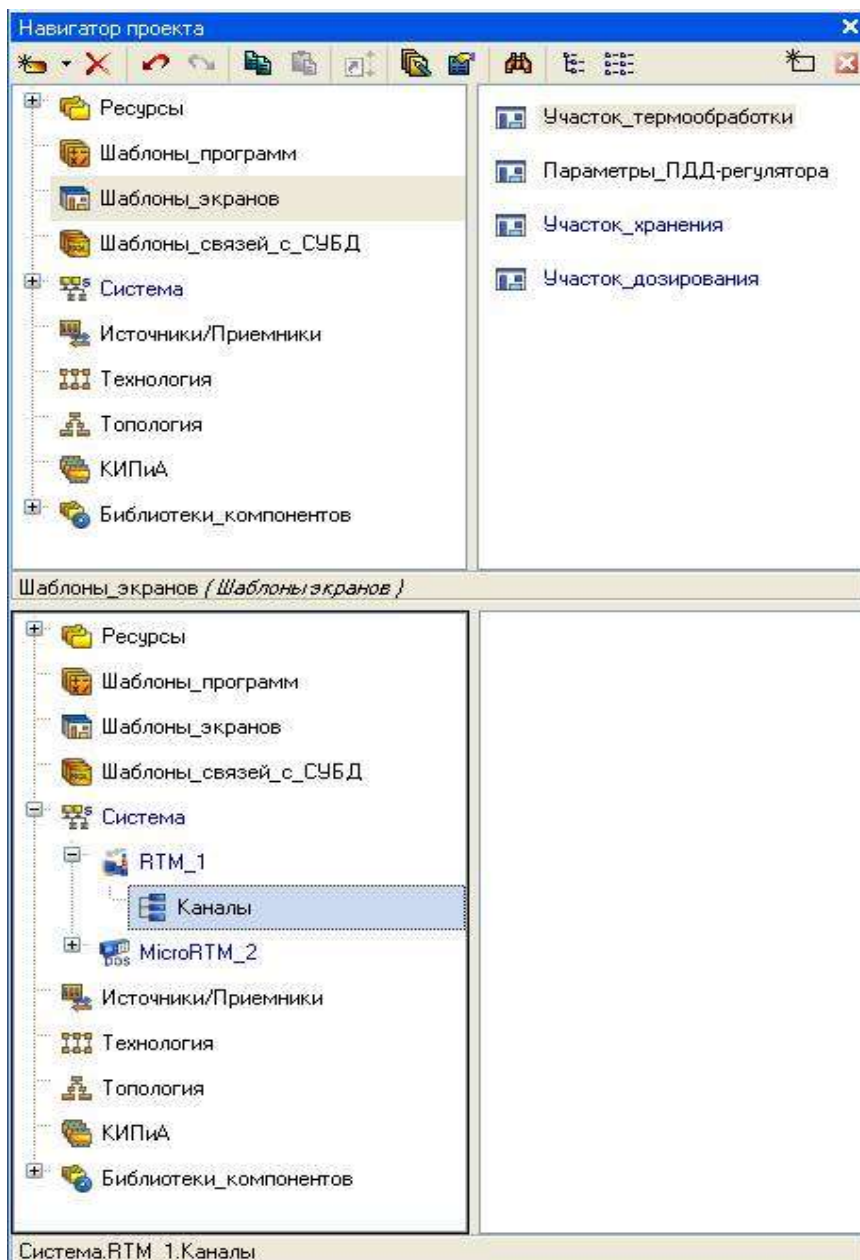
Произвести выбор слоя Система щелчком ЛК мыши. Далее с помощью ПК создать узлы МРВ для АРМ и МикроМРВ для РС-based контроллера.



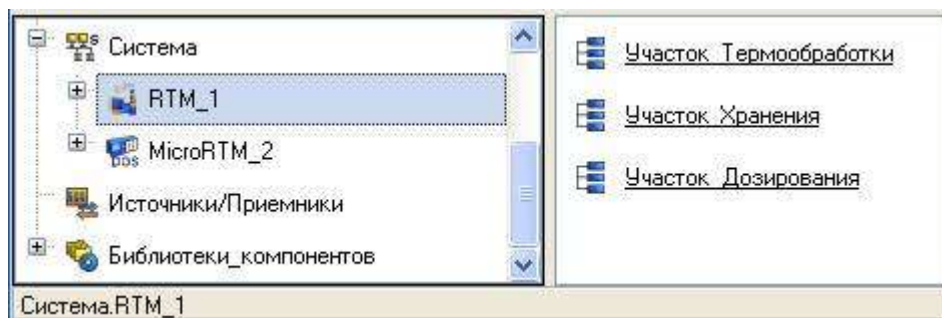
В результате выполненных действий в слое Система будут созданы два узла проекта.



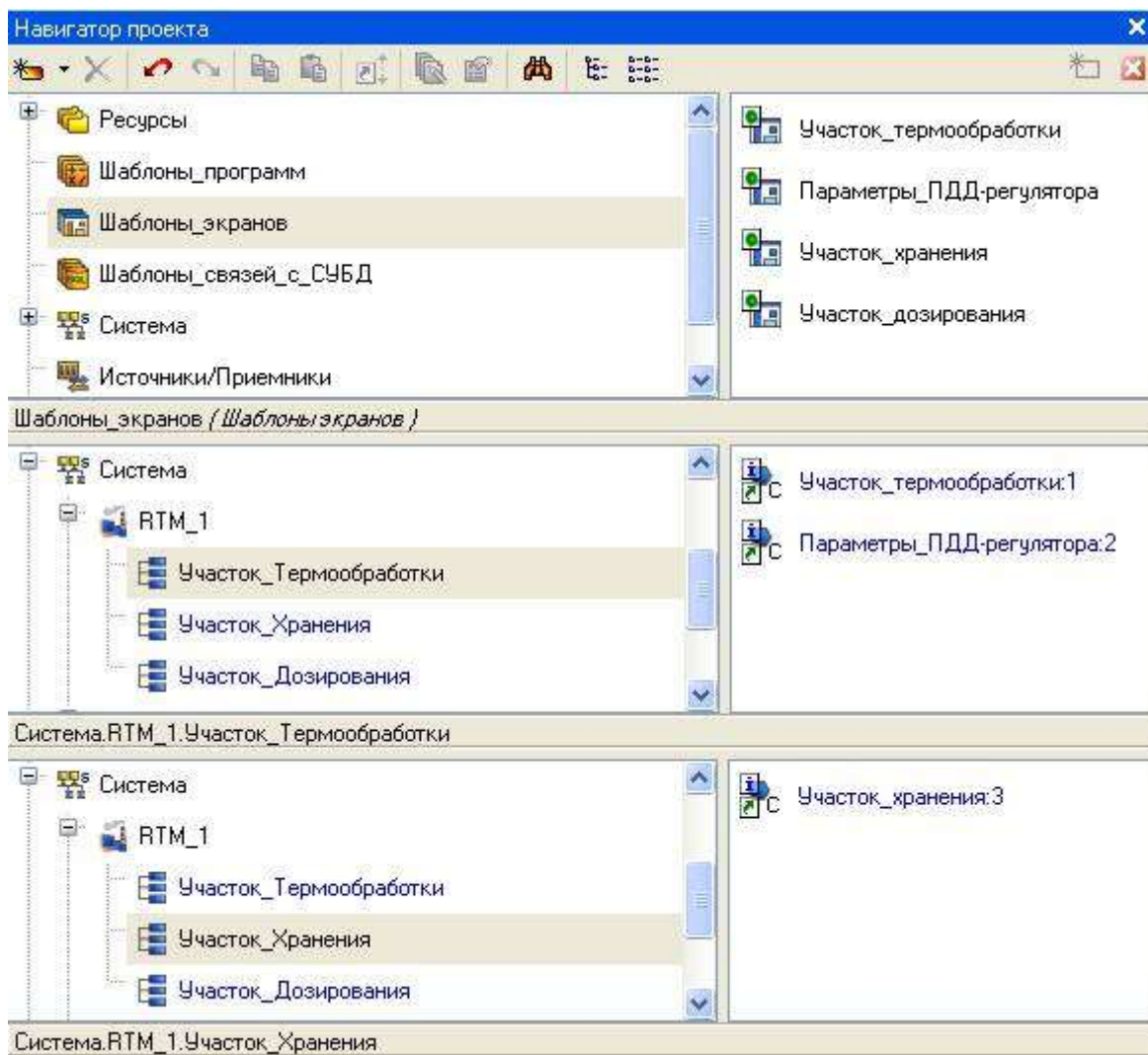
С помощью иконки  создать дополнительное окно Навигатора проекта и открыть в верхнем слой **Шаблоны_экранов**, а в нижнем – группу компонентов **Каналы** вновь созданного узла АРМ **RTM_1**.



В узле **RTM_1** изменить название группы **Каналы** на **Участок_Термообработки** и создать группы **Участок_Хранения** и **Участок_Дозирования**.

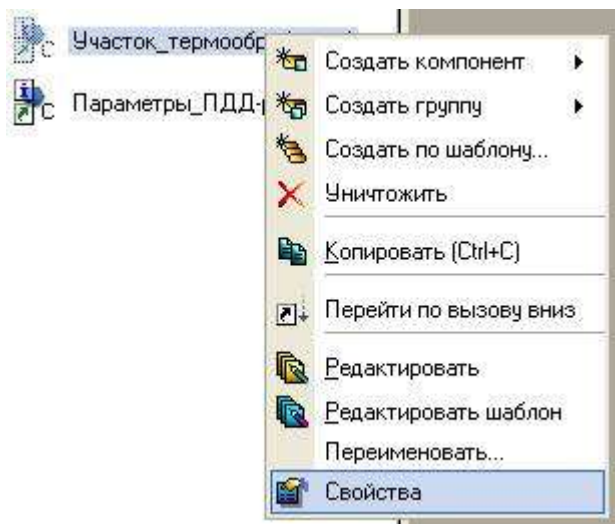



Затем, выделяя ЛК шаблоны экранов и удерживая ЛК, перетащить их в группы узла **RTM_1**.

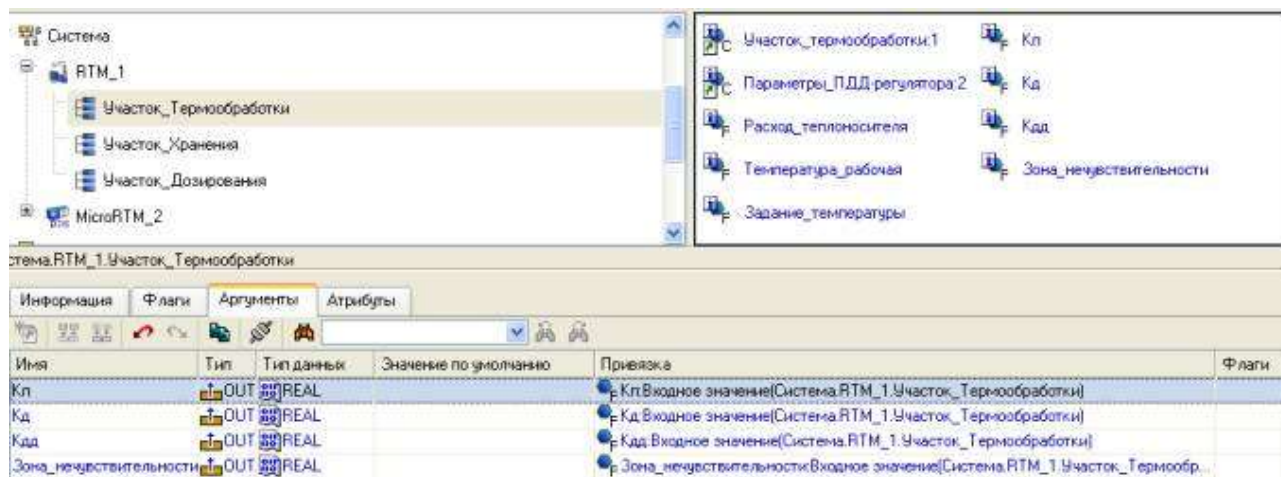
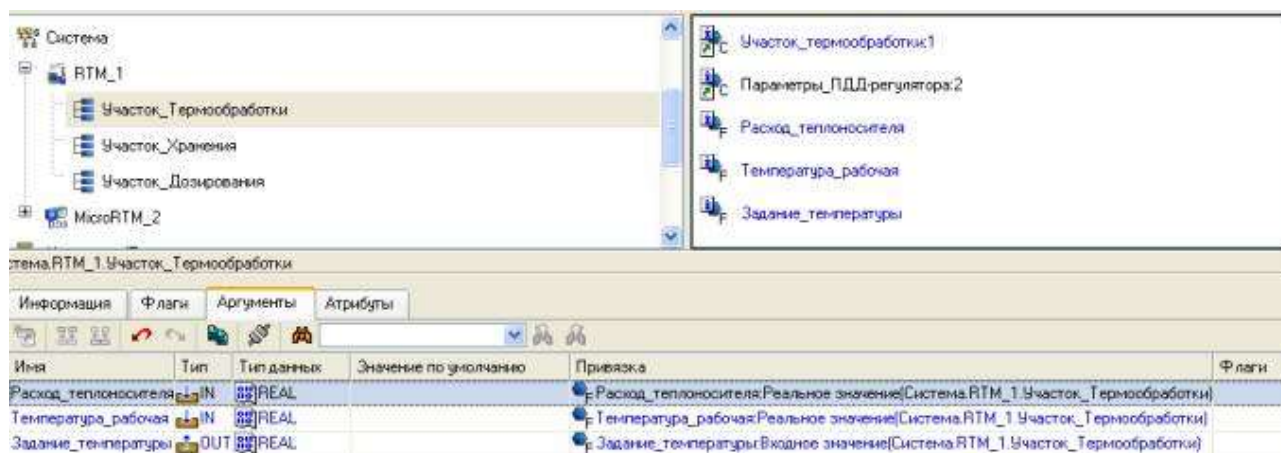


На следующем шаге создать каналы по аргументам разработанных шаблонов экранов. Для этого войти в группу каналов АРМ – узла **RTM_1**

Участок_Термообработки и вызвать свойства канала класса **Вызов Участок_термообработки:1**.



Перейти во вкладку **Аргументы**, выделить ЛК первый аргумент и с помощью щелчка ЛК мыши на иконке  создать каналы в выбранной группе и автоматически связать их атрибуты с аргументами шаблона экрана.



Подобным же образом создать каналы в группах **Участок_Хранения** и **Участок_Дозирования**. Предварительно для канала класса **Вызов Участок_Хранения** задать аргументам **Слой_основной** и **Слой_тренд** значения по умолчанию – соответственно **0** и **1**. После выполнения процедуры автопостроения получится следующее.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Уровень	IN	REAL		Уровень:Реальное значение(Система.RTM_1.Участок_Хранения)	
Температура	IN	REAL		Температура:Реальное значение(Система.RTM_1.Участок_Хранения)	
Давление	IN	REAL		Давление:Реальное значение(Система.RTM_1.Участок_Хранения)	
Влажность	IN	REAL		Влажность:Реальное значение(Система.RTM_1.Участок_Хранения)	
Дверь	IN	USINT		Дверь:Реальное значение(Система.RTM_1.Участок_Хранения)	
Вентиляция	IN	USINT		Вентиляция:Реальное значение(Система.RTM_1.Участок_Хранения)	
Пож_сигнализация	IN	USINT		Пож_сигнализация:Реальное значение(Система.RTM_1.Участок_Хранения)	
Событие_Дверь	IN/OUT	REAL			NP
Событие_Вент	IN/OUT	REAL			NP
Событие_Пож	IN/OUT	REAL			NP
Слой_основной	IN	USINT	0		NP
Слой_тренд	IN	USINT	1		NP

Для группы **Участок_Дозирования**.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Расход_продукта	IN	REAL		Расход_продукта:Реальное значение(Система.RTM_1.Участок_Дозирования)	
Задание_объема	OUT	REAL		Задание_объема:Входное значение(Система.RTM_1.Участок_Дозирования)	
Старт_Стоп	OUT	USINT		Старт_Стоп:Входное значение(Система.RTM_1.Участок_Дозирования)	
Объем_продукта_текущий	IN	REAL		Объем_продукта_текущий:Реальное значение(Система.RTM_1.Участок_Дозиров...	
Насос	IN	USINT		Насос:Реальное значение(Система.RTM_1.Участок_Дозирования)	
Сброс_объема	OUT	USINT			NP

5. Контрольные вопросы.

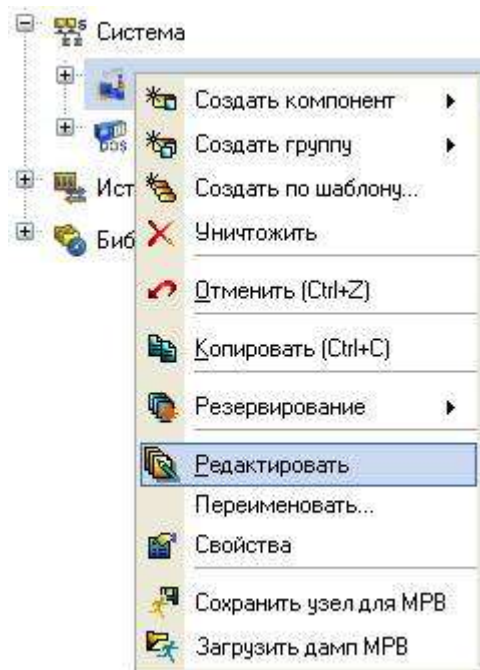
5.1. Какие действия необходимо выполнять со слоем Система?

5.2. Как создать каналы по аргументам разработанных шаблонов экрана?

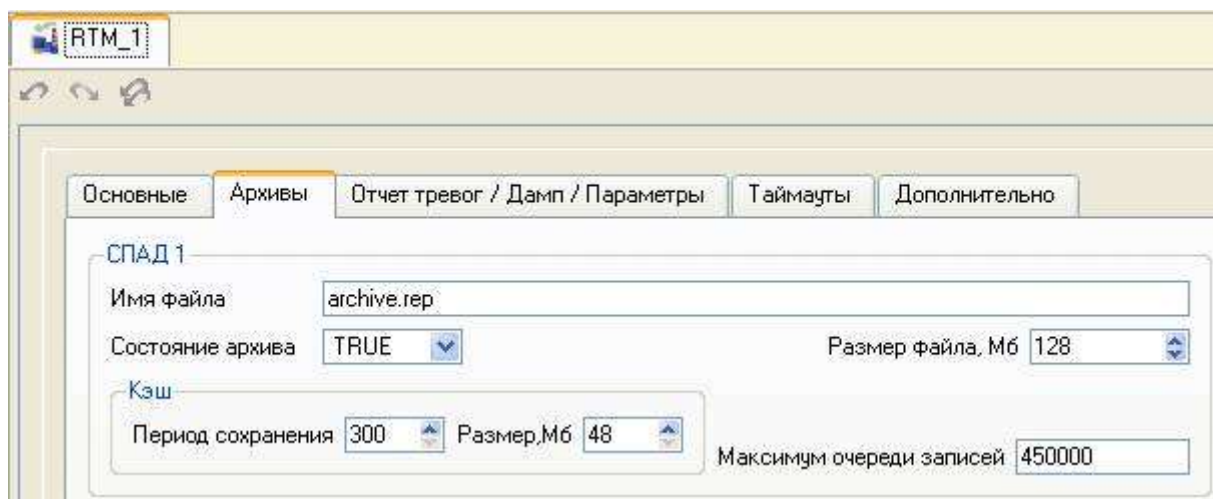
ЛАБОРАТОРНАЯ РАБОТА №7

Создание архива и отчета тревог

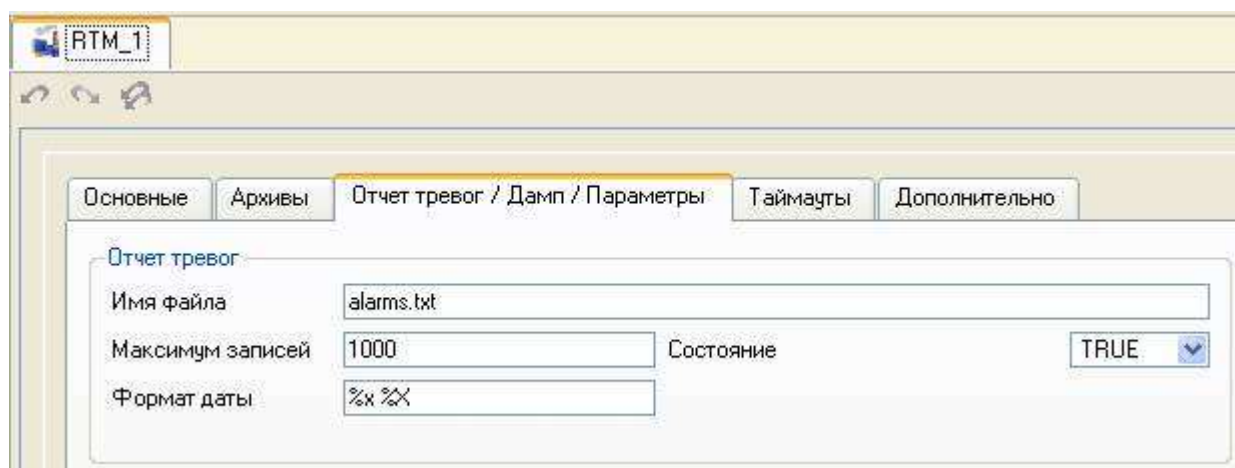
1. Цель работы – освоение принципов создания архива и отчета тревог.
2. Задание по лабораторной работе: изучить сохранение реальных значений каналов в архив **SIAD/SQL 6** и ведения отчета тревог по таким событиям, как пересечение заданных для канала уставок и границ.
3. Методические указания по выполнению лабораторной работы.
 - 3.1. Сформировать архив и отчет тревог в соответствии с п. 4.
 - 3.2. Разработать шаблоны экранов и программ (п. 4).
4. Процедура создания архива и отчета тревог.
Вызвать на редактирование узел **RTM_1**, определить файлы архива, отчета тревог и задать настройки для сетевого обмена.



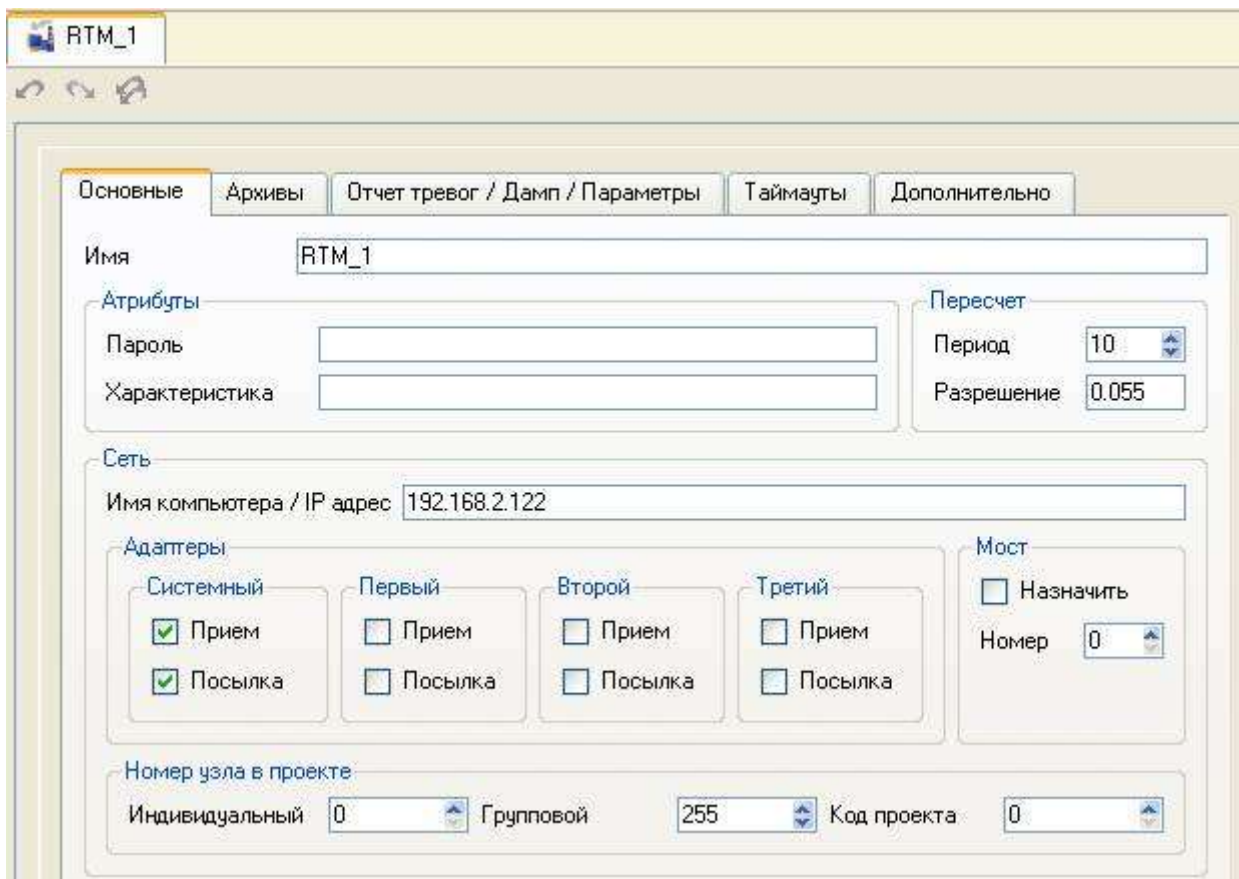
Определить архивирование каналов, связанных с технологическими параметрами, в архив **СПАД 1**. Для этого перейти во вкладку **Архивы** и отредактировать бланк **СПАД 1** следующим образом.



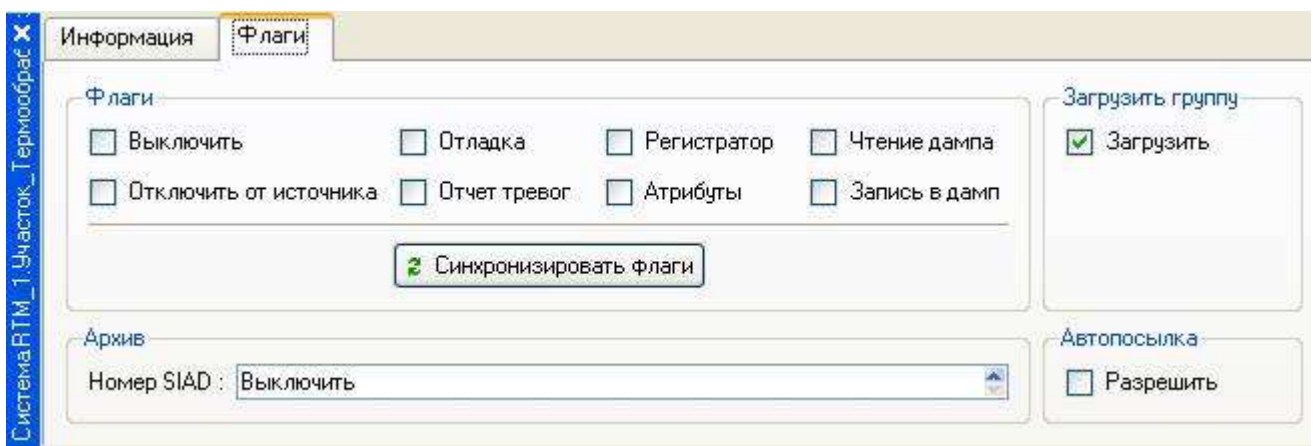
Во вкладке **Отчет тревог/Дамп/Параметры** определить следующие параметры.



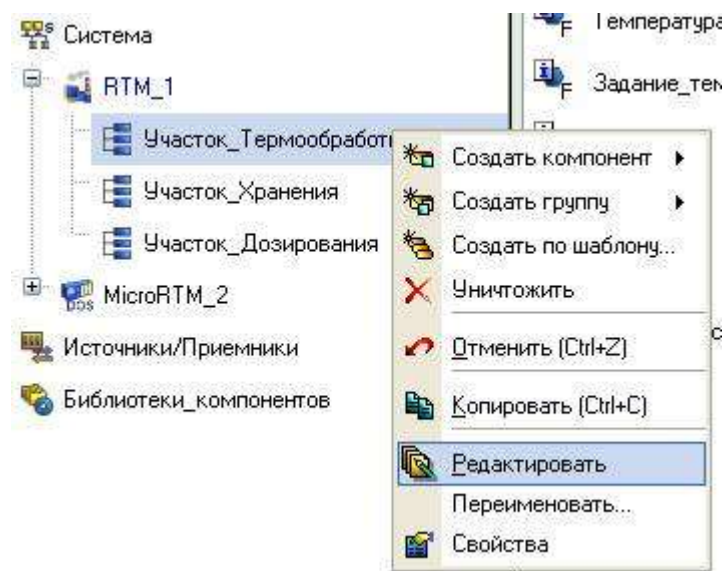
В бланке **Основные** задать IP-адрес АРМ и направления обмена данными по сети. Следует иметь в виду, что IP-адрес в бланке можно не задавать ввиду того, что при запуске отладчик или МРВ самостоятельно определяют данный параметр узла и используют его при сетевом обмене.



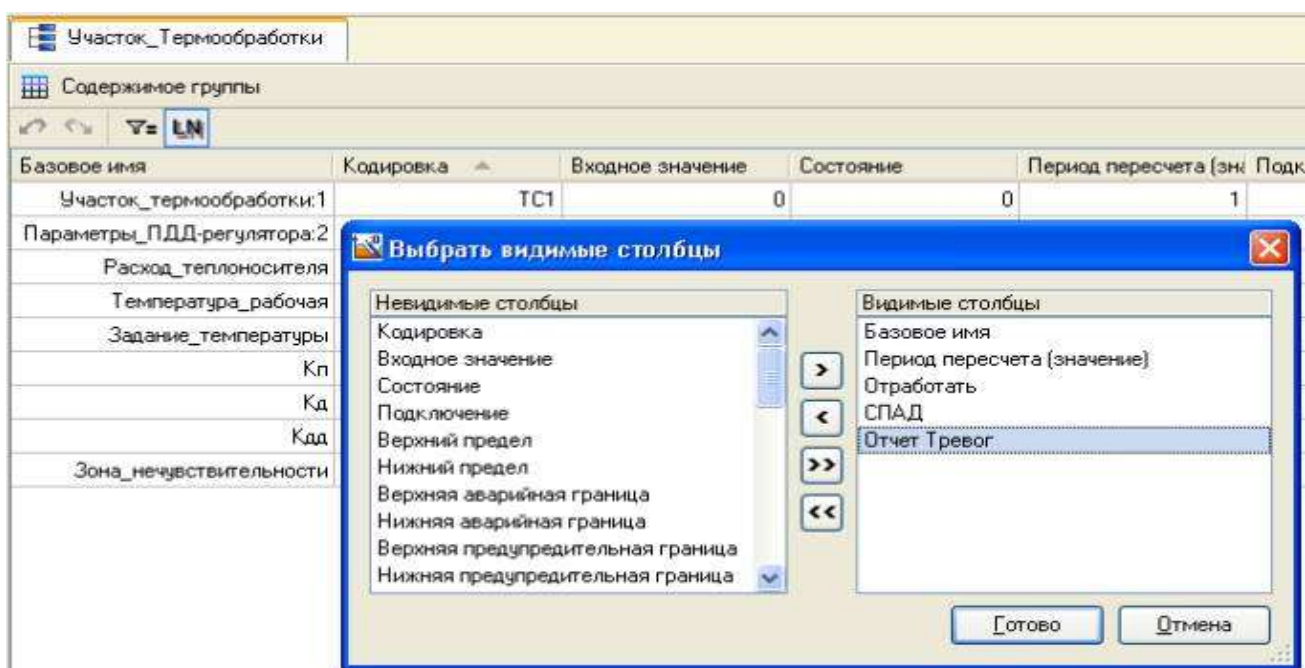
Закрывать бланк редактирования узла **RTM_1**. Для получения возможности просмотра в отладчике сведенных в группы каналов открыть свойства группы компонентов **Участок_Термообработки** и установить во вкладке **Флаги** флаг **Загрузить группу**. Подобным образом поступить в отношении других групп.



Для помещения каналов группы **Участок_Термообработки** в архив и отчет тревог вызвать на редактирование данную группу.



В открывшемся окне щелчком ПК на строке атрибутов каналов вызвать настроечное меню отображения атрибутов и в правом окне сформировать список отображаемых в таблице атрибутов.



Выделить с помощью ЛК (не отпуская ее) группу каналов и, удерживая при этом клавишу **Ctrl**, двойным щелчком ЛК на выделенных каналах в столбце **СПАД** задать значение **1**.

Участок_Термообработки					
Содержимое группы					
Базовое имя	Период пересчета (зн)	Отработать	СПАД	Отчет Тревог	
Участок_термообработки:1	1	0	No	0	
Параметры_ПДД-регулятора:2	1	0	No	0	
Расход_теплоносителя	1	0	No	0	
Температура_рабочая	1	0	No	0	
Задание_температуры	1	1	No	0	
Кп	1	1	1	0	
Кд	1	1	2	0	
Кдд	1	1	3	0	
Зона_нечувствительности	1	1	No	0	

В результате для выделенных каналов будет определена запись их реальных значений в архив СПАД 1. Для группы **Участок_Термообработки** атрибуты будут установлены следующим образом.

Участок_Термообработки					
Содержимое группы					
Базовое имя	Период пересчета (зн)	Отработать	СПАД	Отчет Тревог	
Участок_термообработки:1	1	0	No	0	
Параметры_ПДД-регулятора:2	1	0	No	0	
Расход_теплоносителя	1	0	1	0	
Температура_рабочая	1	0	1	0	
Задание_температуры	1	1	1	0	
Кп	1	1	1	0	
Кд	1	1	1	0	
Кдд	1	1	1	0	
Зона_нечувствительности	1	1	1	0	

Для группы **Участок_Хранения**.

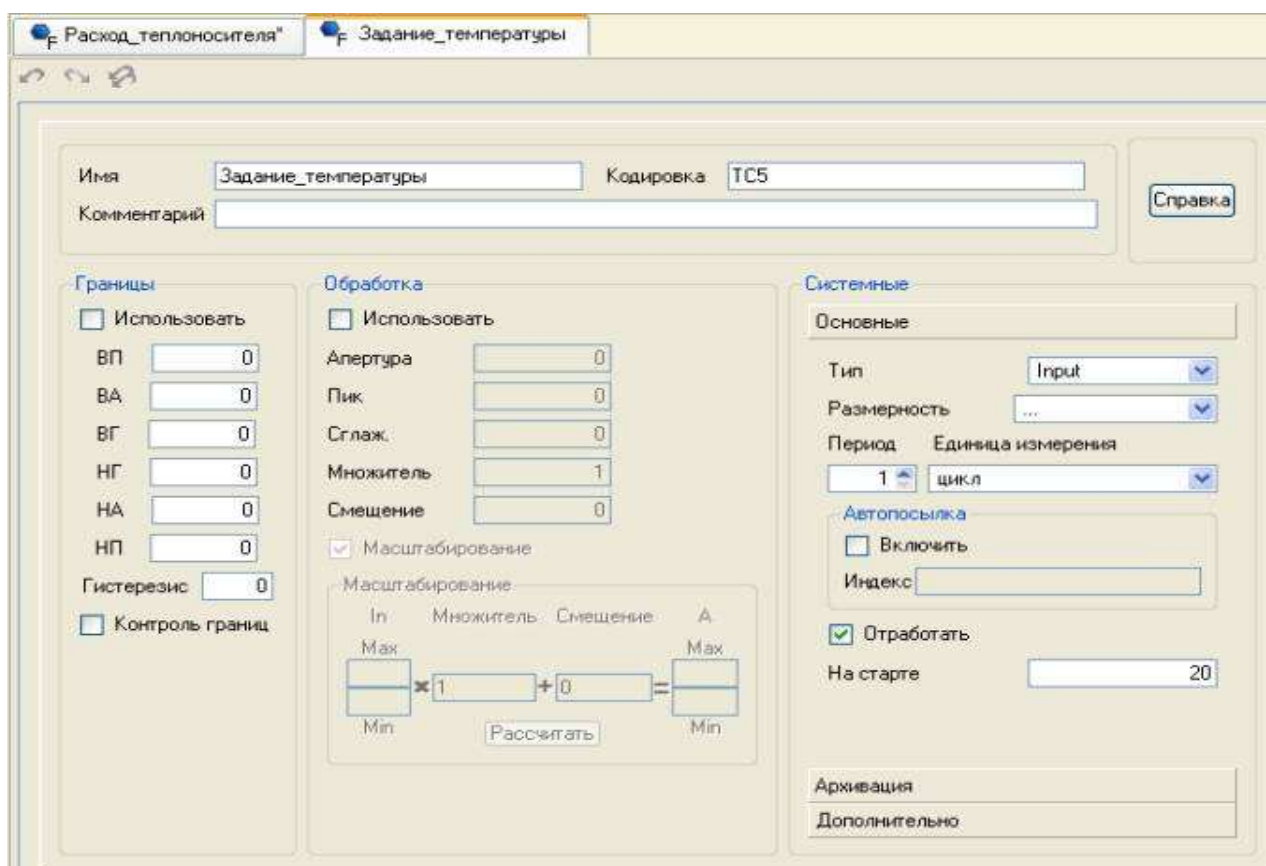
Участок_Хранения					
Содержимое группы					
Базовое имя	Период пересчета (зн)	Отработать	СПАД	Отчет Тревог	
Участок_хранения:3	1	0	No	0	
Уровень	1	0	1	1	
Температура	1	0	1	1	
Давление	1	0	1	1	
Влажность	1	0	1	1	
Дверь	1	0	1	1	
Вентиляция	1	0	1	1	
Пож_сигнализация	1	0	1	1	

Для группы **Участок_Дозирования**.

Базовое имя	Период пересчета (зн)	Отработать	СПАД	Отчет	Тревог
Участок_дозирования:4	1	0	No	0	0
Расход_продукта	1	0	1	0	0
Задание_объема	1	0	1	0	0
Старт_Стоп	1	0	1	0	0
Объем_продукта_текущий	1	0	1	0	0
Насос	1	0	1	0	0

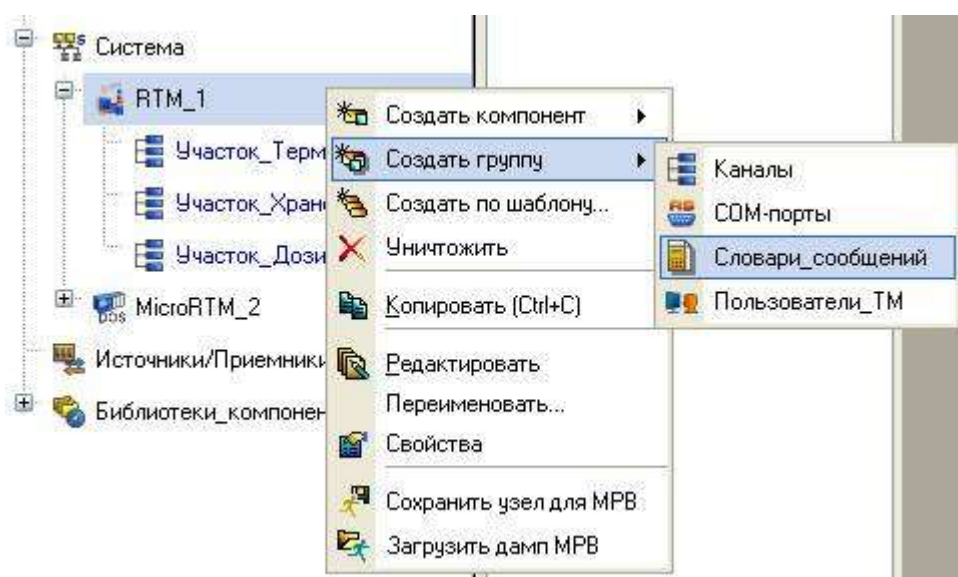
Для каналов класса **Float** установить границы и определить процедуру пересчета в физические величины, учитывая то обстоятельство, что все данные поступают от 12-разрядных АЦП, т.е. в кодах, лежащих в диапазоне (0-4095). Так, например, для канала **Расход_теплоносителя** в группе **Участок_Термообработки** это будет выглядеть следующим образом.

Для каналов, формирующих задания для PC-based контроллера, необходимо задать начальные значения. Так, например, для канала **Задание_температуры**.

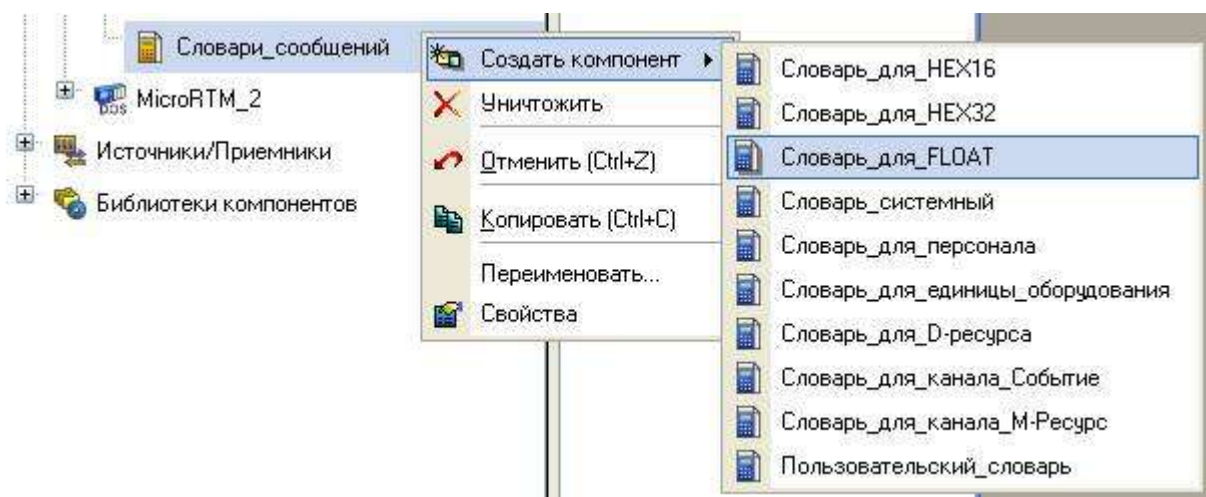


Для остальных каналов указать: $K_p=1$, $K_d=0.1$, $K_{dd}=0.01$, Зона_нечувствительности=0.5 и Задание_объема=50.

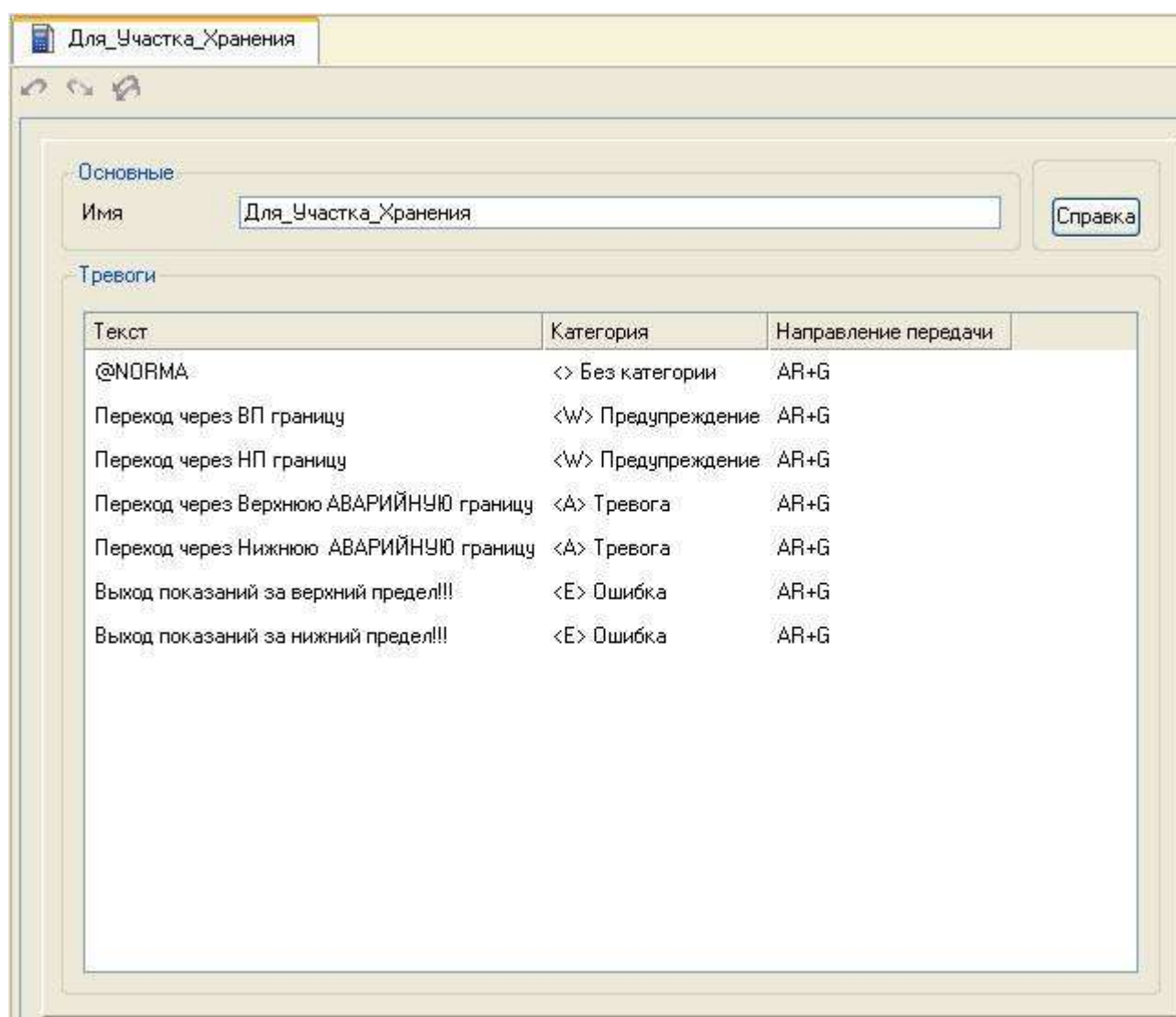
Для каналов класса **Float** группы каналов **Участок_Хранения** задать сообщения в отчет тревог. С этой целью создать для узла **RTM_1** новую группу **Словари сообщений**.



В данной группе создать новый компонент – **Словарь_для_Float**.

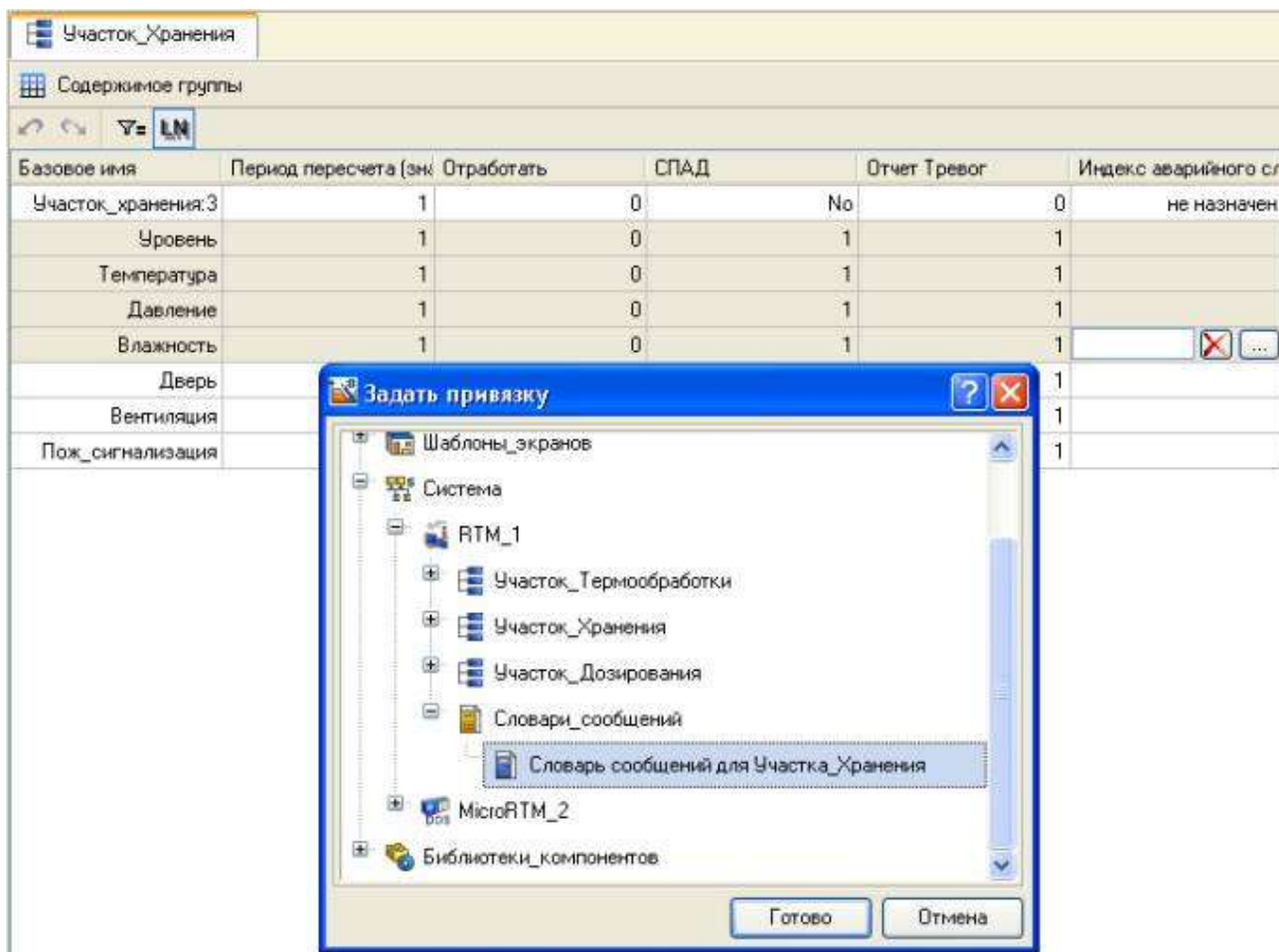


Двойным щелчком ЛК открыть его для редактирования, изменить его имя и сформировать сообщения для помещения в отчет тревог.

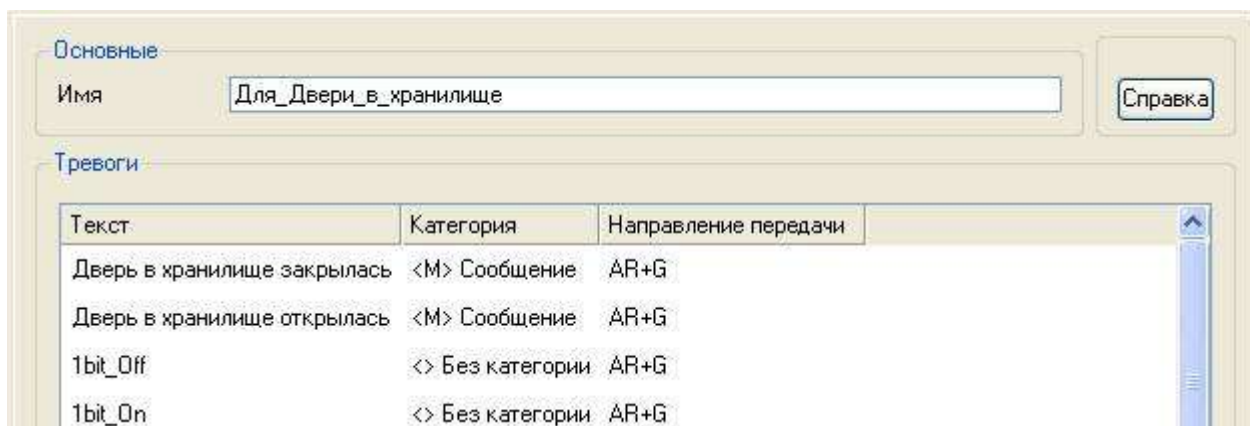


Подключать созданный словарь к каналам класса **Float** группы **Участок_Хранения** можно либо последовательно, открывая каналы на

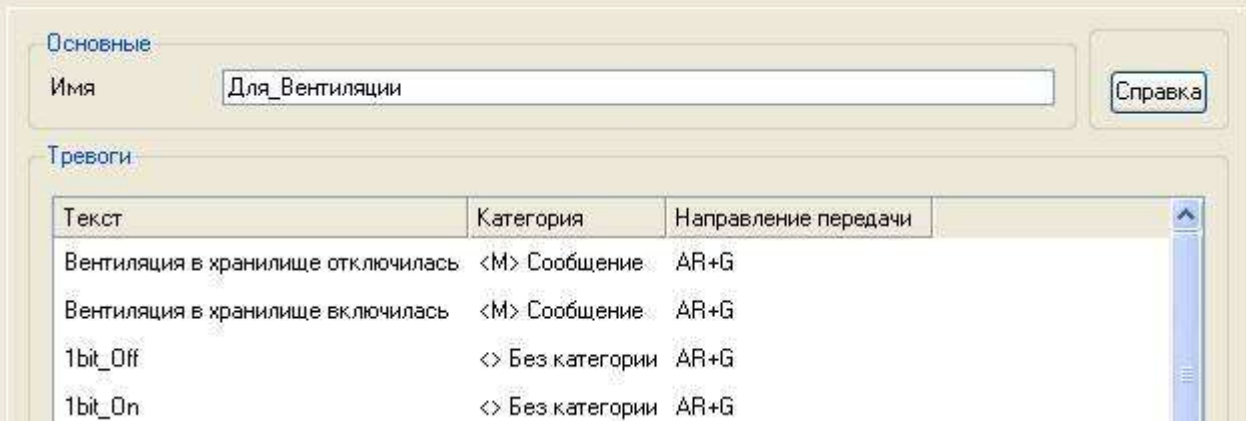
редактирование во вкладке **Архивация** и указывая индекс аварийного словаря, либо воспользоваться для этого процедурой группового редактирования.



Для каналов группы **Участок_Хранения**, принимающих от PLC дискретные сигналы, также необходимо создать словари для помещения сообщений в отчет тревог, но уже выбрав за основу словарь для каналов класса **HEX16**. В нем определить сообщения для нулевого бита. Для канала **Дверь** имеем:



Для канала **Вентиляция**.



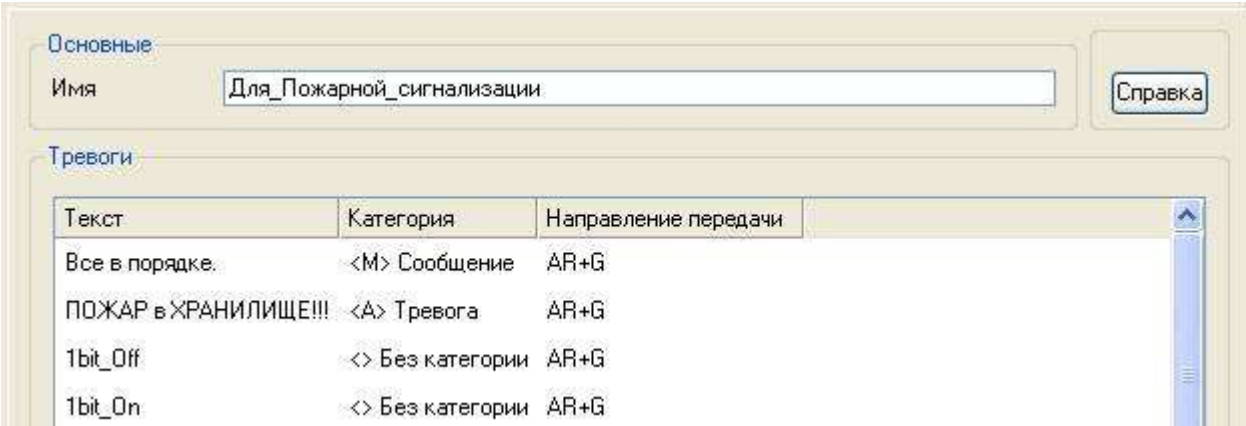
Основные

Имя

Тревоги

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

Для канала **Пожарная_сигнализация**.




Основные

Имя

Тревоги

Текст	Категория	Направление передачи
Все в порядке.	<M> Сообщение	AR+G
ПОЖАР в ХРАНИЛИЩЕ!!!	<A> Тревога	AR+G
1bit_Off	<> Без категории	AR+G
1bit_On	<> Без категории	AR+G

Подключить созданные словари к каналам таким же образом, как и в рассмотренном выше случае. Сохранить созданный проект, нажав ЛК .

5. Контрольные вопросы.
 - 5.1. Что содержит вкладка Архивы?
 - 5.2. Что содержит вкладка Отчет тревог?
 - 5.3. Какова процедура пересчета значений в физические величины?

ЛАБОРАТОРНАЯ РАБОТА 8

Организация вывода времени на графических экранах и фиксация событий

1. Цель работы – освоение принципов организации вывода времени и фиксации событий.

2. Задание по лабораторной работе: организовать вывод времени на графический экран; фиксацию событий по изменению дискретных сигналов.

3. Методические указания по выполнению лабораторной работы.

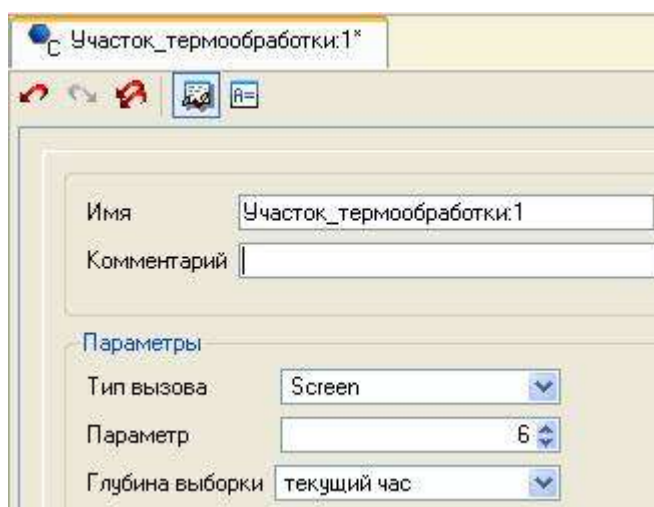
3.1. Организовать вывод времени на графическом экране (п. 4.1).

3.2. Спроектировать процесс фиксации событий (п. 4.2).

4. Процедура организации вывода времени и фиксация событий.

4.1. Организация вывода времени на графических экранах.

Существуют несколько способов вывода времени на графических экранах. Наиболее простой из них заключается в задании для каналов класса **Вызов**, настроенных на вызов графических экранов, атрибута **Параметр**.



Например, при его равенстве **6** в заголовке окна будет выводиться следующее.

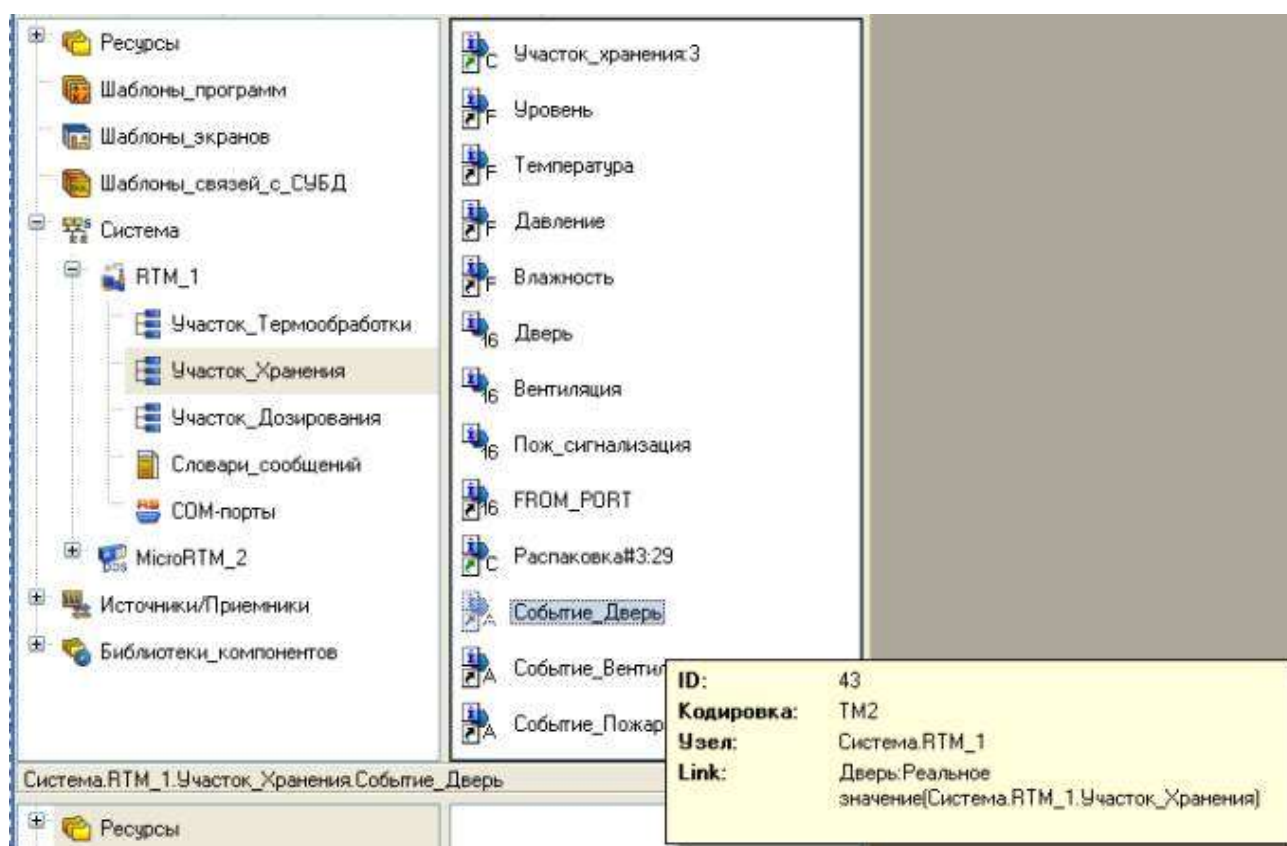


Другой способ заключается в создании системных переменных **@t_Seconds**, **@t_Minutes** и **@t_Hours** в подгруппе компонентов **Системные** группы **Диагностика_и_Сервис** в слое **Источники/Приемники** и связывании

их с аргументами шаблонов экрана, которые, в свою очередь, с помощью ГЭ Текст отображают системное время – часы, минуты и секунды в выбранном месте графического экрана.

4.2. Фиксация событий.

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 квитирования создать в группе **Участок_Хранения** узла АРМ три канала класса **Событие**. Переименовать их и, открыв дополнительное окно Навигатора проекта на той же группе, связать с помощью мыши с соответствующими каналами класса **НЕХ16 Дверь, Вентиляция и Пожарная_сигнализация**.



Двойным щелчком ЛК мыши открыть на редактирование канал **Событие_Дверь** и установить атрибут **Размер стека аварий** равным 20. Таким же образом поступить в отношении других двух каналов.

Событие_Дверь

Имя: Событие_Дверь Кодировка: TM2

Комментарий: _____

Справка

Параметры

Размер стека аварий: 20

Квитирование в сеть

Системные

Основные

Тип: Input

Размерность: ...

Период: 1 Единица измерения: цикл

Автопосылка

Включить


Индекс: _____

Отработать



На старте: 0

Архивация

Дополнительно

Так как предварительно уже был подготовлен ГЭ События  на шаблоне экрана **Участок_Хранения**, то выполнить привязку атрибутов вновь созданных каналов класса **Событие** к аргументам канала класса **Вызов Участок_Хранения**.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Уровень	IN	REAL		Уровень:Реальное значение(Система.RTM_1.Участок_Хранения)	
Температура	IN	REAL		Температура:Реальное значение(Система.RTM_1.Участок_Хранения)	
Давление	IN	REAL		Давление:Реальное значение(Система.RTM_1.Участок_Хранения)	
Влажность	IN	REAL		Влажность:Реальное значение(Система.RTM_1.Участок_Хранения)	
Дверь	IN	USINT		Дверь:Реальное значение(Система.RTM_1.Участок_Хранения)	
Вентиляция	IN	USINT		Вентиляция:Реальное значение(Система.RTM_1.Участок_Хранения)	
Пож_сигнализация	IN	USINT		Пож_сигнализация:Реальное значение(Система.RTM_1.Участок_Хранения)	
Событие_Дверь	IN/OUT	REAL		Событие_Дверь:Реальное значение(Система.RTM_1.Участок_Хранения)	NP
Событие_Вент	IN/OUT	REAL		Событие_Вентиляция:Реальное значение(Система.RTM_1.Участок_Хранения)	NP
Событие_Пож	IN/OUT	REAL		Событие_Пожар:Реальное значение(Система.RTM_1.Участок_Хранения)	NP
Слой_основной	IN	USINT	0		NP
Слой_тренд	IN	USINT	1		NP

Также выставить значения по умолчанию для аргументов **Слой_основной** и **Слой_тренд** соответственно **0** и **1** с тем, чтобы при открытии данного графического экрана слой, содержащий ГЭ Тренд , не отображался. Сохранить выполненную работу, нажав ЛК .

5. Контрольные вопросы.

5.1. Как использовать системные переменные в связи с аргументами шаблонов экрана?

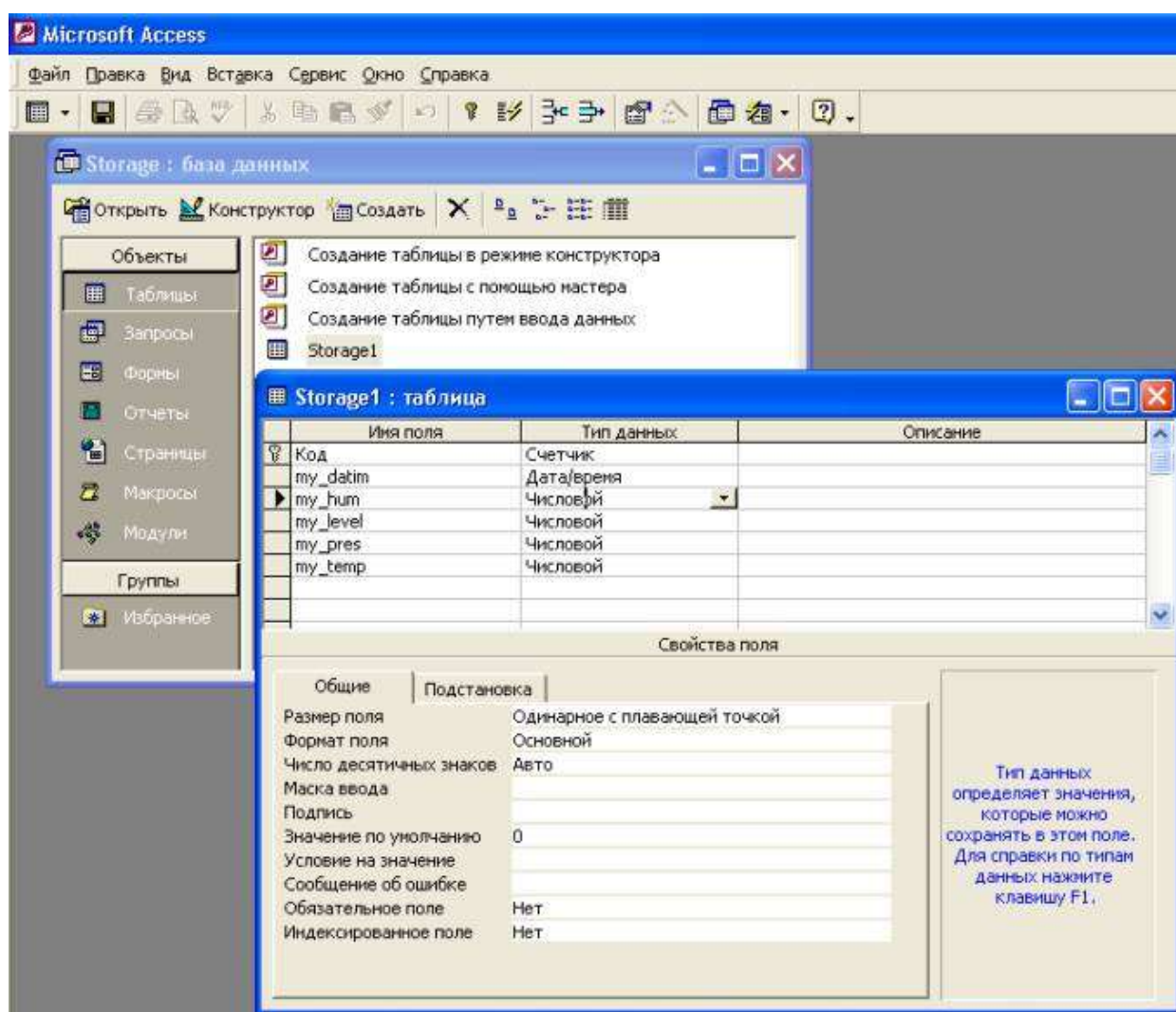
5.2. Как выполнить фиксацию событий?

ЛАБОРАТОРНАЯ РАБОТА № 9

Связь с СУБД MS Access

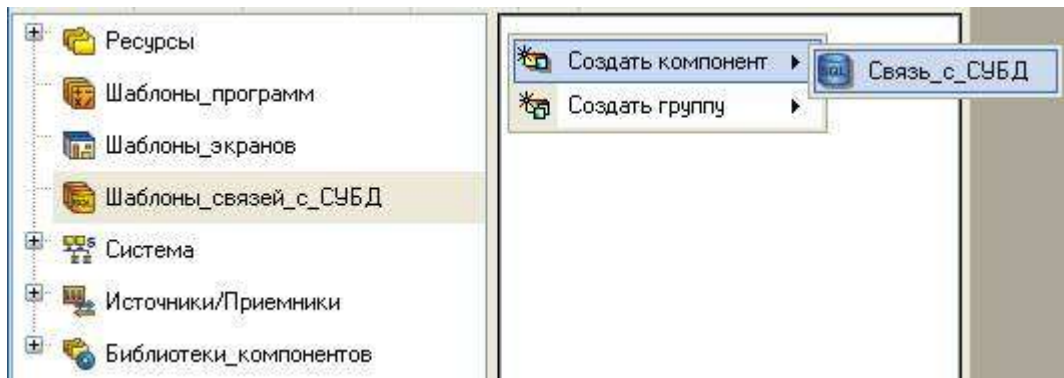
1. Цель работы – освоение принципов организации связи с СУБД MS Access.
2. Задание по лабораторной работе: провести программирование записи данных, получаемых от контроллера, в таблицу реляционной базы данных.
3. Методические указания по выполнению лабораторной работы.
 - 3.1. Организовать связь с СУБД MS Access (п. 4).
4. Процедура организации связи с СУБД MS Access.

Пусть необходимо данные, получаемые от PLC по участку хранения, записывать в таблицу реляционной базы данных. Структура таблицы **Storage1** файла базы данных MS Access **Storage.mdb** для накопления значений технологических параметров, поступающих в реальном времени от АРМ, будет иметь следующий вид.



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**.

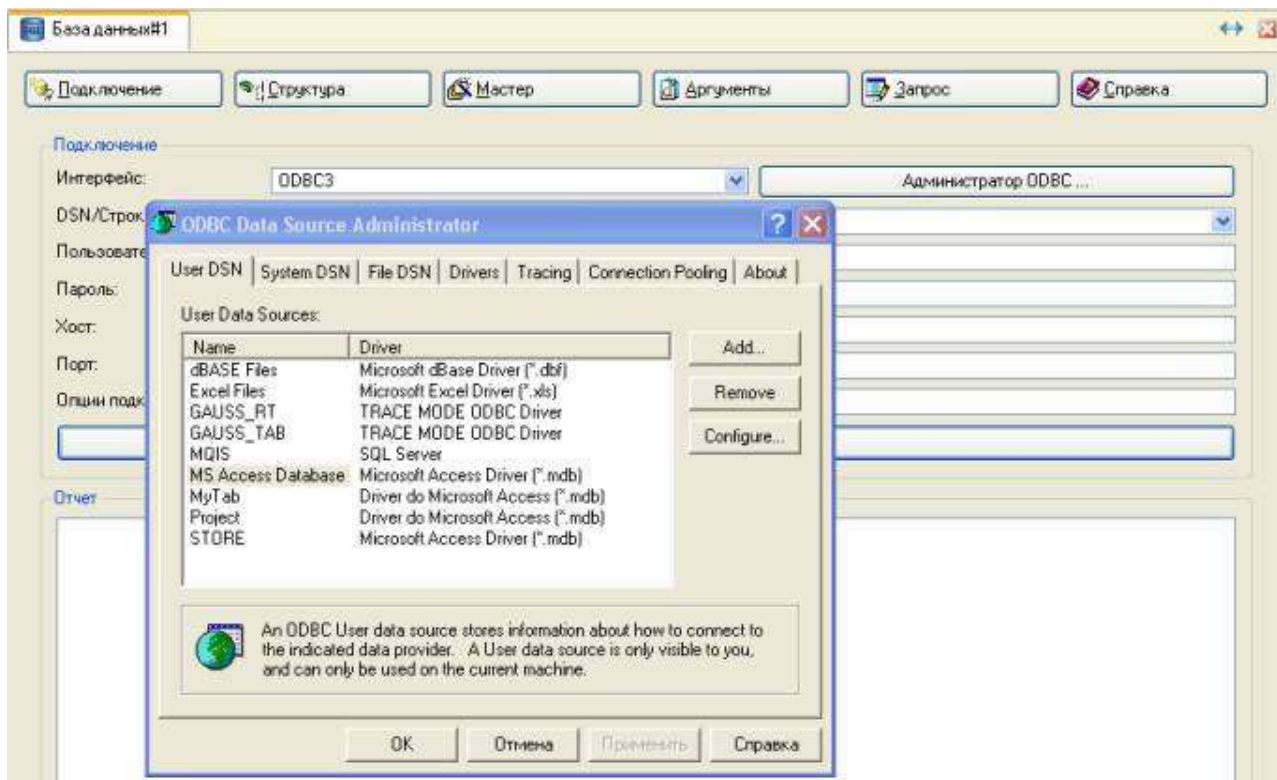
Далее необходимо приступить к разработке SQL-запроса и встраиванию его в проект. Для этого в Навигаторе проекта открыть слой **Шаблоны_связей_с_СУБД** и создать новый компонент **База_данных#1**.



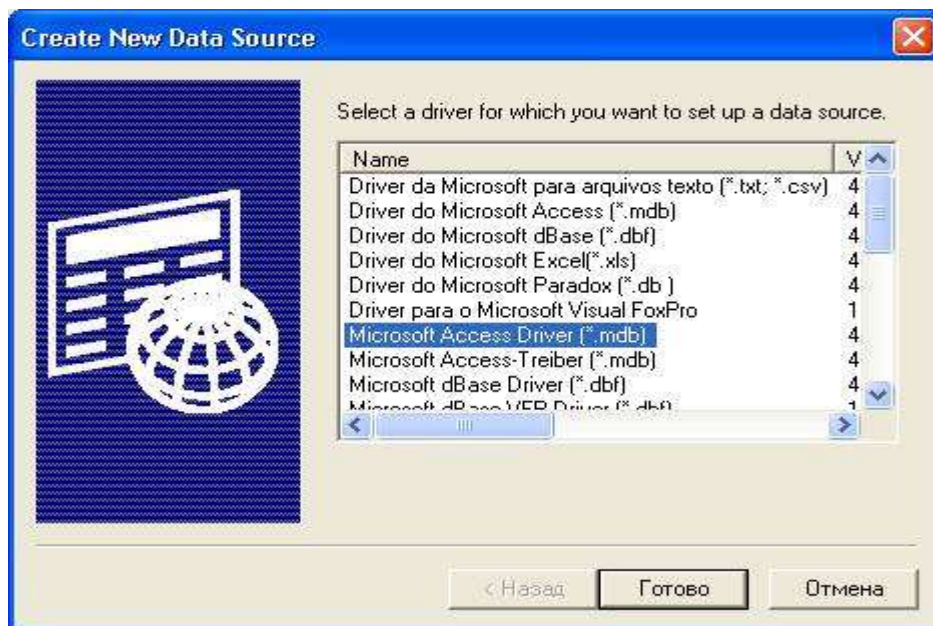
Открыть свойства шаблона и задать ему следующие аргументы.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Влажность	IN	REAL			
Уровень	IN	REAL			
Временная_отметка	IN	REAL			
Давление	IN	REAL			
Температура	IN	REAL			

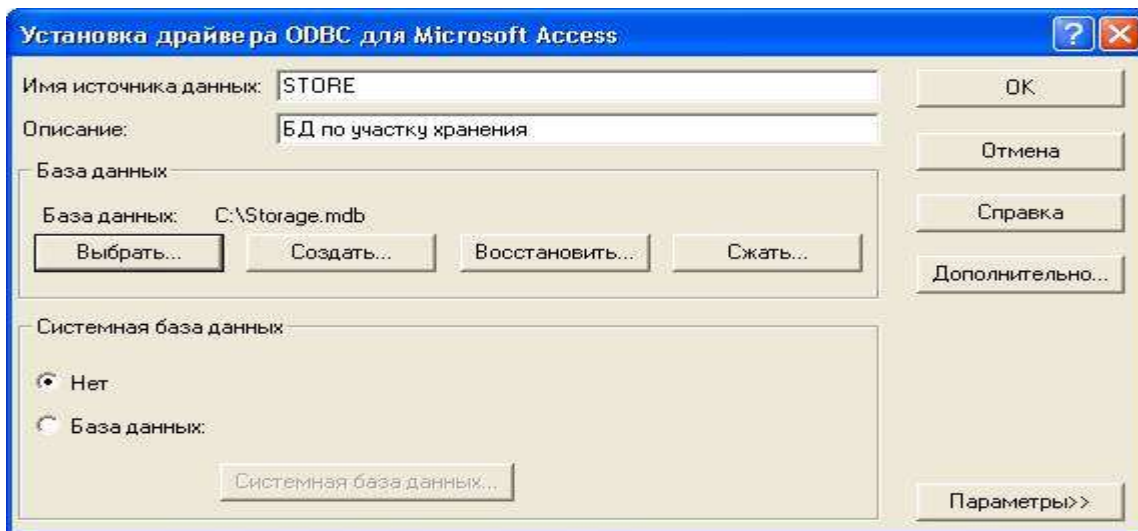
Закрывать свойства и двойным щелчком ЛК открыть шаблон на редактирование. Перед реализацией записи в файл БД MS Access C:\Storage.mdb данных по участку хранения зарегистрировать данный файл в качестве источника/приемника данных. По щелчку ЛК на экранной кнопке **Администратор ODBC...** будет вызван диалог для создания пользовательского **DSN**.



Создать с помощью клавиши **Add** новый источник/приемник данных - **STORE**.



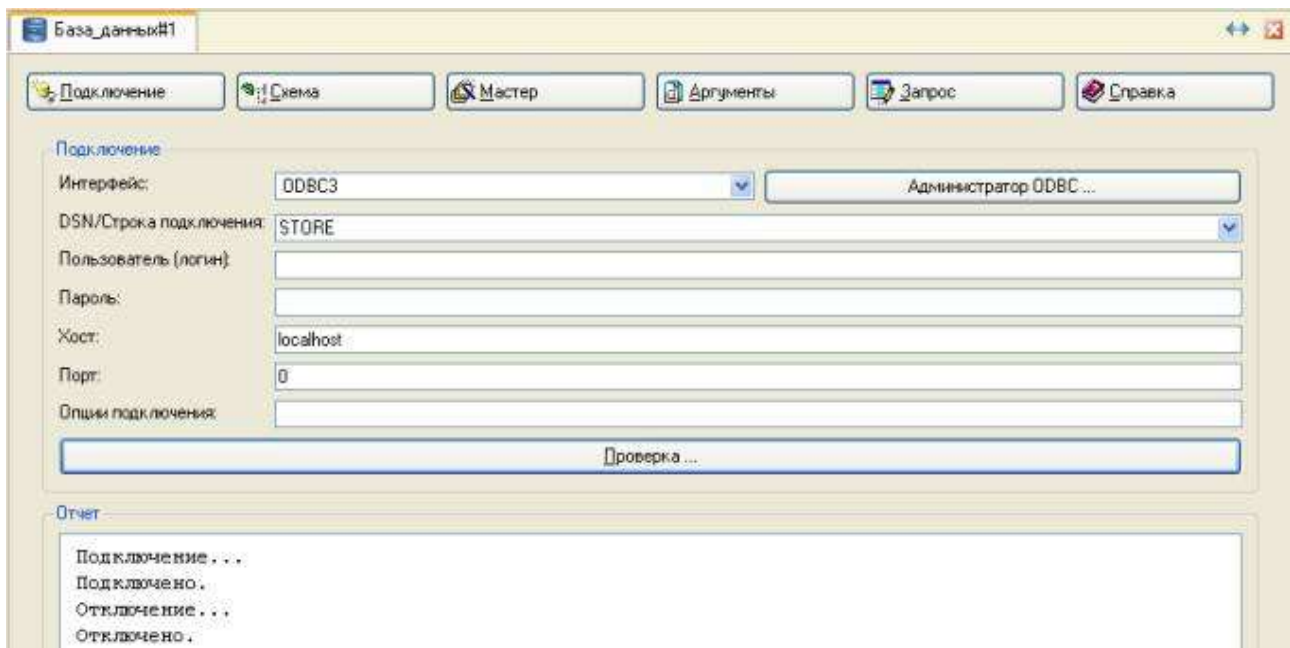
Далее настроить его следующим образом.



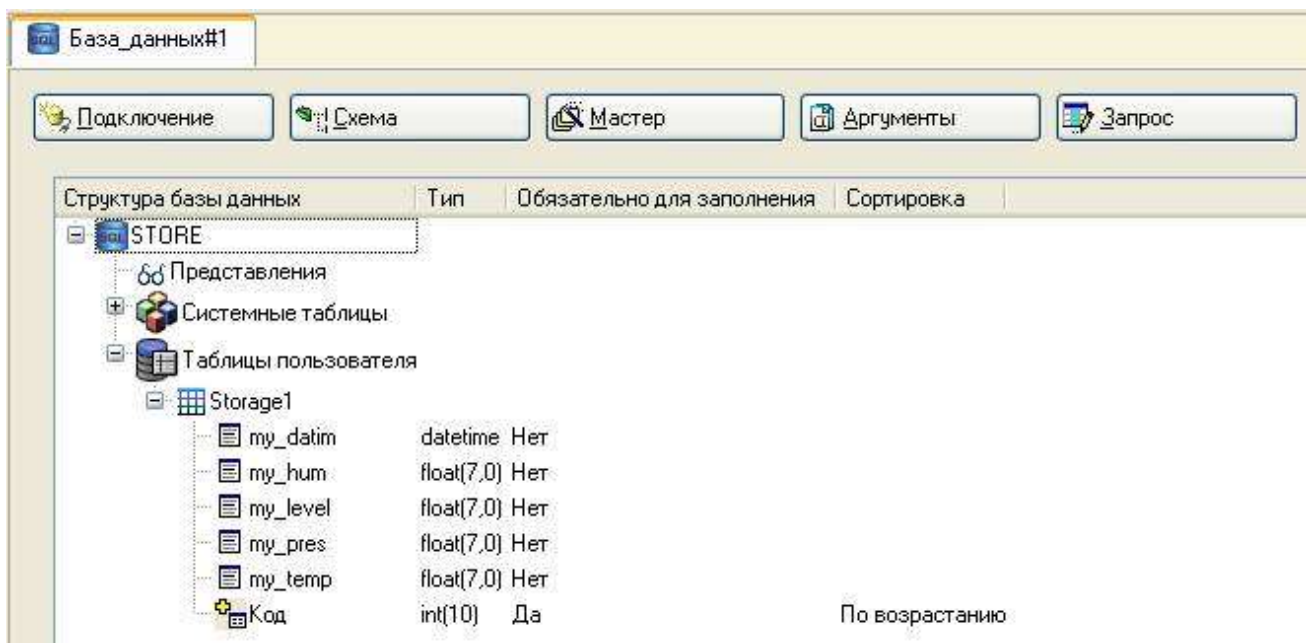
Нажать кнопку **ОК**. Таким образом, в числе прочих пользовательских **DSN** появится новый – **STORE**.



Нажать **ОК** и далее продолжить работу в редакторе шаблона **База_данных#1**. В строке **DSN/Строка подключения** указать созданный пользовательский **DSN**, в нашем случае **STORE**, и проверить с помощью экранной клавиши **Проверка** правильность подключения к файлу базы данных в окне **Отчет**.



С помощью экранной кнопки **Схема** просмотреть структуру подключенной БД.



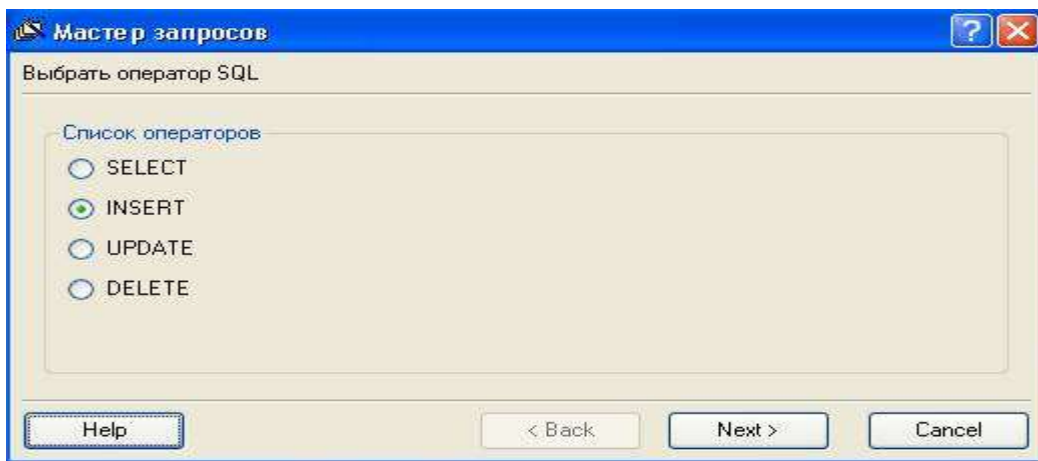
Далее либо с помощью **Мастера запросов**, либо непосредственно во вкладке **Запрос** создать на языке **SQL** запрос, помещающий связанные с аргументами шаблона атрибуты каналов в таблицу **Storage1** файла Storage.mdb с созданием новых записей.

```

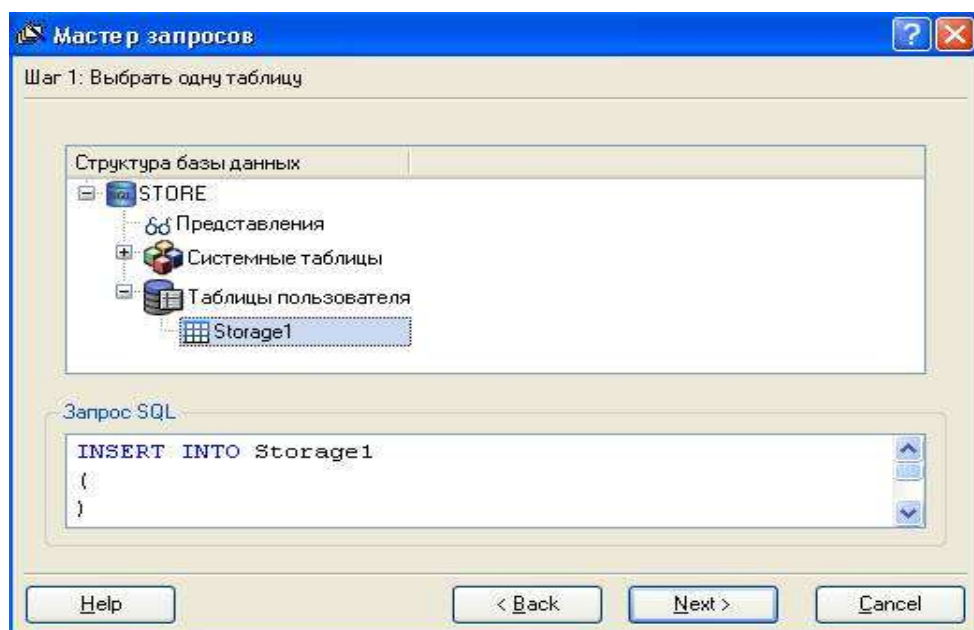
INSERT INTO Storage1
(
    my_hum,
    my_level,
    my_datim,
    my_pres,
    my_temp
)
VALUES
(
    '#Влажность#',
    '#Уровень#',
    '#Временная_отметка#',
    '#Давление#',
    '#Температура#'
)

```

Для использования **Мастера запросов** нажать ЛК экранную кнопку **Мастер** и в открывшемся окне указать тип создаваемого запроса **INSERT**.

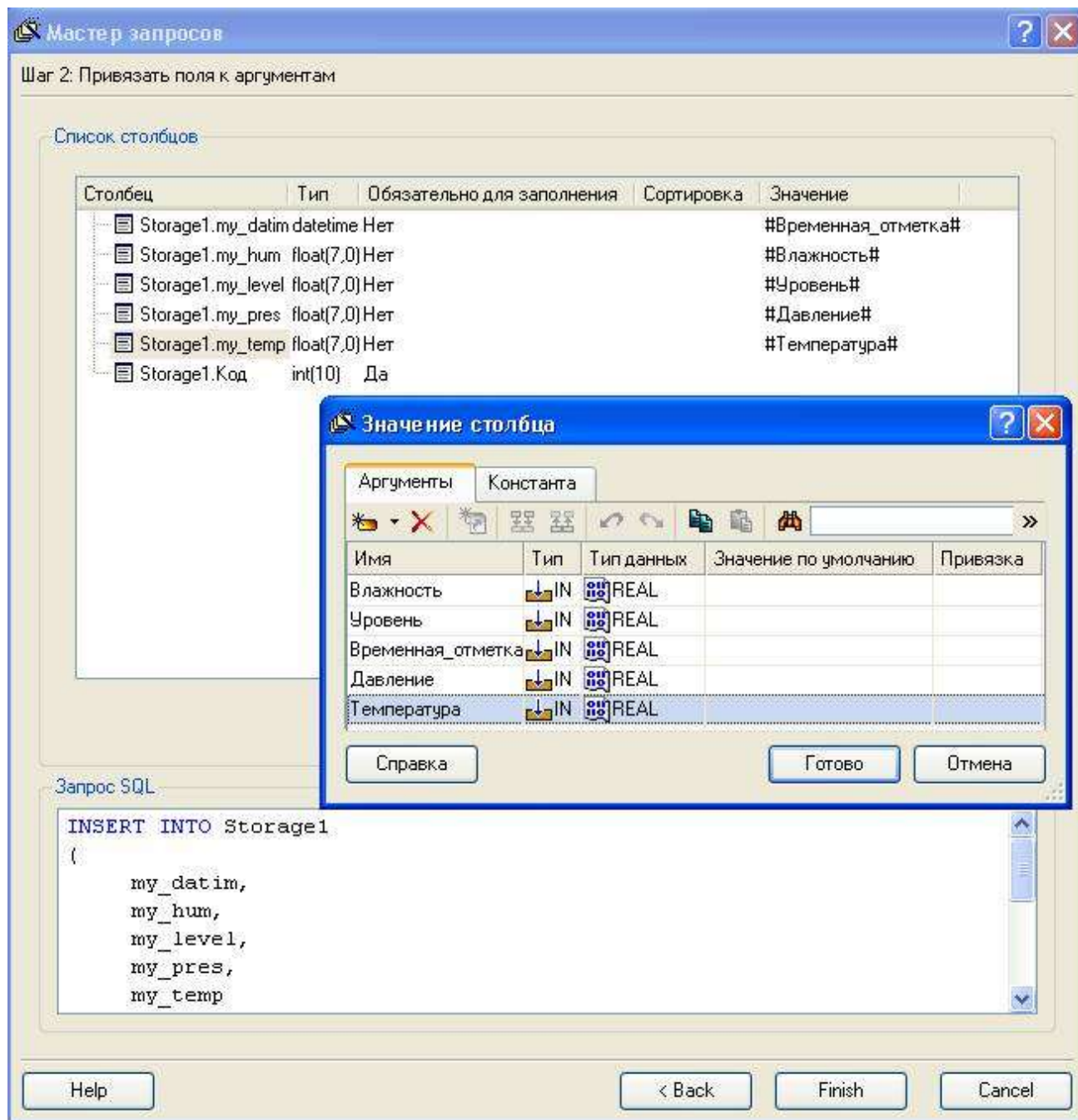


Нажать кнопку **Next**; выбрать таблицу **Storage1**.

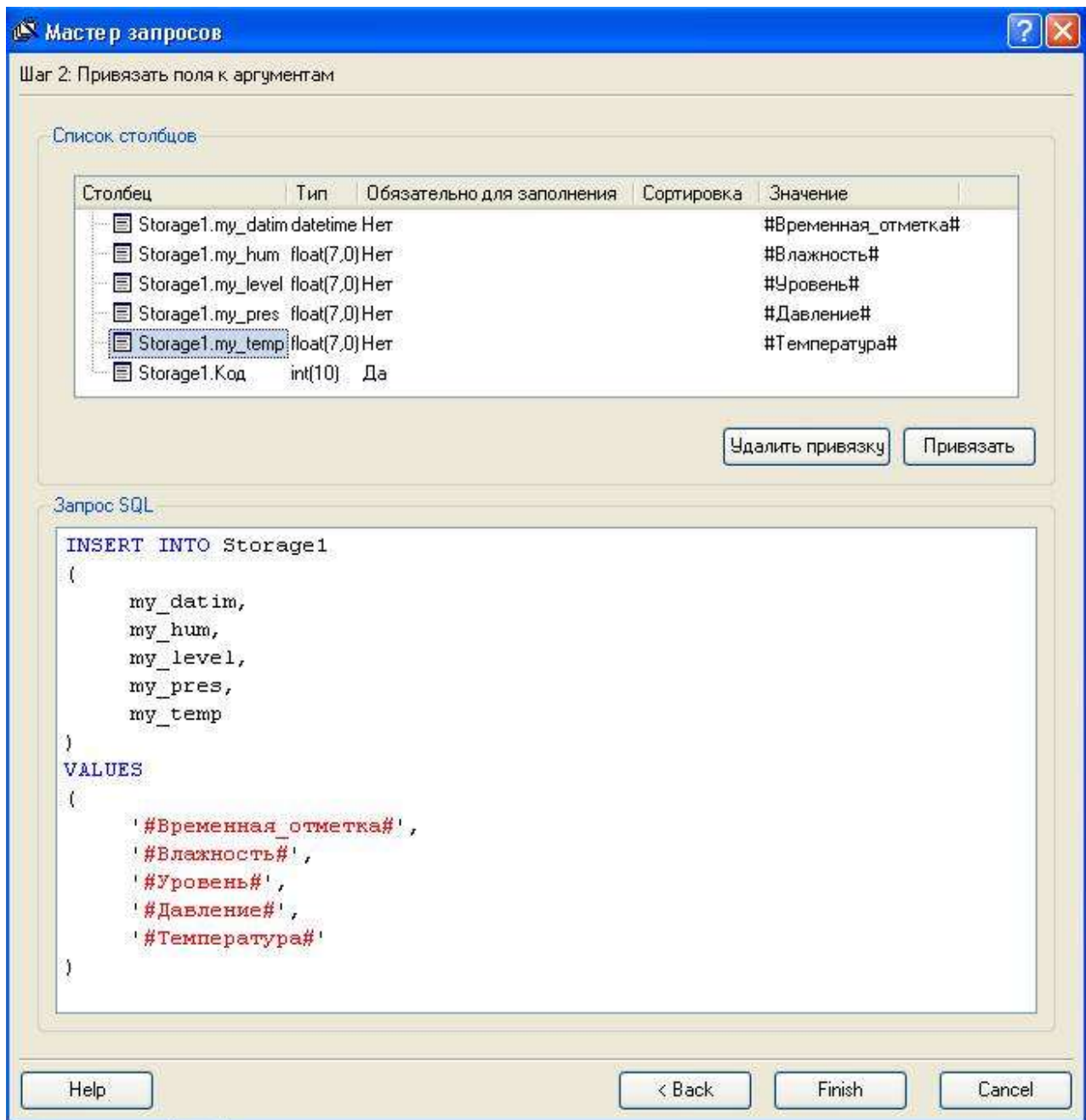


Перейти к следующему шагу нажатием кнопки **Next**.


Далее произвести последовательную привязку к полям БД соответствующих аргументов шаблона. При выделении ЛК поля БД из списка и последующем нажатии кнопки **Привязка** в открывающемся окне **Значение столбца** указывается привязываемый аргумент.

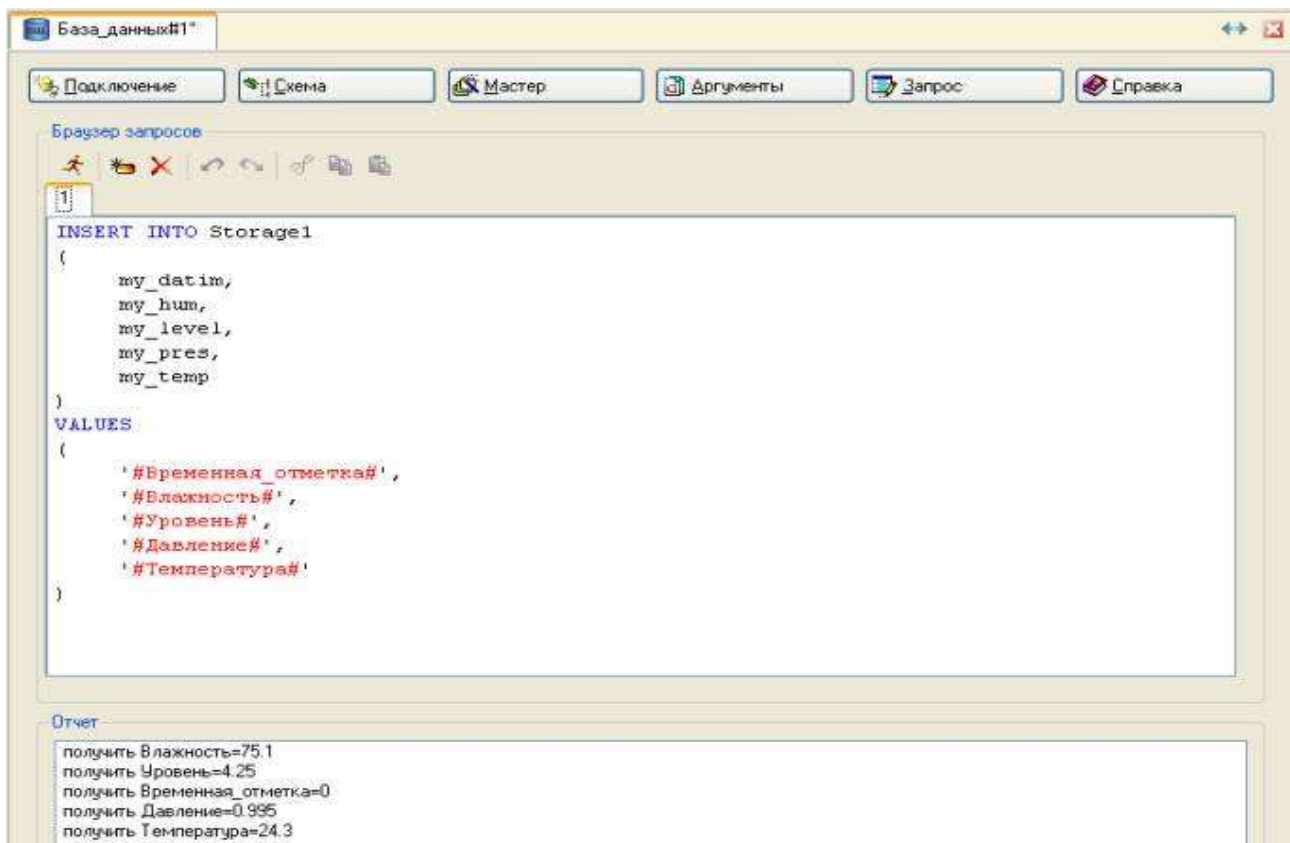


В нижнем окне **Мастера запросов** при этом будет формироваться следующий запрос.

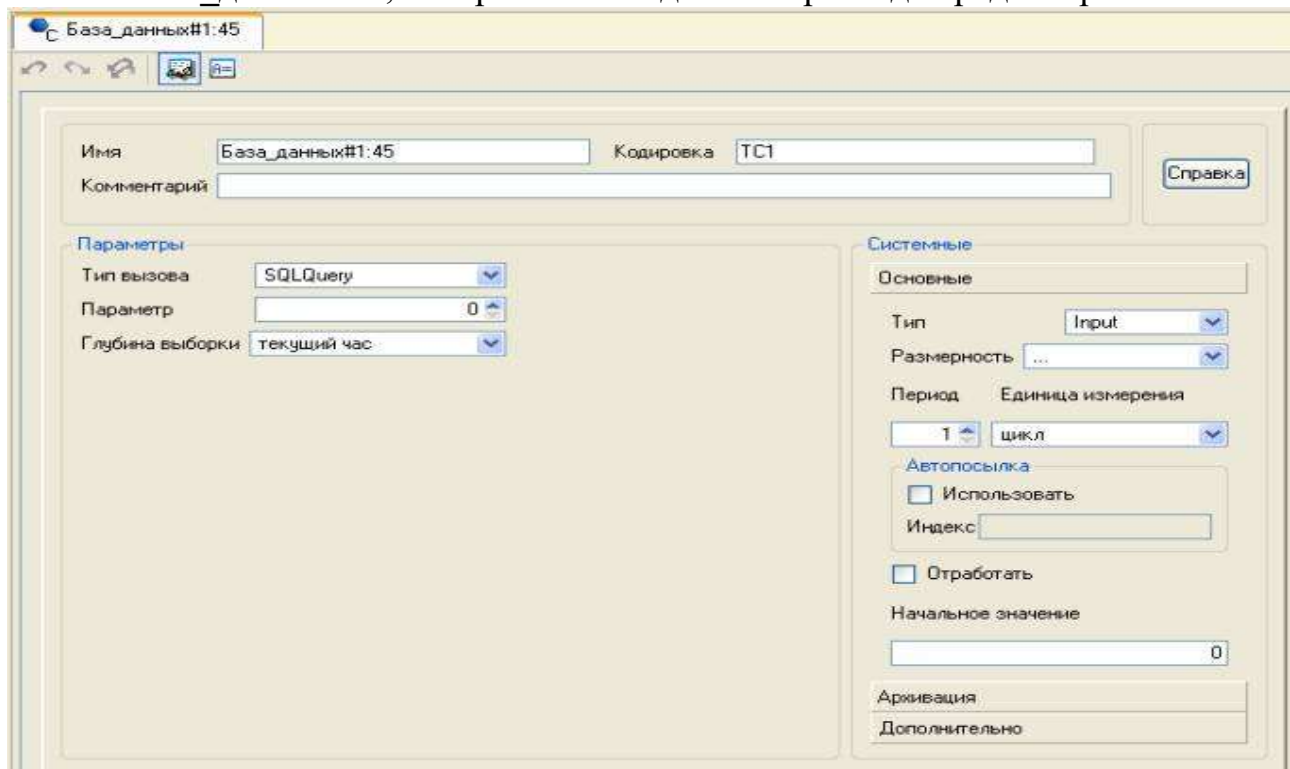



Завершить создание запроса нажатием экранной кнопки **Finish**.

Отладка разработанного шаблона производится во вкладке **Запрос** с помощью иконки . Предварительно необходимо задать значения аргументов по умолчанию в соответствующем столбце табличного редактора аргументов. Ход выполнения запроса отображается в окне **Отчет**:



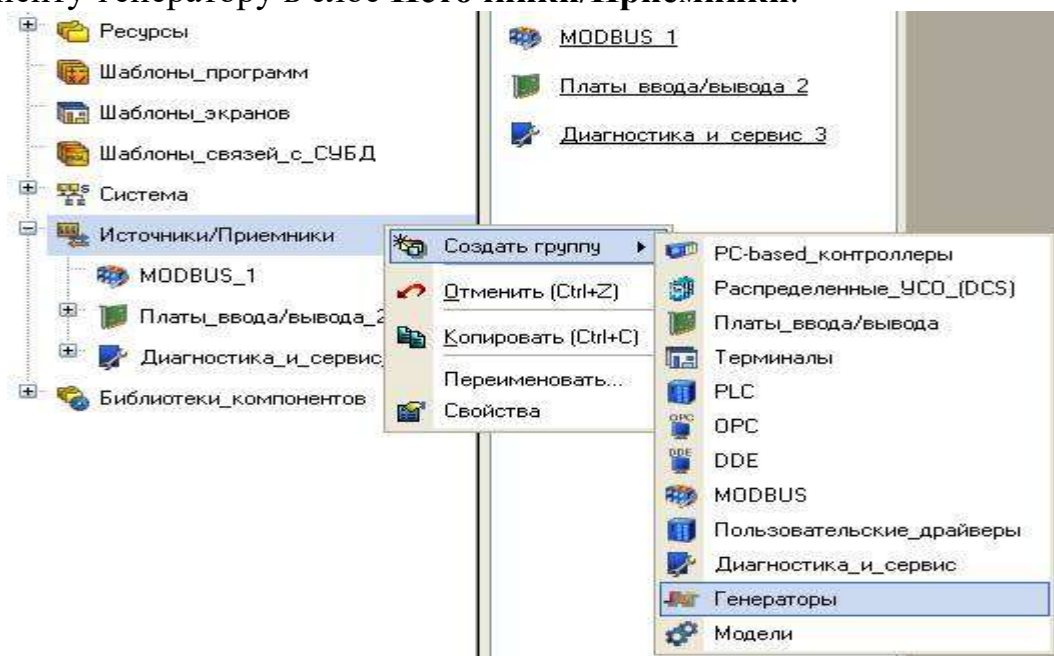
Открыть дополнительное окно Навигатора проекта, настроив его на отображение группы **Участок_Хранения** узла АРМ. Выделить ЛК шаблон **База_данных#1** и, удерживая ЛК нажатой, перетащить шаблон в группу **Участок_Хранения**. В группе **Участок_Хранения** будет создан канал класса **Вызов База_данных#1**, который необходимо открыть для редактирования.

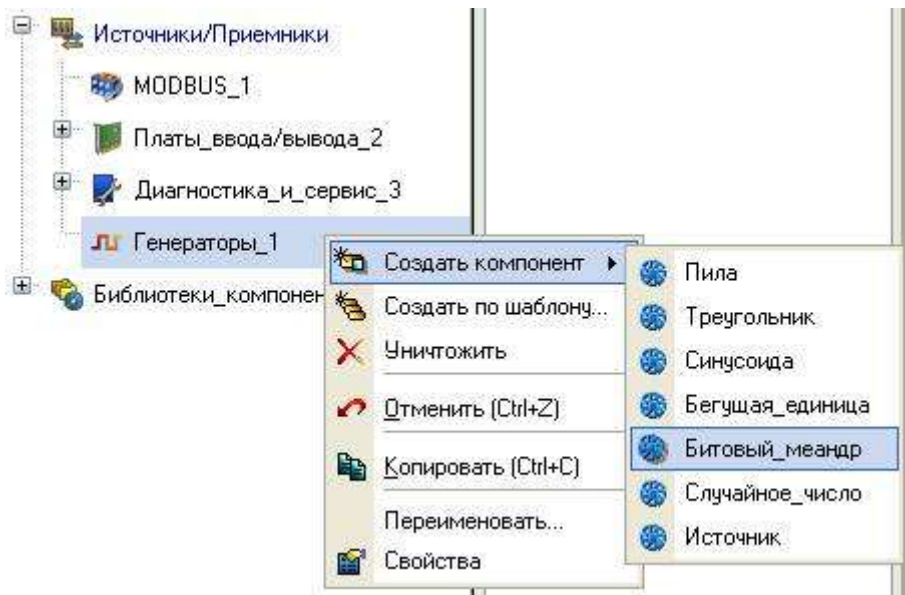


Для связывания аргументов созданного запроса с атрибутами каналов перейти с помощью иконки  в табличный редактор аргументов и выполнить привязку следующим образом.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Влажность	IN	REAL		Влажность:Реальное значение(Система RTM_1,Участок_Хранения)
Уровень	IN	REAL		Уровень:Реальное значение(Система RTM_1,Участок_Хранения)
Временная_отметка	IN	REAL		Битовый_меандр#1:Время изменения(Система RTM_1,Участок_Хранения)
Давление	IN	REAL		Давление:Реальное значение(Система RTM_1,Участок_Хранения)
Температура	IN	REAL		Температура:Реальное значение(Система RTM_1,Участок_Хранения)

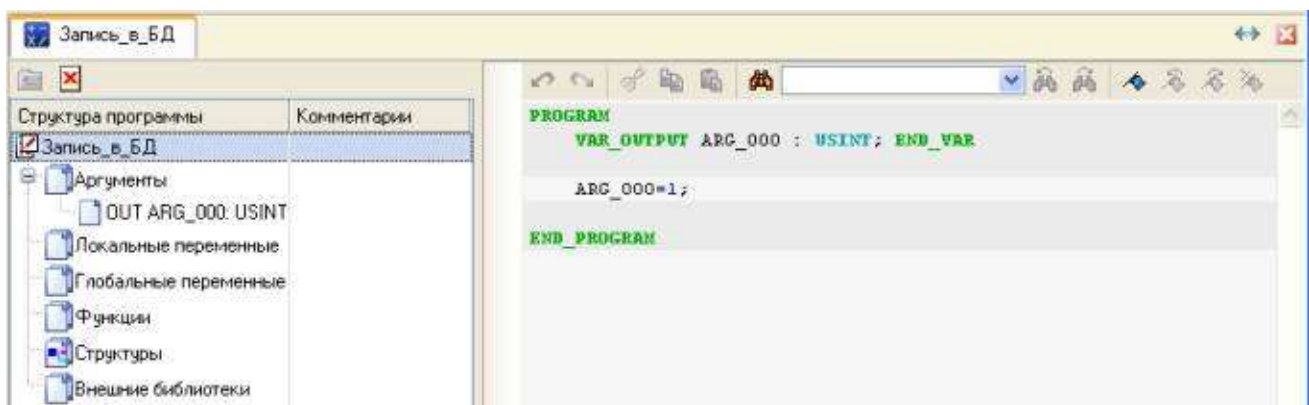
Для формирования временной отметки используется атрибут **Время изменения** канала **Битовый_меандр#1**, созданного в группе **Участок_Хранения** методом автопостроения по соответствующему компоненту-генератору в слое **Источники/Приемники**.



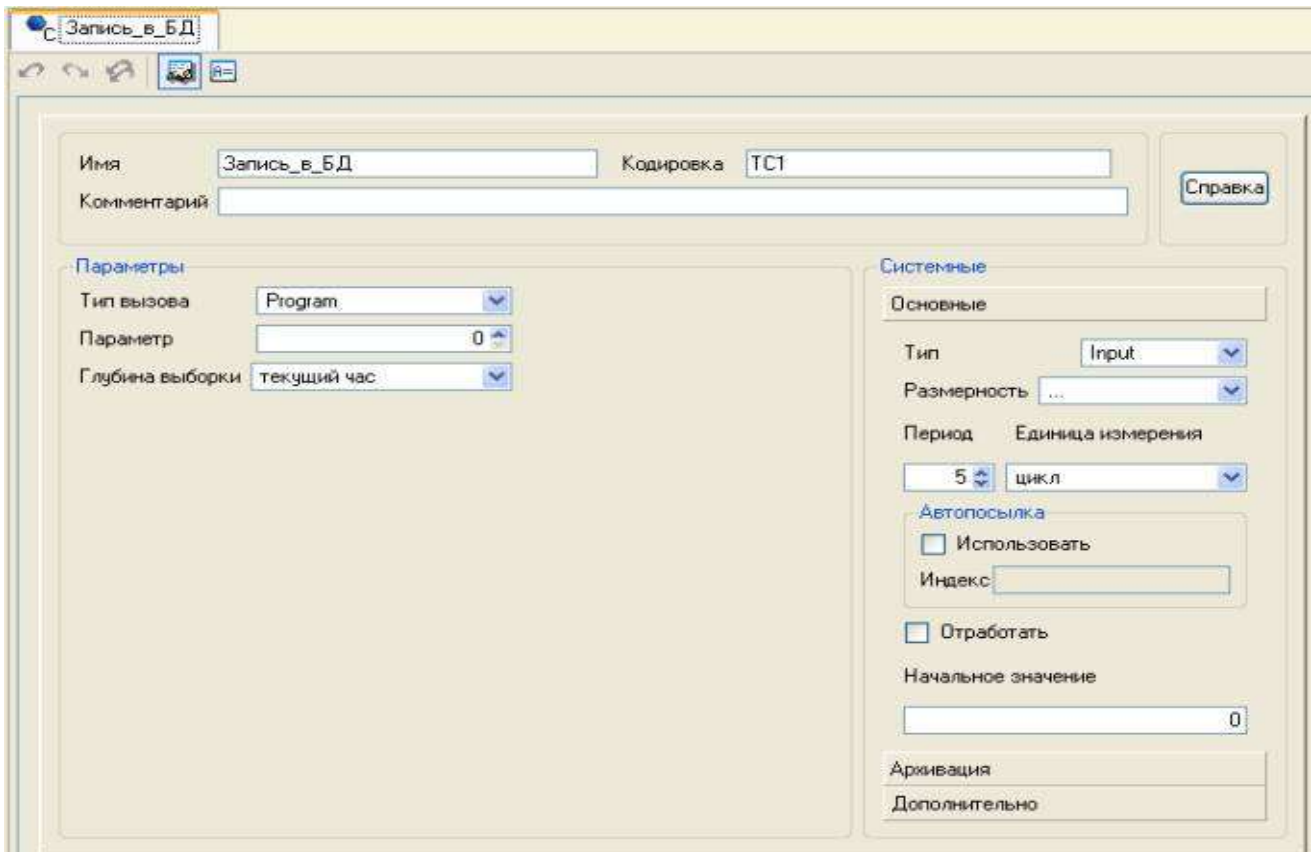



Перетащить созданный компонент в группу **Участок_Хранения**. Поскольку на каждом цикле пересчета базы каналов реальное значение канала **Битовый_меандр#1** будет изменяться, то при записи в файл БД будет сохраняться временная отметка по текущему времени.

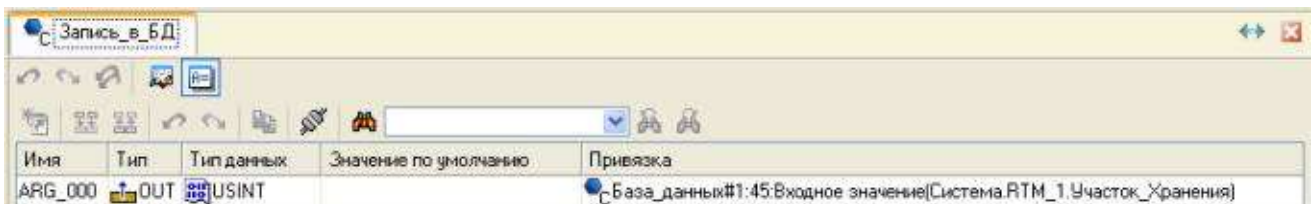
Теперь для того чтобы при запуске на АРМ МРВ в файл БД **Storage.mdb** производилась запись данных по участку хранения, необходимо во входное значение канала класса **Вызов База_данных#1** подавать с требуемой периодичностью номер выполняемого запроса, в нашем случае **1**. Сделать это можно, например, с помощью программы или ГЭ Кнопка. Создать в слое **Шаблоны_программ** новую программу **Запись_в_БД** и реализовать ее следующим образом.



Откомпилировать программу и, убедившись в корректности компиляции, перетащить в группу **Участок_Хранения** узла **RTM_1**. Открыть на редактирование канал класса **Вызов Запись_в_БД** и установить в его основном бланке атрибут **Период**, равный, например, пяти циклам.



Перейти с помощью иконки  к аргументам канала и произвести привязку следующим образом.



Изменяя период пересчета данного канала, можно регулировать темп записи величин технологических параметров участка хранения в файл БД MS Access.

Сохранить выполненную работу, нажав ЛК .

5. Контрольные вопросы.
 - 5.1. Как разработать SQL – запрос и встроить его в проект?
 - 5.2. Как создать новый источник/приемник данных?

ЛАБОРАТОРНАЯ РАБОТА № 10

Использование имитаторов аппаратных средств

1. Цель работы – освоение принципов использования имитаторов аппаратных средств.

2. Задание по лабораторной работе: провести имитацию технологического объекта и сигналов, передаваемых от контроллера.

3. Методические указания по выполнению лабораторной работы.

3.1. Выполнить подготовительные операции (п. 4.1).

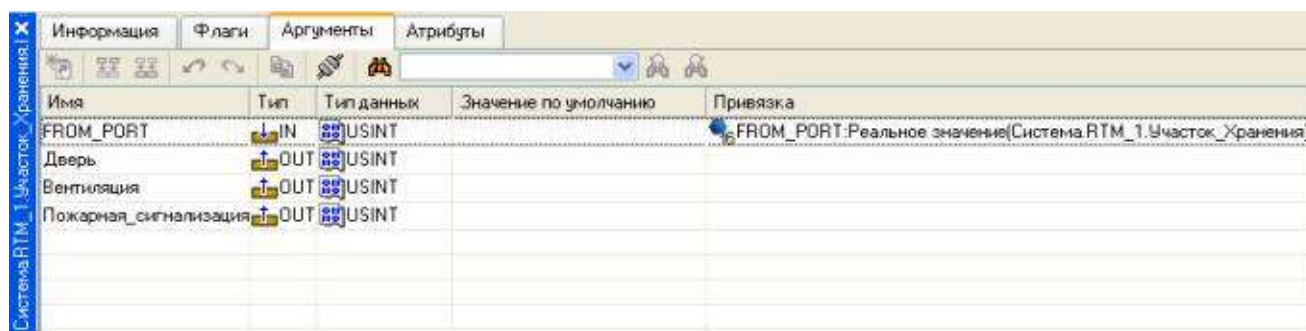
3.2. Разработать программы имитаторов, встроить их в проект (п. 4.2).

4. Процедура использования имитаторов аппаратных средств.

4.1. Подготовительные операции.

Для проверки работоспособности созданного проекта можно прибегнуть к имитации как технологического объекта на участке термообработки, сигналов, передаваемых от ПЛК участка хранения, и расходомера готового продукта участка дозирования.

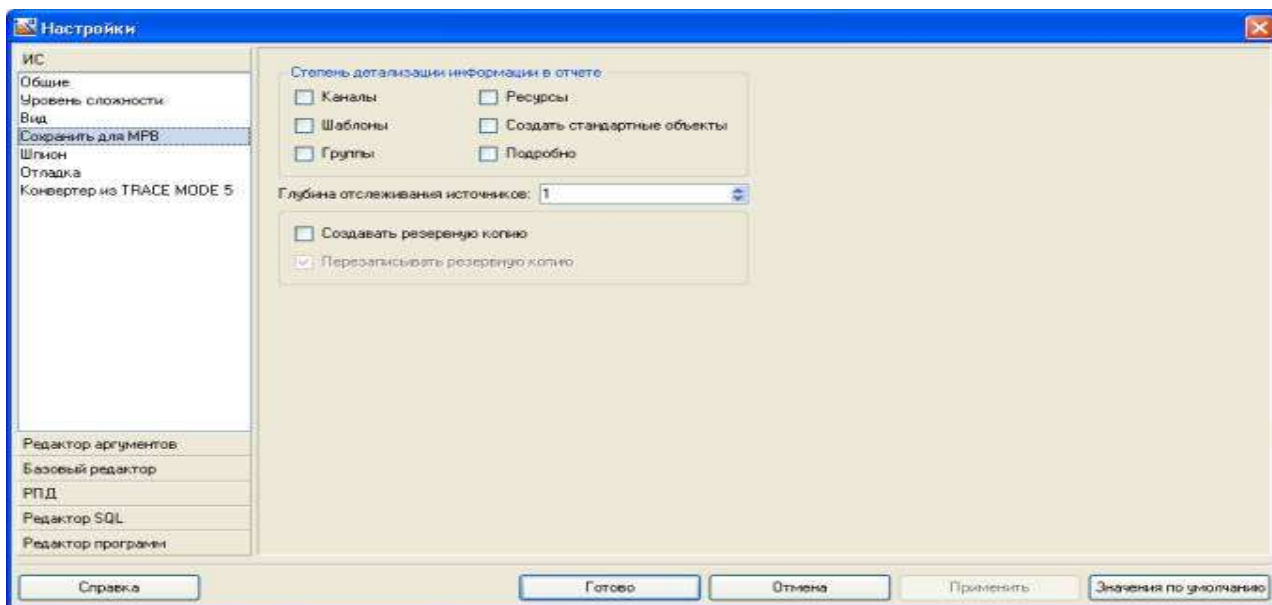
Предварительно необходимо произвести ряд подготовительных операций. Загрузить в интегрированную среду разработки проект **QS_Lesson_2.prj** и открыть слой **Источники/Приемники**. С помощью клавиши **Del** удалить группы **MODBUS_1** и **Платы_ввода/вывода_2**, тем самым удалить привязку для каналов, непосредственно взаимодействующих с аппаратурой ввода/вывода. В группе **Участок_Хранения** узла **АРМ RTM_1** выделить канал класса **Вызов Распаковка#3** и, открыв его свойства, во вкладке **Аргументы** убрать привязки аргументов к атрибутам каналов следующим образом.



Имя	Тип	Тип данных	Значение по умолчанию	Привязка
FROM_PORT	IN	USINT		FROM_PORT:Реальное значение(Система.RTM_1.Участок_Хранения)
Дверь	OUT	USINT		
Вентиляция	OUT	USINT		
Пожарная сигнализация	OUT	USINT		

На практике столь радикальный подход может и не применяться, особенно в тех случаях, когда при проведении отладки выполненного проекта (одного из узлов) по тем или иным причинам нет аппаратных средств, реализующих каналы ввода/вывода, но при этом вполне уместна их замена либо источником-генератором сигнала, либо имеющимся в распоряжении другим аппаратным средством. В подобной ситуации наиболее рационально прибегнуть к двойной привязке источников с указанием перед проведением

процедуры сохранения проекта для MPB глубины отслеживания источников в настройках ИС.



Так, например, для подмены аппаратного источника **PCL_channel#1** – входа АЦП на плате ввода/вывода **A-8111** генератором синусоиды достаточно в группе компонентов **Генераторы** в слое **Источники/Приемники** создать источник **Синусоида#1** и с помощью механизма **drag-and-drop** связать его с имеющимся уже источником **PCL_channel#1**. Подмена источников реализуется путем указания перед компиляцией проекта в настройках ИС **глубины отслеживания источников** равной **2**. Однако следует учесть, что при работе в реальном времени изменять глубину отслеживания источников нельзя.

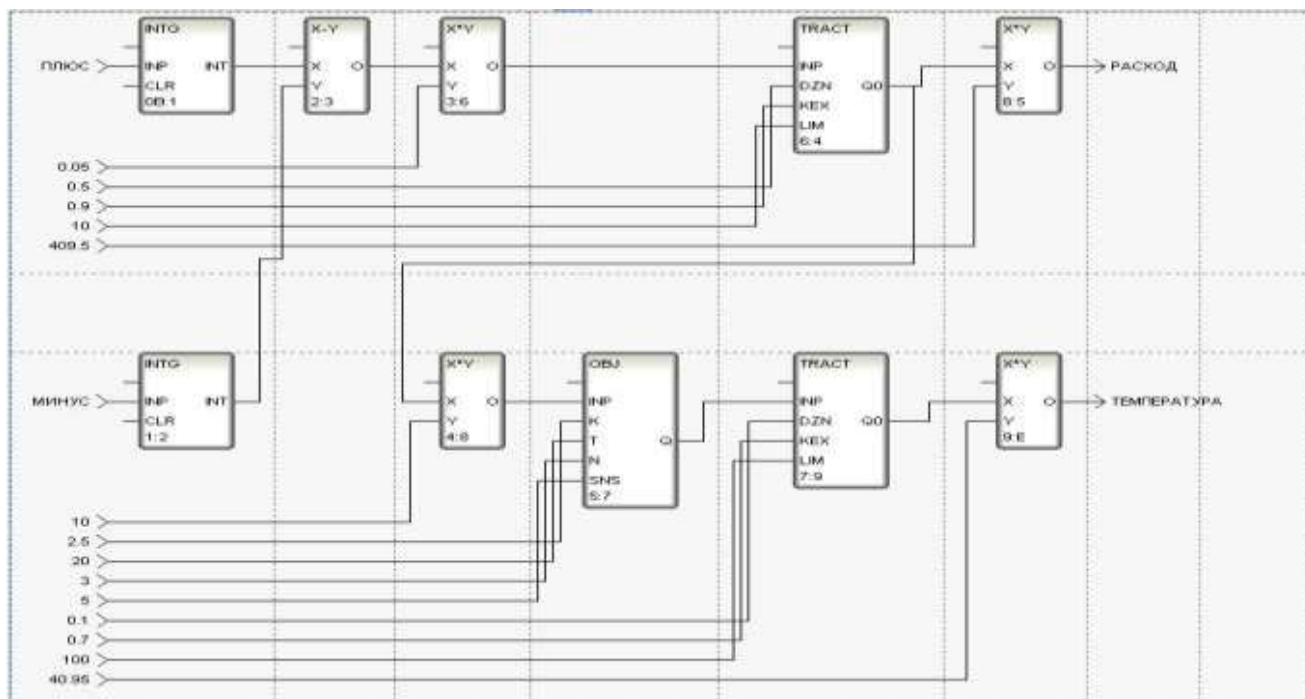
4.2. Разработка программ имитаторов, встраивание их в проект.

Далее необходимо приступить к созданию имитатора объекта участка термообработки. В слое **Шаблоны программ** создать программу **Имитатор_Термообработки** и задать ей следующие аргументы.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	флаги
ПЛУС	↓ IN	USINT			
МИНУС	↓ IN	USINT			
РАСХОД	↑ OUT	REAL			
ТЕМПЕРАТУРА	↑ OUT	REAL			

Входные аргументы **ПЛУС** и **МИНУС** – сигналы от ПДД-регулятора с ШИМ, а выходные аргументы **РАСХОД** и **ТЕМПЕРАТУРА** – соответственно расход теплоносителя и температура в аппарате. Построить математическую модель аппарата, основываясь на предположении, что его функционирование описывается классическим 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 назначить следующими: коэффициент передачи (усиления) 2.5, постоянная времени 20 (в тактах пересчета), запаздывание 3 (в тактах пересчета), а также наложить на его выход гармонические и случайные помехи. В качестве языка программирования применить **Техно FBD**.



Функциональные блоки **TRACT** применены для сглаживания и ограничения формируемых выходных значений, а функциональные блоки умножения **X*Y 8** и **9** – для приведения значений к выходному диапазону 12-разрядного АЦП. Откомпилировать созданную программу.

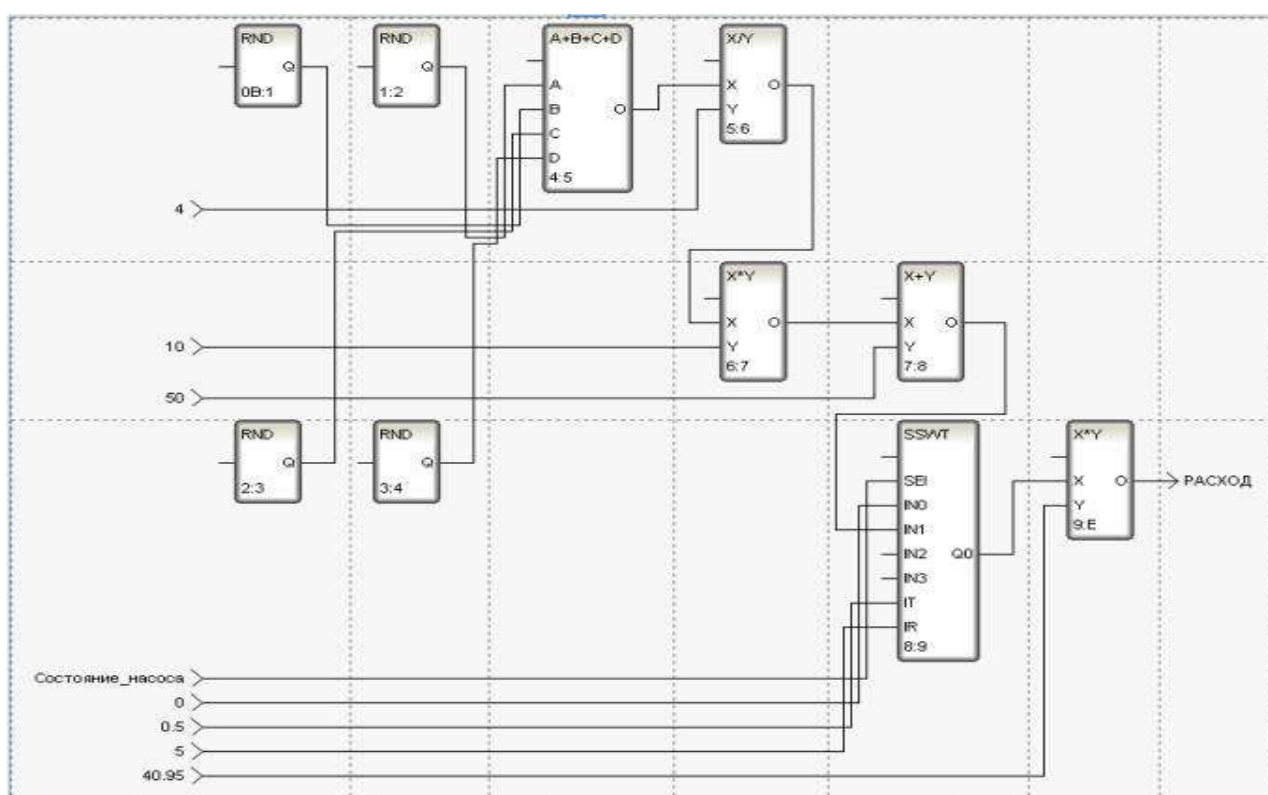
Открыть дополнительное окно Навигатора проекта на группе **Участок_Термообработки** узла PC-based контроллера **MicroRTM_2** и перетащить в него шаблон программы. Для созданного канала класса **Вызов Имитатор_Термообработки** выполнить привязку атрибутов каналов к аргументам.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
ПЛЮС	IN	USINT		ВЫХОД_ПЛЮС:Реальное значение(Система:MicroRTM_2.Участок_Термообработки)
МИНУС	IN	USINT		ВЫХОД_МИНУС:Реальное значение(Система:MicroRTM_2.Участок_Термообработки)
РАСХОД	OUT	REAL		Расход_теплоносителя:Входное значение(Система:MicroRTM_2.Участок_Термообработки)
ТЕМПЕРАТУРА	OUT	REAL		Входной_параметр:Входное значение(Система:MicroRTM_2.Участок_Термообработки)

Для имитации расхода продукта создать шаблон программы **Имитатор_Расход_Продукта**, для чего также воспользоваться языком **Техно FBD**. Аргументы программы определить следующим образом.

Имя	Тип	Тип данных	Значение по умолчанию
Состояние_насоса	IN	USINT	
РАСХОД	OUT	REAL	

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



Откомпилировать программу.

В дополнительном окне навигатора проекта выделить группу **Участок_Дозирования** узла **MicroRTM_2** и перетащить в него шаблон программы. Для созданного таким образом канала класса **Вызов Имитатор_Расход_Продукта** выполнить привязку атрибутов каналов к аргументам.

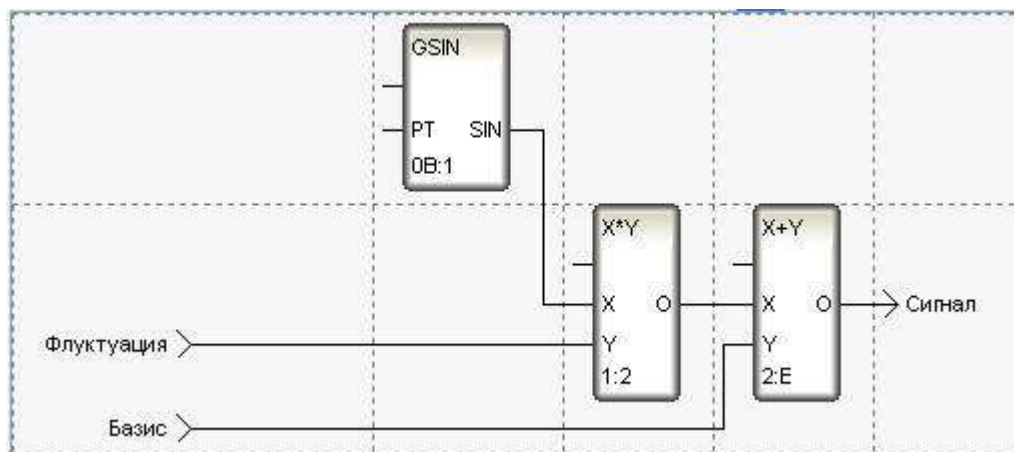
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Состояние_насоса	IN	USINT		Насос: Реальное значение[Система: MicroRTM_2. Участок_Дозирования]
РАСХОД	OUT	REAL		Расход_продукта: Входное значение[Система: MicroRTM_2. Участок_Дозирования]

Так как от PLC-контроллера Siemens S7-200 поступают данные как аналоговые, так и дискретные, то для их имитации на АРМ необходимо создать две соответствующие программы, причем варьируя значения аргументов, задаваемых как константы – в поле **Значение по умолчанию** табличного редактора аргументов, устанавливать величины параметров и их "случайные" флуктуации.

Создать в слое **Шаблоны программ** новую программу – **Имитатор_УХ_Аналог**. Объявить для нее следующие аргументы.

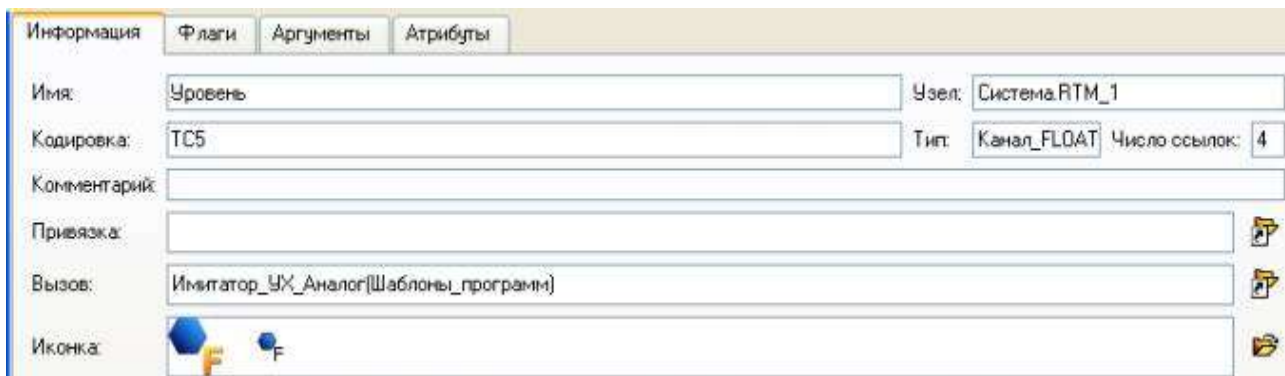
Имя	Тип	Тип данных	Значение по умолчанию
Сигнал	↑ OUT	REAL	
Флуктуация	↓ IN	REAL	
Базис	↓ IN	REAL	

Программу разработать на языке программирования **Техно FBD**.

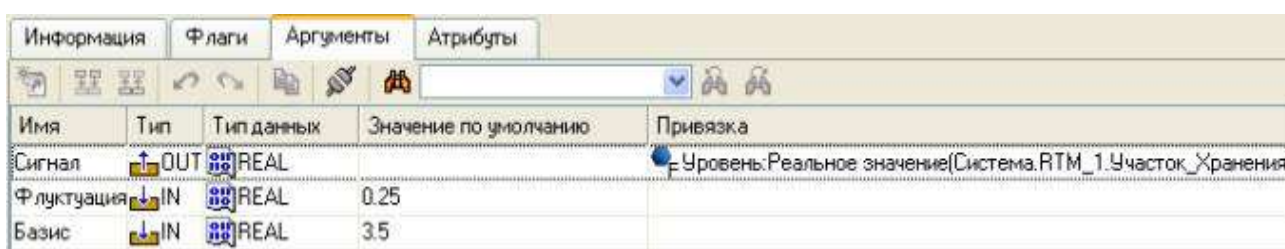


Изменяющийся по синусоидальному закону "случайный" компонент в диапазоне (-1,1) масштабировать, задавая аргумент **Флуктуация**, и добавлять к аргументу **Базис**, определяющему установившееся значение. Откомпилировать программу нажатием функциональной клавиши **F7**.

Для привязки разработанной программы выделить в окне Навигатора проекта группу **Участок_Хранения** узла **RTM_1**. Выделить канал **Уровень** и открыть его свойства. Во вкладке **Информация** определить свойство канала **Вызов** путем указания в диалоге шаблона программы **Имитатор_УХ_Аналог**.

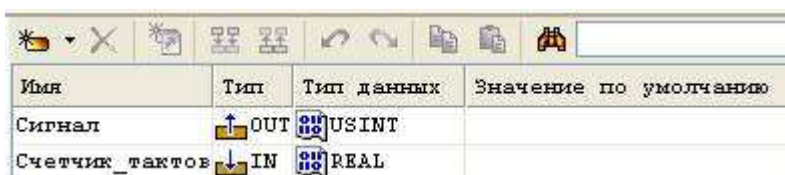


Перейти в появившуюся после выполненного действия вкладку **Аргументы**, привязать аргументы программы к атрибуту канала и задать константы в столбце **Значения по умолчанию**.

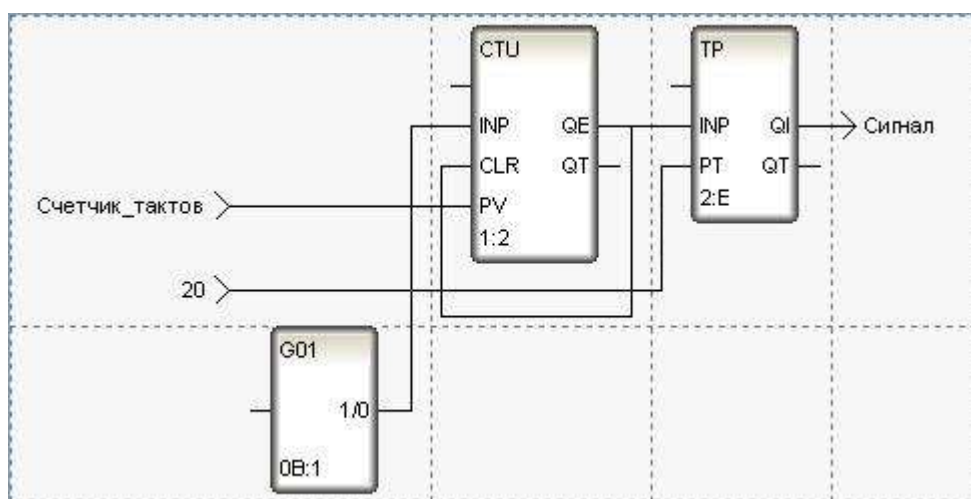


Подобным образом поступить в отношении каналов **Температура**, **Давление** и **Влажность**, указав для них соответственно аргументы **Флуктуация** и **Базис** как (0.25,24), (0.025,0.98) и (0.2,70).

Для имитации дискретных сигналов создать шаблон программы **Имитатор_УХ_Дискрет** с заданными аргументами.

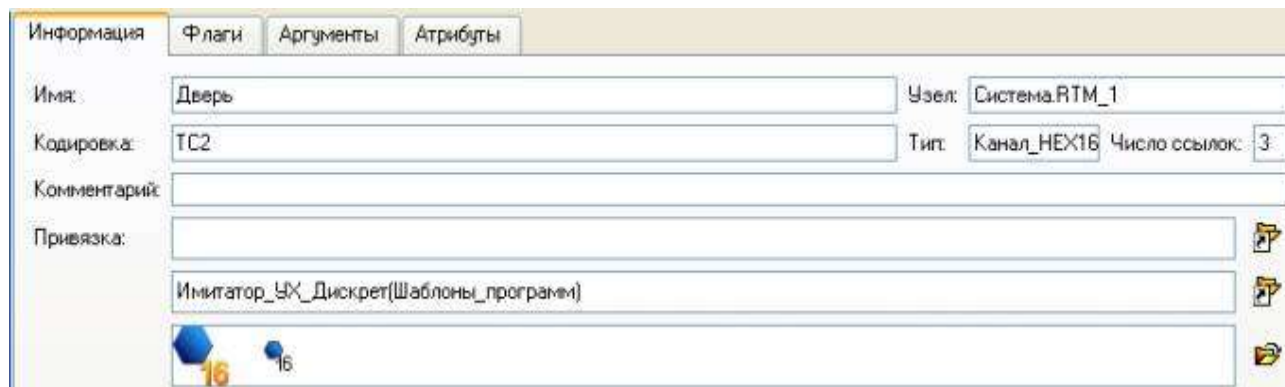


Разработанная на **Техно FBD** программа будет иметь следующий вид.

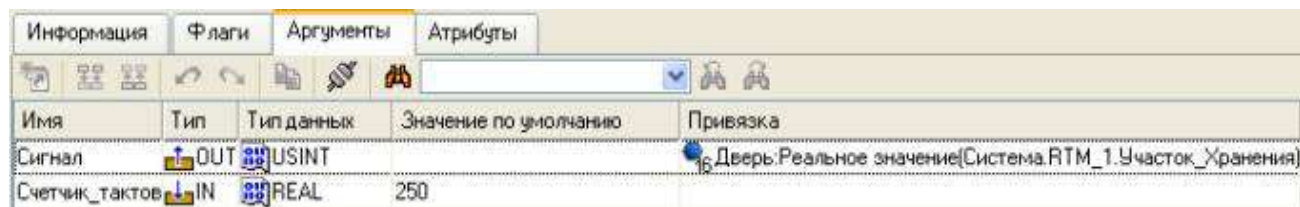


Данная программа будет периодически имитировать срабатывание дискретного датчика. Задавая входным аргументом **Счетчик_тактов** количество тактов, поступающих от генератора **G01**, до момента срабатывания датчика (установки триггера **TR**), при его достижении выходной аргумент **Сигнал** будет удерживаться в состоянии **1** в течение последующих 20 тактов, затем произойдет сброс его в состояние **0**, а далее картина будет повторяться.

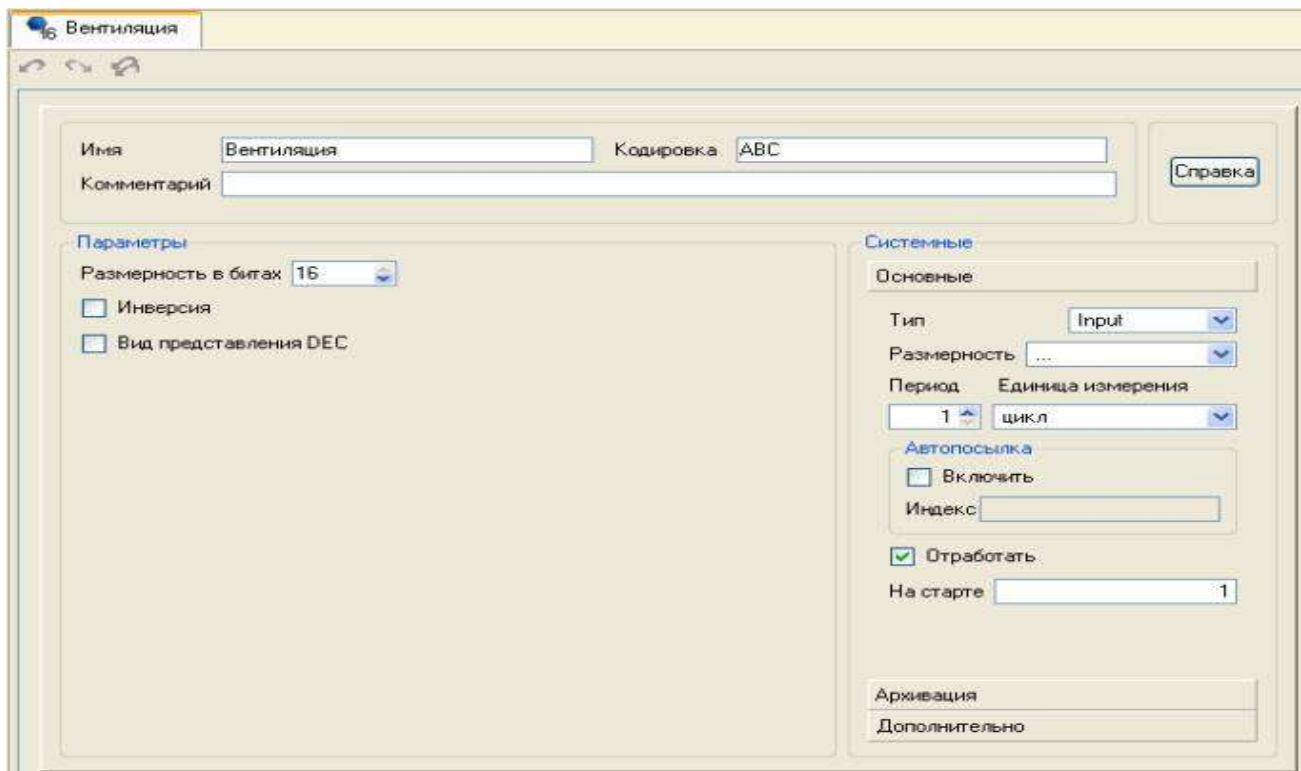
После компиляции программы выделить канал **Дверь** в группе **Участок_Хранения** узла **RTM_1** и в бланке **Информация** определить свойство **Вызов** следующим образом.




Во вкладке **Аргументы** связать выходной аргумент программы с атрибутом **Реальное значение** текущего канала и задать период "срабатывания" датчика в качестве значения по умолчанию.



Для канала **Пожарная_сигнализация** поступить аналогичным образом, определив для него периодичность срабатывания как 1000. В отношении канала **Вентиляция** поступить несколько иначе – задать ему постоянное значение, соответствующее включенному состоянию системы вентиляции.



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

Сохранить созданный проект на диск с именем **DEMO_PRJ.prj** и, затем, с помощью щелчка ЛК мыши на иконке  панели инструментов создать файлы для загрузки на целевые платформы для проведения отладки проекта.

5. Контрольные вопросы.
- 5.1. Как разработать программу имитатора?
- 5.2. Как встроить в проект программу имитатора?

ЛАБОРАТОРНАЯ РАБОТА № 11

Отладка многоузлового проекта

1. Цель работы – освоение принципов отладки многоуровневого проекта.

2. Задание по лабораторной работе: произвести отладку многоузлового проекта без использования аппаратных средств ввода/вывода.

3. Методические указания по выполнению лабораторной работы.

3.1. Выполнить настройку сетевых протоколов (п. 4.1).

3.2. Подготовить папки проекта к отладке (п. 4.2).

3.3. Провести отладку проекта (п. 4.3).

4. Процедура отладки многоузлового проекта.

4.1. Настройка сетевых протоколов.

Отладку многоузлового проекта без использования аппаратных средств ввода/вывода можно проводить следующим образом:

- последовательно перетащить узлы проекта в слой **Технология**;

- создать в слое **Система** новый узел, например, **RTM_Common** ;

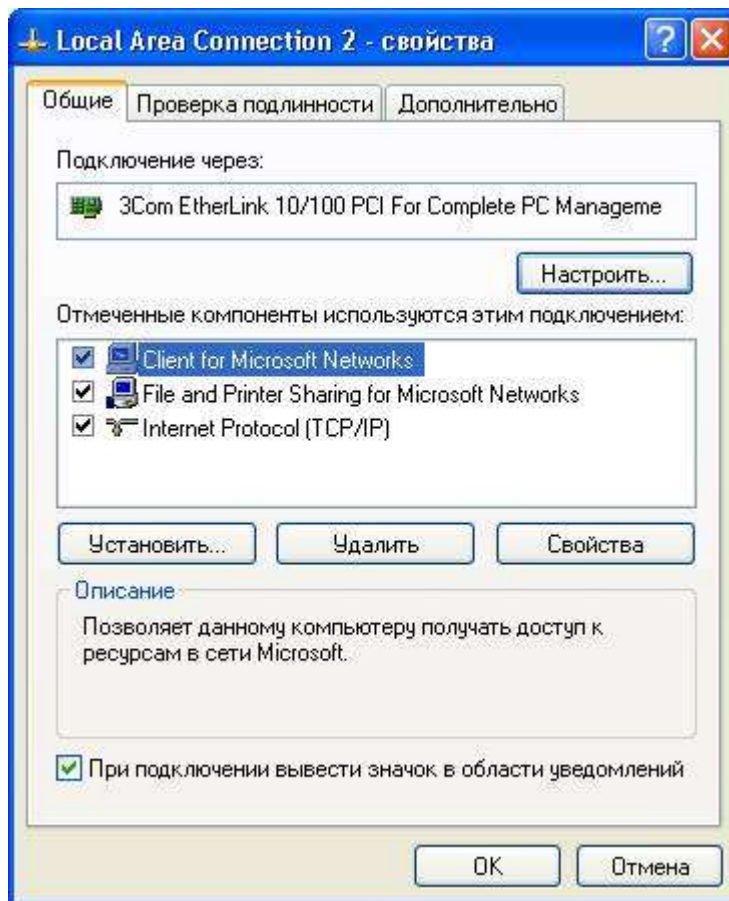
- перетащить слой **Технология** во вновь созданный узел в слое **Система**;

- выполнить для данного узла команду **Сохранить для MPB** из контекстного меню;

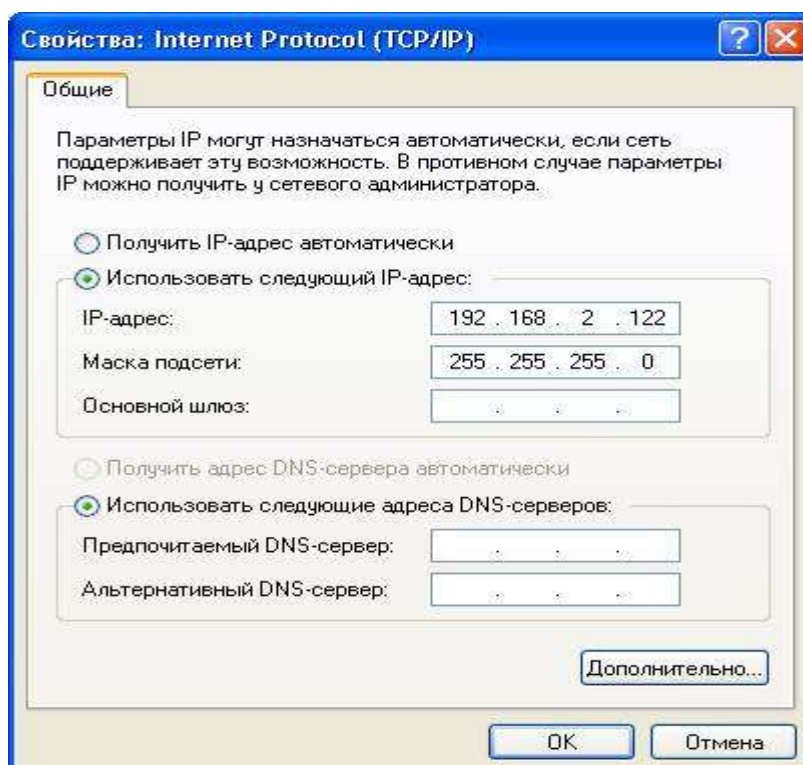
- запустить узел **RTM_Common** на исполнение, для просмотра атрибутов каналов использовать встроенные возможности отладчиков с поддержкой/без поддержки графических экранов;

- после проведения отладки и внесения необходимых изменений, производимых в "реальных" узлах проекта, узел **RTM_Common** можно удалить.

Для запуска проекта в демо-режиме необходимо иметь два компьютера с установленной на них **ОС MS Windows**, которые объединены в сеть с настроенным протоколом **TCP/IP**.

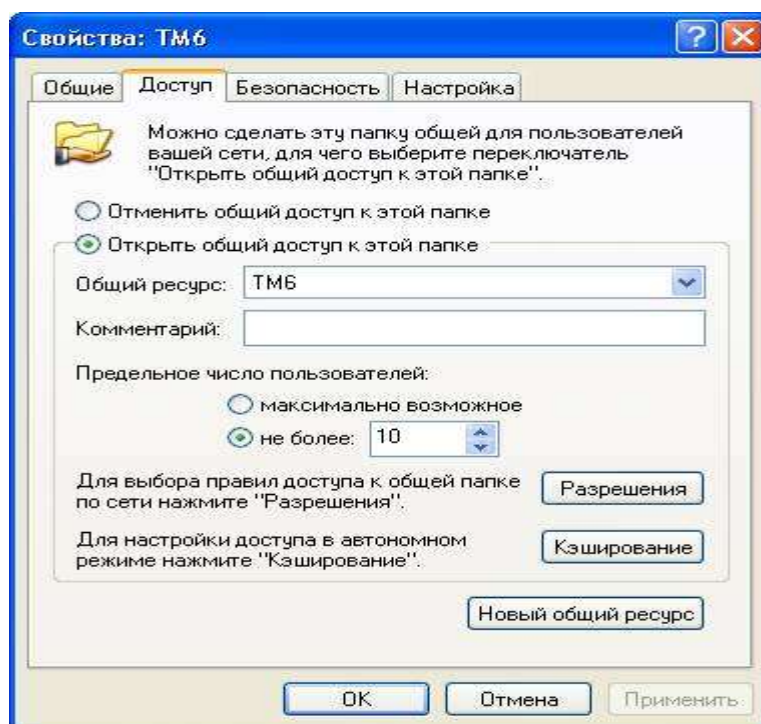


При настройке протокола **TCP/IP** необходимо задать реальные IP-адреса компьютеров, объединенных в сеть, например, для компьютера, предназначенного для запуска узла **RTM_1**, IP-адрес 192.168.2.122, а для узла **MicroRTM_2** – 192.168.2.123.




4.2. Подготовка папки проекта к отладке.

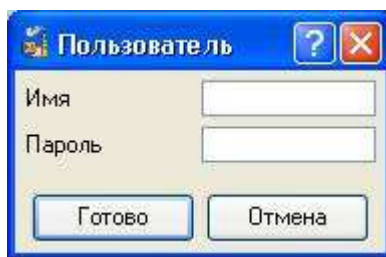
Для осуществления доступа к файлам, созданным для узлов проекта, с других компьютеров рекомендуется сделать папку, содержащую инструментальную систему **TRACE MODE 6**, общедоступной, задав при этом разрешения на доступ к ней пользователям, входящим в систему на других компьютерах, в разделе **Разрешения** бланка.






Это позволит запускать на удаленных компьютерах отладчики без установки на них **TRACE MODE 6** – с поддержкой графических экранов **rtc.exe** и без поддержки графических экранов **rtmg32.exe** (например, для узла **MicroRTM_1**).

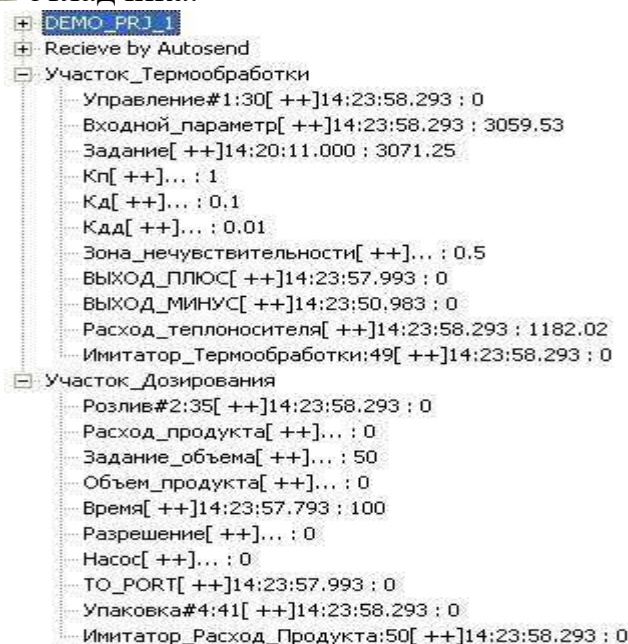
4.3 Проведение отладки.

На компьютере с установленной инструментальной системой **TRACE MODE 6** запустить отладчик с поддержкой графических экранов двойным щелчком ЛК мыши на иконке рабочего стола Windows  или с помощью проводника Windows в текущей директории – **C:\TM6**. В открывшемся окне отладчика в пункте **Файл** основного меню выбрать **Открыть** и указать файл **DEMO_PRJ_0.dbb** из подкаталога **\RTM_1** для загрузки узла АРМ проекта. После загрузки на экран будет выведен диалог входа в систему, в котором необходимо задать определенные на этапе разработки **Имя** и **Пароль** введенных в проект пользователей – **admin** или **oper**.

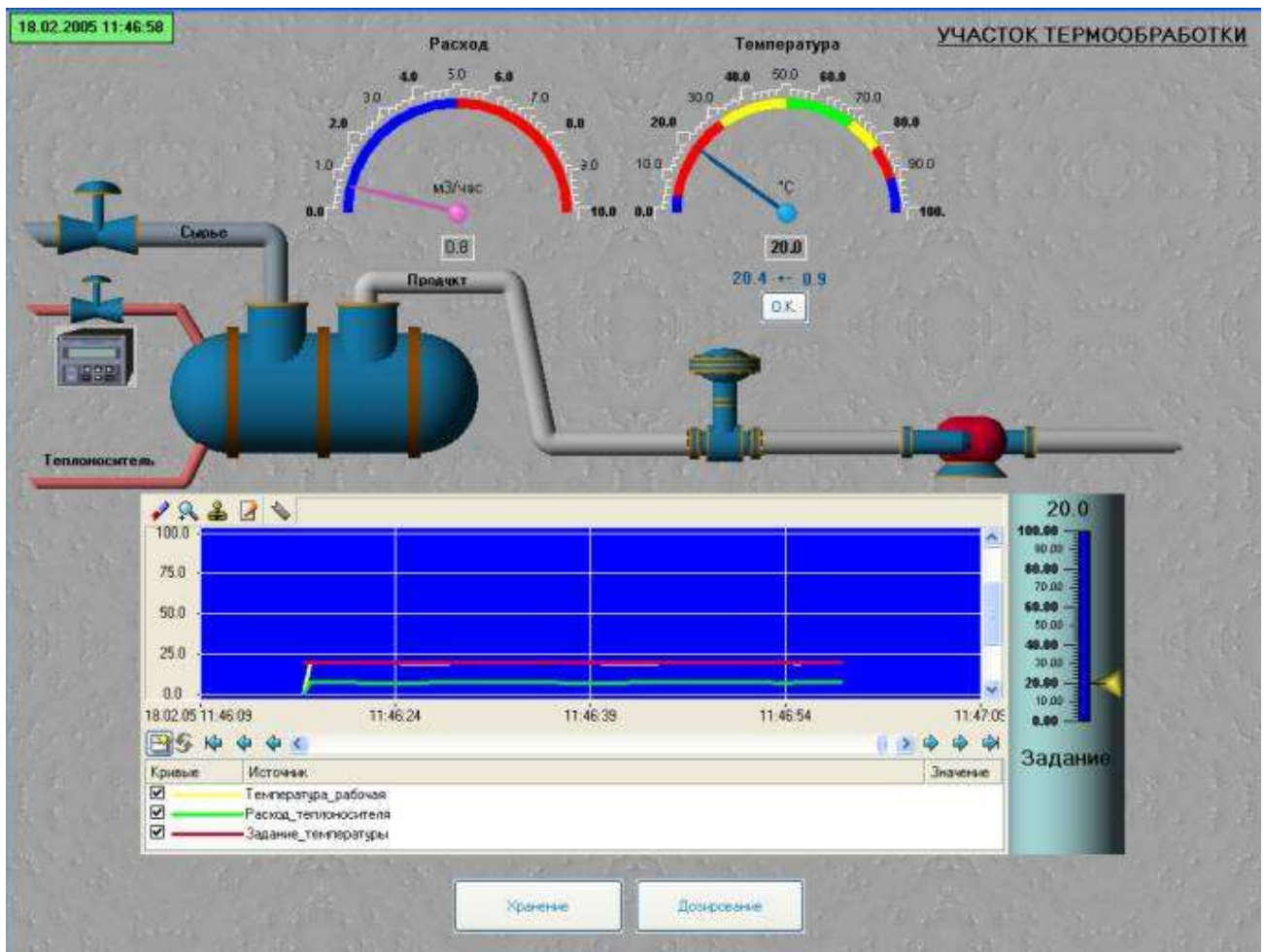


Для запуска узла можно использовать комбинацию клавиш **Ctrl+R** или воспользоваться иконкой  панели инструментов отладчика. Запустить узел на исполнение можно также, находясь в интегрированной среде разработки, путем выделения ЛК мыши в слое **Система** узла **RTM_1** и нажав затем ЛК иконку  на панели инструментов.

Для запуска узла PC-based контроллера необходимо на удаленном компьютере с помощью проводника Windows войти в открытую для доступа папку \\Имя_компьютера_с_установленной_ИС\TM6 и запустить в ней файл **rtmg32.exe**, указав в пункте меню **Открыть** файл **ДЕМО_PRJ_1.dbb** из подкаталога **\MicroRTM_2**. Запуск на исполнение производится щелчком ЛК мыши по иконке  отладчика.



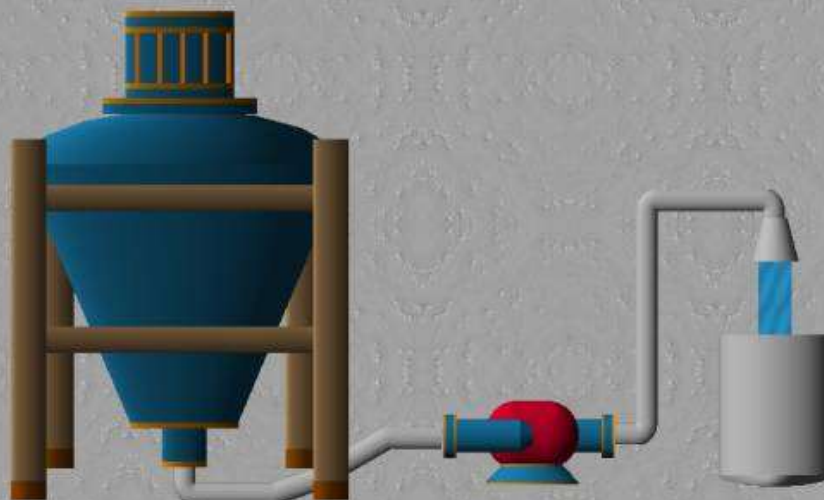
В случае если сетевые настройки указаны верно, графический экран узла АРМ, отображающий участок термообработки, "оживет". Меняя задание регулятору и его настройки, на тренде можно наблюдать реакцию модели объекта.



Перемещаясь по экранам, можно производить "мониторинг" участка хранения, "отпускать" готовый продукт, фиксировать "произошедшие" события и т.д.

18.02.2005 13:24:20

УЧАСТОК ДОЗИРОВАНИЯ

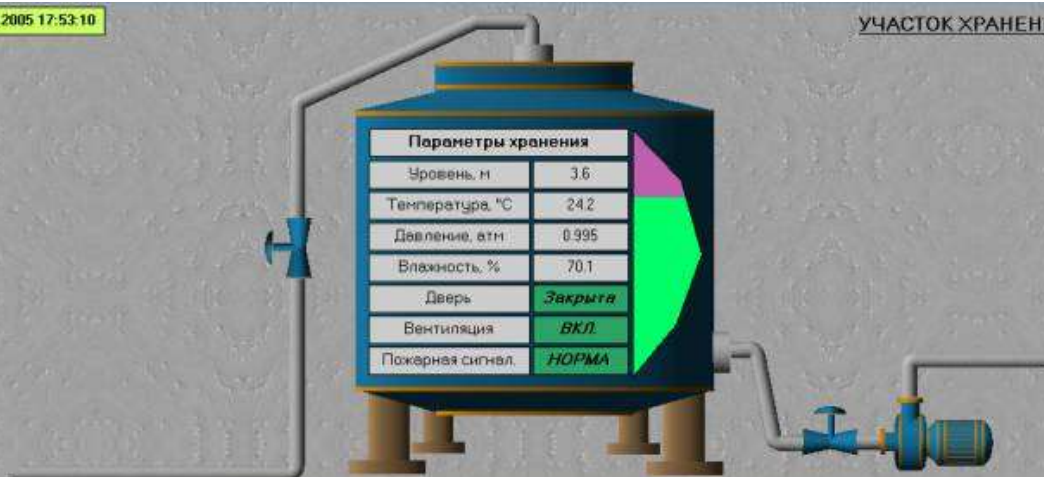


Расход продукта	56.8
Задание объема	50.0
Объем продукта текущий	19.1
ОТПУСТИТЬ ПРОДУКТ	

Термообработка Хранение

17.02.2005 17:53:10

УЧАСТОК ХРАНЕНИЯ



Параметры хранения	
Уровень, м	3.6
Температура, °C	24.2
Давление, атм	0.995
Влажность, %	70.1
Дверь	Закрота
Вентиляция	ВКЛ
Пожарная сигнал	НОРМА

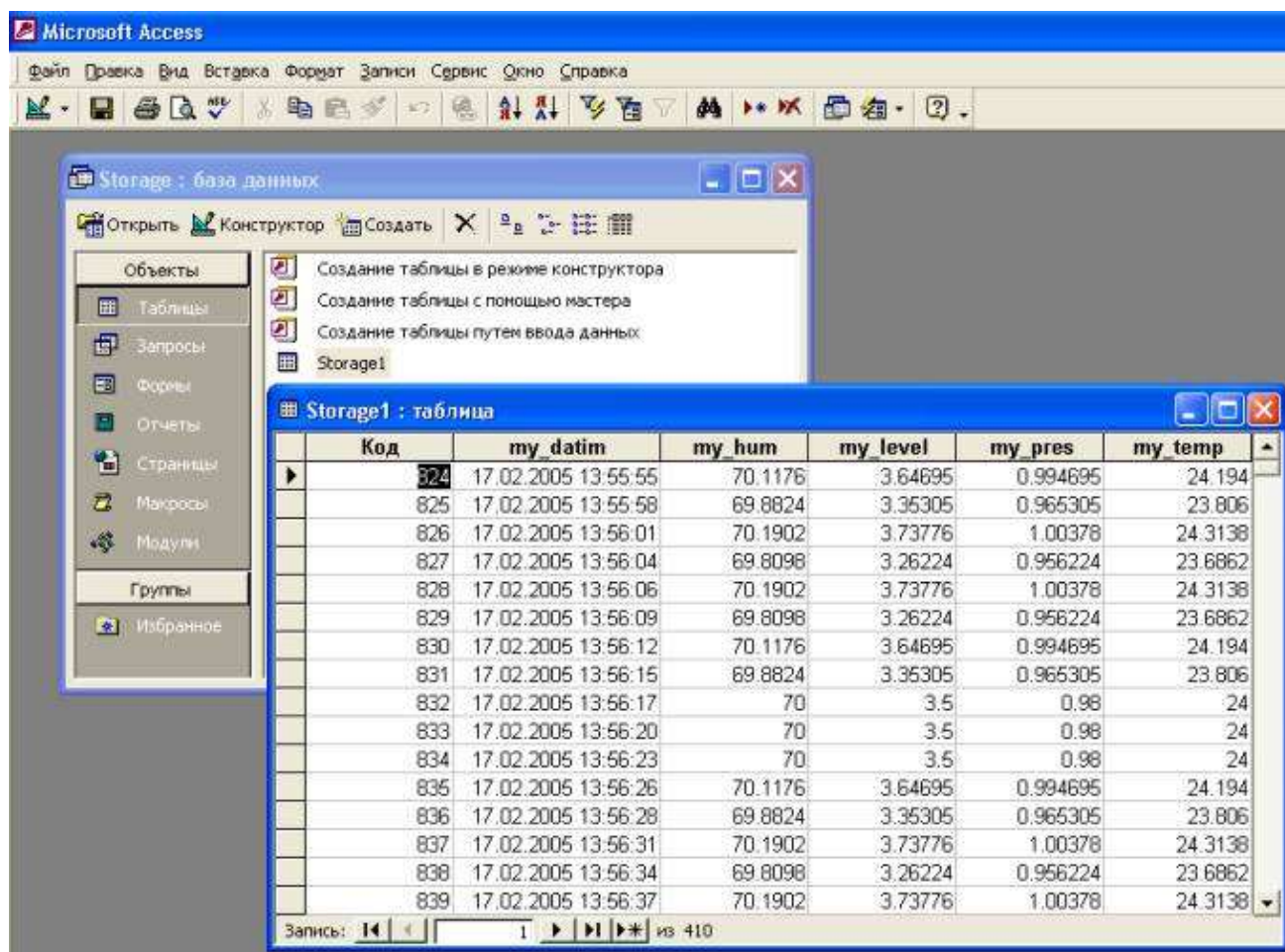
Events			
Имя	Время возникновения	Время исчезновения	Время калитрования
<input checked="" type="checkbox"/> Событие_Дверь	17.02.2005 17:51:19.297	17.02.2005 17:51:30.313	17.02.2005 17:53:06
<input type="checkbox"/>	17.02.2005 17:51:19.297	17.02.2005 17:51:30.313	...
<input type="checkbox"/>	17.02.2005 17:46:43.844	17.02.2005 17:46:54.960	...
<input checked="" type="checkbox"/> Событие_Венти...	17.02.2005 17:42:09.0		
<input type="checkbox"/> Событие_Пожар	...		

Термоподготовка Дозирование


Данные, записываемые в отчет тревог узла АРМ – файл **alarms.txt**, доступны для просмотра, например, с помощью **Блокнота**:

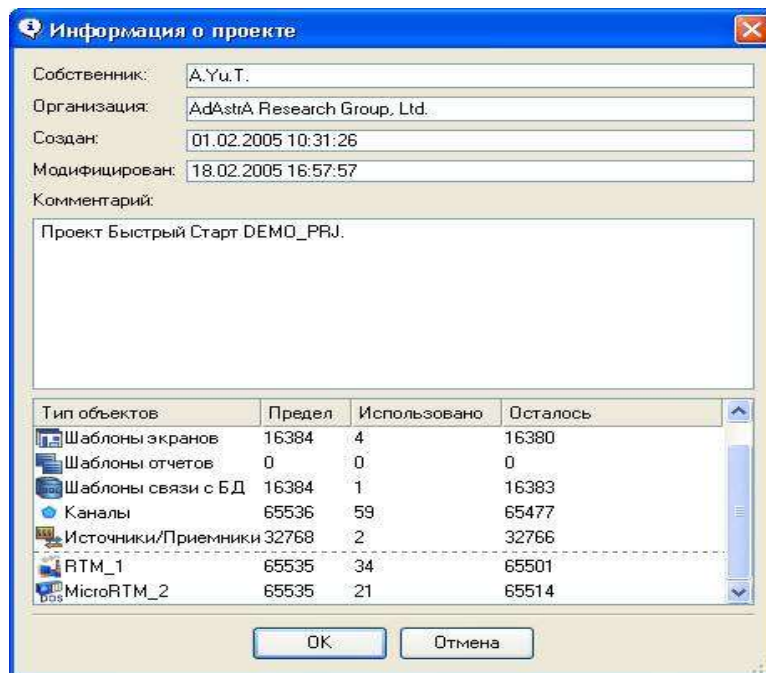
17.02.2005 17:42:02.0	admin	Login	
17.02.2005 17:42:07.0	DEMO_PRJ_0	START	
17.02.2005 17:46:43.8	М Дверь	TC2	Дверь в хранилище открылась
17.02.2005 17:46:54.8	М Дверь	TC2	Дверь в хранилище закрылась
17.02.2005 17:51:19.2	М Дверь	TC2	Дверь в хранилище открылась
17.02.2005 17:51:30.3	М Дверь	TC2	Дверь в хранилище закрылась

Записи, заносимые в БД по участку хранения, можно просматривать, в том числе, и в реальном времени.



Завершить работу узла можно щелчком ЛК мыши на иконке .

Для получения информации, необходимой для выбора "мощности" МРВ (по количеству каналов) узлов разработанного проекта, предназначена иконка  панели инструментов. При нажатии на нее ЛК открывается окно, отображающее как общее количество компонентов проекта, так и их распределение по узлам.



5. Контрольные вопросы.

5.1. Какова процедура настройки сетевых протоколов?

5.2. Как подготовить папку проекта к отладке?

5.3. Как провести отладку проекта?

ОГЛАВЛЕНИЕ

Общие организационно – методические указания	3
Перечень рекомендуемой литературы	5
Лабораторная работа №1. Создание простейшего проекта в пакете TRACE MODE	6
Лабораторная работа №2. Добавление в рабочий проект функции управления	16
Лабораторная работа №3. Связь по протоколу DDE с приложением MS Windows на примере Excel	28
Лабораторная работа №4. Построение операторского интерфейса: мониторинг, управление, регулирование	31
Лабораторная работа №5. Написание программ на языках программирования ПЛК	60
Лабораторная работа №6. Создание узлов проекта и баз каналов	67
Лабораторная работа №7. Создание архива и отчета тревог	72
Лабораторная работа №8. Организация вывода времени на графических экранах и фиксация событий	82
Лабораторная работа №9. Связь с СУБД MS Access	86
Лабораторная работа №10. Использование имитаторов аппаратных средств	98
Лабораторная работа №11. Отладка многоузлового проекта	106

Учебное издание

Владимир Иванович Устич

ПРОЕКТИРОВАНИЕ SCADA - СИСТЕМ

Редактор Г. А. СМИРНОВА

Подписано в печать 14.04. 2019 г. Формат 60х90 (1/16). Уч.-изд. л.5,3 .
Печ. л. 7,25 . Тираж 50 экз. Заказ

Издательство федерального государственного бюджетного
образовательного учреждения
высшего образования
"Калининградский государственный технический университет"
236022, Калининград, Советский проспект, 1