

СОЗДАНИЕ АСУТП НЕФТЕПЕРЕРАБАТЫВАЮЩЕЙ СТАНЦИИ

Задание на разработку модели технологического объекта управления

Цель работы: разработка модели технологического объекта управления (ТОУ). В качестве ТОУ предлагается нефтеперерабатывающая станция, состоящая из двух ёмкостей, соединённых трубой с клапаном.

В процессе выполнения лабораторной работы требуется:

1. Создать графический элемент (ГЭ), моделирующий процесс управления уровнем жидкости в первой ёмкости (резервуар). Уровень в резервуаре устанавливается пользователем.

2. Разместить на экранной форме анимационные эффекты, отображающие события, происходящие в ТОУ.

2.1. При уровне жидкости в резервуаре, превышающей заданную границу, например, 20 единиц, в трубе, выходящей из резервуара, слева от клапана требуется включить анимацию первого потока нефти.

2.2. При уровнях жидкости выше 70 и 90 единиц требуется автоматически включить соответствующие анимации утечки нефти, и вывести на экран соответствующие сообщения об аварии.

2.3. Разместить на экранной форме графический элемент (ГЭ), моделирующий работу управляемого клапана.

2.4. При уровне жидкости в резервуаре больше заданной величины и при открытом клапане, требуется включить анимацию второго потока (справа от клапана).

2.5. При выполнении условий пункта 2.4. автоматически включить анимацию заполнения нефтью второй ёмкости, предназначенной для первичной обработки.

2.6. После заполнения второй ёмкости клапан требуется автоматически закрыть и над нагревателем включить анимацию, соответствующую этому событию.


2.7. После окончания заполнения второй ёмкости требуется выполнить моделирование слива нефти из второй ёмкости потребителю, например, в бензовоз.

3. Создать архивы технологической информации и организовать запись в них сообщений об уровне жидкости в первой ёмкости и о состоянии клапана. Изменения уровня жидкости во времени отображать с помощью графического элемента «ТРЕНД».


СОЗДАНИЕ ПРОЕКТА SCADA-СИСТЕМЫ

Загрузка интегрированной среды разработки и создание узла проекта

Разработка проекта ведется в интегрированной среде (ИС) Trace Mode 6 московской фирмы АДАСТРА. Загрузка интегрированной среды разработки проекта Trace Mode

6 осуществляется двойным щелчком левой клавишей (ЛК) мыши по иконке  рабочего стола Windows, либо из меню Пуск - Trace Mode 6(base). Если при этом открывается диалог с вариантами уровня сложности: **Простой/Стандартный/Комплексный**. Для реализации данного проекта достаточно выбрать уровень **"Простой"**.

Если этот диалог не открывается, то значит по умолчанию в настройках ИС уже задан уровень сложности. В лаборатории кафедры АТ установлен стандартный уровень.

Затем требуется создать проект щелчком ЛК мыши на экранной кнопке "Создать" . В результате в левом окне навигатора проекта появится дерево проекта с predetermined информационными слоями, соответствующими выбранному уровню сложности.

Затем в навигаторе проекта необходимо создать узел в слое "Система". Для создания узла щелчком правой клавиши (ПК) мыши на строке навигатора "Система" вызываем контекстное меню, в котором выбираем опцию «Создать узел» - RTM как показано на рисунке 1.

Примечание: узел проекта - это информационная структура, состоящая из набора компонентов проекта. Узлы проекта создаются как корневые группы информационного слоя **Система**. Предопределенное название узла указывает на семейство мониторов реального времени (MPV), для которых данный узел предназначен (см. [Модификации мониторов \[1\]](#)). Узел может содержать только те компоненты, которые поддерживаются мониторами соответствующего семейства. В общем случае, узлы могут выполняться под управлением различных мониторов.

Как правило, узел выполняется на отдельном аппаратном средстве (контроллер, компьютер). В случае запуска двух и более узлов на одном аппаратном средстве оно должно быть оборудовано соответствующим количеством сетевых карт.

Параметры узлов задаются в соответствующем редакторе.

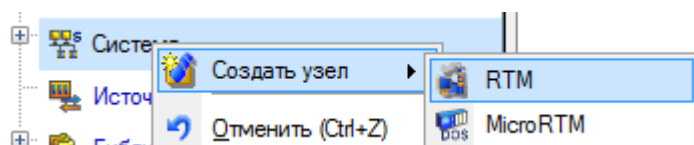


Рисунок 1- Создание узла

В результате будет создан узел с именем RTM_1, который ещё не заполнен компонентами проекта, то есть пустой узел. В процессе разработки проекта в узле разместим каналы, источники/приёмники, наборы ресурсов, словари сообщений и другие компоненты. Имя узла можно задать любое по смыслу проекта. Как правило, имя узла соответствует названию технологического объекта, например, бункер, транспортёр, печь, мельница.

Создание и редактирование экрана оператора АСУТП

Создадим экран с помощью контекстного меню узла RTM_1. По умолчанию экрану автоматически присваивается имя "Экран#1". Переименуем экран, присвоим ему имя "Станция" как показано на рисунке 2.

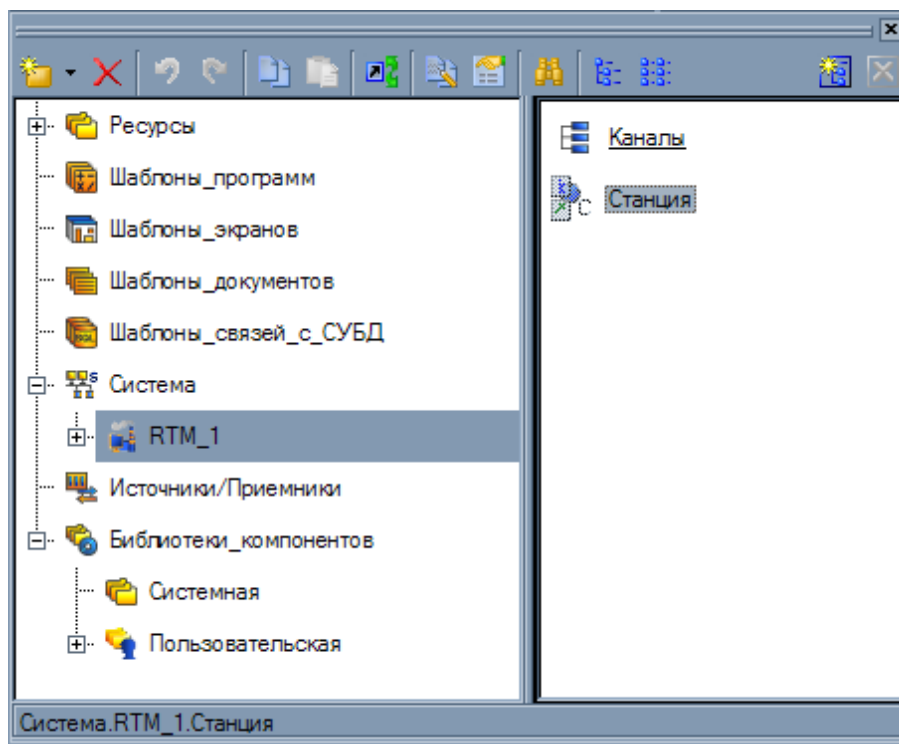


Рисунок 2- Создание экрана

Примечание: созданный экран будет находиться не в узле, а в слое «Шаблоны экранов». В слое «Система» в узле RTM_1 будет находиться только канал типа CALL, вызывающий шаблон экрана. При разработке крупных проектов, состоящих из большого количества узлов, экраны создают в слое «Шаблоны экранов», а не в узле. Это связано с тем, что экранная форма может использоваться в нескольких узлах. Поэтому не имеет смысла одну и ту же экранную форму создавать в каждом узле заново. После разработки всех экранных форм, предусмотренных проектом и размещённых в слое «Шаблоны экранов», их перетаскивают ЛК методом «**drag and drop**» в соответствующий узел проекта. Эта технология называется разработкой от шаблонов.

Таким образом, в узле проекта RTM_1 появился канал "Экран#1" типа CALL со свойством «вызов», которому мы присвоили имя «Станция». Двойным щелчком ЛК на канале **Станция** (рисунок 2) откроем окно графического редактора. Двойной щелчок ЛК по рабочему полю экрана позволяет выбрать настройки фона и размеров экрана (рисунок 3).

Параметры экрана:

Текущий слой: Слой

Копировать Вставить Заменить

Свойство	Значение
<u>Размеры</u>	1024x768
Масштабировать содержимое	True
<u>Фон</u>	Цвет
Цвет	
Положение источника света (%) X	25
Положение источника света (%) Y	25
Код доступа	0x0
Горячая клавиша	Нет
<u>Всплывающее окно</u>	False
Загружать	False
Показать в меню экранов	False
<u>Верхний объект</u>	False
<u>Нижний объект</u>	False
<u>Левый объект</u>	False
<u>Правый объект</u>	False

Рисунок 3- Редактирование экрана

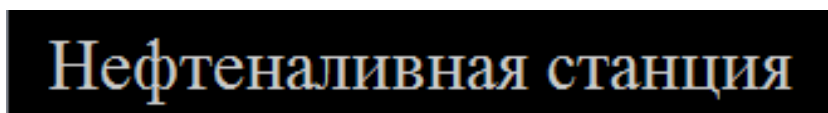
Размер экрана выбирают в соответствии с техническими характеристиками применяемых аппаратных средств таким образом, чтобы максимально использовать графические возможности дисплея. Цвет фона экрана и цвет располагаемых на нём графических элементов необходимо подбирать в соответствии с требованиями эргономики.

Размещение графических элементов на экране «СТАНЦИЯ»

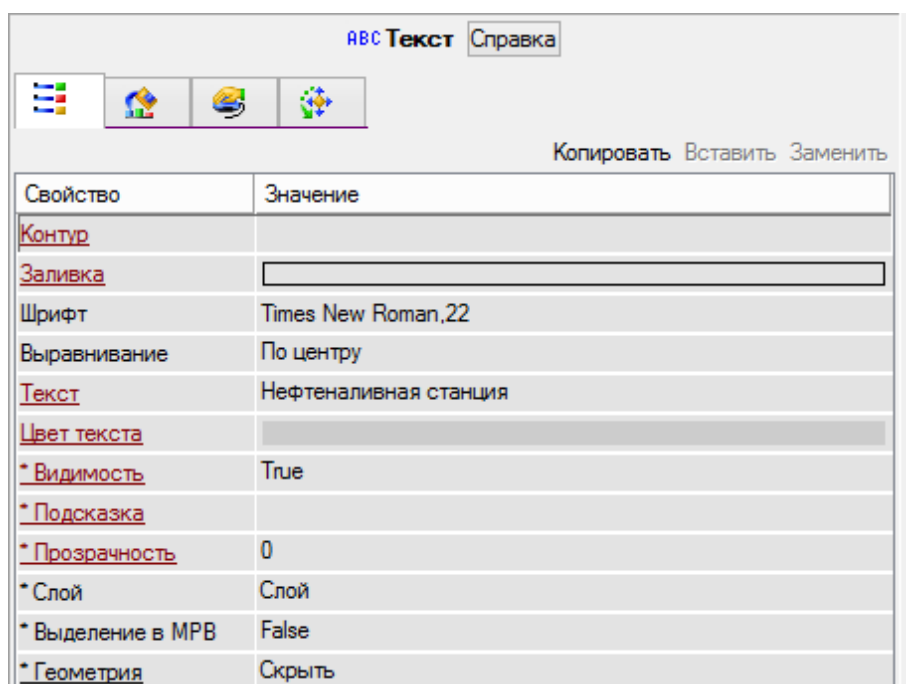
Разместим в верхней части экрана статический текст "Нефтеналивная станция". Для этого воспользуемся инструментальной панелью "Графические элементы":



Статический текст на панели обозначается иконкой "ABC". Выделим её и разместим прямоугольник на экране. Затем в свойствах укажем необходимые параметры. В частности, зададим размер и вид шрифта, цвет текста и фона, а также уберем контур вокруг текста. Внешний вид полученной надписи может быть, например, таким:




Вариант настройки свойств графического элемента «текст» показан на следующем рисунке:



Далее, с помощью группы элементов "Объемные фигуры" создадим на экране необходимые графические элементы (ГЭ). Выбор конкретного вида графического элемента осуществляется из списка, выпадающего по щелчку ЛК по иконкам панели "Графические элементы".



Размещение графического элемента «ёмкость»

Выберем графический элемент (ГЭ) "ёмкость" , после чего с помощью ЛК разместим элемент на экране. Выделив полученный ГЭ, можно изменить его размеры и свойства. Свойства редактируются в таблице свойств, показанной на рисунке 4 слева от ГЭ. Результат настройки ГЭ приведен на рисунке 4 справа.

Разместив остальные графические элементы и предварительно подписав их, получаем экранную форму, показанную на рисунке 5. Свойства графических элементов, такие как материал и прозрачность, идентичны свойствам, показанным на предыдущем рисунке.

Свойства графических элементов, дизайн экранной формы студенты могут выбирать самостоятельно с учетом требований эргономики.

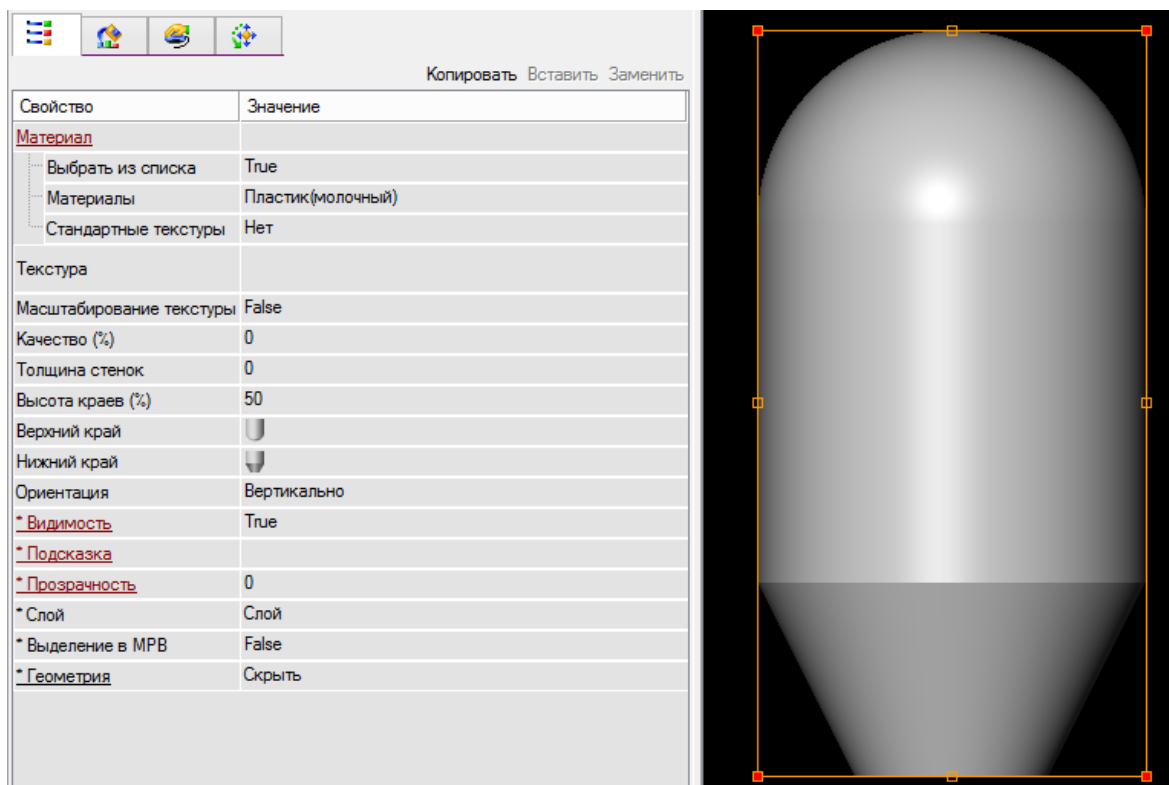


Рисунок 4- Редактирование графического элемента «ёмкость»

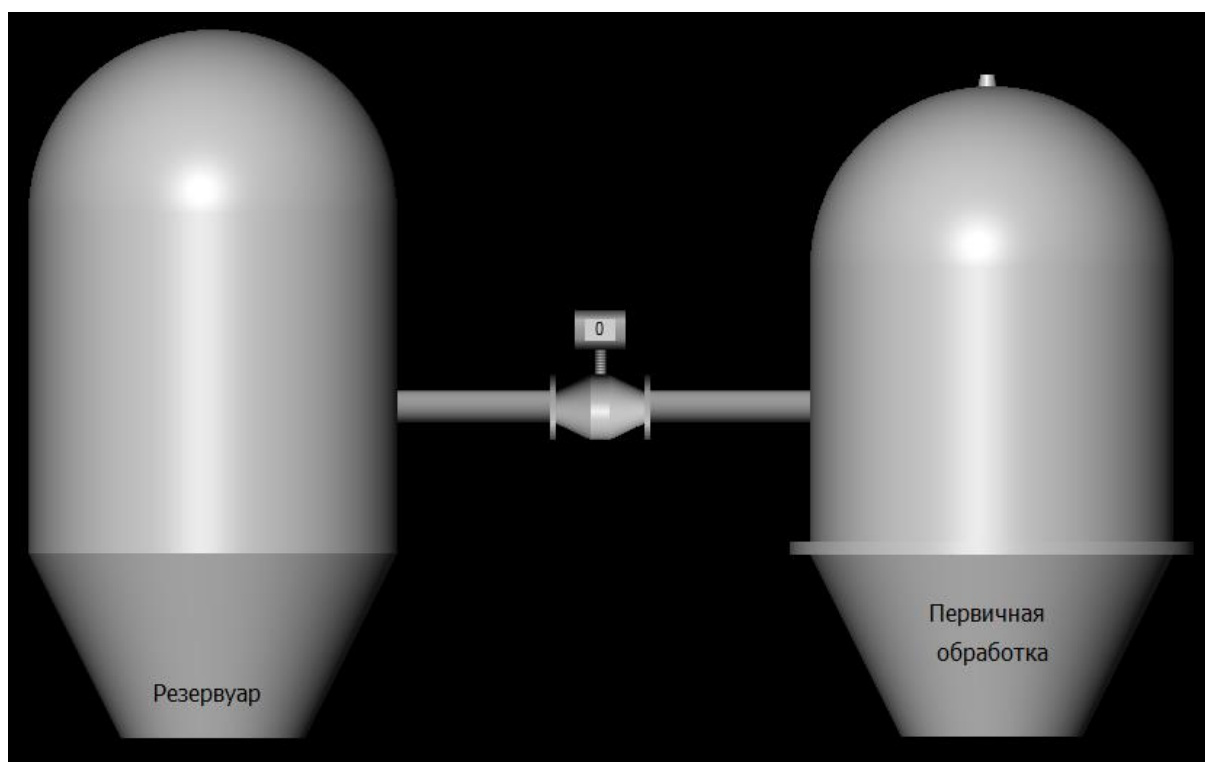



Рисунок 5- Вариант оформления графического экрана

Моделирование процесса изменения количества нефти в резервуаре выполним с помощью ГЭ "Ползунок". Значение уровня нефти в резервуаре, задаваемое "Ползунком", в дальнейшем свяжем с соответствующим каналом.

Графический элемент "Ползунок" разместим поверх резервуара, выбрав на панели значок "Приборы" . Пример настройки свойств ползунка показан на рис.6 справа. Свойства разделены на несколько групп: группа свойств "Ползунок" задает параметры ползунка (размер, форма), группа свойств "Полоса" - цвет и размер секций на шкале, "Шкала" - размер штрихов, шрифт, число десятичных знаков и другие параметры подписей к шкале, "Рамка" - свойства рамки этого объекта.

После задания соответствующих свойств получим изображение, показанное на рисунке 6 (справа).

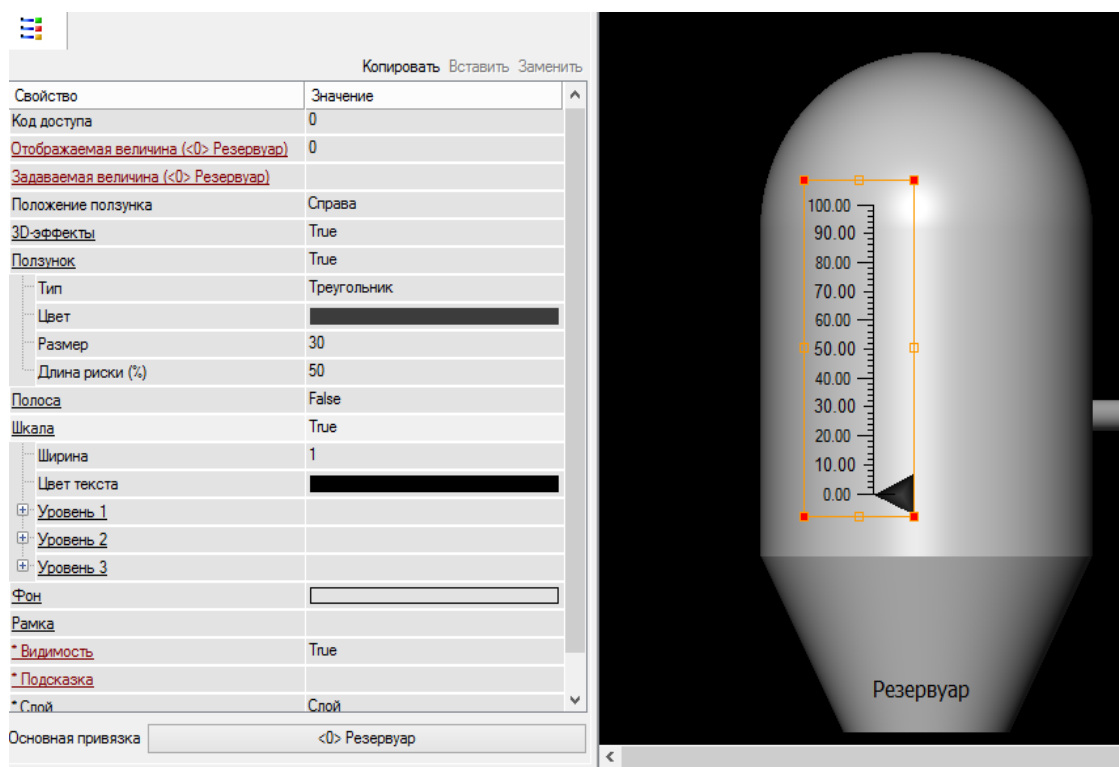




Рисунок 6- Редактирование свойств графического элемента "Ползунок"

Размещение на экране анимационных изображений

Для размещения анимационных изображений (видеоклипов) необходимо предварительно выполнить следующие действия:

- в навигаторе проекта выделить слой "Ресурсы";
- с помощью контекстного меню создать в слое "Ресурсы" группу "Анимация";
- с помощью контекстного меню создать в группе "Анимация" компонент "Библиотека видеоклипов";
- после двойного щелчка ЛК по созданному компоненту справа откроется пустое окно библиотеки;
- в контекстном меню библиотеки выбираем опцию "Импортировать";
- в открывшемся диалоге выбрать каталог Lib/Animation/, и оттуда загрузить файлы fluid_black.mng (жидкость) и flame_small.mng (пламя) (файлы импортируются по отдельности). Если потребуются другие анимационные изображения, то нужно использовать другие директории по указанию преподавателя.

Разместим выбранные видеоклипы на экране "Станция" с помощью кнопки "Ви-

деоклип" , которую можно выбрать, щёлкнув по значку "Ресурсы"  на панели "Графические элементы". Откроется созданная нами библиотека компонентов, содержащая выбранные видеоклипы, показанные на рис. 7.

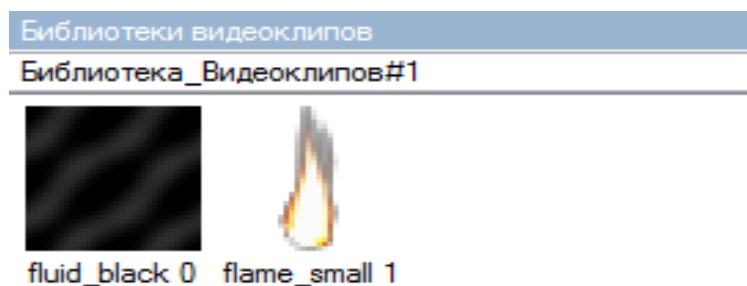


Рисунок 7- Содержание библиотеки анимационных компонентов (видеоклипов)

Над второй ёмкостью, в которой предполагается первичная обработка нефти, разместим видеоклип пламени или иной видеоклип, отображающий факт процесса обработки, например, перекачку потребителю.

Поверх труб разместим видеоклипы, имитирующие потоки нефти справа и слева от клапана.

Необходимые размеры элементам можно придать, выделив и растянув их после размещения. В результате получим мнемосхему, один из вариантов которой показан на рисунке 8.

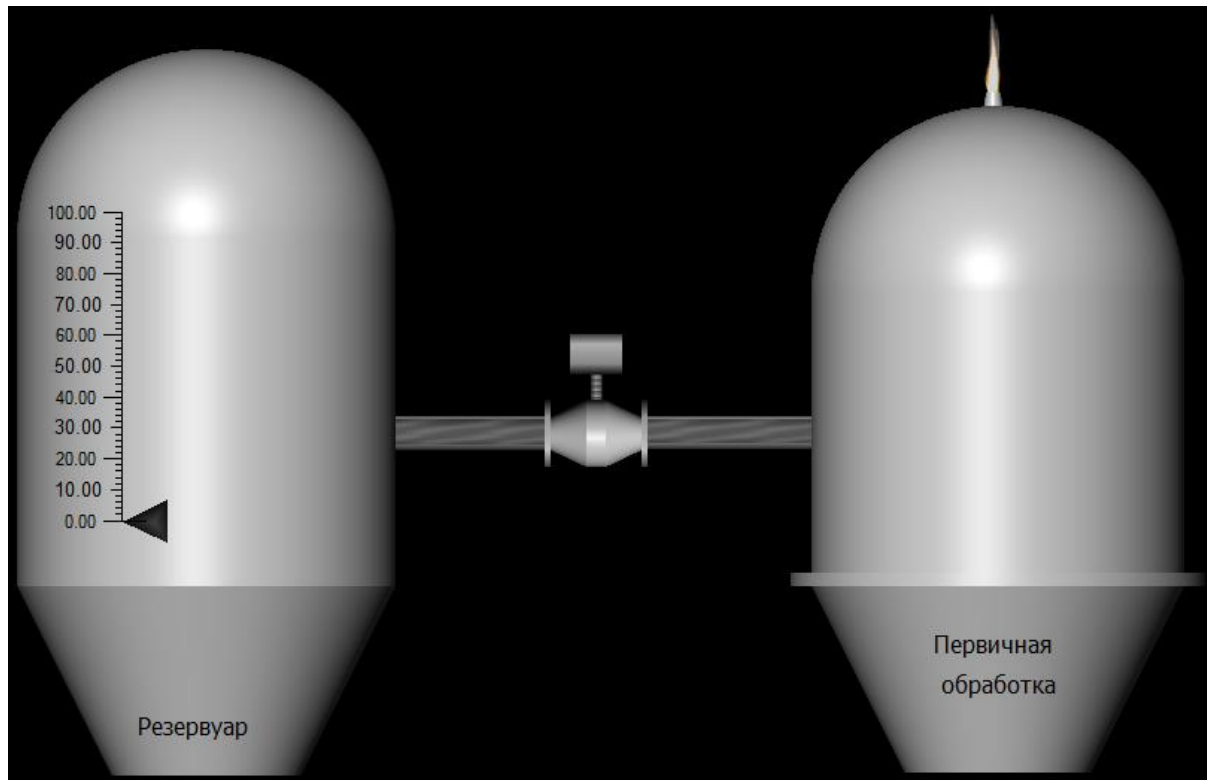



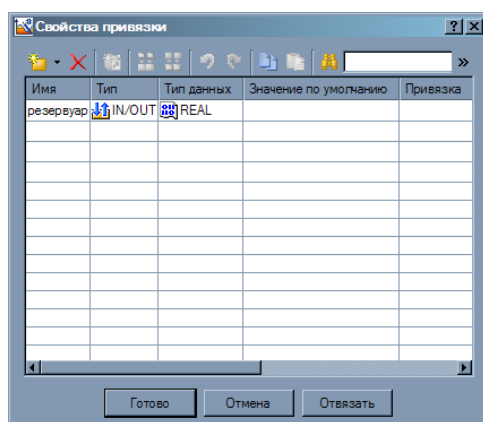
Рисунок 8- Вариант мнемосхемы разрабатываемого технологического процесса

При разработке дизайна экранных форм студенты могут использовать любые доступные видеоклипы, соответствующие мнемосхеме конкретного технологического процесса.

Задание динамических свойств элементов и создание базы каналов

Графические элементы в Trace Mode 6 имеют динамические свойства. Настроим динамические свойства для элементов "Ползунок", "Клапан", а также анимационных изображений жидкости и пламени.

Выделим на экране элемент "Ползунок". Двойным щелчком ЛК откроем его свойства. Щелчком ЛК на экранной кнопке "Основная привязка" откроем окно табличного редактора аргументов шаблона экрана. Кнопкой "Создать аргумент (Ins)"  создадим в таблице новую запись – аргумент шаблона экрана. Имя – "Резервуар", тип "In/Out", тип данных "Real":



Нажав на кнопку "Готово", вернёмся к свойствам элемента. Двойным щелчком по строке "Задаваемая величина" получим доступ к свойству "Привязка" и выберем в свойствах привязки уже созданный аргумент "Резервуар", подтвердив выбор нажатием кнопки "Готово".

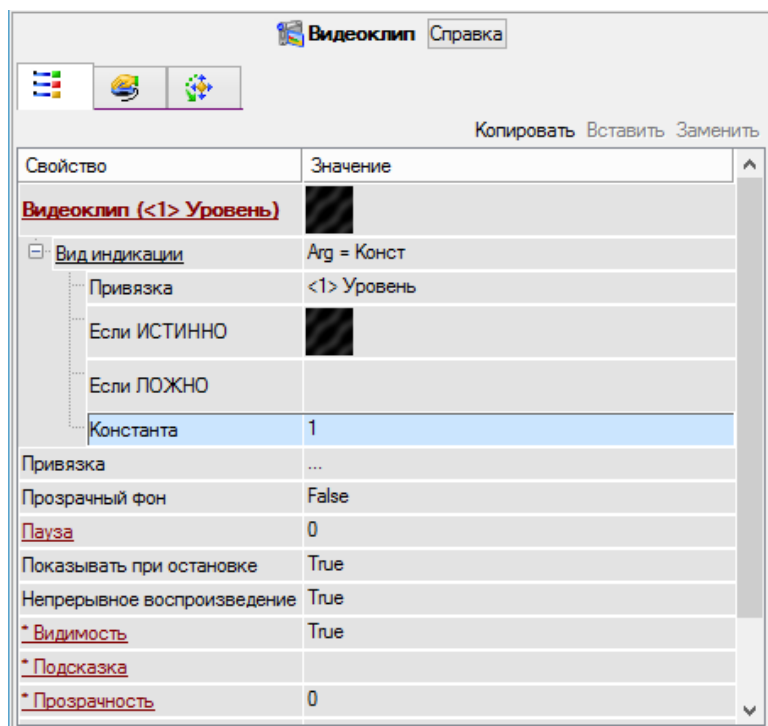
Настройка видеоклипов потока жидкости в трубе

Выберем первый видеоклип потока жидкости (слева от клапана). Зададим для него булевый атрибут видимости, зависящий от уровня жидкости в резервуаре. Для этого в свойствах элемента, как показано на рисунке ниже, выберем "Видеоклип". Далее выбираем "Вид индикации" - "Arg=Конст".

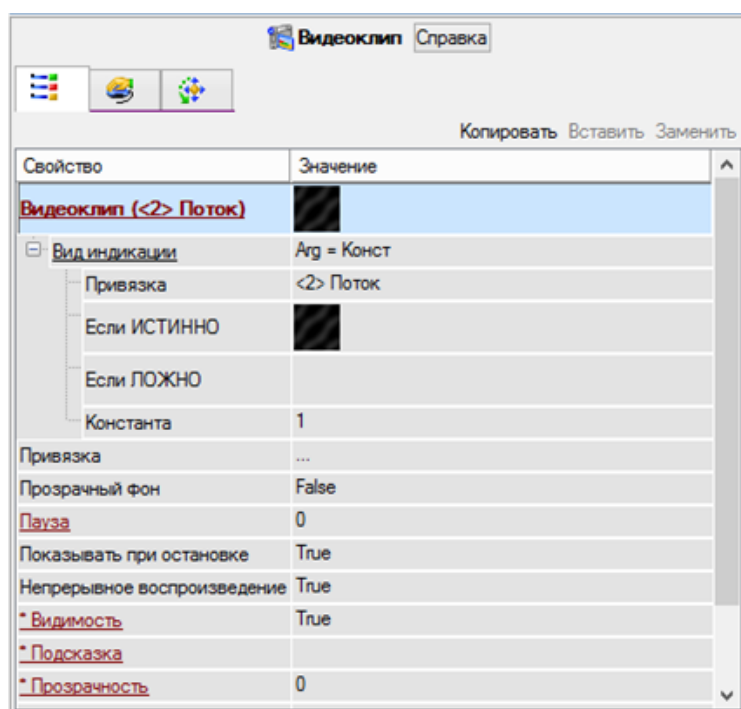
В предложенном списке выбираем значение константы, равное единице (1).

В строке «Если ИСТИННО» выбираем из библиотеки видеоклип жидкости.

Щелчком ЛК по свойству "Привязка" открываем уже знакомое окно табличного редактора аргументов шаблона экрана и создаём новый атрибут экрана "Уровень" с типом "In/Out" и типом данных "Bool". Подтвердим создание атрибута нажатием кнопки "Готово".



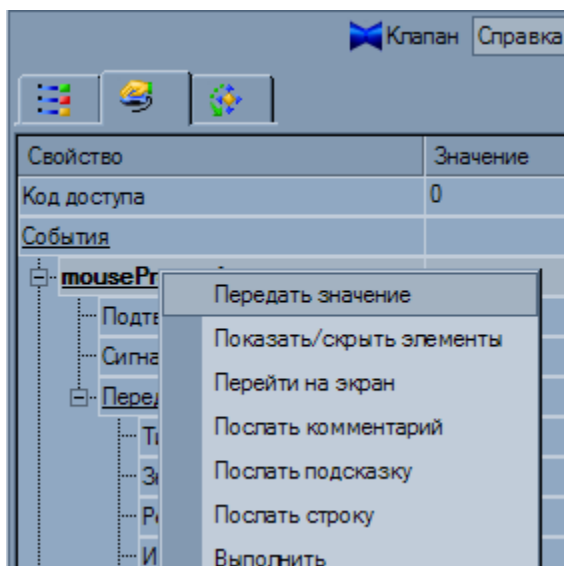
Аналогично настроим второй видеоклип жидкости, размещённый справа от клапана, создав для него аргумент экрана "Поток" с типом "Out" и типом данных "Bool":



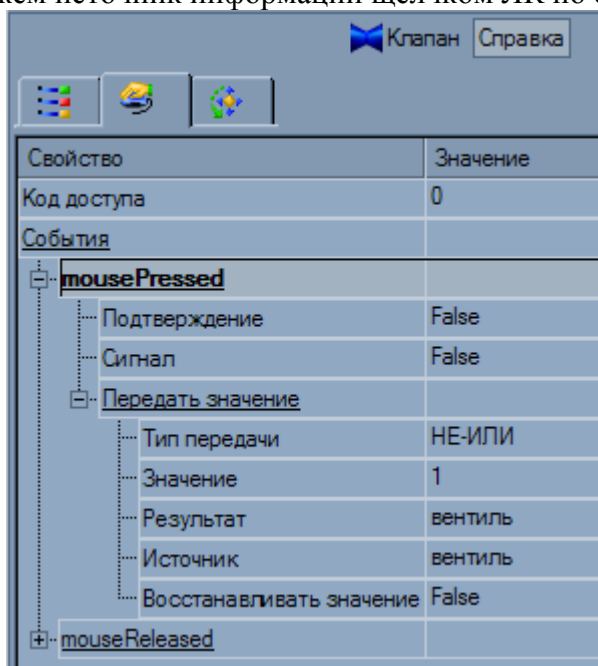
Настройка графического элемента «Клапан»

В свойствах ГЭ «Клапан» перейдем на вкладку "Действия" и зададим свойства события "mousePressed" - нажатие ЛК мыши.

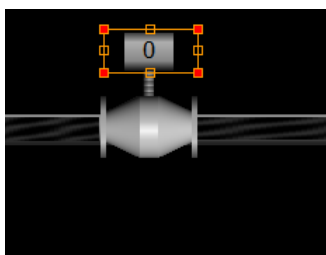
Щёлкнув ПК по свойству "mousePressed", выберем из открывшегося списка вариант "Передать значение":

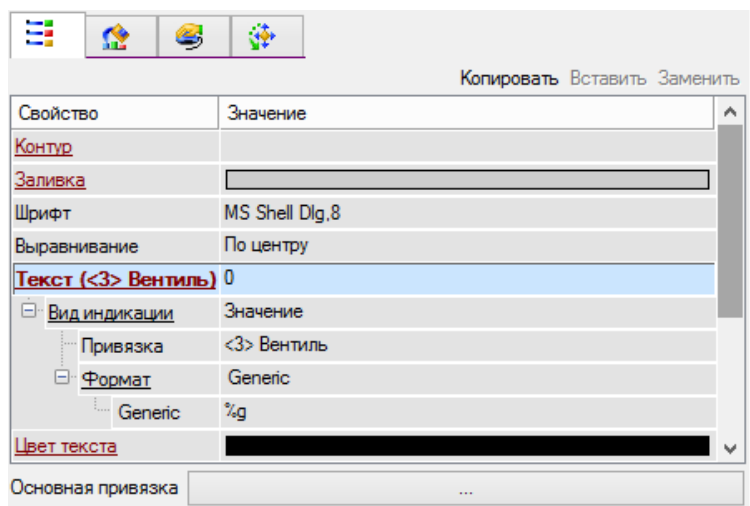


Зададим тип передачи "НЕ-ИЛИ" (XOR). Щёлкнув ЛК по свойству "Результат", переходим к редактированию аргументов шаблона экрана. Создадим новый аргумент экрана. Присвоим ему название "Вентиль", направление передачи "In", типом данных "Bool". Подтвердим создание нового аргумента нажатием кнопки "Готово". К этому же аргументу привяжем источник информации щелчком ЛК по строке свойств "Источник":



Для отображения состояния клапана (открыт или закрыт) на исполнительном устройстве клапана разместим ГЭ «Текст». Зададим динамизацию текста в зависимости от значения булевого атрибута аргумента "Вентиль". Когда вентиль открыт в элементе "Текст" будет отображаться 1, когда закрыт отображается 0. Для этого настроим свойства элемента "Текст" как показано на следующих рисунках:






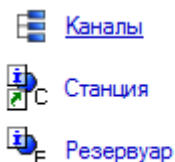
В результате выполненных действий созданы необходимые аргументы экрана. Через эти аргументы экран может получать информацию от других компонентов проекта, например, от программ, и передавать информацию в другие компоненты. Для получения информации и для передачи информации в другие компоненты проекта требуются каналы.

Примечание: канал *Trace Mode* это информационная структура, состоящая из набора переменных и процедур для обработки информации. Через каналы информация поступает в SCADA-систему от внешних устройств, например, от датчиков технологической информации, передаётся на исполнительные устройства АСУТП. Каналы определяют алгоритм работы проекта. В *Trace Mode* 6 определены каналы нескольких классов.

На основании созданных аргументов экрана создадим 4 одноимённых канала. Для этого выберем в навигаторе проекта экран "Станция" и откроем его свойства. Перейдём на вкладку "Аргументы".

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги	Группа	Единицы измерения	Комментарий	Кодировка
резервуар	IN/OUT	REAL							
уровень	IN/OUT	BOOL							
поток	IN/OUT	BOOL							
вентиль	IN	BOOL							

Щелчком ЛК выделим аргумент "Резервуар" и с помощью иконки  выполним автопостроение канала; в результате, в узле "RTM_1", будет создан канал с именем "Резервуар":



Аналогично методом автопостроения создадим каналы "Уровень", "Вентиль", "Поток" по соответствующим аргументам экрана.

Информация Флаги Аргументы Атрибуты				
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
резервуар	IN/OUT	REAL		резервуар:Входное значение (Система.RTM_1)
уровень	IN/OUT	BOOL		уровень:Входное значение (Система.RTM_1)
поток	IN/OUT	BOOL		поток:Входное значение (Система.RTM_1)
вентиль	IN	BOOL		вентиль:Реальное значение (Система.RTM_1)

Отредактировать привязку можно двойным щелчком ЛК по соответствующему полю аргумента.

Примечания:

- Не рекомендуется использование нескольких каналов с одним именем. Система в этом случае не диагностирует ошибку и будет работать корректно, различая каналы по их внутреннему ID, а вас это запутает.
- Цифры, которые появляются в именах элементов в навигаторе проекта после значка #, присваиваются системой для удобства различения их человеком. Можно удалять их через переименование или менять, ID элемента это не изменит.

Для контроля, зададим границы допустимых значений уровня в канале "Резервуар". Выберем в навигаторе проекта узел "RTM_1", затем канал "Резервуар". Выберем из контекстного меню пункт "Редактировать". В открывшемся окне зададим следующие границы:

Границы

☒ Использовать

ВП

ВА

ВГ

НГ

НА

НП

Гистерезис

☐ Контроль границ

Реализация алгоритмов управления

Для программирования алгоритмов функционирования разрабатываемого проекта АСУ в TRACE MODE 6 включены языки **Техно ST**, **Техно SFC**, **Техно FBD**, **Техно LD** и **Техно IL** [1]. Данные языки являются модификациями языков ST (Structured Text), SFC (Sequential Function Chart), FBD (Function Block Diagram), LD (Ladder Diagram) и IL (Instruction List) стандарта IEC61131-3.

Программы и некоторые их компоненты (функции, шаги и переходы SFC и т.п.) могут быть разработаны на любом из встроенных языков в соответствующем редакторе, при этом языки для программы и ее компонентов выбираются независимо.

Для создания и редактирования свойств аргументов, переменных, функций и структурных типов программы, а также для использования в программе функций из внешних библиотек в интегрированную среду разработки проекта встроены специальные табличные редакторы.

TRACE MODE 6 имеет также средства для отладки программ.

Основным языком программирования TRACE MODE 6 является **Техно ST**. Программы, разработанные на языках **Техно LD**, **Техно SFC** и **Техно FBD**, перед компиляцией транслируются в **Техно ST**. **ПЛ**-программы перед компиляцией частично транслируются в **ST**, частично – в ассемблер. Отсюда следует, например, что ключевые слова **Техно ST** являются таковыми и для всех других языков.

Более подробно принципы работы с языками программирования Trace Mode 6 можно изучить в справочной системе [1].

В лабораторной работе будем использовать два из них: **Техно FBD** и **Техно ST**.

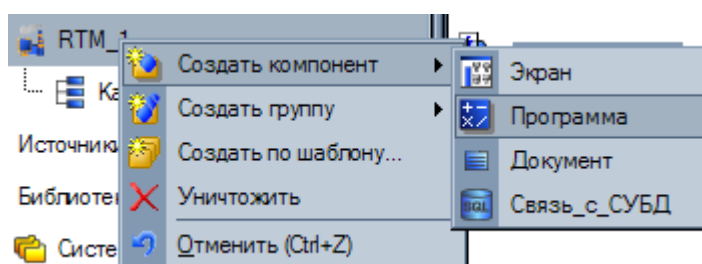
Создание FBD-программы

Создаваемая программа реализует следующий алгоритм. Канал **"уровень"** принимает два значения 0 или 1 в зависимости от того, есть в резервуаре нефть или нет, то есть значение в канале **"уровень"** зависит от значения в канале **"резервуар"**. Пусть для определённости сливное отверстие резервуара располагается на уровне 20 единиц. Если уровень нефти в резервуаре больше 20 единиц, то можно осуществлять откачку нефти. Значение канала **"уровень"** при этом становится равным единице. Если уровень нефти в резервуаре меньше 20, то откачка нефти прекращается, а значение канала **"уровень"** обнуляется. Для реализации этого алгоритма создадим FBD-программу. FBD-программы представляют собой мощный инструмент для решения различных логических задач АСУТП.

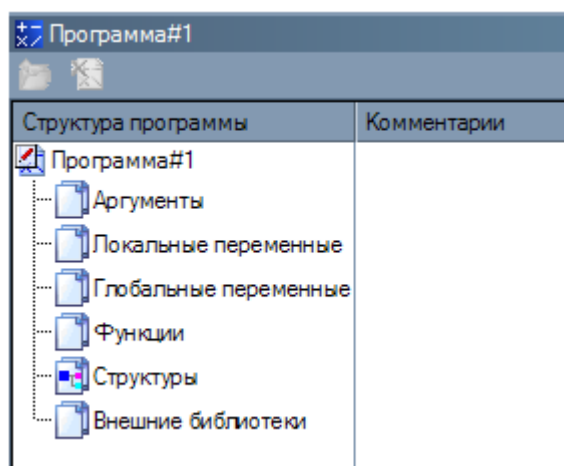
Кроме языка FBD в Trace Mode 6 реализованы ещё 4 языка программирования.

Вызвав контекстное меню в узле RTM_1 навигатора проекта, создадим компонент "программа" с именем "п_уровень".

***Примечание:** аналогично процессу создания экрана созданная программа будет находиться не в узле, а в слое «Шаблоны программ». В слое «Система» в узле RTM_1 будет находиться только канал типа CALL, вызывающий шаблон программы. При разработке крупных проектов программы создают в слое «Шаблоны программ», а не в узле. После разработки всех программ, предусмотренных проектом и размещённых в слое «Шаблоны программ», их перетаскивают ЛК методом **«drag and drop»** в соответствующий узел проекта. Это связано с тем, что программа может использоваться в нескольких узлах. Поэтому не имеет смысла одну и ту же программу создавать в каждом узле заново. Такая технология называется разработкой от шаблонов.*







Двойным щелчком ЛК по компоненту "п_уровень" перейдем в режим редактирования программы:

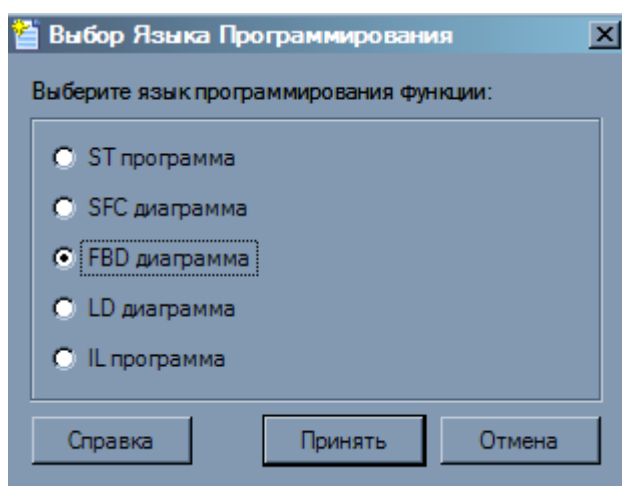


Выделением ЛК в дереве шаблона "п_уровень" строки "Аргументы" вызовем табличный редактор аргументов.


С помощью иконки  создадим в редакторе аргументов два аргумента с именами "пр_резервуар" и "пр_уровень" с типами IN, REAL и OUT, BOOL соответственно.

Имя	Тип	Тип данных
пр_резервуар	 IN	 REAL
пр_уровень	 OUT	 BOOL

Выделим ЛК в дереве шаблона строку "п_уровень" и в открывшемся диалоге "Выбор языка" выберем язык FBD.





По нажатию клавиши "Принять" откроется поле редактирования диаграммы.

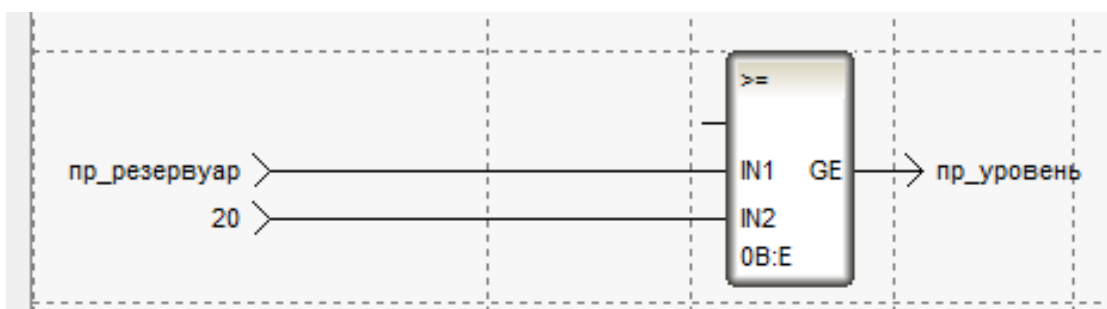
Если палитра FBD-блоков не открылась, её можно вызвать с помощью кнопки . В палитре перейдем на вкладку "Сравнение", выберем блок "больше либо равно" (\geq) и перенесём его в окно редактирования программы (FBD-диаграммы).

Входы и выходы блоков FBD-диаграммы нужно привязать соответственно к источникам и приёмникам информации.

Двойным щелчком ЛК (или щелчок ПК - Привязать) по выводам выбранного блока зададим привязки: вход In1 привяжем к аргументу "пр_резервуар", на вход In2 подадим константу (значение) "20", а выход соединим с аргументом "пр_уровень".

Откомпилировать и отладить программу можно с помощью кнопки  на панели инструментов. Для отображения сообщений, возникающих при компиляции нужно активи-

ровать пункт меню Вид - Сообщения . Полученная FBD-диаграмма выглядит следующим образом:



Выполним привязку аргументов созданной программы к атрибутам каналов. Для этого:

- вызовем свойства программы (компонент "п_уровень") через контекстное меню;
- выберем вкладку "Аргументы";
- двойным нажатием в поле "Привязка" свяжем аргументы программы с атрибутами каналов:
 - аргумент "пр_резервуар" к реальному значению канала "резервуар",
 - аргумент "пр_уровень" к входному значению канала "уровень".





Примечание:

Информация в канал от любого источника всегда подаётся на входное значение канала, а передаётся в любой приёмник информации с реального значения канала.

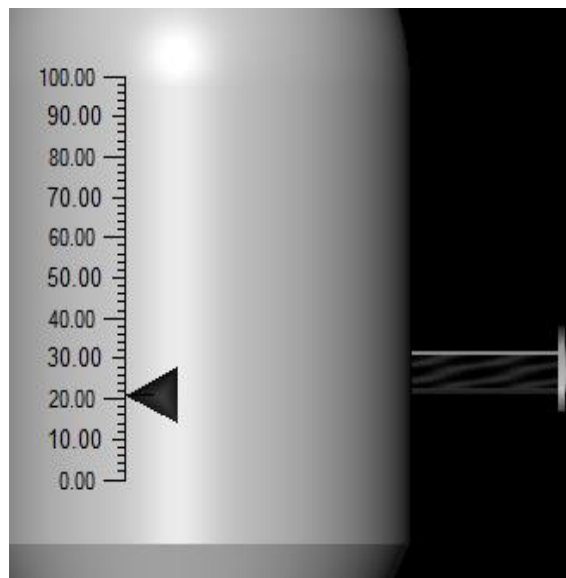
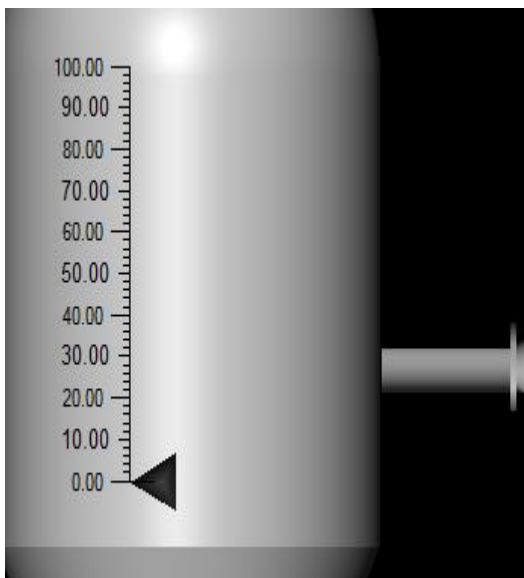
В результате привязки аргументов программы к каналам имеют следующий вид:

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
пр_резервуар	IN	REAL		Резервуар:Реальное значение (Система.RTM_1)
пр_уровень	OUT	BOOL		Уровень:Входное значение (Система.RTM_1)

Проверим и отладим работу программы. Для этого выполним следующие процедуры:

- сохраним проект с помощью иконки ;
- на инструментальной панели выберем ЛК иконку «сохранить для МРВ»  и подготовим тем самым проект для запуска в реальном времени;
- с помощью иконки  на инструментальной панели запустим профайлер или монитор реального времени (МРВ), в который автоматически загружается разработанный проект (**при этом в навигаторе проекта обязательно должен быть выделен соответствующий узел!!!**);
- запустим проект на исполнение в окне профайлера с помощью такой же кнопки .

После запуска проекта установим уровень в резервуаре с помощью ползунка выше 20. Если после этого появилась визуализация жидкости в трубе, то есть заработал видеоклип, значит программа и привязка аргументов настроены правильно.



Повторным нажатием на кнопку  остановим исполнение программы и закроем профайлер.

Создание ST-программы

Алгоритм перекачки нефти во второй резервуар для первичной обработки реализуем программой на языке **Техно-ST**. Перекачка может осуществляться, если уровень нефти в первом резервуаре выше сливного (20) и открыт клапан. Значение канала "поток" зависит от значения каналов "уровень" и "вентиль". Значение канала "поток" устанавливается в единицу если открыт вентиль и уровень в резервуаре выше сливного отверстия. В этом случае в обоих названных каналах "уровень" и "вентиль" должно быть значение "Истина" (1). Реализуем данный алгоритм с помощью ST-программы.

Вызвав контекстное меню в узле RTM_1 навигатора проекта, создадим компонент "программа" с именем "п_поток". Аналогично FBD-программе, создадим для программы следующие три аргумента: "поток" (Out, Bool), "вентиль" (In, Bool), "уровень" (In, Bool).


В навигаторе проекта выделим ЛК строку "Программа#2" и в открывшемся диалоге "Выбор языка" выберем язык **ST**.

В открывшемся окне редактирования программы создадим программный код, показанный на рисунке:

```
PROGRAM
  VAR_INPUT уровень : BOOL; END_VAR
  VAR_OUTPUT поток : BOOL; END_VAR
  VAR_INPUT вентиль : BOOL; END_VAR

  поток = уровень and вентиль;

END_PROGRAM
```

Откомпилировать и отладить программу можно с помощью кнопки  на панели инструментов.

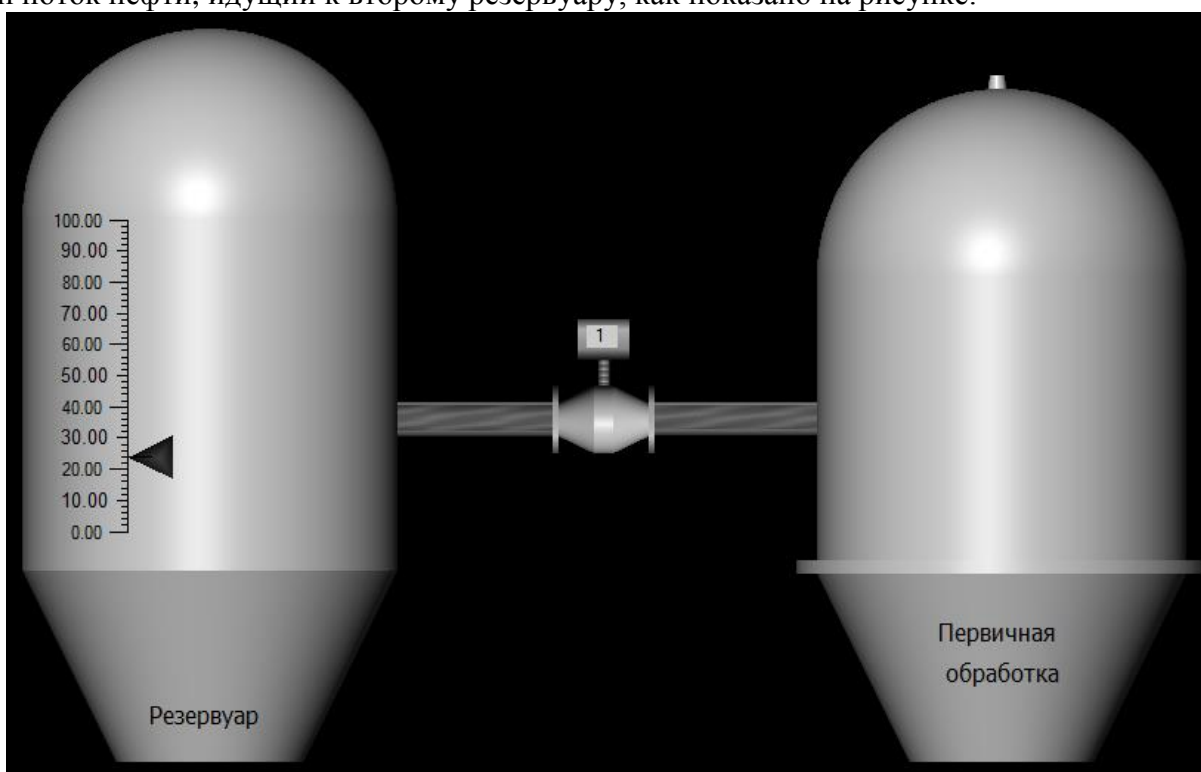
Примечание: программный код создаётся на белом поле между блоком объявления переменных и директивой `END_PROGRAM`. Объявление переменных делается не вручную, а через пункт "Аргументы" графического представления структуры программы.

Выполним привязку аргументов созданной программы к атрибутам каналов аналогично разработке FBD-программы. Требуется привязать аргумент "Уровень" к каналу "Уровень", аргумент "Вентиль" к каналу "Вентиль", аргумент "Поток" к каналу "Поток". Выполняя привязку не забываем о том, что:

информация в канал от источника всегда подаётся на входное значение, а передаётся в приёмник информации с реального значения.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Уровень_R	IN	BOOL		Уровень:Реальное значение (Система.RTM_1)
Поток_In	OUT	BOOL		Поток:Входное значение (Система.RTM_1)
Вентиль_R	IN	BOOL		Вентиль:Реальное значение (Система.RTM_1)

Сохраним проект и запустим на исполнение в профайлере. Если всё сделано корректно, при достаточном уровне нефти в резервуаре (выше 20) и при открытом клапане виден поток нефти, идущий к второму резервуару, как показано на рисунке:



Доработка функционала и внешнего вида проекта

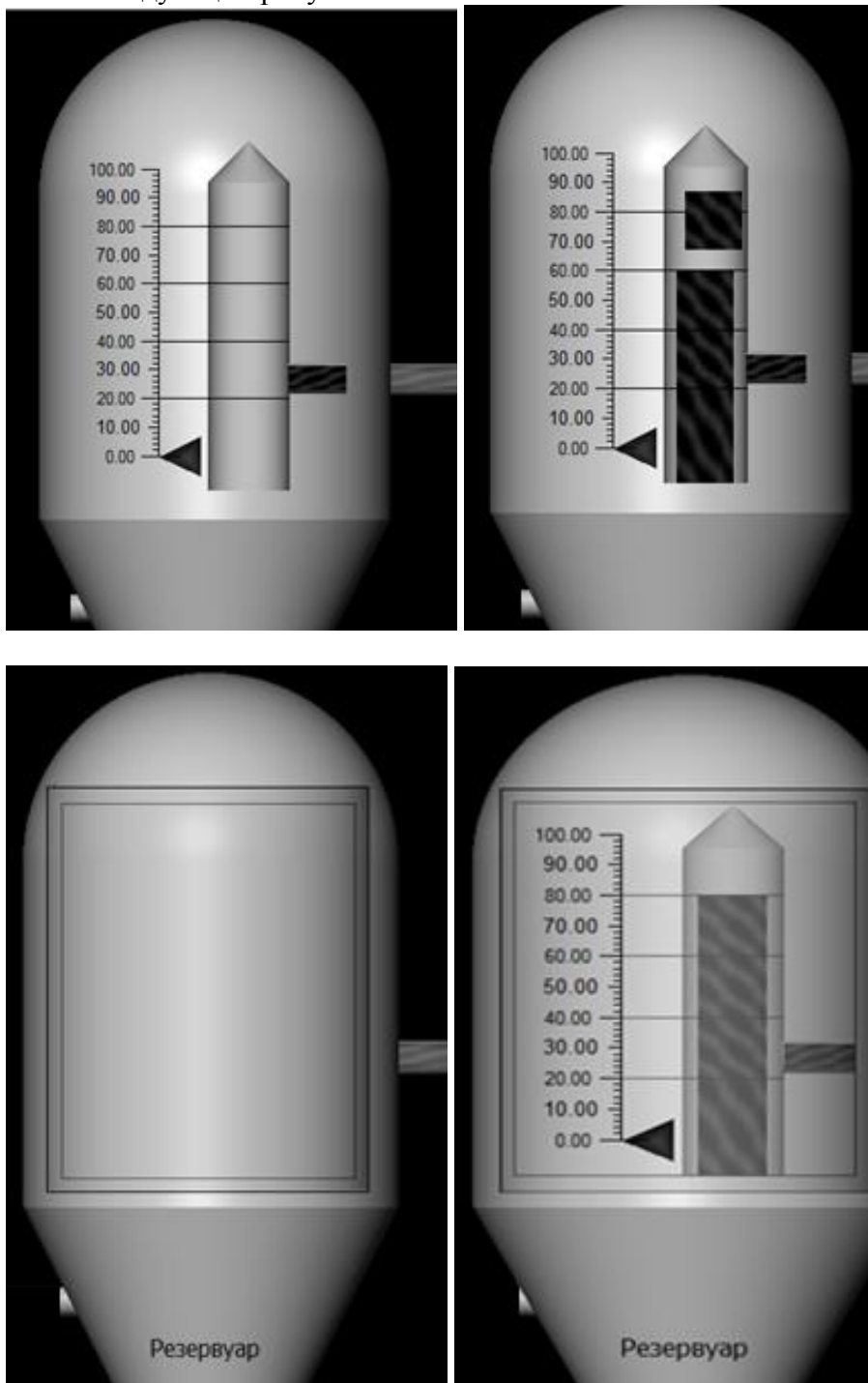
Для большей наглядности мнемосхемы технологического процесса выполним следующие действия. Отобразим на ёмкости "Резервуар" элементы, имитирующие процессы, происходящее внутри. Создадим ещё одну ёмкость, фрагмент трубы и с помощью линий продлим черные отметки ползунка рядом с отметками 20, 40, 60 и 80 единиц до новой ёмкости как показано на рисунке.

Отобразим на ней видеоклипы, соответствующие уровням ползунка (20, 40, 60, 80), то есть нужно добавить 4 видеоклипа. Далее создадим элемент, имитирующий стекло, через которое будем наблюдать происходящие процессы. С помощью линий вычерчиваем два контура, как показано на рисунке, и сверху накладываем цилиндр со следующими

свойствами: цвет серый, прозрачность 50. Этот полупрозрачный цилиндр и является имитатором стекла.

Элементы дизайна (контурные линии, полупрозрачный цилиндр («стекло») и т.д.) студенты могут заменить на другие, подходящие по смыслу для данного технологического процесса. Допускаются изменения и в алгоритме работы проекта, но в этом случае изменятся и разрабатываемые ST и FBD-программы.

Последовательность создания предлагаемого дизайна экранной формы резервуара показана на следующем рисунке.



Для визуализации процесса заполнения резервуара разделим его условно на четыре части, соответствующие значениям 20, 40, 60 и 80. При уровне нефти выше 0 и ниже 20 единиц отображается первый видеоклип. Аналогично отображаются следующие видеоклипы выше первого при соответствующих уровнях. Для этого требуется задать соот-

ветствующие свойства всех видеоклипов. Свойства нижнего видеоклипа показаны на рисунке:

Копировать Вставить Заменить	
Свойство	Значение
Видеоклип (<0> Резервуар)	
Вид индикации	Arg >= Констант
Привязка	<0> Резервуар
Если ИСТИННО	
Если ЛОЖНО	
Константа	0
Привязка	...
Прозрачный фон	False
Пауза	0
Показывать при остановке	True
Непрерывное воспроизведение	True
* Видимость	True
* Подсказка	
* Прозрачность	0
* Слой	Слой
* Выделение в MPB	False
* Геометрия	Скрыть

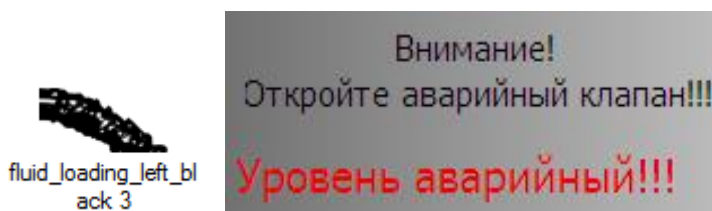
Моделирование аварийных ситуаций

Для перекачки нефти из первого резервуара во второй требуется выполнение двух условий:

- уровень нефти в резервуаре выше заданного значения 20;
- открыт вентиль.

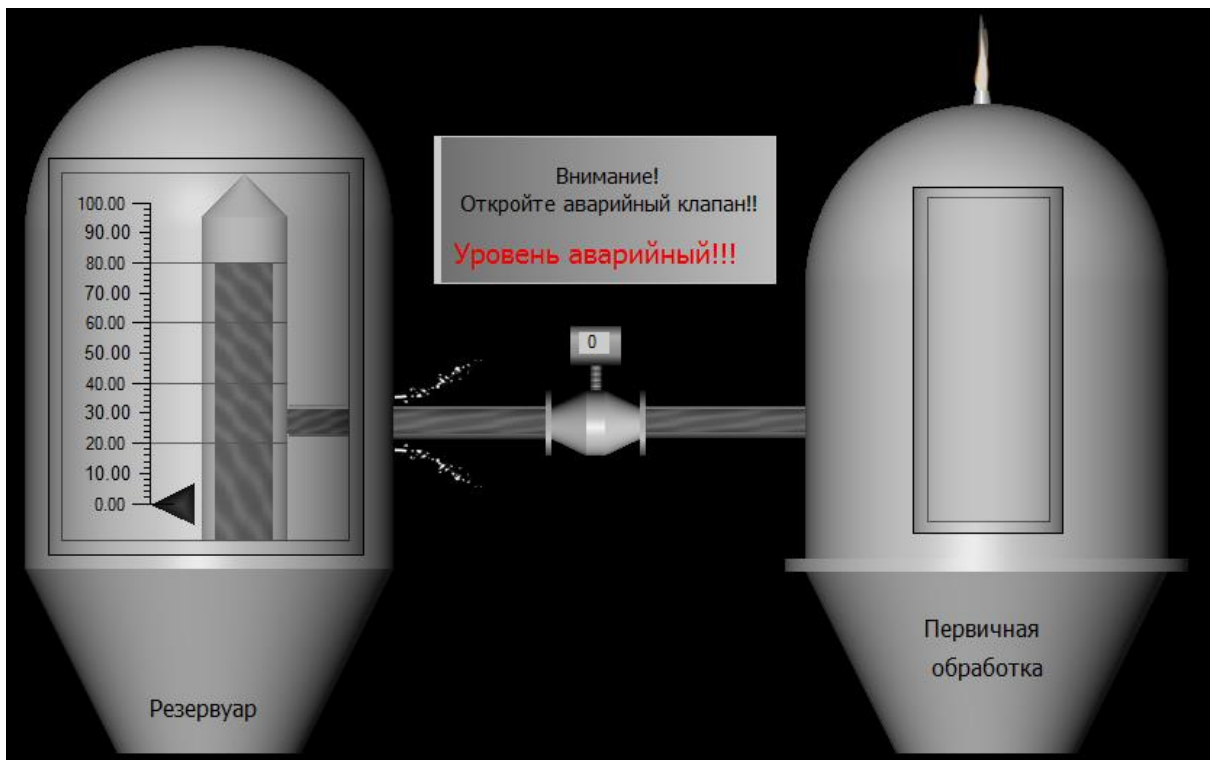
При уровне нефти выше 70 и выше 90 требуется вывести сообщения об утечке и соответствующую анимацию.

Видеоклип утечки и сообщение будут следующими:

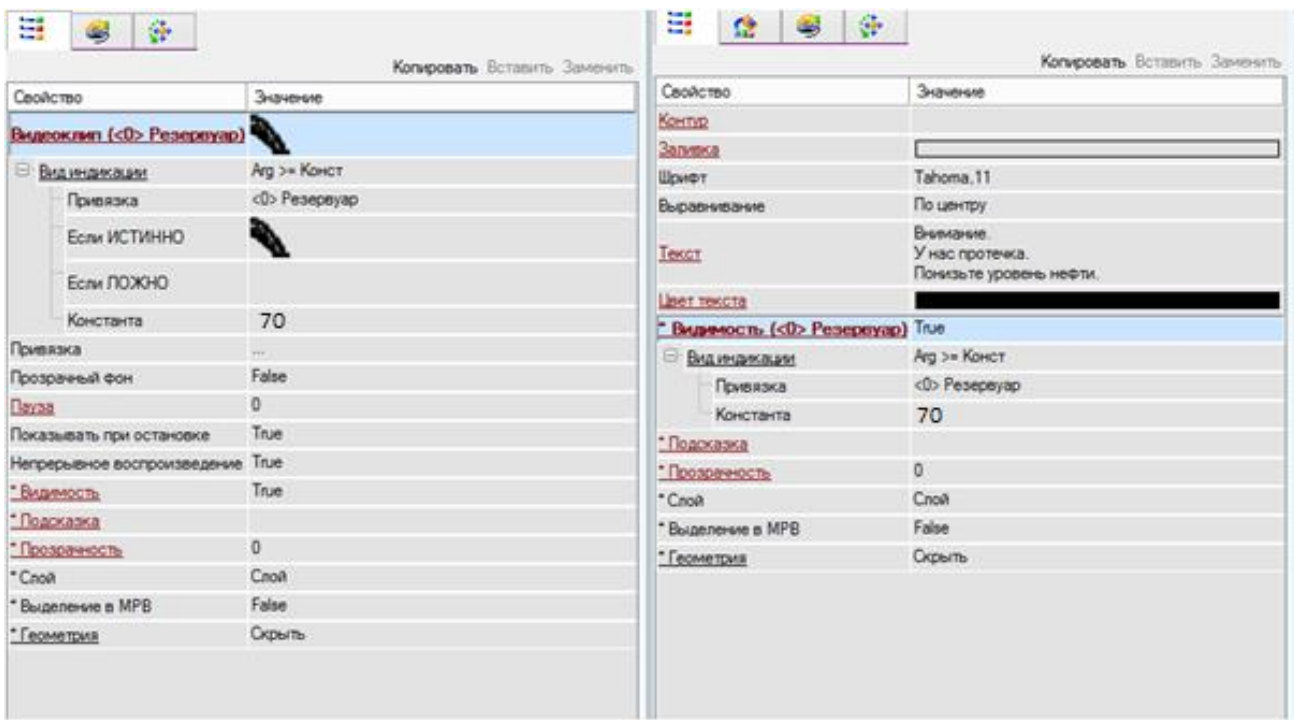


Примечание: элементы визуализации аварийной ситуации (видеоклипы, текстовые сообщения) кроме предлагаемых в методических указаниях студенты могут выбирать самостоятельно из имеющихся библиотек.

В итоге экранная форма принимает следующий вид:



Видеоклипы утечки и сообщение будут выводиться после того, как ползунок перейдет определенный уровень. Например, нижняя струя нефти и табличка с сообщением “Внимание! Откройте аварийный клапан!!!” появятся при значении уровня выше 70. Появление текста и самой таблички задаётся через поле видимость окна свойств соответствующего элемента, как показано на рисунке:



Аналогично настраивается анимация верхней струи нефти и второе сообщение об утечке «Уровень аварийный!!!» при уровне нефти выше 90.

Вторую ёмкость оформим следующим образом. Разместим на ёмкости гистограмму произвольной формы с помощью иконки “Многоугольник”. Поверх этой диаграммы разместим элемент имитирующий стекло, также как делали это ранее.

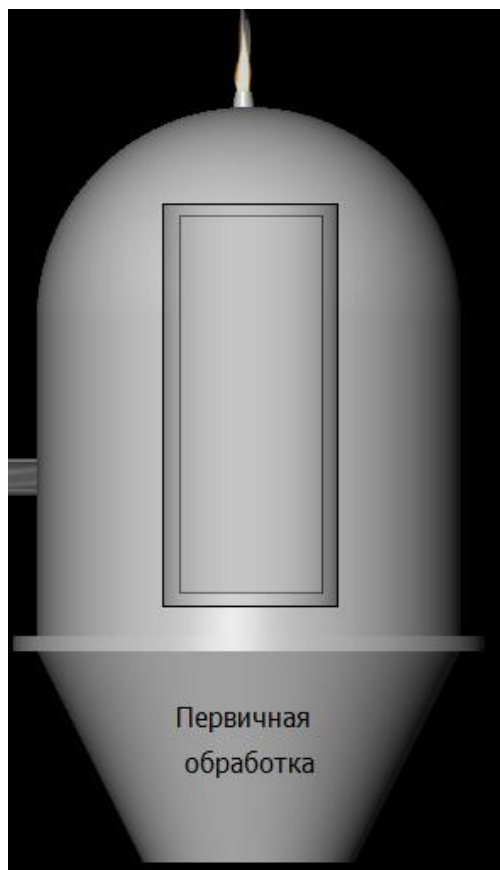
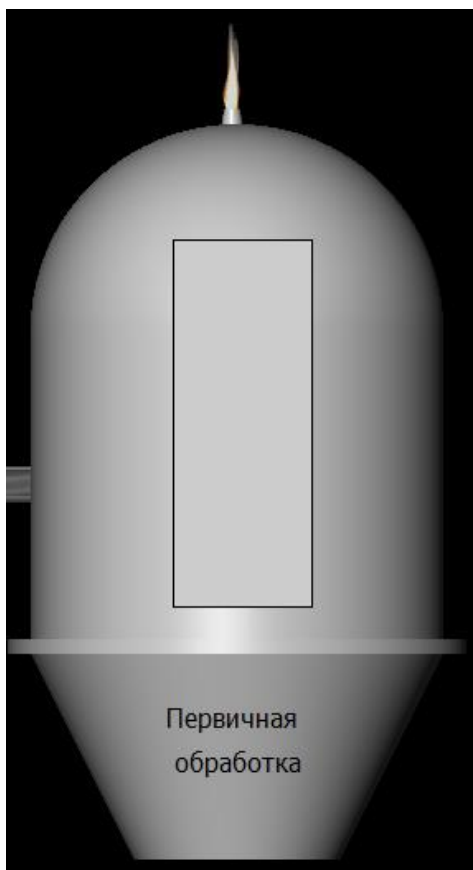

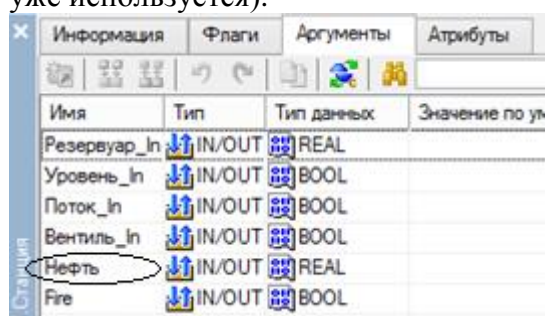


Диаграмма должна начинать заполняться при поступлении нефти во вторую ёмкость. Видеоклип «пламя» (или другой, например, мигающая лампа) появился после того как диаграмма будет полной . После того как над второй ёмкостью появится видеоклип, уровень нефти в ней начнет постепенно уменьшаться, так как нефть будет поступать потребителю. При этом вентиль автоматически будет перекрыт и нефть не будет поступать во вторую ёмкость.

Для реализации этого алгоритма, создадим дополнительные аргументы экрана (Уровень2 и Fire). Выполним следующие действия: **вид->аргументы экрана->** и в появившемся окне нажмем кнопку «создать аргумент» .

Имя	Тип	Тип данных
Резервуар	IN/OUT	REAL
Уровень	IN/OUT	BOOL
Поток	IN/OUT	BOOL
Вентиль	IN/OUT	BOOL
Уровень2	IN/OUT	REAL
Fire	IN/OUT	BOOL

Откроем свойства компонента “Станция” и перейдем на вкладку аргументы. Переименуем аргумент “Уровень2”, присвоив ему новое имя – “Нефть” (имя «Уровень» уже используется).

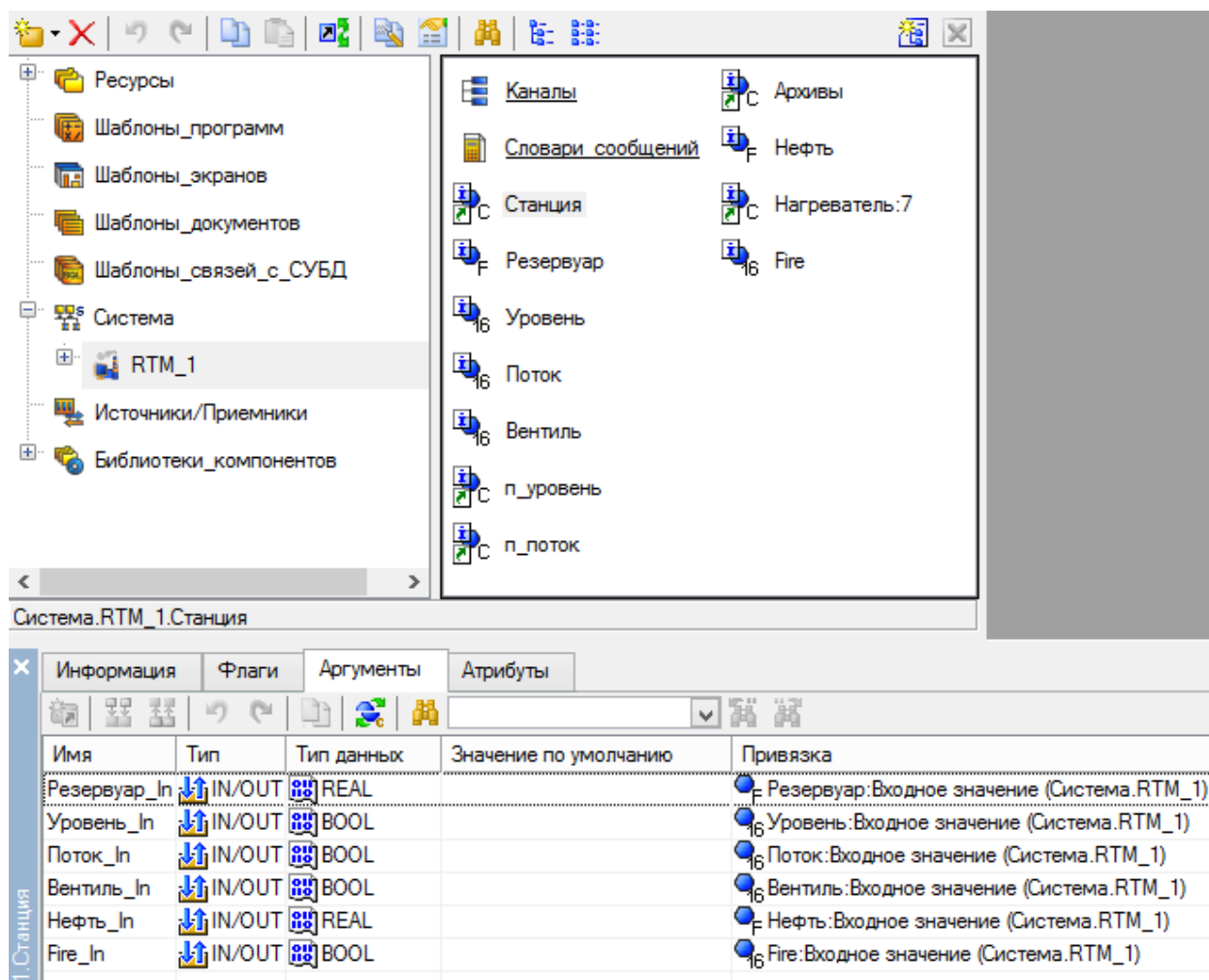


Далее создадим каналы, соответствующие аргументам, с помощью кнопки – «Создать по аргументам каналы с привязкой» так, как это делали ранее.

Настройка свойств видеоклипа и гистограммы показана на рисунках ниже.

Для реализации алгоритма функционирования системы управления первичной обработкой, происходящей во второй ёмкости, создадим FBD-программу, которая условно названа **Нагреватель:7**.



Структура проекта после создания аргументов, FBD-программы и каналов, связанных с аргументами



Привяжем видеоклип пламени к аргументу Fire, а заполнение гистограммы «Многоугольник» к аргументу Уровень2.

Настройка свойств видеоклипа


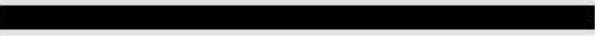
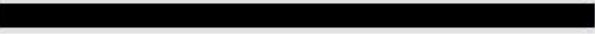
Копировать Вставить Заменить



Свойство	Значение
Видеоклип (<5> Fire)	
<input type="checkbox"/> Вид индикации	Arg = Конст
Привязка	<5> Fire
Если ИСТИННО	
Если ЛОЖНО	
Константа	1
Привязка	...
Прозрачный фон	True

Настройка свойств гистограммы «Многоугольник»

☒ Разрешено

Копировать Вставить Заменить

Свойство	Значение
Направление	Вверх
Слой	
<input type="checkbox"/> Слой (<4> Уровень2)	
Имя	
Привязка	<4> Уровень2
Тип заливки	Цвет
Цвет заливки	
Стиль заливки	
Макс	100
Мин	0
Мин=НП, Макс=ВП	False

Далее в окне редактора FBD-программы создадим локальные переменные, необходимые для функционирования программы. Для этого зайдем в раздел редактора FBD-программы «**Локальные переменные**» и в окне справа нажатием кнопки «Создать»  создадим требуемые переменные. Переменным зададим определенные значения, показанные на рисунке ниже. Затем создадим аргументы программы с помощью такой же кнопки  в правом окне раздела **Аргументы**.

Создание локальных переменных для FBD-программы Нагреватель:7 и задание им начальных значений

Имя	Тип данных	Начальное значение	Комментарий
RateUp	REAL	1	Скорость заполнения
RateDown	REAL	1.5	Скорость сгорания
LevelMax	REAL	100	Максимальное значение
LevelMin	REAL	0	Минимальное значение
LevelEnableFire	REAL	90	
LevelDisableFire	REAL	0	

Имя	Тип	Тип данных
Enable	IN/OUT	BOOL
Level	IN/OUT	REAL
Fire	IN/OUT	BOOL

Далее выполним привязку аргументов созданной программы к соответствующим каналам. Вызовем свойства компонента "Нагреватель:7" через контекстное меню и перейдем на вкладку «Аргументы».

Аргументы FBD-программы Нагреватель:7

Информация				
Флаги				
Аргументы				
Атрибуты				
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Enable	IN/OUT	BOOL		
Level	IN/OUT	REAL		
Fire	IN/OUT	BOOL		

Привяжем созданные аргументы программы через поле «привязка» к аргументам экрана (Enable к Вентиль, Level к Нефть, Fire к Fire). При этом создадутся одноименные каналы и аргументы поменяют “имя” на то, которое соответствует их привязке. Результат показан на рисунке ниже:

Варианты экранных форм для различных ситуаций работы программы показаны на следующих рисунках:

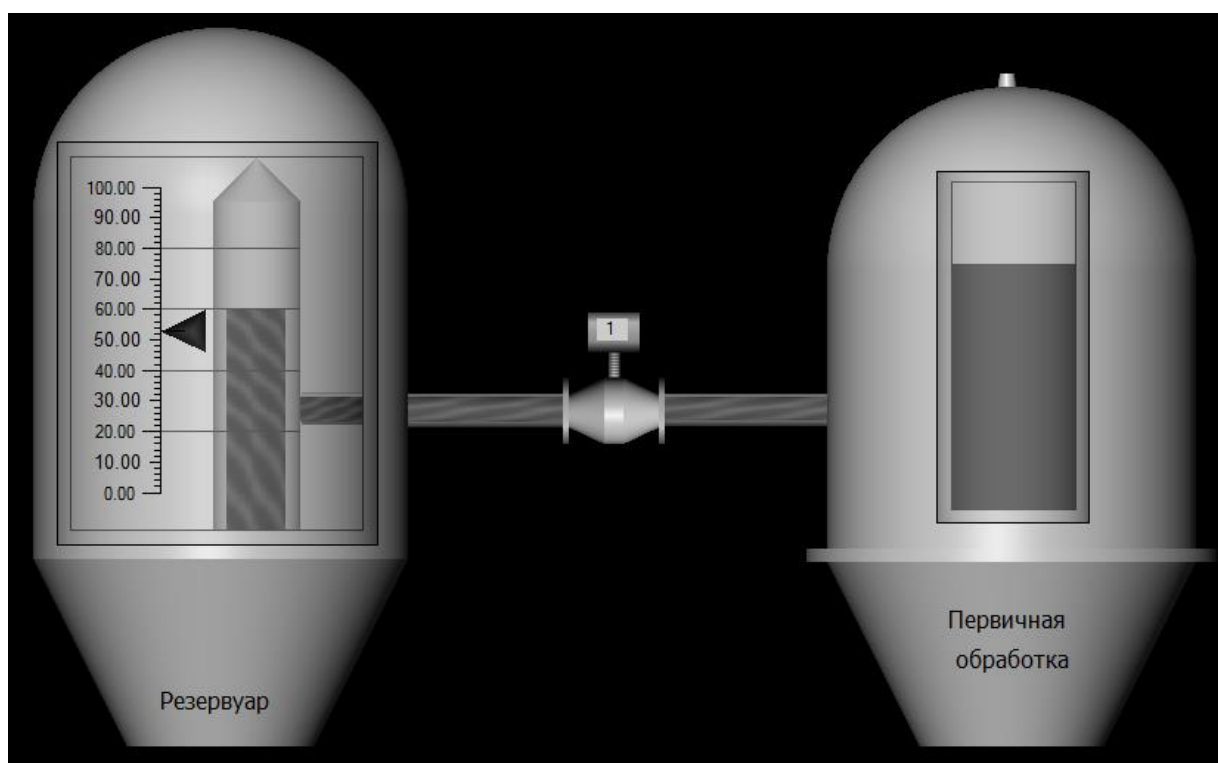


Рисунок 10. Уровень выше сливного и вентиль открыт. Заполняется вторая ёмкость.

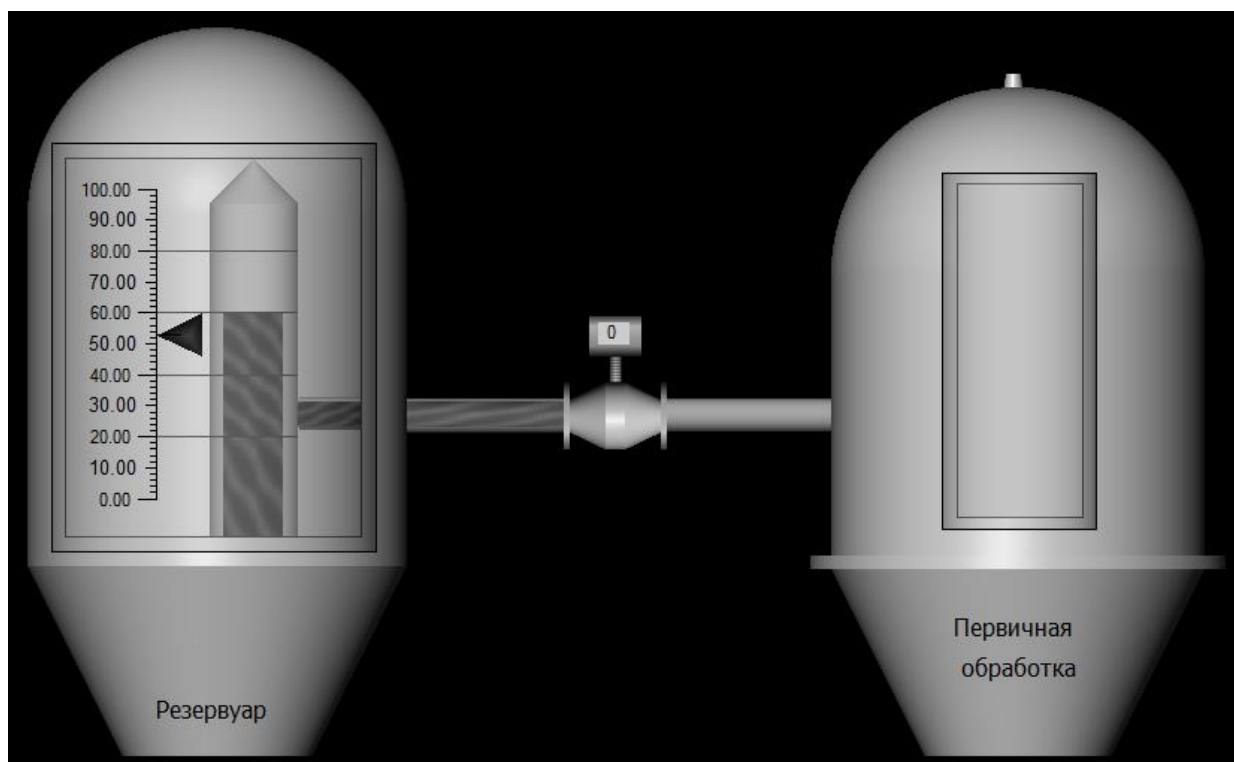


Рисунок 11. Уровень выше сливного, вентиль закрыт. Происходила откачка нефти из второй ёмкости. В данном случае откачка уже закончилась.

АРХИВИРОВАНИЕ ТЕХНОЛОГИЧЕСКОЙ ИНФОРМАЦИИ

Создание экрана архивов

Мониторы TRACE MODE поддерживают функцию записи значений атрибутов каналов в архивы СПАД (структурированный промышленный архив данных, SIAD) и в архив «Отчет тревог» (см. Архивирование каналов узла [1]).

СПАД - архив отображается в виде тренда. Данные можно просматривать за весь период работы АСУТП. Это позволяет, например, проанализировать причины возникновения аварийных и предаварийных ситуаций, перерасхода энергии, полуфабрикатов и т.д.

Архив «Отчет тревог» (ОТ) – это текстовый файл (ASCII), в который заносятся сообщения, генерируемые в различных ситуациях при работе АСУ из словарей сообщений, подготовленных при создании проекта SCADA-системы. Сообщения можно просматривать в реальном масштабе времени в процессе работы АСУТП с помощью специальных экранных форм.

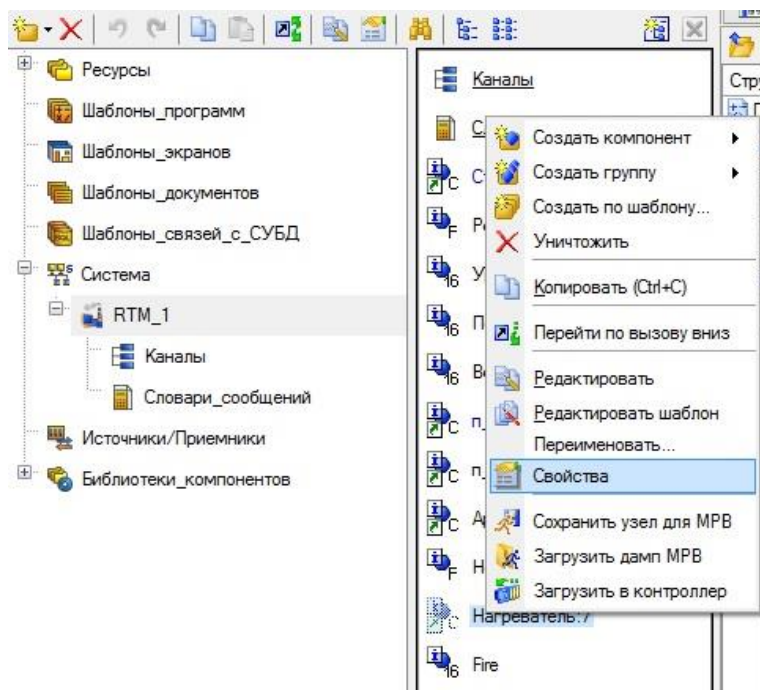
Создание архива СПАД

Для сохранения реальных значений каналов в архив СПАД (SIAD/SQL 6) и ведения архива «Отчет тревог» по таким событиям, как пересечение заданных для канала границ, необходимо:

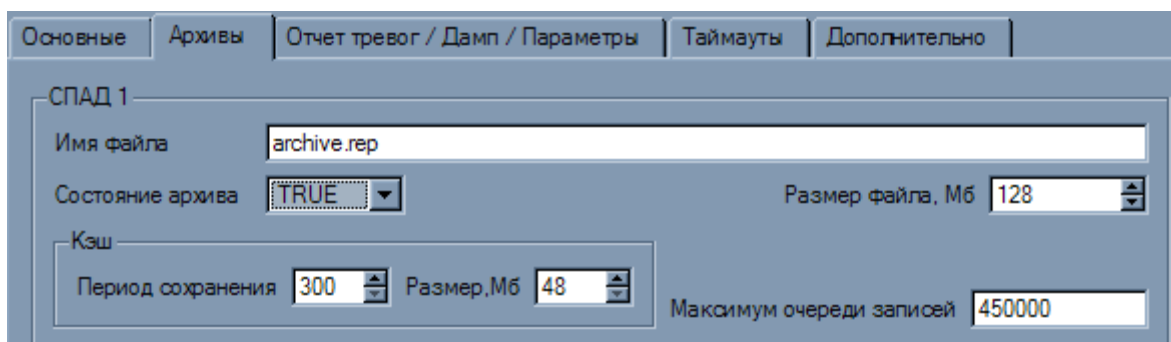
- Настроить узел на архивацию. Для узла проекта нужно предварительно задать ряд общих параметров.
- Настроить каналы на архивацию. Для каналов индивидуально или групповым способом установить соответствующие атрибуты СПАД-архива и/или архива «Отчет тревог».

Для настройки узла "RTM_1" на архивирование требуется выполнить следующие действия:

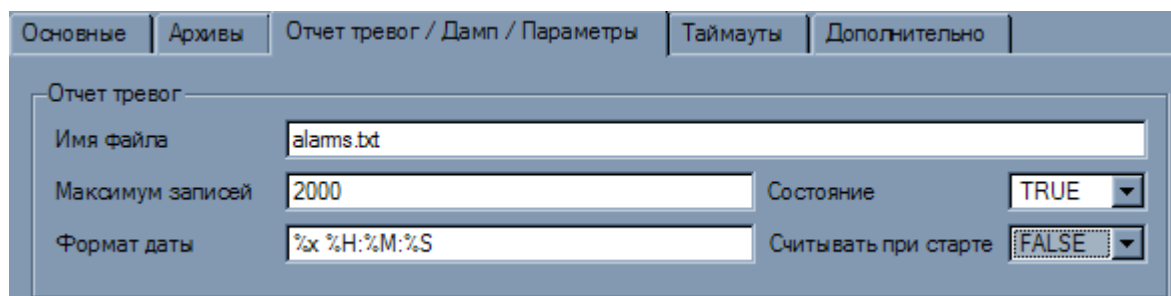
- выделить узел;
- кликнув ПК по выделенному узлу войдём в опцию «Редактировать» контекстного меню узла "RTM_1";
- определим файлы СПАД-архива, архива «Отчет тревог» и зададим настройки для сетевого обмена как показано на следующих рисунках:



Реальные значения каналов, связанных с технологическими параметрами, изменяющимися в реальном масштабе времени, будем записывать (архивировать) в архив СПАД1. Для этого перейдем во вкладку "Архивы" узла и отредактируем бланк СПАД 1 следующим образом:

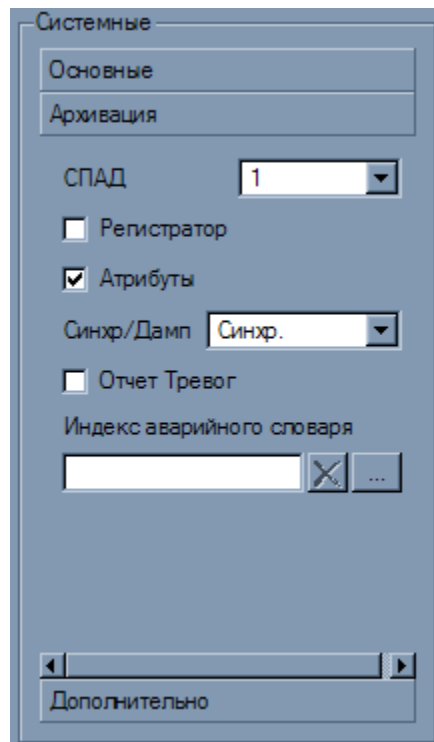


Во вкладке "Отчет тревог/Дамп/Параметры" определим следующие параметры:



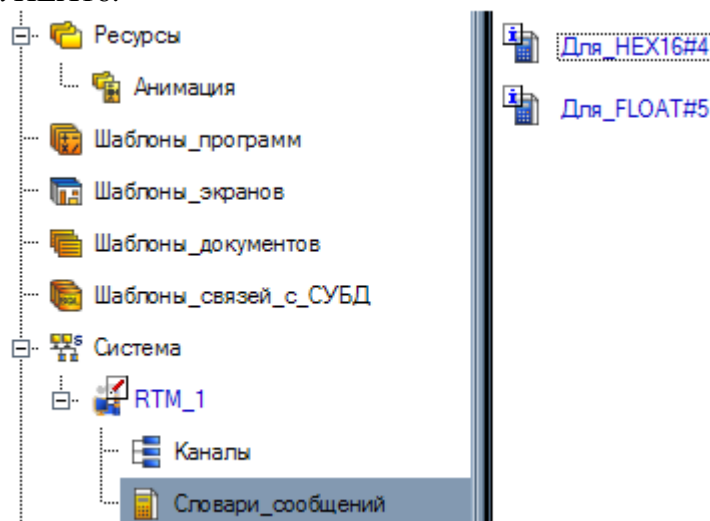
Настройка каналов на архивирование

Выделим канал "Резервуар" и в контекстном меню выберем его редактирование. В разделе "Системные" настроим параметры архивации:

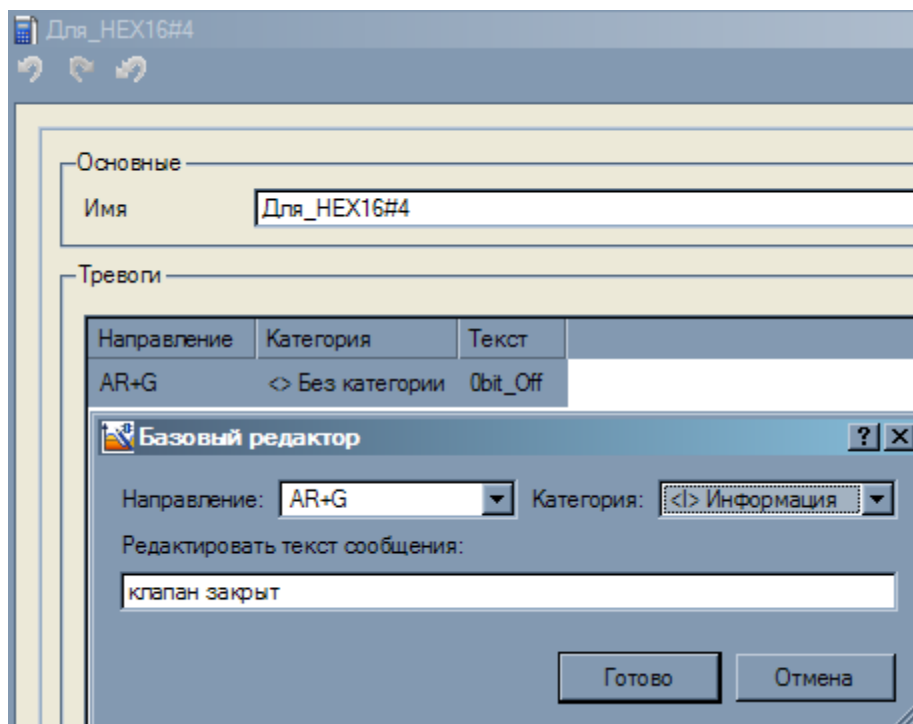


Создание отчёта тревог

Для каналов "Вентиль" и "Резервуар" зададим сообщения в отчет тревог. С этой целью создадим для узла "RTM_1" новую группу "Словари сообщений" (контекстное меню - "Создать группу"). В группе создадим два словаря: один для каналов типа FLOAT и второй для каналов типа HEX16.



Откроем словарь HEX16. Выделим первую запись, соответствующую состоянию "FALSE" (0bit_Off) и зададим текст сообщения «клапан закрыт». Установим этому сообщению категорию I (Информация).



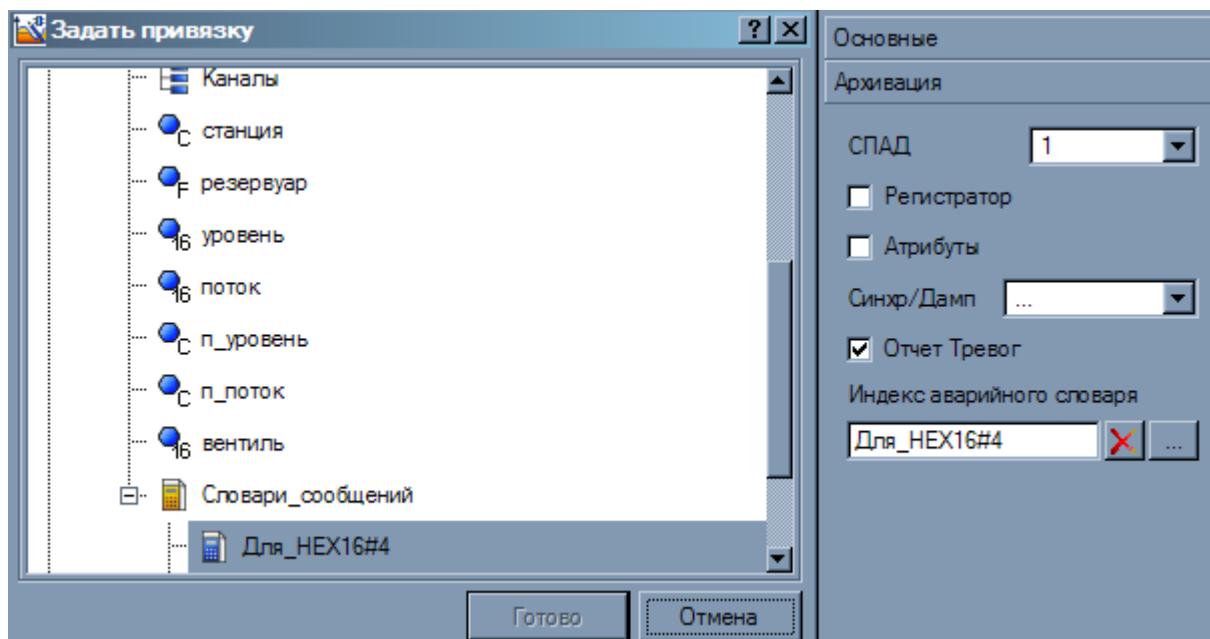
Для 0bit_On сделаем сообщение того же типа I (Информация) с текстом "клапан открыт". Получим:

Тревоги		
Направление	Категория	Текст
AR+G	<I> Информация	клапан закрыт
AR+G	<I> Информация	клапан открыт

Откроем окно редактирования словаря FLOAT (двойной щелчок ЛК или "Редактировать" в контекстном меню) и зададим для него следующий набор сообщений:

Для_FLOAT#5*		
Основные		
Имя: Для_FLOAT#5		
Тревоги		
Направление	Категория	Текст
AR+G	<I> Информация	уровень нормальный
AR+G	<W> Предупре...	уровень превышен
AR+G	<W> Предупре...	уровень недостаточный
AR+G	<A> Тревога	уровень значительно превышен
AR+G	<A> Тревога	резервуар почти пуст

Созданные словари подключим к соответствующим каналам. С этой целью откроем на редактирование канал "Вентиль". В разделе "Архивация" выберем опцию "Индекс аварийного словаря" "Для_HEX16".



Аналогично поступим с каналом "Резервуар", связав его со словарём FLOAT.

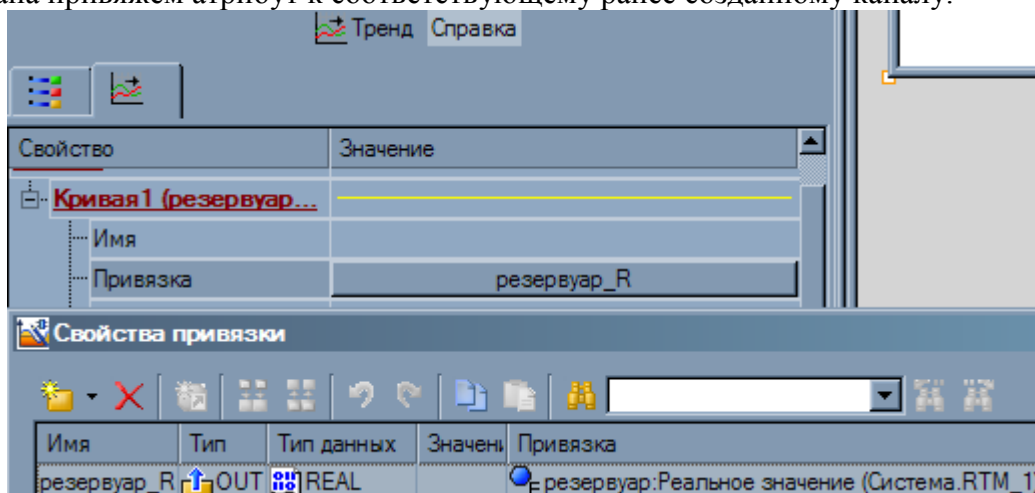
Размещение графических элементов

С помощью контекстного меню узла RTM_1 создадим еще один экран. Созданный "Экран#2" переименуем, присвоив имя "Архивы". Разместим на экране элементы "Тренд" и "Отчёт тревог узла".



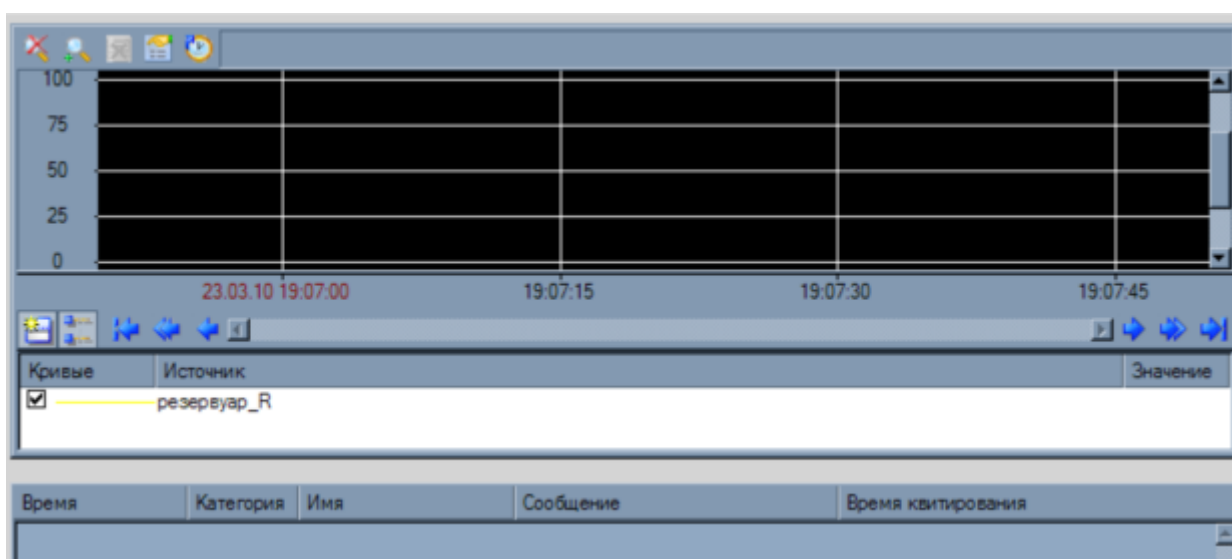
Перейдем на закладку "Кривые" свойств "Тренда" и создадим кривую (щелчок правой кнопкой мыши и выбор "Кривая").

Внешний вид кривой настраивается произвольно в соответствии с дизайном экранной формы и требованиям эргономики. Для созданной кривой создадим привязку к каналу "Резервуар". Для этого создадим атрибут "Резервуар" экрана "Архивы", а затем в свойствах экрана привяжем атрибут к соответствующему ранее созданному каналу.




В режиме работы переход между текущими и архивными данными "Тренда" осуществляется по кнопке

Вид тренда отчета тревог показан на следующем рисунке:

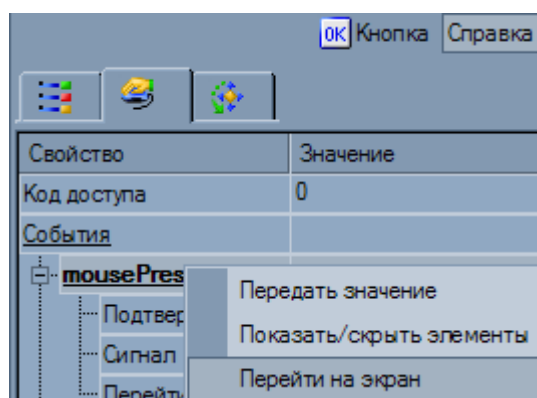


Реализация переходов между экранами

Для выполнения переходов между экранами в реальном масштабе времени, то есть для навигации по проекту, необходимо разместить на каждом из них кнопку  с соответствующими свойствами.

1. В свойствах кнопки в разделе «Общие свойства» разместим на кнопке текст, соответствующий имени экрана, на который осуществляется переход (в строке "Текст"). В данном случае на кнопке размещается текст «Архивы».

2. В свойствах кнопки «События» добавим для кнопки событие "mousePressed", для которого выберем действие "Перейти на экран" и из выпадающего списка выбираем соответствующий экран.







Примечание: переход на требуемый экран будет осуществляться только в том случае, если у этого экрана есть хотя бы один аргумент. Если аргументов нет, то нужно создать. Этот аргумент не будет использоваться, то есть «пустой аргумент», зато функция перехода будет реализована.

Перед запуском следует выполнить два вида сохранения проекта:

- обычное сохранение в соответствующий файл с расширением **.prg**;
- сохранение в формате монитора реального времени (MPB).

Запуск проекта

Для запуска проекта следует последовательно выполнить следующие действия:

- сохранить проект с помощью иконки ;
- на инструментальной панели выбрать ЛК иконку «Сохранить для МРВ» . В результате этого действия проект будет подготовлен для запуска в специальной среде, называемой монитором реального времени (МРВ) или профайлером;
- в окне навигатора проекта выделить соответствующий узел, в данном случае узел RTM_1;
- с помощью иконки  на инструментальной панели запустим МРВ;
- проект на исполнение запустим в окне профайлера (МРВ) с помощью такой же иконки .

Литература

1. Справочная система по SCADA-системе Trace Mode 6// Adastrа Research Group, Ltd – Москва, 2013.