

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования

**«МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)»
ВОЛЖСКИЙ ФИЛИАЛ МАДИ**

УТВЕРЖДАЮ
Зав. кафедрой ИиТТП
Петрова А.В.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ
ПО ДИСЦИПЛИНЕ
«ИНТЕРФЕЙСЫ АВТОМАТИЗИРОВАННЫХ СИСТЕМ
ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ»**

Направление подготовки
09.03.01 «Информатика и вычислительная техника»

Направленность (профиль)
***Автоматизированные системы обработки информации и
управления в отраслях транспортно-дорожного комплекса***

Квалификация
Бакалавр

Кафедра: ИиТТП

Чебоксары

ВВЕДЕНИЕ

Система TRACE MODE 6 предназначена для автоматизации промышленных предприятий, энергетических объектов, интеллектуальных зданий, объектов транспорта, систем энергоучета и т.д. Масштаб систем автоматизации, создаваемых в TRACE MODE, может быть любым – от автономно работающих управляющих контроллеров и рабочих мест операторов (АРМ), до территориально распределенных систем управления, включающих десятки контроллеров и АРМ, обменивающихся данными с использованием различных коммуникаций – локальная сеть, интранет/интернет, последовательные шины на основе RS-232/485, выделенные и коммутируемые телефонные линии, радиоканал и GSM-сети.

Динамические характеристики и надежность создаваемого в TRACE MODE 6 программного обеспечения (ПО) АРМ и контроллеров позволяют применять разработанные системы автоматизации в таких отраслях промышленности как нефтехимия, металлургия, энергетика, машиностроение, коммунальное хозяйство, пищевая промышленность, транспорт, а также при проведении научных исследований.

Основные термины и определения Программные продукты TRACE MODE 6 подразделяются на интегрированную среду разработки (инструментальную систему) и исполнительные модули.

Инструментальная система TRACE MODE устанавливается на рабочем месте инженера-разработчика АСУ и предназначена для создания системы автоматизации и отладки всех ее компонентов.

Исполнительные модули TRACE MODE предназначены для запуска проекта TRACE MODE в реальном времени, т.е. для эксплуатации на действующем объекте автоматизации. Основным исполнительным модулем TRACE MODE для АРМ является МРВ – монитор реального времени, реализующий такие основные функции как непрерывный сбор данных, их математическую обработку и визуализацию. Для запуска проекта в контроллерах используются исполнительные модули МикроМРВ, которые различаются по типу контроллеров (разрядность процессора, операционная система, использование сетевого взаимодействия и др.).

Каждому компьютеру/контроллеру, запускаемому под управлением исполнительного модуля в проекте TRACE MODE сопоставлен отдельный узел. Максимальное количество узлов в проекте – 255. В рамках узла создаются каналы – основные информационные единицы для ввода и первичной обработки данных. Различают два типа каналов – INPUT и OUTPUT.

Источники и приемники данных представляют собой описатели точек ввода-вывода, то есть связей с контроллерами, платами УСО, интеллектуальными датчиками и т.д. Каждая точка ввода-вывода может быть одним аналоговым сигналом или группой (до 16-ти) дискретных сигналов.

Количество проектов, разрабатываемых с помощью одной инструментальной системы TRACE MODE, как и время работы в ней не ограничено.

В состав инструментальной системы TRACE MODE 6 входят:

- интегрированная среда разработки TRACE MODE 6 IDE (файл **tmdevenv.exe**);
- профайлеры – отладочные МРВ (файл **rtc.exe** – с поддержкой графических экранов; файл **rtmg32.exe** – без поддержки графических бесплатный набор драйверов устройств ввода\вывода);
- библиотека компонентов (файл **tmdevenv.tmul** и набор ресурсов – обои, логотипы, анимации в каталоге \Lib папки инструментальной системы);
- электронная документация (встроенная справочная система);
- демонстрационные проекты TRACE MODE (примеры систем автоматизации, основанные на имитации технологического процесса).

Конечным результатом работы инструментальной системы TRACE MODE 6 является набор файлов, предназначенных для исполнения задач АСУ в мониторах реального времени на АРМах и в контроллерах. В рассматриваемых далее случаях в качестве МРВ для АРМ будет использоваться профайлер с поддержкой графических экранов **rtc.exe**, а для контроллера - без поддержки графических экранов **rtmg32.exe**, расположенные в директории инструментальной системы TRACE MODE 6.

Цель данного пособия – научить студентов самостоятельно создавать ПО в среде TRACE MODE 6., помочь усвоить лекционный курс информационных дисциплин, приобрести навыки самоконтроля и развития мышления.

Лабораторная работа № 1.

Интерфейс ПО «Система мониторинга с узлом АРМ»

1. Цель работы

Создать систему мониторинга, содержащую один узел АРМ (автоматизированное рабочее место) с использованием механизма автопостроения каналов TRACE MODE методом «от шаблона экрана».

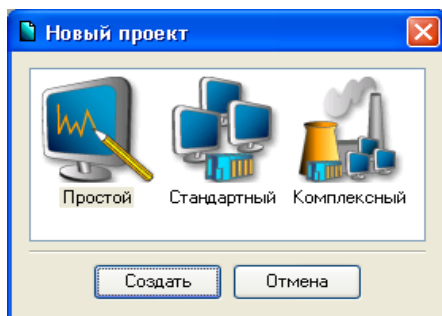
2. Порядок выполнения работы

- 1). Изучите теоретическую часть. Выполните задание и продемонстрируйте его преподавателю
- 2). Оформите отчет по лабораторной работе (см. приложение 1).

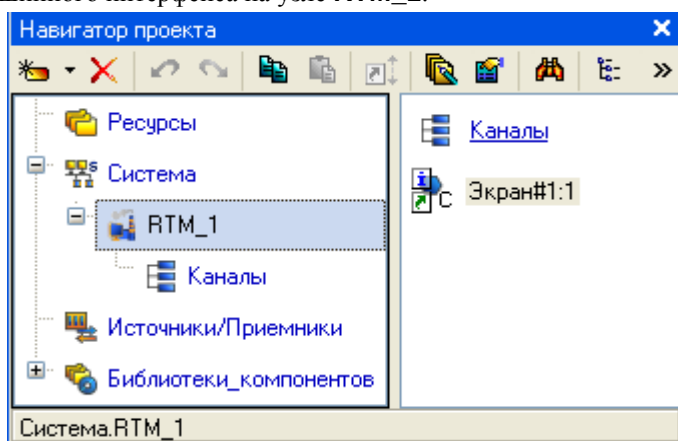
3. Образец выполнения работы

Создание узла АРМ

Загрузим инструментальную систему двойным щелчком левой клавиши (ЛК) мыши по иконке  рабочего стола Windows и с помощью иконки  инструментальной панели создадим новый проект. При этом в открывшемся на экране диалоге



выберем стиль разработки **Простой**. После нажатия ЛК мыши на экранной кнопке **Создать**, в левом окне Навигатора проекта отобразится дерево проекта, содержащее слои **Ресурсы**, **Система** (с созданным узлом АРМ **RTM_1**), **Источники/Приемники** и **Библиотеки_компонентов**. В правом окне Навигатора проекта отобразится содержимое узла **RTM_1** – пустая группа **Каналы** и один канал класса **CALL Экран#1**, вызывающий соответствующий компонент – шаблон экрана, предназначенный для отображения с помощью графических элементов (ГЭ) средств человеко-машинного интерфейса на узле **RTM_1**:



Создание графического экрана

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

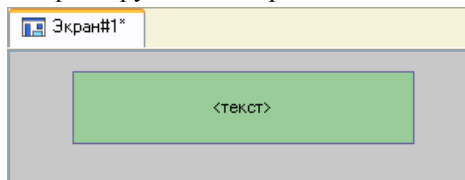
Создание статического текста

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

Для этого выполним следующие действия: на панели инструментов графического редактора ЛК мыши выделим иконку ГЭ Текст ;

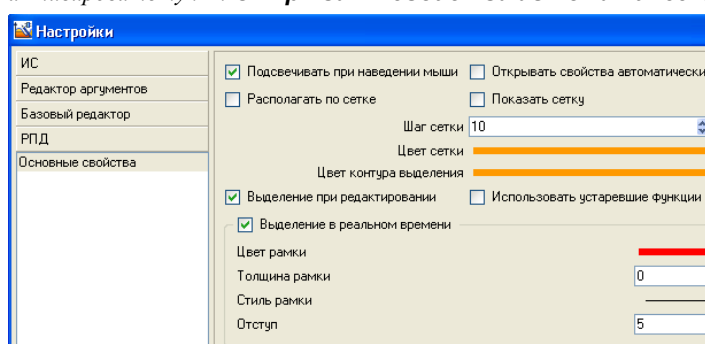
в поле графического редактора установим прямоугольник ГЭ, для чего

- зафиксируем ЛК *точку привязки* - левый верхний угол;
- развернем прямоугольник движением курсора до необходимого размера;
- зафиксируем ЛК выбранный ГЭ:



– Для перехода в режим редактирования атрибутов размещенного ГЭ выделим ЛК иконку  на панели инструментов;

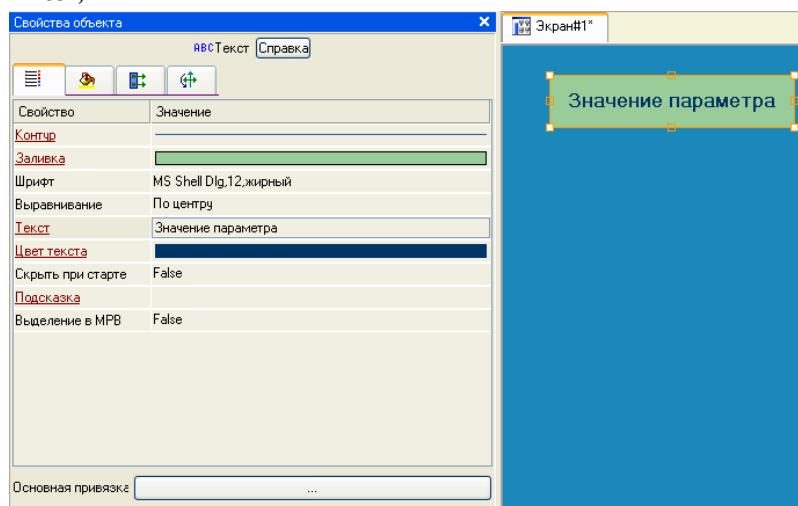
1. Для автоматического вывода окна свойств ГЭ по завершению его размещения необходимо в настройках интегрированной среды разработки в разделе **РПД/Основные свойства** активировать пункт **Открывать свойства автоматически**.



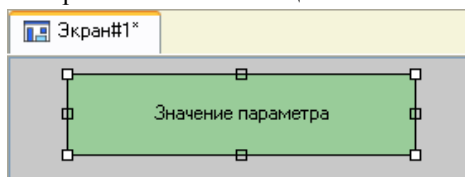
2.

– двойным щелчком ЛК по ГЭ откроем окно его свойств;

– в правом поле строки **Текст** наберем **Значение параметра** и нажмем на клавиатуре клавишу **Enter**;



– закроем окно свойств щелчком ЛК по иконке , ГЭ примет следующий вид:

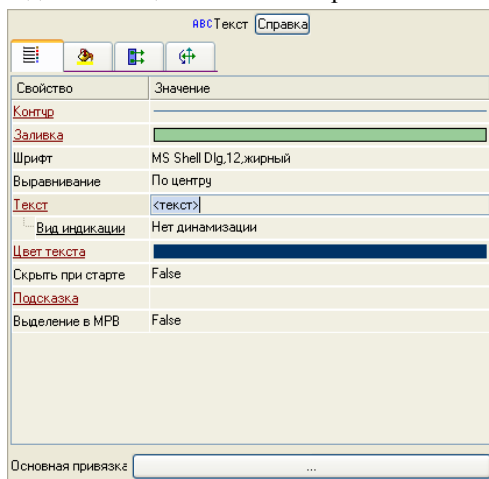


Если введенный Вами текст не уместился в прямоугольнике ГЭ, выделите его и растяните до нужного размера с помощью мыши.

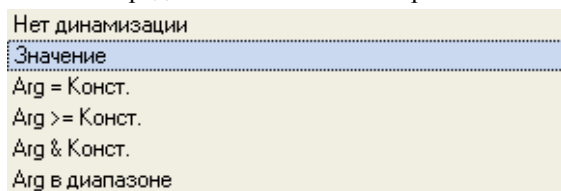
Создание динамического текста, создание аргумента экрана в процессе настройки динамического текста

Подготовим на экране вывод динамического текста для отображения численного значения какого-либо источника сигнала – внешнего или внутреннего - путем указания динамизации атрибута **Текст** ГЭ. Определим назначение аргумента шаблона экрана. Для этого необходимо проделать следующие действия:

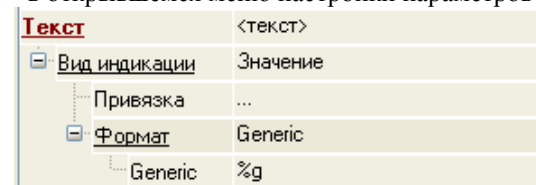
- создадим и разместим новый ГЭ  справа от ГЭ с надписью **Значение параметра**;
- откроем свойства вновь размещенного ГЭ;
- двойным щелчком ЛК на строке **Текст** вызовем меню **Вид индикации**;



- в правом поле строки щелчком ЛК вызовем список доступных типов динамизации атрибута;
- из всех предлагаемых типов выберем ЛК **Значение**;

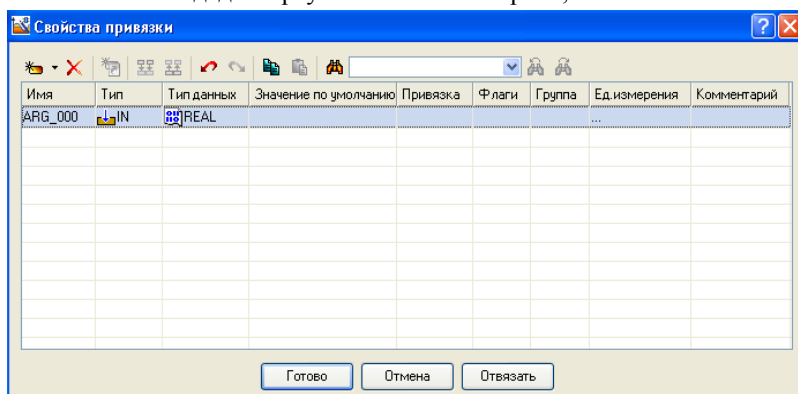


- в открывшемся меню настройки параметров динамизации:



- выполним щелчок ЛК в правом поле строки **Привязка**;

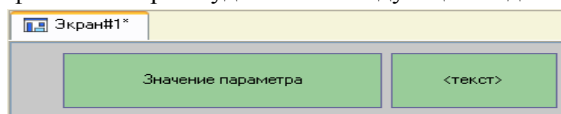
- в открывшемся окне **Свойства привязки**, нажмем ЛК по иконке  на панели инструментов и тем самым создадим аргумент шаблона экрана;



- двойным щелчком ЛК выделим имя аргумента и изменим его, введя с клавиатуры **Параметр** (здесь и в дальнейшем ввод данных с клавиатуры будем завершать нажатием клавиши **Enter**);

- подтвердим связь атрибута **Текст** ГЭ с данным аргументом щелчком ЛК по экранной кнопке **Готово**;
- закроем окно свойств ГЭ.

Графический экран будет иметь следующий вид:



Создание стрелочного прибора, привязка к аргументу

Применим для отображения параметра новый тип ГЭ – Стрелочный прибор. Для этого необходимо выполнить последовательность действий:

- выделим двойным щелчком ЛК на инструментальной панели графического редактора иконку



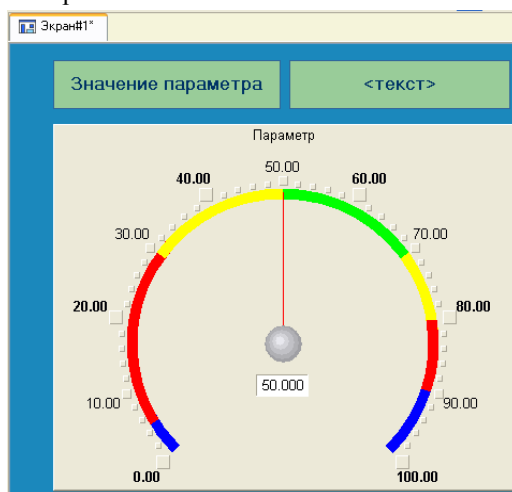
и выберем из появившегося меню иконку стрелочного прибора



- установим ГЭ , выбрав его размер таким, чтобы все элементы графики и текста на нем были разборчивы и симметричны;

- перейдем в режим редактирования и откроем окно свойств ГЭ ;
 - щелчком ЛК на экранной кнопке **Основная привязка** откроем окно табличного редактора аргументов шаблона экрана;
- ЛК выберем уже имеющийся аргумент **Параметр**;
- подтвердим выбор щелчком ЛК на кнопке **Готово**;
- двойным щелчком ЛК откроем атрибут **Заголовок** и в строке **Текст** введем надпись **Параметр**;

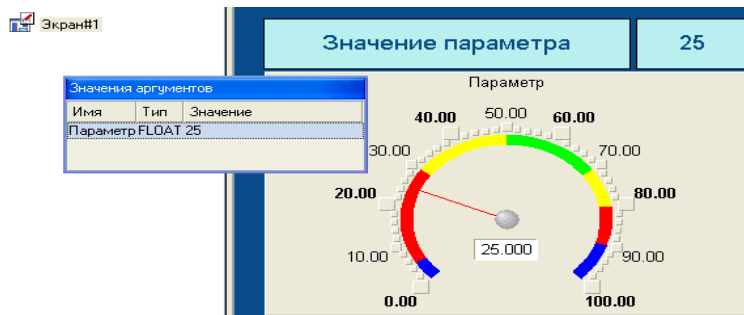
- закроем окно свойств ГЭ .



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

Значения аргументов		
Имя	Тип	Значение
Параметр	FLOAT	0

Так, введя значение 25, наблюдаем:

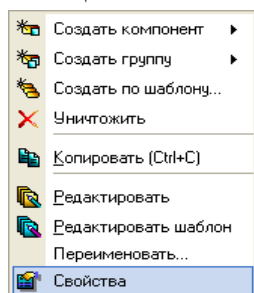


В нашем случае оба ГЭ отображают введенное значение – привязки выполнены правильно. Выход из режима эмуляции – повторное нажатие ЛК по иконке

Автопостроение канала

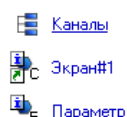
Для создания канала в узле проекта по аргументу шаблона экрана воспользуемся процедурой *автопостроения*. Для этого:

- 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**:



Задание границ и уставок

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

Имя: Параметр Кодировка: TC5 Справка

Комментарий:

Границы

☒ Использовать

ВП: 90 ВА: 80 ВГ: 70 НГ: 50 НА: 30 НП: 5 Гистерезис: 0 ☒ Контроль границ

Обработка

☐ Использовать

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

☒ Масштабирование

Масштабирование: In Мин Макс Множитель Смещение А Мин Макс

Max: 1 x 1 + 0 = Min: (Рассчитать)

Системные

Основные

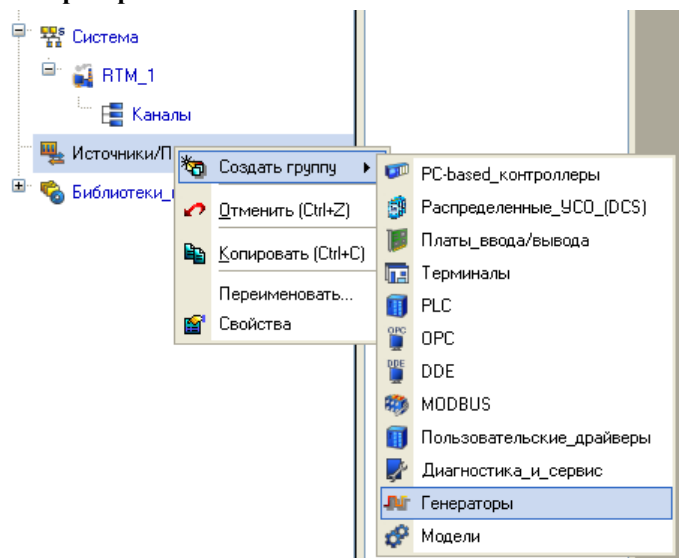
Тип: Input Размерность: ... Период: 1 Единица измерения: цикл Автопосылка: ☐ Включить Индекс: 0 ☐ Отработать На старте: 0

Архивация: Дополнительно:

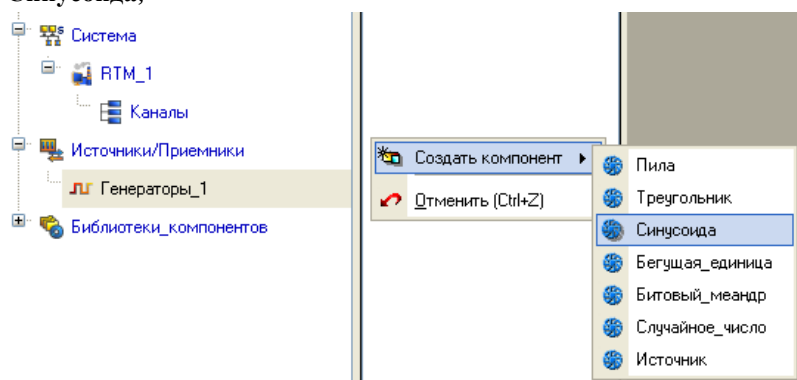
Создание генератора синуса и привязка его к каналу

Введем в состав проекта источник сигнала – внутренний генератор синусоиды, свяжем его с созданным каналом и опробуем в работе выполненные средства отображения. Для этого сделаем следующие действия:

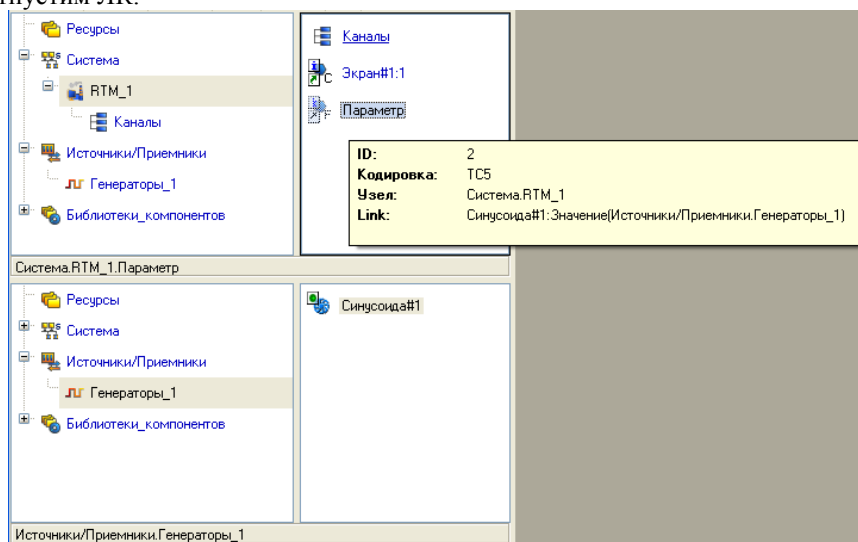
- откроем слой **Источники/Приемники** и через ПК создадим в нем группу компонентов **Генераторы**:



- двойным щелчком ЛК откроем группу **Генераторы_1** и через ПК создадим в ней компонент **Синусоида**;



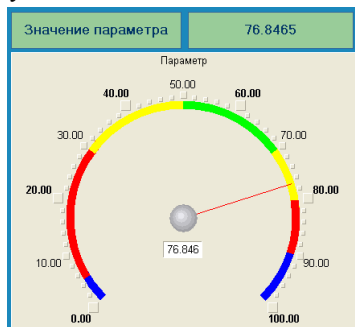
- захватим с помощью ЛК созданный источник и, не отпуская ЛК, перетащим курсор на узел **RTM_1** в слое **Система**, а затем, в открывшемся окне компонентов, на канал **Параметр**. Отпустим ЛК.



Запуск проекта

- Сохраним проект с помощью иконки , задав в открывшемся окне имя **QS_Lesson_1.prj**;
- на инструментальной панели выберем ЛК иконку  и подготовим проект для запуска в реальном времени;
- ЛК выделим в слое **Система** узел **RTM_1**, а после, нажав ЛК иконку  на инструментальной панели, запустим профайлер;
- запуск/останов профайлера осуществляется с помощью иконки  на его инструментальной панели или клавишной комбинации **Ctrl+R**.

В открывшемся окне ГЭ справа от надписи «Значение параметра» должно показываться изменение синусоидального сигнала. То же значение должен отображать и стрелочный прибор:



После проверки работоспособности созданного проекта остановим профайлер с помощью иконки  и закроем данное приложение с помощью клавишной комбинации **Alt+F4**.

Лабораторная работа № 2.

Функции управления ПО «Узел АРМ»

1. Цель работы

В систему мониторинга, содержащую один узел АРМ (автоматизированное рабочее место), добавить функции управления.

2. Порядок выполнения работы

- 1). Изучите теоретическую часть. Выполните задание и продемонстрируйте его преподавателю
- 2). Оформите отчет по лабораторной работе.

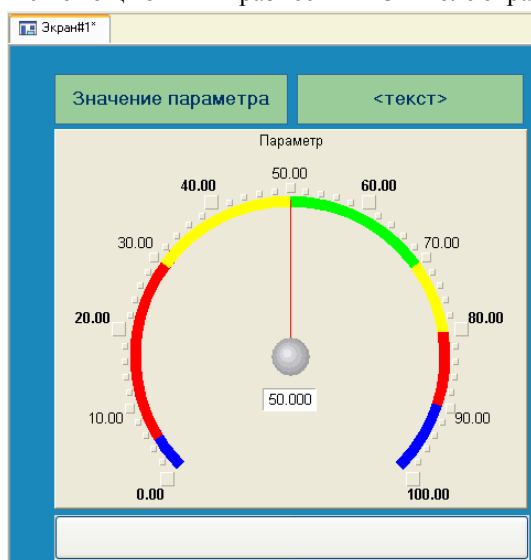
3. Образец выполнения работы

Введем в состав графического экрана ГЭ, позволяющий реализовать ввод числовых значений с клавиатуры. Создадим новый аргумент шаблона экрана для их приема.

Редактирование графического экрана

Для этого:

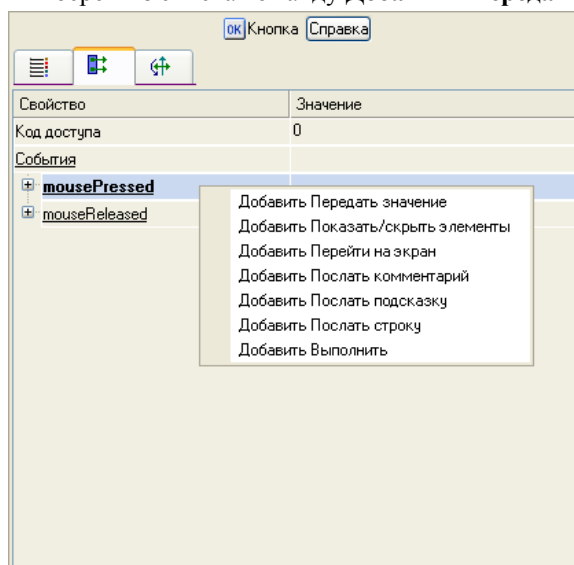
- вызовем графический экран на редактирование;
- на инструментальной панели графического редактора выберем ЛК иконку ГЭ Кнопка -  ;
- с помощью мыши разместим ГЭ в поле экрана под ГЭ  ;



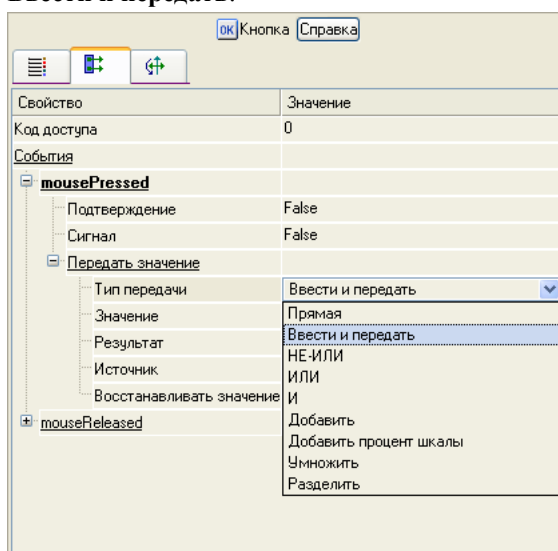
- перейдем в режим редактирования  , выделим ГЭ  ЛК и вызовем окно его свойств:

Свойство	Значение
Два состояния	False
Плоская	False
Текст	Управление
Шрифт	MS Shell Dlg.12, жирный
Цвет текста	
Заливка	
Изображение	
Истинный размер изображения	False
Скрыть при старте	False
Подсказка	

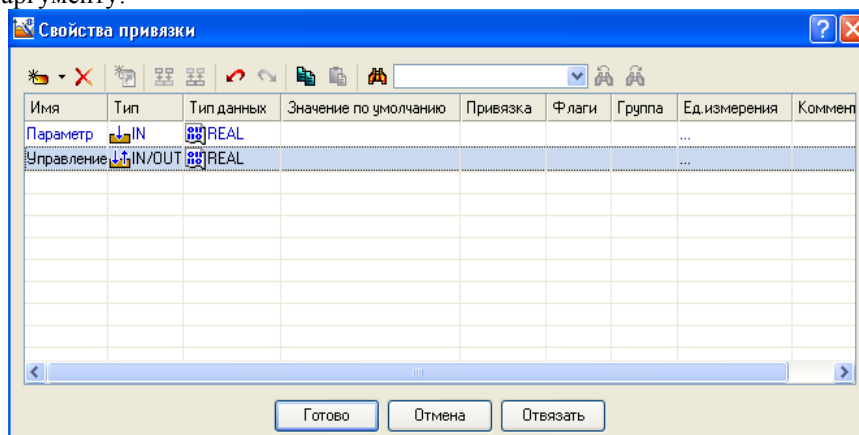
- в поле **Текст** введем надпись **Управление**;
- откроем бланк **События**  и ПК раскроем меню **По нажатию (mousePressed)**;
- выберем из списка команду **Добавить Передать значение**;



- в раскрывшемся меню настроек выбранной команды в поле **Тип передачи** выберем из списка **Ввести и передать**:



- щелчком ЛК в поле **Результат** вызовем табличный редактор аргументов;
- создадим еще один аргумент и зададим ему имя **Управление**;
- изменим тип аргумента на **IN/OUT**, кнопкой **Готово** подтвердим привязку атрибута ГЭ к этому аргументу:



- закроем окно свойств ГЭ с помощью щелчка ЛК по иконке .

Далее выполним размещение ГЭ Текст для отображения вводимого с клавиатуры значения. Воспользуемся уже имеющимся на графическом экране ГЭ путем его копирования/вставки и перепривязки. Для этого:

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



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

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

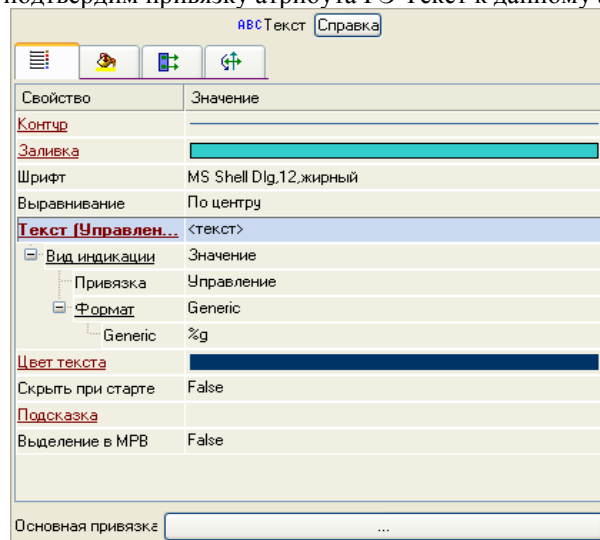
- переместим, удерживая нажатой ЛК, копию ГЭ Текст справа от размещенного на экране ГЭ Кнопка;

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

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

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

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



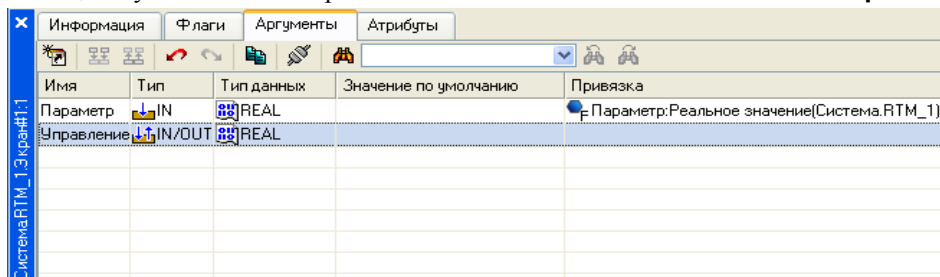
- закроем окно свойств ГЭ Текст.

Привязка аргумента экрана к каналу

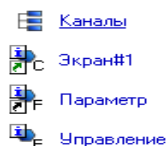
Создадим по аргументу **Управление** шаблона экрана новый канал, отредактируем привязку атрибута канала к аргументу шаблона экрана. Для этого:

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

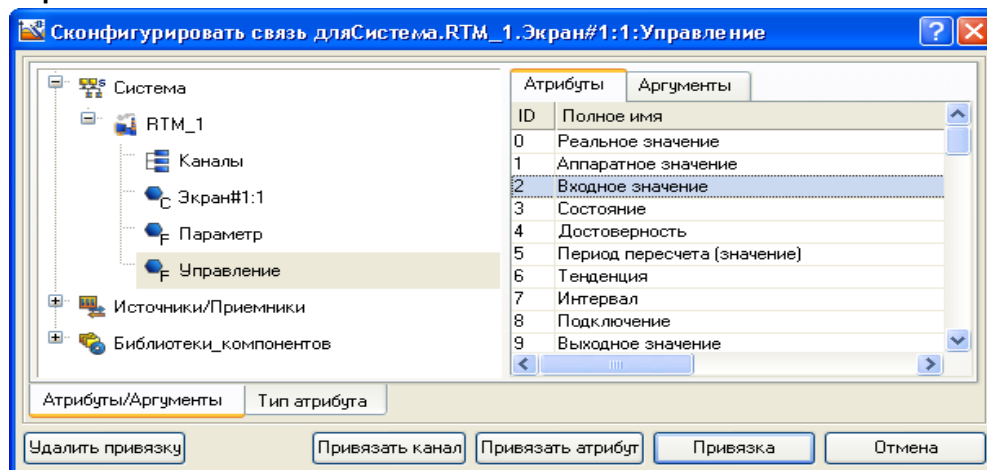
- по щелчку ПК вызовем через контекстное меню свойства компонента **Экран#1**:



- выберем вкладку **Аргументы**, ЛК выделим аргумент **Управление** и с помощью иконки  выполним автопостроение канала;
- в результате, в узле **RTM_1**, будет создан канал с именем **Управление**:



- двойным щелчком ЛК в поле **Привязка** аргумента **Управление** вызовем окно настройки связи, выберем в нем атрибут **Входное значение** канала **Управление** и кнопкой **Привязка** подтвердим связь аргумента экрана **Управление** с атрибутом **Входное значение** канала **Управление**:

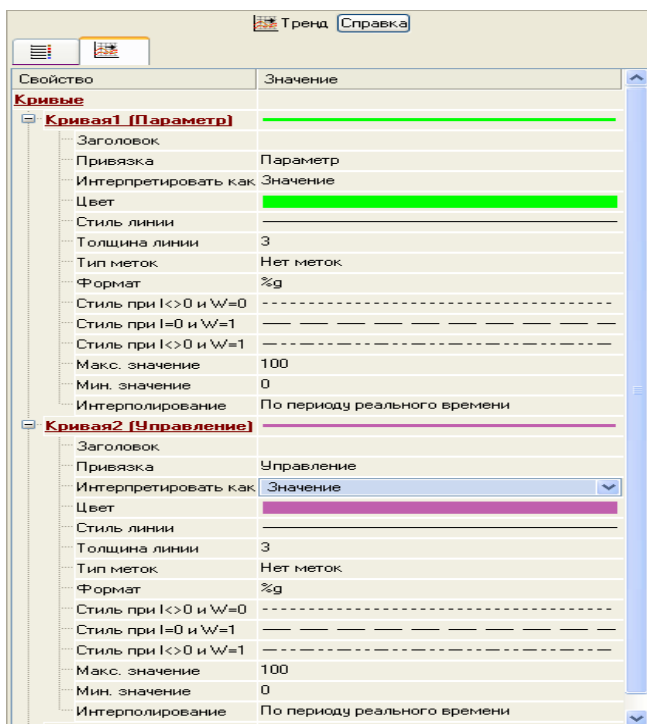


- закроем окно свойств компонента **Экран#1**.

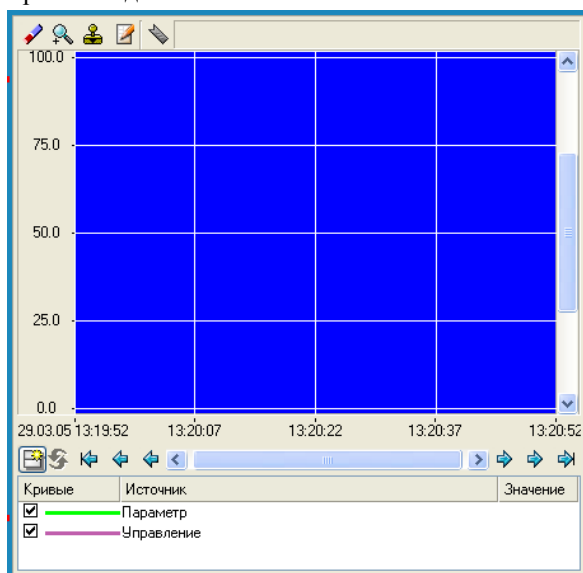
Размещение ГЭ Тренд

Дополним созданный экран новым ГЭ для совместного просмотра изменений значений каналов узла во времени и отслеживании предыстории – трендом.

В правой части графического экрана разместим ГЭ Тренд  для вывода значений **Параметр** и **Управление**. Основные свойства ГЭ  оставим заданными по умолчанию. Перейдем во вкладку , выделив ЛК строку **Кривые**, с помощью ПК создадим две новых кривых. Настроим для них привязки к существующим аргументам, толщину и цвет линий:



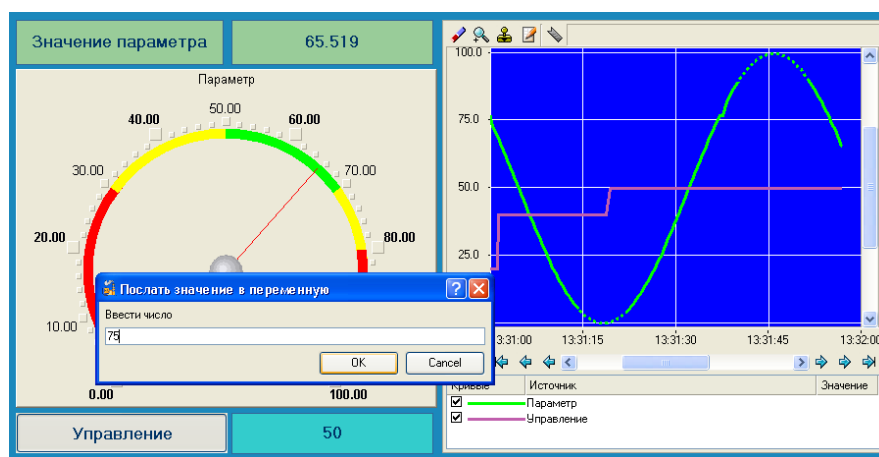
ГЭ примет вид:



Запуск проекта

- Сохраним проект с помощью иконки  ;
- на инструментальной панели выберем ЛК  и подготовим тем самым проект для запуска в реальном времени;
- с помощью иконки  на инструментальной панели запустим проект на исполнение.

Теперь с помощью кнопки **Управление** будем вводить величину «управляющего воздействия» и наблюдать результат в соседнем поле и тренде:



Лабораторная работа № 3.

Форматирование экранной формы ПО «Узел АРМ»

1. Цель работы

С помощью нового компонента проекта – шаблона программы связать два имеющихся канала операцией сложения.

2. Порядок выполнения работы

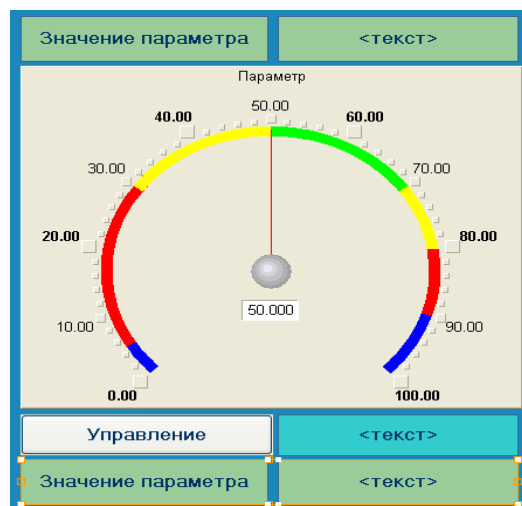
- 1). Изучите теоретическую часть. Выполните задание и продемонстрируйте его преподавателю
- 2). Оформите отчет по лабораторной работе.

3. Образец выполнения работы

Будем 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.

Доработка графического экрана

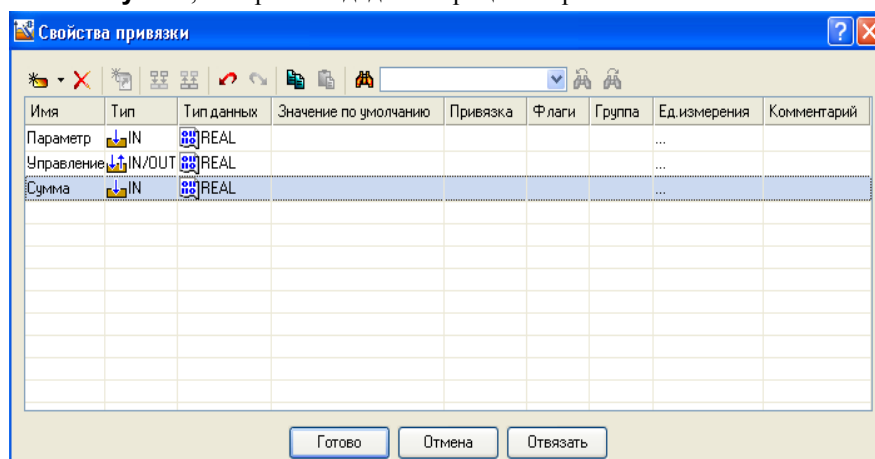
– Скопируем два первых ГЭ – «Значение параметра» и «текст» и разместим их ниже ГЭ Кнопка;



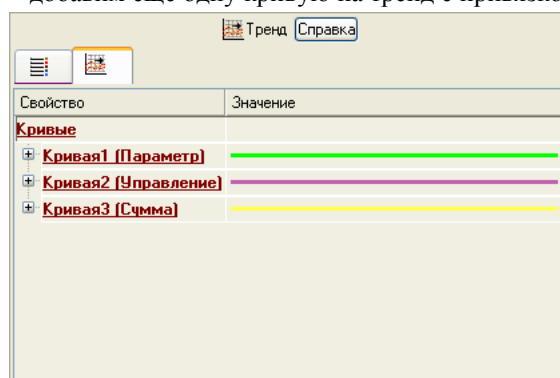
- изменим статический текст первого ГЭ на **Сумма** ;



– динамику второго ГЭ привяжем к новому - третьему аргументу шаблона экрана типа **IN** с именем **Сумма**, который создадим в процессе привязки:



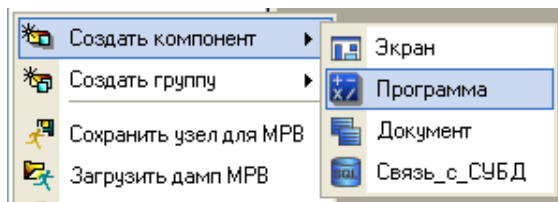
– добавим еще одну кривую на тренд с привязкой к аргументу **Сумма**.



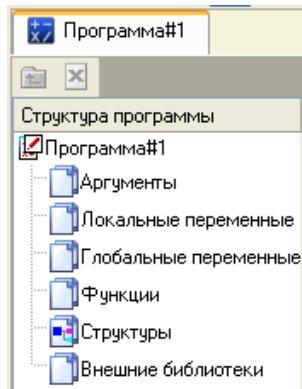
Создание программы на языке Техно ST

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

– Двойным щелчком ЛК откроем узел **RTM_1**:



- создадим в нем компонент **Программа**;
- двойным щелчком ЛК по компоненту **Программа#1** перейдем в режим редактирования программы:

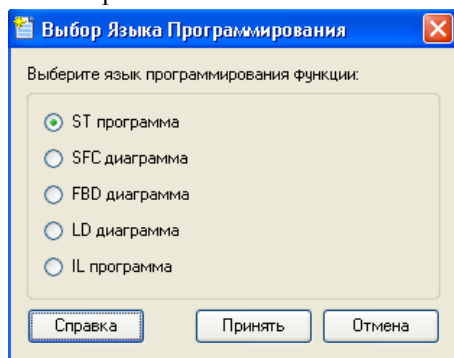


- выделением ЛК в дереве шаблона **Программа#1** строки **Аргументы** вызовем табличный редактор аргументов;

- с помощью иконки  создадим в редакторе аргументов три аргумента с именами **Параметр**, **Управление** и **Сумма**. При этом первые два аргумента должны быть типа **IN**, а третий – **OUT**;

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

- выделим ЛК в дереве шаблона строку **Программа#1** и в открывшемся диалоге **Выбор языка** выберем язык **ST**:



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

```

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

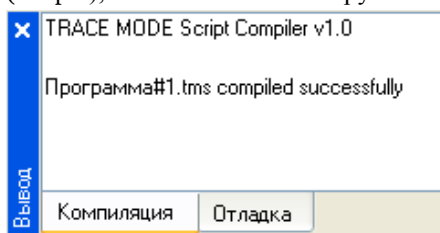
  Сумма=Параметр+Управление;

END_PROGRAM

```

- с помощью иконки  на инструментальной панели редактора или нажатием «горячей

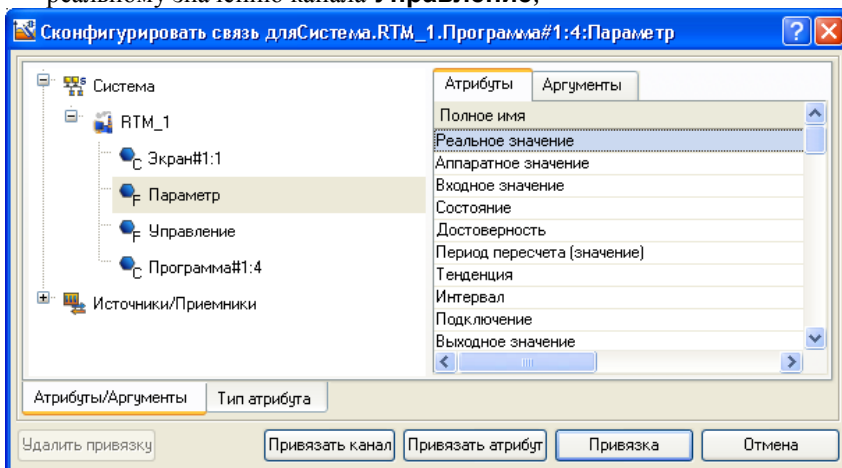
клавиши» **F7** скомпилируем программу и убедимся в успешной компиляции в окне **Выход** (Output), вызываемом из инструментальной панели с помощью иконки :



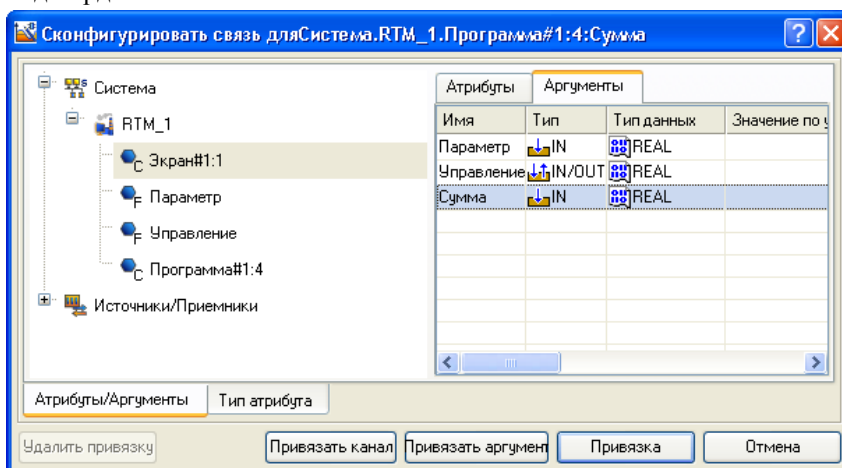
Привязка аргументов программы

Выполним привязку аргументов программы к атрибутам каналов:

- вызовем свойства компонента **Программа#1** через контекстное меню;
- выберем вкладку **Аргументы**;
- двойным нажатием в поле **Привязка** свяжем 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** вызовем окно настройки связи, выберем в левом окне канал класса **CALL Экран#1**, а в правом откроем вкладку **Аргументы** и укажем в ней аргумент **Сумма**, затем щелчком ЛК по экранной кнопке **Привязка** подтвердим связь:



В результате, будем иметь:

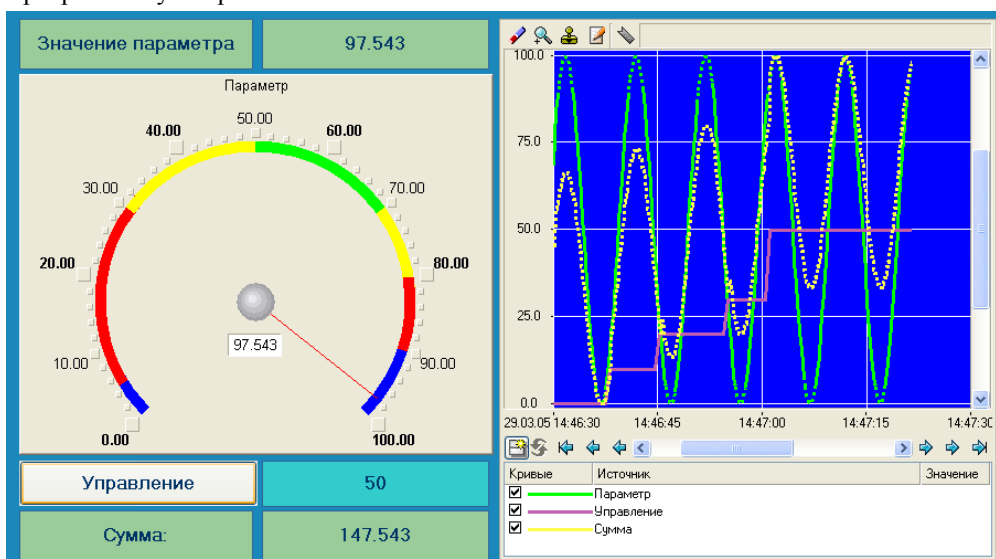
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Параметр	IN	REAL		Параметр:Реальное значение(Система.RTM_1)
Управление	IN	REAL		Управление:Реальное значение(Система.RTM_1)
Сумма	OUT	REAL		Экран#1:1:Сумма(Система.RTM_1)

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

Запуск проекта

- Сохраним проект с помощью иконки  ;
- на инструментальной панели выберем ЛК иконку  и подготовим тем самым проект для запуска в реальном времени;
- с помощью иконки  на инструментальной панели запустим режим исполнения.

Вводя теперь с помощью кнопки «Управление» «управляющие воздействия» будем наблюдать соответствующее изменение реального значения канала **Управление** и результата работы программы суммирования:



Лабораторная работа № 4.

Связь по протоколу DDE с математическим редактором.

Подключение реального внешнего модуля ввода сигналов

1.Цель работы

Организовать работу запроса реальных значений с математическим редактором. Внедрить в созданный проект модуль удаленного ввода I-7017.

2. Порядок выполнения работы

- 1). Изучите теоретическую часть. Выполните задание и продемонстрируйте его преподавателю
- 2). Оформите отчет по лабораторной работе.

3. Образец выполнения работы

Связь по протоколу DDE с математическим редактором.

MPB как DDE-сервер

Организуем запрос реальных значений каналов узла разработанного проекта приложением MS Windows в качестве, которого выберем книгу MS Excel. Для этого выполним:

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

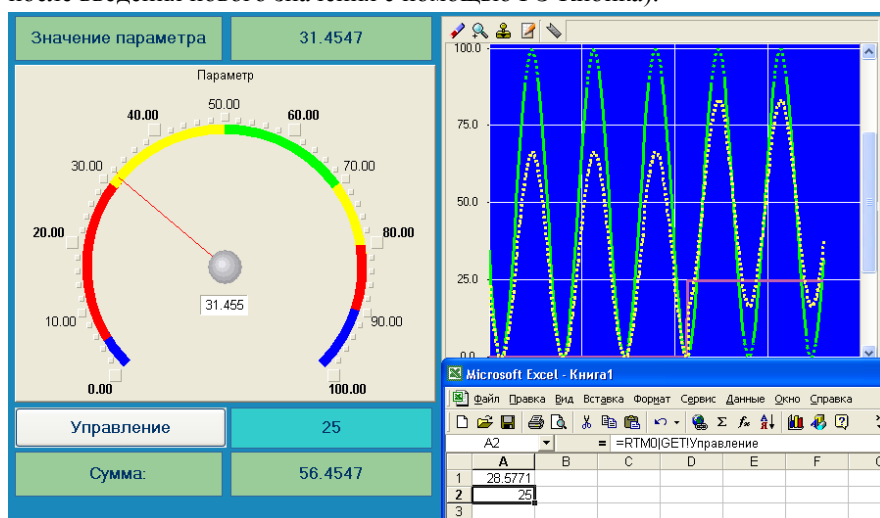
=RTM0|GET!Параметр

=RTM0|GET!Управление

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

- запустим на исполнение узел **APM RTM_1**;
- в меню таблицы MS Excel **Правка** выберем команду **Связи**, выделим оба параметра и нажмем кнопку **Обновить**, после чего закроем окно кнопкой **ОК**.

Убедимся, 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):



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:

The screenshot shows the configuration dialog for a DDE component. The 'Основные' (Basic) tab is active. The 'Имя' (Name) field is set to 'Excel.Sheet1.R3C3'. The 'Кодировка' (Encoding) field is set to 'TWO'. The 'Комментарий' (Comment) field is empty. The 'Параметры' (Parameters) section shows 'Режим' (Mode) set to 'ADVISE', 'A' set to 0, 'B' set to 0, 'Направление' (Direction) set to 'Input', and 'Формат' (Format) set to 'Аналог' (Analog). A 'Справка' (Help) button is visible next to the 'Кодировка' field.

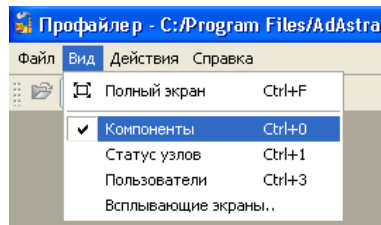
где в атрибуте **Имя**:

- Excel – имя приложения;
- Sheet1 – имя листа книги MS Excel;

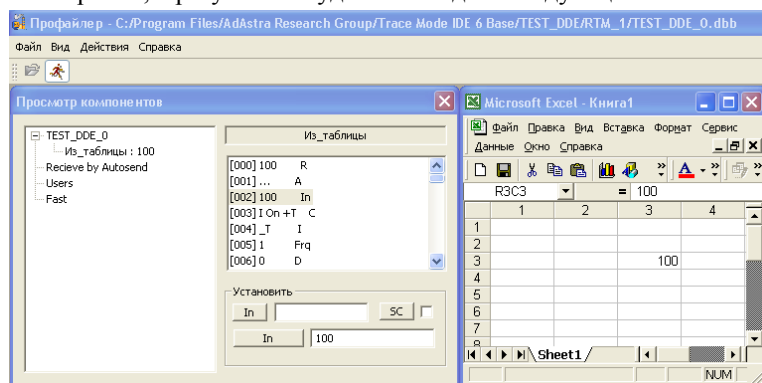
– R3C3 – адрес ячейки.

ADVISE – режим отправки клиенту значения при каждом его изменении.

После 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, запустим MS Excel, а затем узел **APM 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:



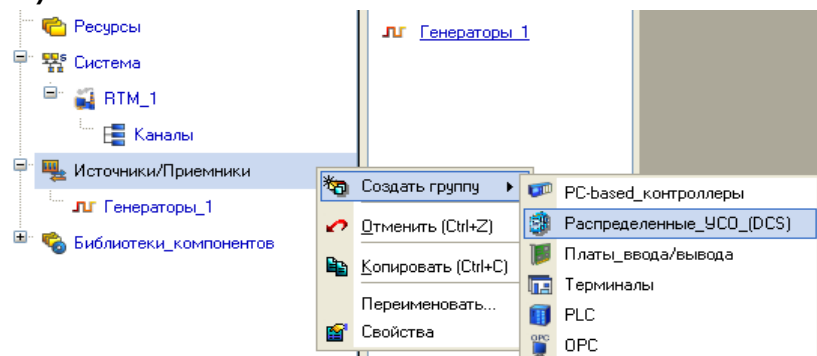
Подключение реального внешнего модуля ввода сигналов

Введем в созданный проект модуль удаленного ввода **I-7017** с подключенным к одному из его входов датчиком напряжения. Предварительно настроим модуль с помощью конфигурационной утилиты, поставляемой с модулем на указанную характеристику, зададим «**инженерный**» формат вывода данных, присвоим ему номер в сети **RS-485** равный **1** и установим формат обмена данными **57600,n,8,1** без контрольной суммы. Подключим модуль к порту **COM1** компьютера через автоматический конвертор интерфейсов **I-7520**, обеспечим питание обоих модулей.

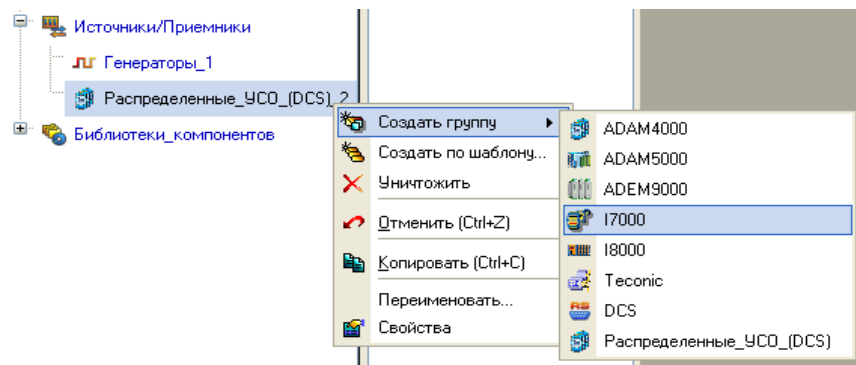
Создание канала аналогового ввода данных от модуля I-7017

Создать один DCS-модуль аналогового ввода:

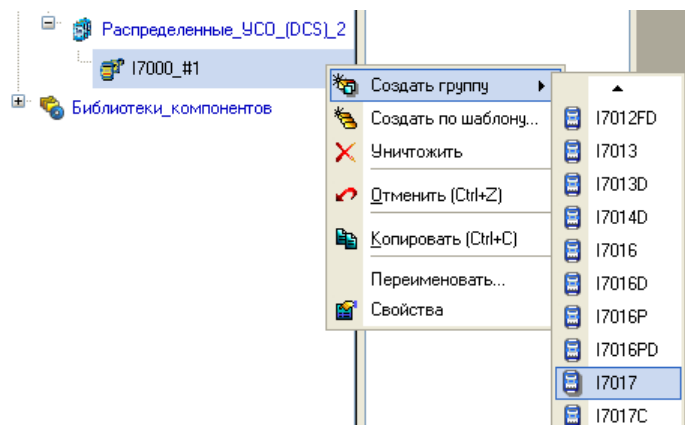
открыть слой **Источники/приемники** и через ПК создать в нем группу **Распределенное УСО (DCS)**:



двойным щелчком ЛК открыть группу **Распределенное УСО (DCS)_1** и через ПК создать в ней группу **I-7000**;



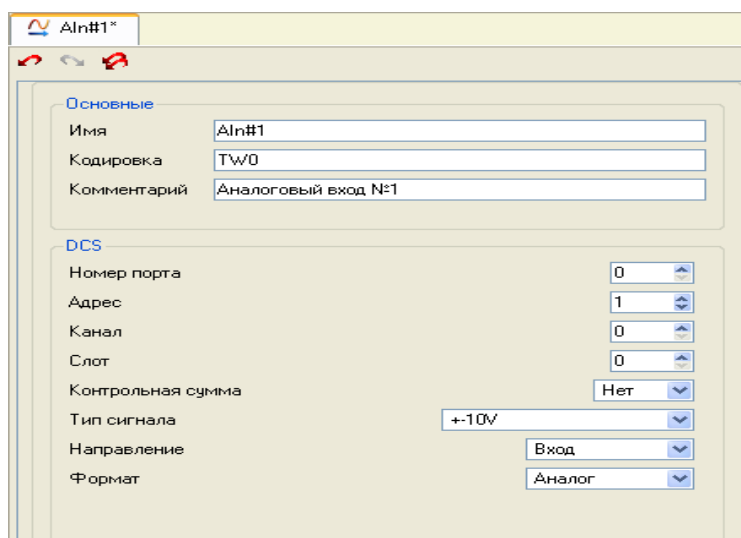
открыть созданную группу **I7000_1** двойным щелчком ЛК и через ПК создать в ней подгруппу **I7017#1**:



открыть созданную подгруппу **I7011#1** двойным щелчком ЛК и перейти к созданным компонентам-источникам модуля I-7017:

 AIn#1
 AIn#2
 AIn#3
 AIn#4
 AIn#5
 AIn#6
 AIn#7
 AIn#8

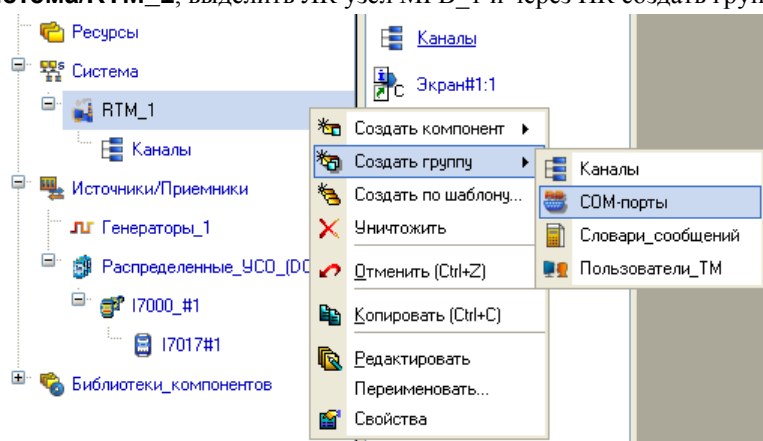
выделить ЛК компонент **AIN#1** и двойным щелчком ЛК перейти в режим редактирования его атрибутов;



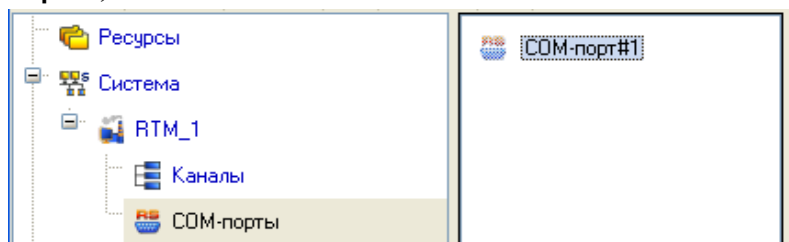
все основные атрибуты, задающие настройки модуля, оставить по умолчанию: **Номер порта** 0 соответствует COM1 компьютера, **Адрес** модуля в нашем случае 1, атрибуты **Канал** и **Слот** для выбранного модуля не задаются, **Контрольная сумма** – отсутствует, **Тип сигнала** - Вход. Изменить из выпадающего меню **Тип сигнала** в соответствии с типом датчика и ввести **Комментарий**.

Создание и настройка COM-порта

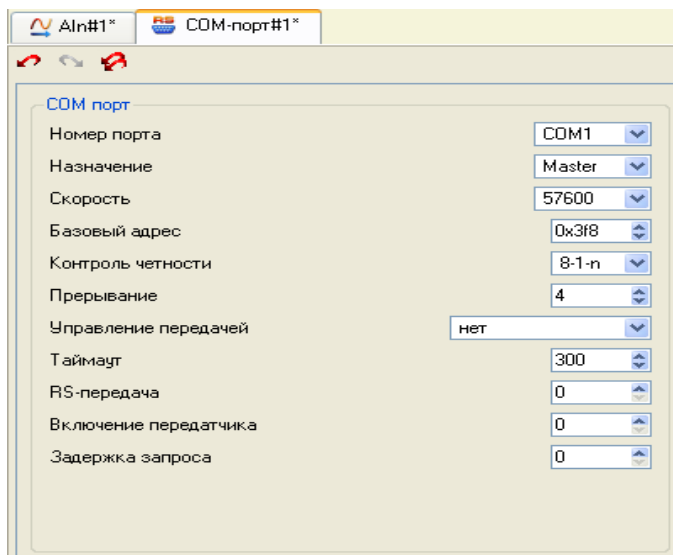
Создать и настроить последовательный порт для узла MPB_1, для этого открыть в окне проекта слой **Система/RTM_1**, выделить ЛК узел MPB_1 и через ПК создать группу **COM-порты**:



двойным щелчком ЛК открыть вновь созданную группу и в окне компонентов выделить ЛК **COM-порт#1**;



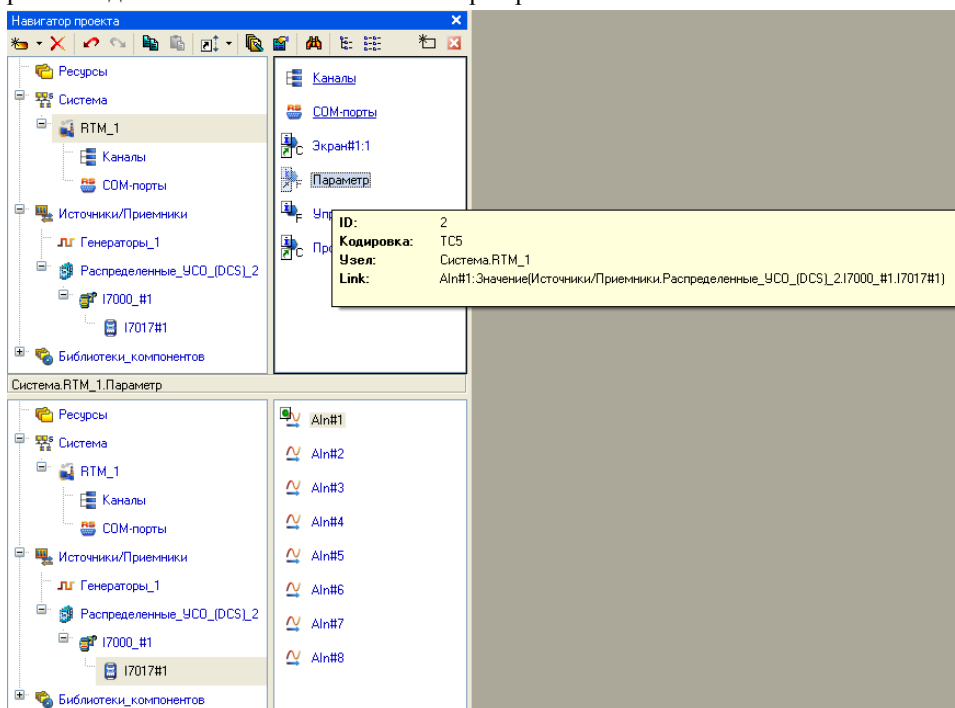
двойным щелчком ЛК на выделенном компоненте **COM-порт#1** открыть для редактирования его атрибуты:



оставить все атрибуты выставленными по умолчанию, изменив лишь **Скорость** на 57600, после закрыть окно атрибутов щелчком ЛК по кнопке .

Изменение привязки канала к источнику данных

Изменим привязку канала **Параметр** с генератора синусоиды на реальный аналоговый входной сигнал. Для этого необходимо открыть с помощью кнопки дополнительное окно Навигатора проекта и открыть в нем слой **Система/RTM_1**, захватить с помощью ЛК компонент-описатель аналогового входа модуля **AIn#1** и, не отпуская ЛК, перетащить курсор на канал **Параметр**, открытый в дополнительном окне Навигатора проекта.



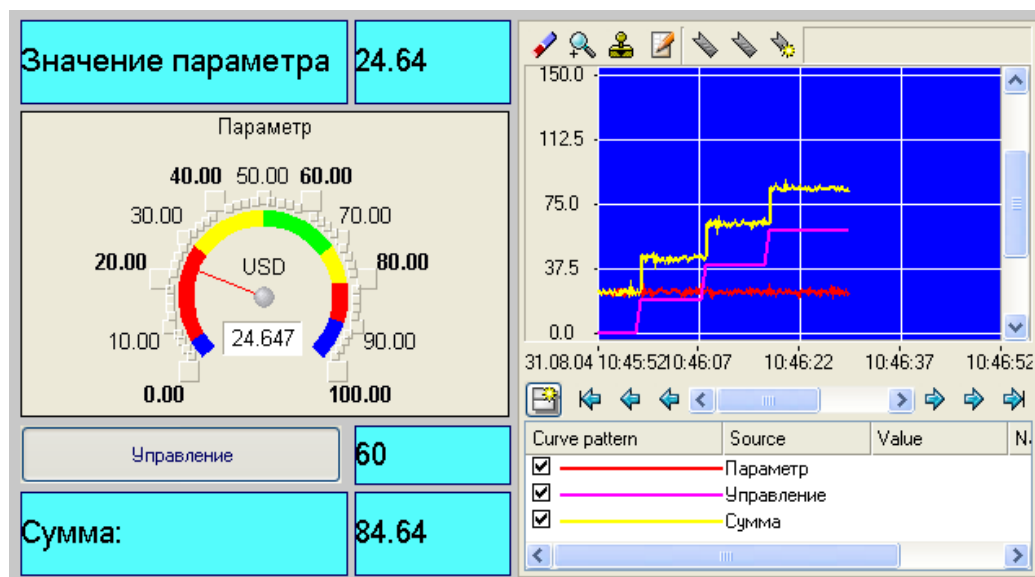
Сохранить выполненные изменения в проекте:

сохранить проект с помощью иконки ;

на панели инструментов выбрать ЛК иконку и скомпилировать тем самым модуль для запуска проекта в реальном времени.

Запуск проекта

Выбрать иконку  на инструментальной панели Редактора проекта и запустить режим эмуляции, вращая ручку задатчика, можно наблюдать изменение сигнала на тренде. Если по каким-либо причинам задатчик отсутствует, то можно использовать любой источник постоянного напряжения, максимальное значение которого не превышает установленный входной диапазон модуля I-7017.



Лабораторная работа № 5. Создание информационной базы.

1. Цель работы

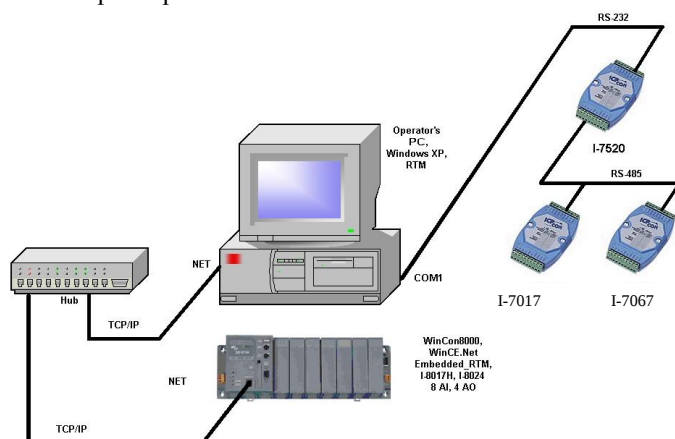
Создать два узла – узел контроллера WinCon8000 и узел APM – MPB для Windows XP.

2. Порядок выполнения работы

- 1). Изучите теоретическую часть. Выполните задание и продемонстрируйте его преподавателю
- 2). Оформите отчет по лабораторной работе.

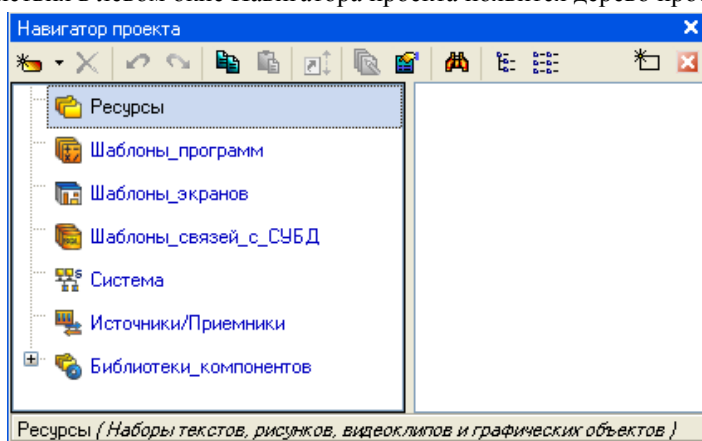
3. Образец выполнения работы

В учебном проекте создадим два узла – узел контроллера WinCon8000, запускаемого под управлением Микро-MPB для ОС WinCE.Net, и узел APM – MPB для Windows XP, связанный по сети с узлом контроллера.

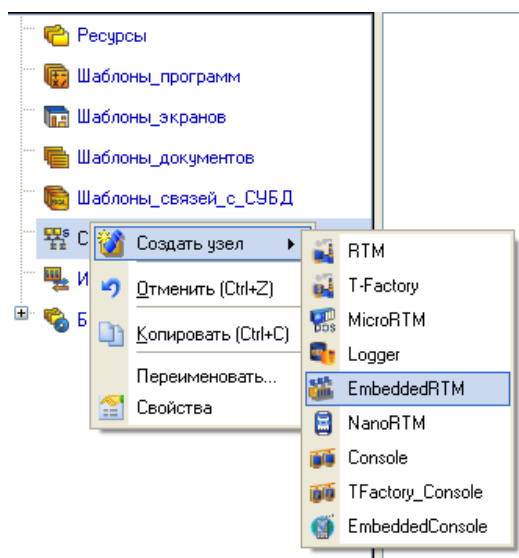


Реализация

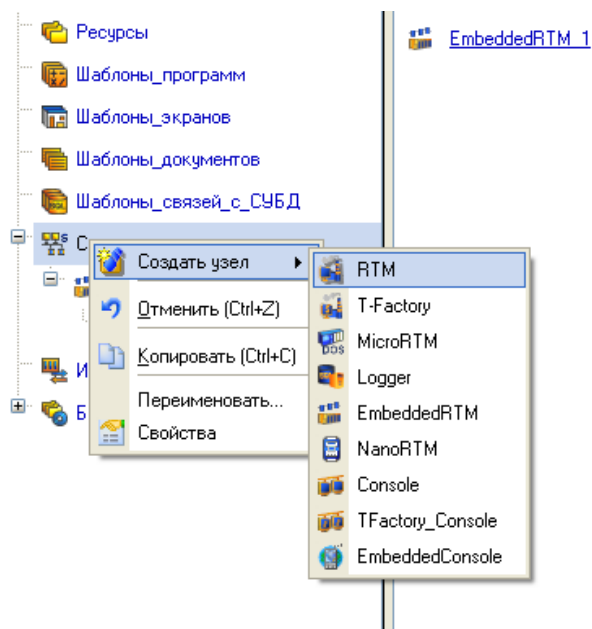
Для начала работы над проектом загрузим инструментальную систему – интегрированную среду разработки TRACE MODE 6 и с помощью иконки  инструментальной панели создать новый проект. В качестве стиля разработки выберем **Стандартный** (Standard). В результате выполненного действия в левом окне Навигатора проекта появится дерево проекта.



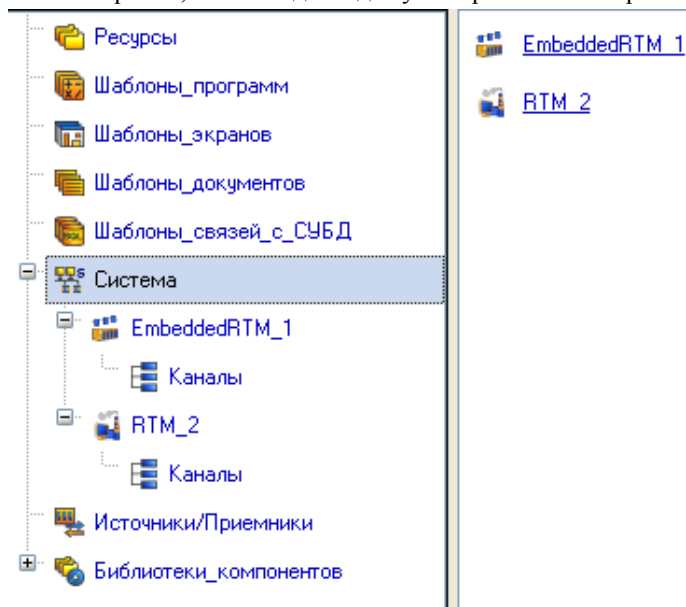
Выделим ЛК мыши в дереве проекта слой **Система**, ПК на правом свободном поле вызовем командное меню и создадим для контроллера узел **EmbeddedRTM**,



Для создания узла АРМ сделаем аналогичную операцию с использованием Навигатора проекта, пользуясь ЛК и ПК мыши:



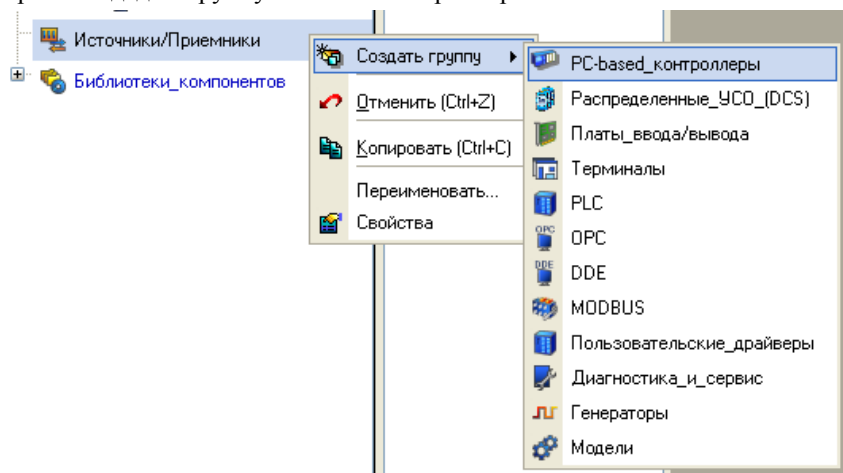
Таким образом, нами созданы два узла проекта – контроллера и АРМ:



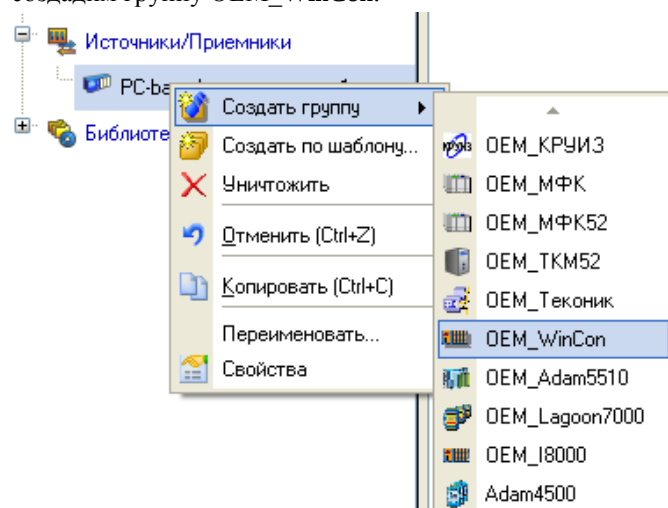
Создание компонентов Источников/приемников контроллера

В качестве контроллера будем использовать PC-based контроллер серии WinCon8000. В контроллере установлено два модуля - I-8017H (8 AI) в слоте 1 и I-8024 (4 AO) в слоте 2. Создадим описание данных модулей.

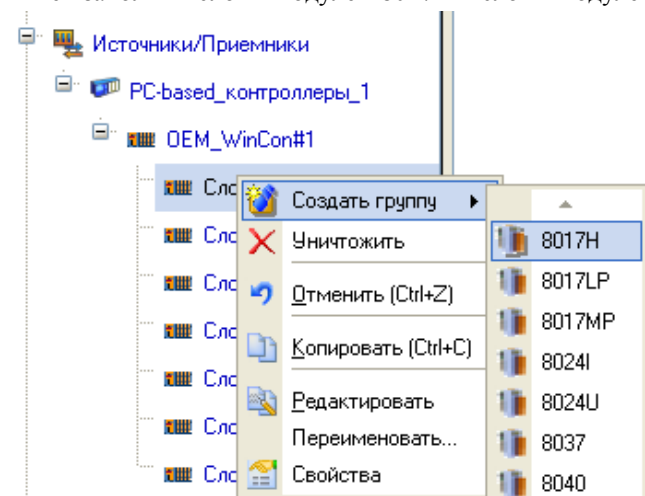
Выделим ЛК слой **Источники/приемники** и ПК вызовем контекстное меню с использованием которого создадим группу PC-based контроллеров:

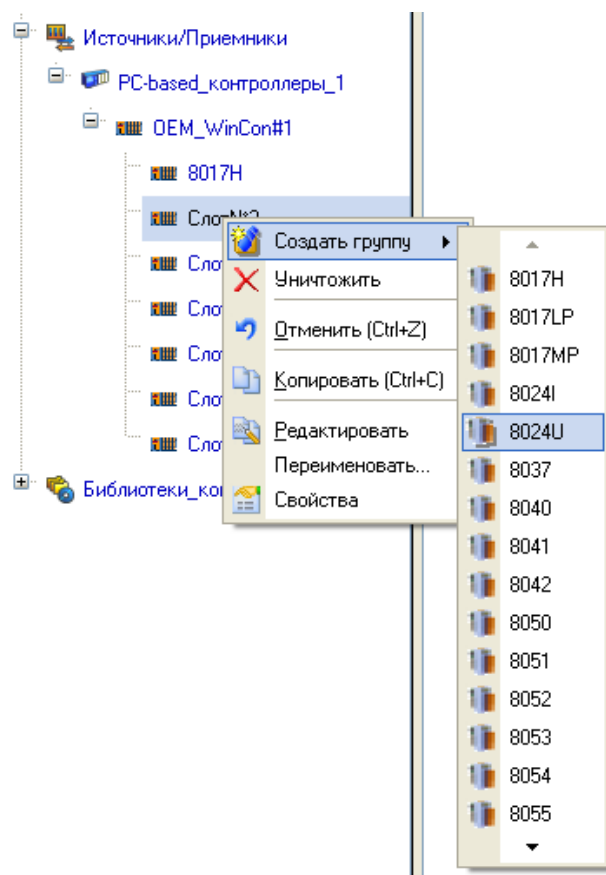


Войдя во вновь созданную группу PC-based_контроллеры_1, также с помощью вызова по ПК меню создадим группу OEM_WinCon:

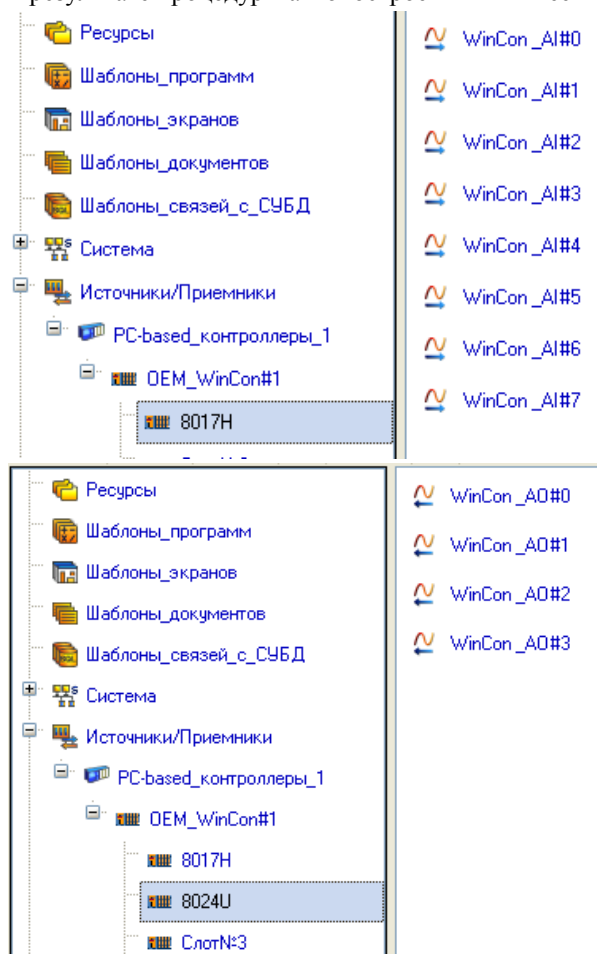


В ней заполним слот 1 модулем 8017H и слот 2 модулем 8024U:





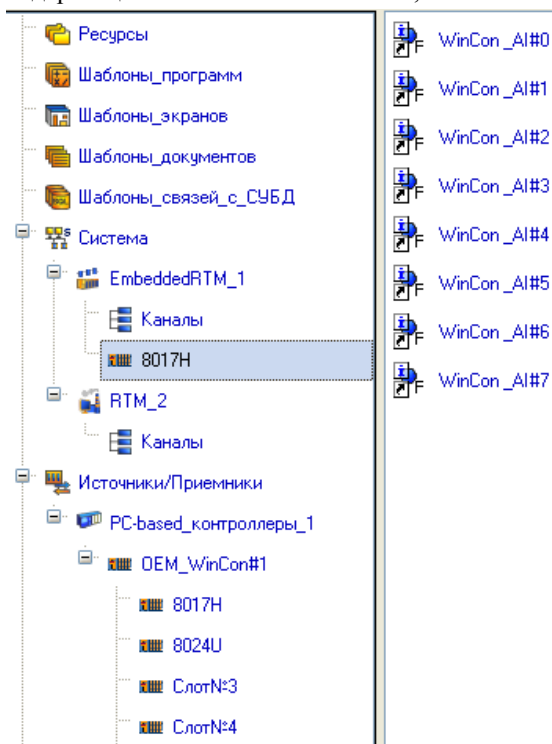
В результате процедуры автопостроения мы имеем источники/приемники для обоих модулей:



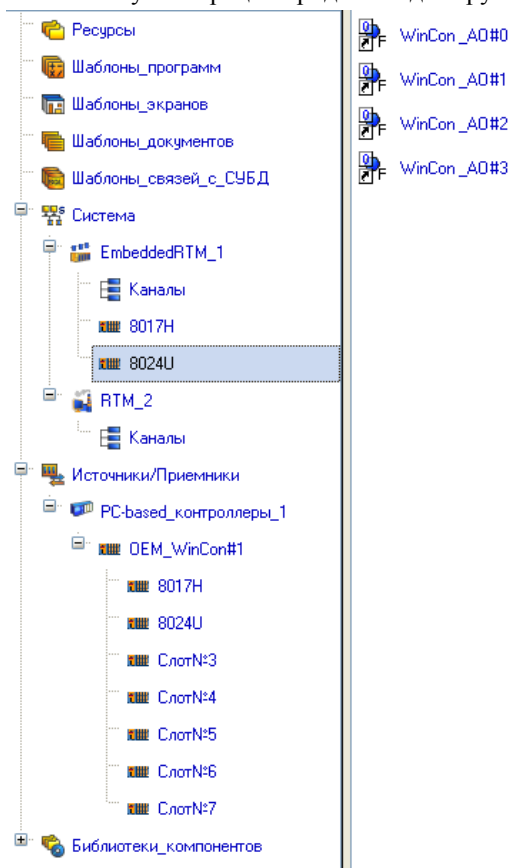
Автопостроение базы каналов в контроллере

Создадим каналы

Для создания в узле контроллера EmbeddedRTM_1 каналов, связанных с модулями необходимо выделить группу 8017H в слое **Источники/приемники**, удерживая ЛК нажатой, перетащим группу 8017H на узел EmbeddedRTM_1 и отпустим ЛК. В узле контроллера будет создана новая группа, содержащая каналы TRACE MODE, связанные с указанной аппаратурой ввода/вывода:



Аналогичную операция проделаем для группы модуля I-8024:



Редактирование каналов

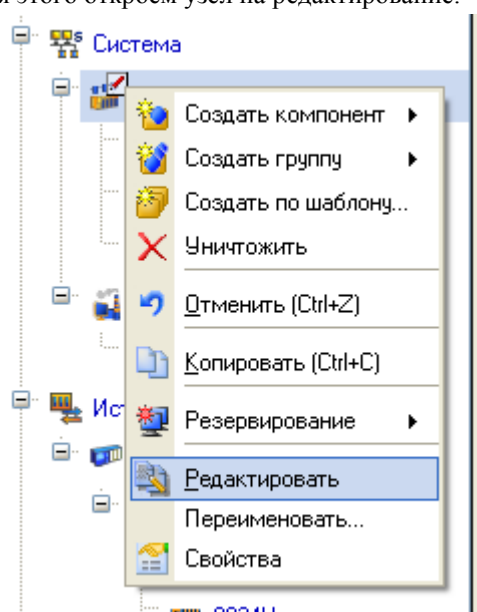
Настройка границ

Зададим границы для созданного канала аналогового ввода **WinCon_AI#0**, для этого двойным щелчком ЛК вызовем редактор канала и заполним бланк Границы следующим образом:

The screenshot shows the 'Границы' (Limits) tab of the WinCon channel editor. The window has a title bar and a menu bar. The main area is divided into three sections: 'Границы' (Limits), 'Обработка' (Processing), and 'Системные' (System). The 'Границы' section contains a list of input channels (ВП, ВА, ВГ, НГ, НА, НП) with their respective values (10, 0, 0, 0, 0, -10) and a 'Гистерезис' (Hysteresis) value of 0. There is a checkbox 'Использовать' (Use) which is checked, and a checkbox 'Контроль границ' (Limit control) which is unchecked. The 'Обработка' section contains a list of processing parameters (Апертура, Пик, Сглаж., Множитель, Смещение) with their respective values (0, 0, 0, 1, 0) and a checkbox 'Использовать' (Use) which is unchecked. There is also a checkbox 'Масштабирование' (Scaling) which is checked. The 'Системные' section contains a list of system parameters (Тип, Размерность, Период, Единица измерения, Автопосылка, Индекс, Отработать, На старте) with their respective values (Input, ..., 1, цикл, Включить, ..., Отработать, 0). There are also checkboxes 'Архивация' (Archiving) and 'Дополнительно' (Additional).

Настройка режима сетевого обмена

Поскольку данные из каналов, созданных в узле контроллера предполагается использовать в узле АРМ, настроим направление сетевого обмена, IP-адрес, период пересчета базы каналов, код проекта. Для этого откроем узел на редактирование:



И установим следующие настройки:

Основные Архивы Отчет тревог / Дамп / Параметры Таймауты Дополнительно

Имя

Атрибуты

Пароль

Характеристика

Пересчет

Период

Разрешение

Сеть

Имя компьютера / IP адрес

Адаптеры

Системный

☐ Прием

☐ Посылка

Первый

☒ Прием

☒ Посылка

Второй

☐ Прием

☐ Посылка

Третий

☐ Прием

☐ Посылка

Мост

☐ Использовать

Искать

Номер узла в проекте

Индивидуальный Групповой Код проекта

Модемы

Телефон №1 Строка инициализации

Телефон №2 Строка инициализации

Регистратор

При старте Статус

Системные

Стороживой таймер Тип дисплея Тип клавиатуры

Лабораторная работа № 6.

Модули подключения технического обеспечения к АРМ

1. Цель работы

Создать компоненты источников/приемников для модулей удаленного ввода/вывода, подключаемых к АРМ

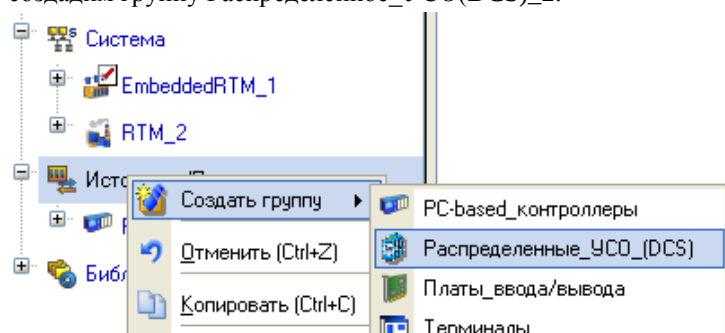
2. Порядок выполнения работы

- 1). Изучите теоретическую часть. Выполните задание и продемонстрируйте его преподавателю
- 2). Оформите отчет по лабораторной работе.

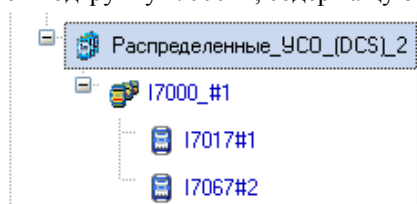
3. Образец выполнения работы

Созданные в предыдущей лабораторной работе два узла WinCon8000 и АРМ подключим по последовательным коммуникациям с модулями удаленного ввода/вывода I-7017 и I-7067.

Для узла АРМ при подключении модулей удаленного ввода/вывода, в слое **Источники/Приемники** создадим группу Распределенное_УСО(DCS)_2:



В ней подгруппу I7000#1, содержащую описатели модулей I-7017 и I-7067:



Групповое редактирование описателей

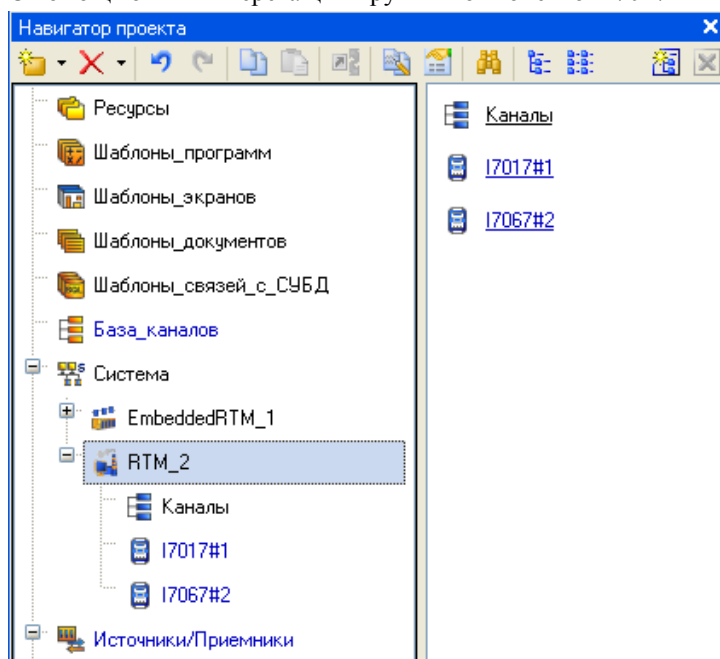
Модули подключены к порту COM1 АРМ и имеют следующие параметры настройки: 57600,n,8,1 без формирования контрольной суммы при обмене данными, адрес модуля I-7017 в сети RS-485 выставлен **1**, I-7067 – **3**. Групповым образом отредактируем описатели обоих модулей:

I7017#1							
Содержимое группы							
Имя							
PORT	ADDR	CH	SLOT	FSC			
AIIn#1	0	1	0	0	Нет		
AIIn#2	0	1	1	0	Нет		
AIIn#3	0	1	2	0	Нет		
AIIn#4	0	1	3	0	Нет		
AIIn#5	0	1	4	0	Нет		
AIIn#6	0	1	5	0	Нет		
AIIn#7	0	1	6	0	Нет		
AIIn#8	0	1	7	0	Нет		

I7017#1 I7067#2							
Содержимое группы							
Имя							
PORT	ADDR	CH	SLOT	FSC			
DO#1	0	3	0	0	Нет		

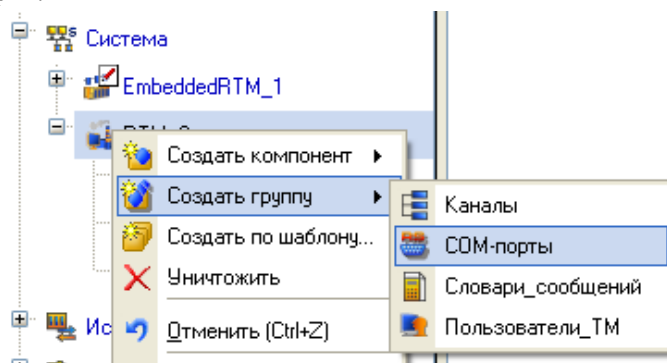
Автопостроение каналов в АРМ

С помощью мыши перетащим группы компонентов I7017#1 и I7067#2 в узел АРМ:

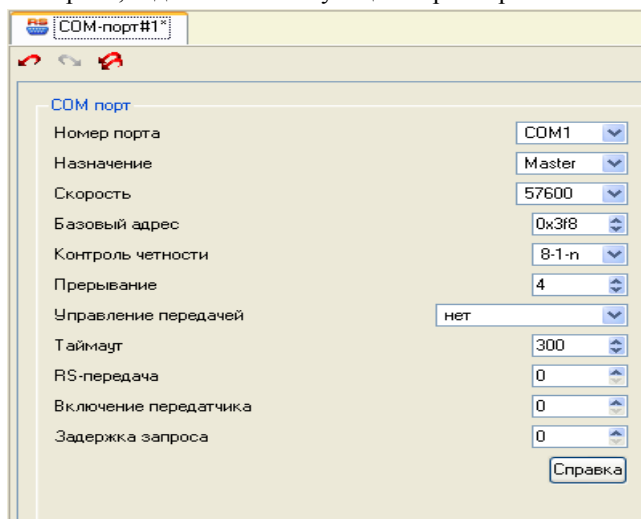


Создание и настройка последовательного порта в АРМ

Создадим и настроим последовательный порт COM1 в узле АРМ. Для этого создадим группу COM-порты:



Войдем во вновь созданную группу и двойным щелчком ЛК откроем для редактирования компонент COM-порт#1, задав соответствующие параметры:



Настройка параметров узла АРМ

Для установления связи с каналами узла контроллера, настроим сетевые настройки узла АРМ.

3. IP-адрес указывается для конкретного компьютера.

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

Имя: RTM_2

Атрибуты

Пароль:

Характеристика:

Пересчет

Период: 10

Разрешение: 0.055

Сеть

Имя компьютера / IP адрес: 192.168.10.1

Адаптеры

Системный

☒ Прием

☒ Посылка

Первый

☐ Прием

☐ Посылка

Второй

☐ Прием

☐ Посылка

Третий

☐ Прием

☐ Посылка

Мост

☐ Использовать

Искать: нет

Номер узла в проекте

Индивидуальный: 2

Групповой: 255

Код проекта: 0

Модемы

Телефон №1:

Строка инициализации:

Телефон №2:

Строка инициализации:

Регистратор

При старте: Включен

Статус: Активен

Системные

Сторожевой таймер: нет

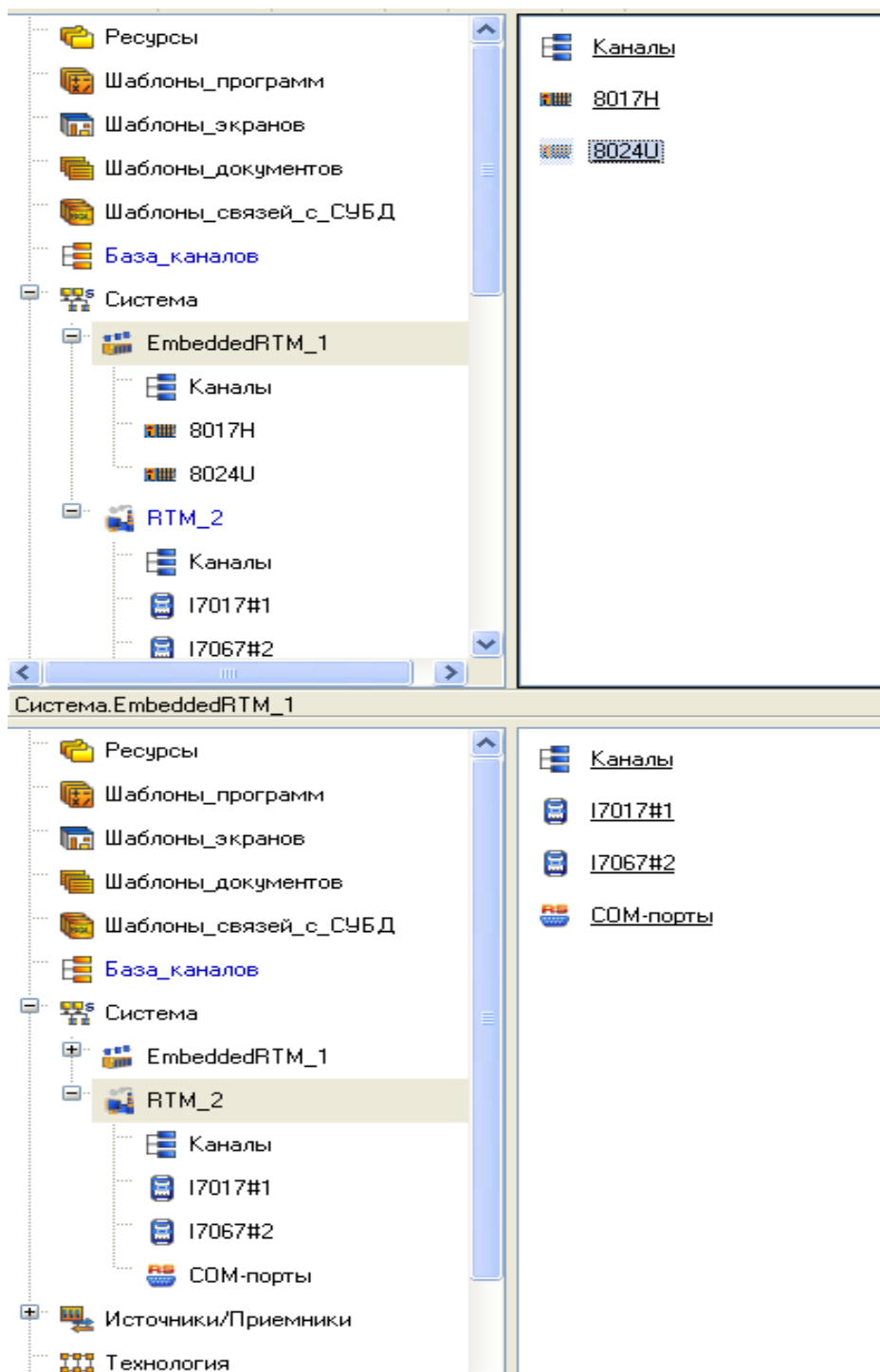
Тип дисплея: VGA

Тип клавиатуры: ...

Закроем бланк редактирования узла.

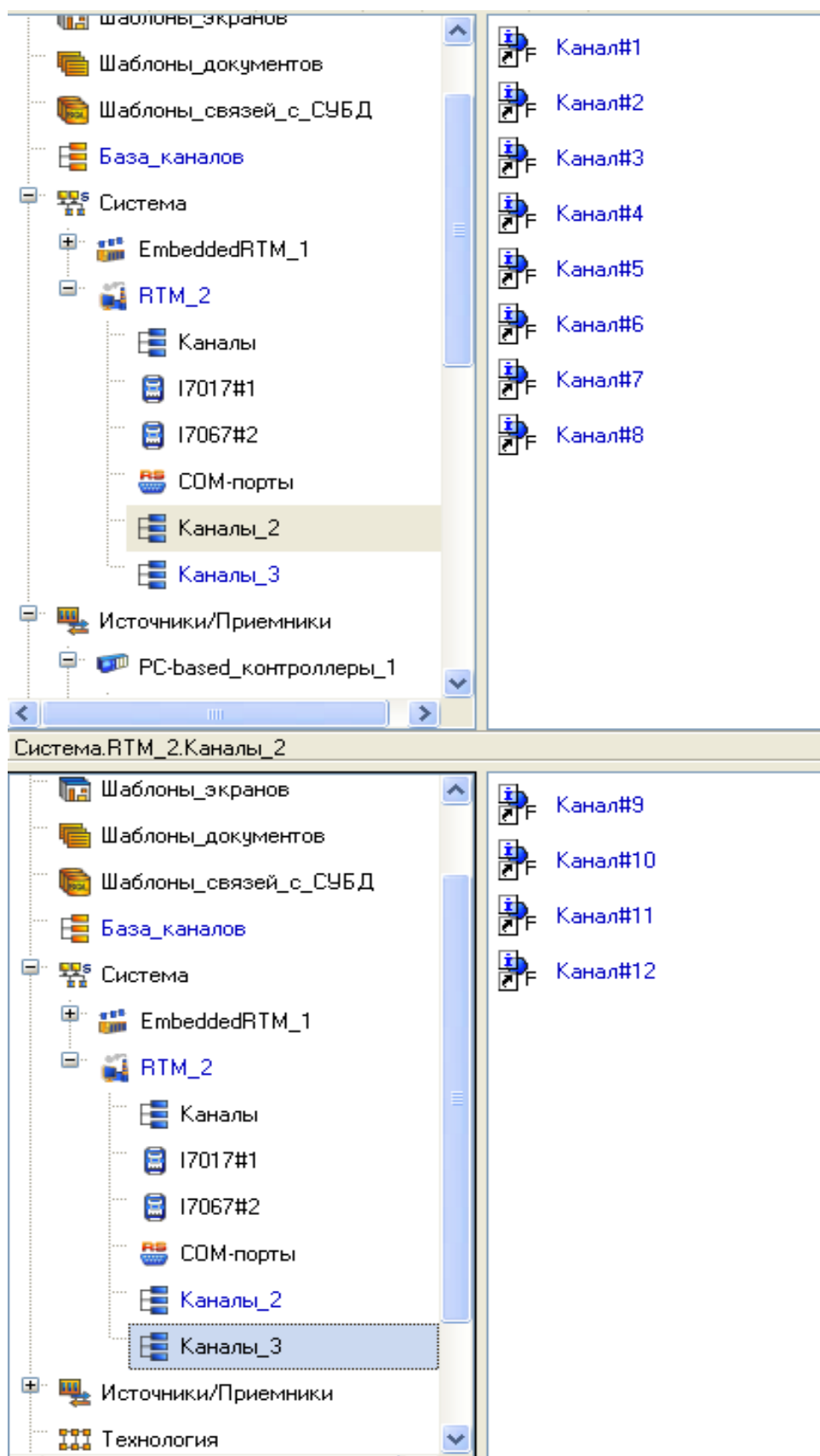
Конфигурирование межузлового обмена

Откроем дополнительную панель Навигатора проекта, спозиционировав верхнюю панель на узел контроллера, нижнюю – на узел АРМ следующим образом:

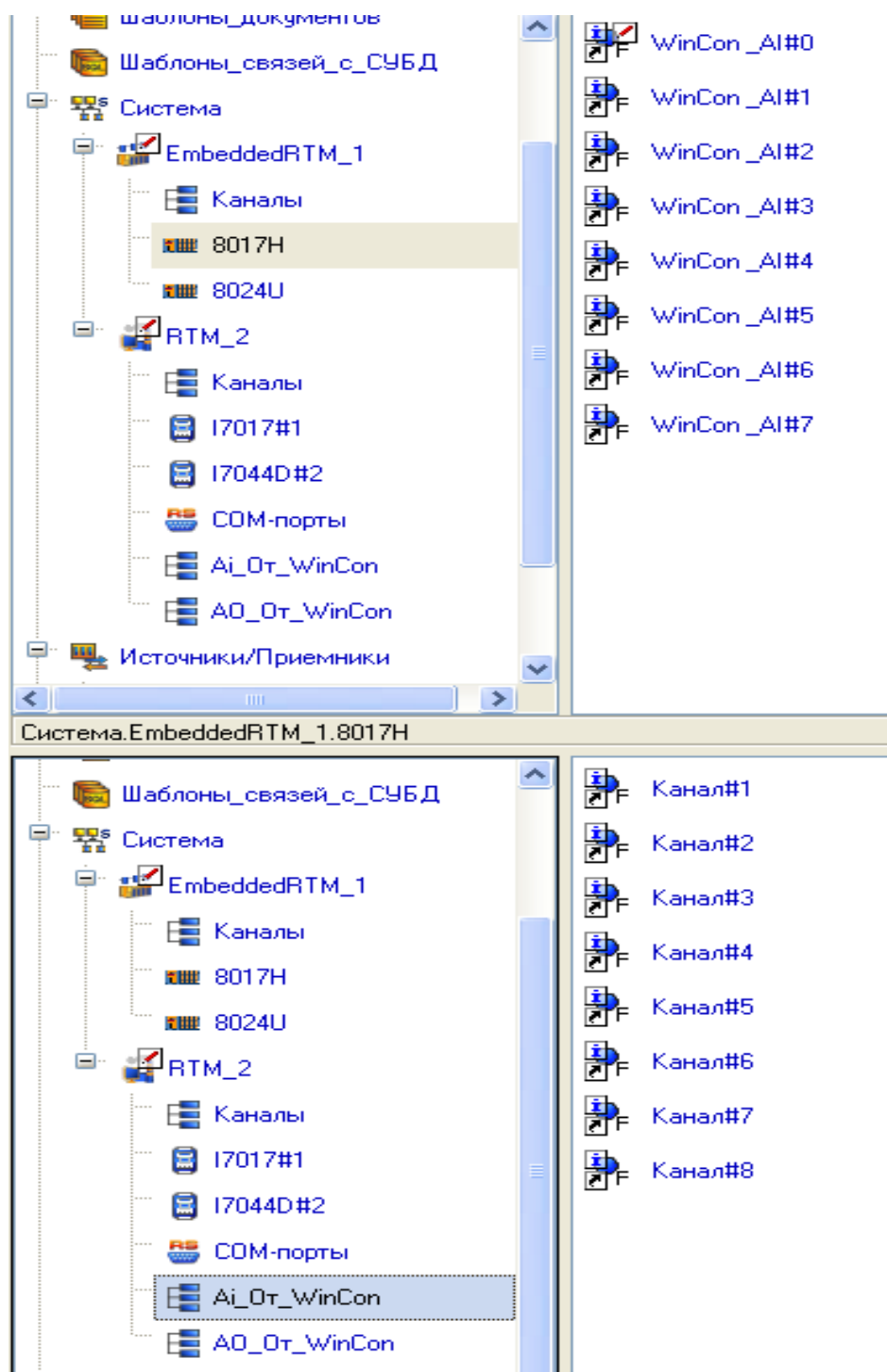


Захватив ЛК и удерживая ее на группе 8017H узла контроллера EmbeddedRTM_1, перетащим группу в узел АРМ. После отпущания ЛК в узле АРМ создастся группа

Каналы_2, содержащая каналы, связанные с каналами узла контроллера. То же самое сделаем с группой 8024U:



Изменим имя группы с Каналы_2 на Ai_От_WinCon, а группы Каналы_3 на АО_От_WinCon. В результате нами подготовлены два узла проекта для последующего создания системы контроля и управления:



Переименуем в узле RTM_2 в группе Ai_от_WinCon канал Канал#1 в Ai_from_WinCon1 и в группе AO_от_WinCon канал Канал#9 в AO_from_WinCon1.

Сохраним выполненный проект с помощью иконки , задав имя для записи **regul**:

Имя файла:	regul	Сохранить
Тип файла:	Project Files (*.prj)	Отмена

Лабораторная работа № 7.

Математическое обеспечение контроллера

1. Цель работы

Разработать математическое обеспечение контроллера в объеме одного контура регулирования с аналоговым клапаном управления.

2. Порядок выполнения работы

- 1). Изучите теоретическую часть. Выполните задание и продемонстрируйте его преподавателю
- 2). Оформите отчет по лабораторной работе.

3. Образец выполнения работы

Для разработки математического обеспечения контроллера в объеме одного контура регулирования с аналоговым клапаном управления будем использовать в качестве входного параметра первый вход модуля 8017Н, а в качестве управляющего сигнала – первый выход модуля I-8024 установленных в контроллер WinCon8000.

Для удобства управления и оценки качества регулирования задание и управление от регулятора мы будем формировать в относительных единицах – от 0 до 100% от шкалы преобразования АЦП и ЦАП. Для этого мы введем соответствующие нормирующие функциональные блоки.

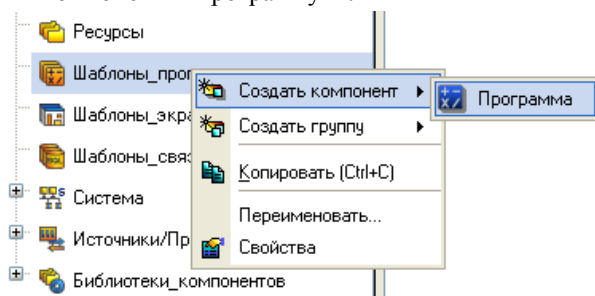
Чтобы исключить высокочастотные колебания в регуляторе, введем элемент ограничения чувствительности.

Для обеспечения возможности переключения режима управления автоматический/ручной введем блок безударного переключения управления исполнительным органом.

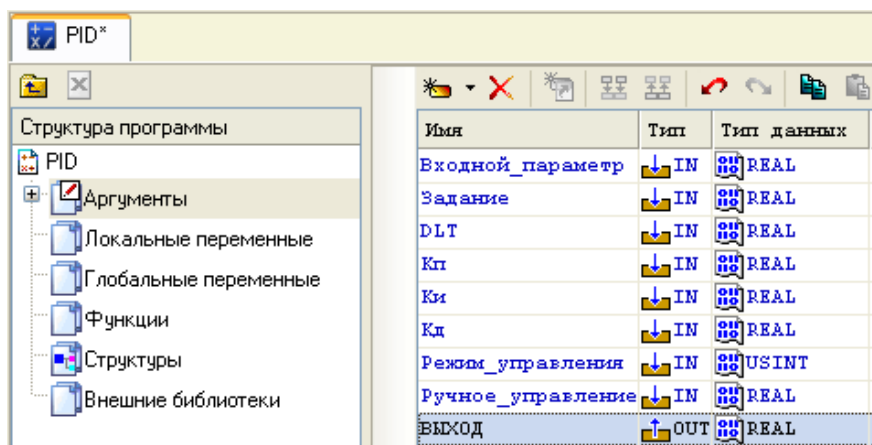
Исходным материалом для проектирования служит информационная база проекта, созданная на предыдущем занятии и сохраненная с именем **regul.prj**.

Разработка программы PID-регулятора

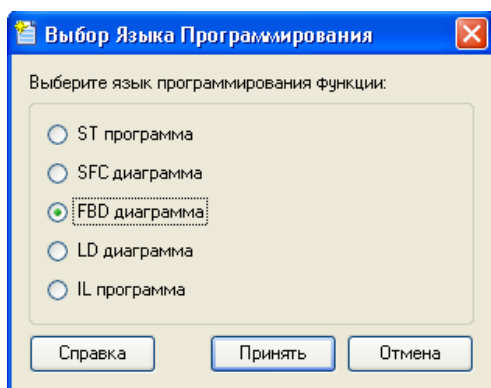
Разработку программы регулятора начнем с запуска интегрированной среды разработки TRACE MODE 6. Для создания программы выделим ЛК слой шаблонов программ и по щелчку ПК создадим новый компонент – Программу#1:



Изменим название программы на **PID**, и двойным щелчком ЛК приступим к редактированию. На первом этапе зададим в соответствующем диалоге аргументы разрабатываемой программы, уделяя внимание используемым типам данных:



Чтобы приступить непосредственно к созданию программы щелчком ЛК по названию программы в редакторе программ. В открывшемся диалоге выбора языка программирования выберем FBD:

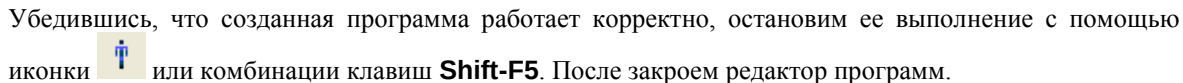


Далее, используя процедуру вызова блоков из соответствующих разделов (библиотек),

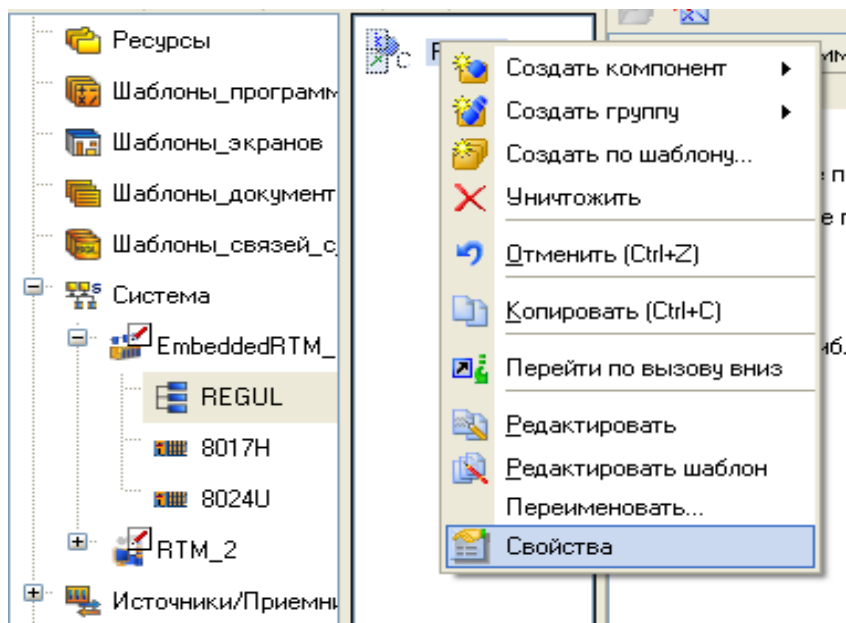
Раздел	Блок
Регулирование	Переключение с динамической балансировкой (SSWT), звено PID (pid)
Алгебраические функции	Перевод в проценты (t%), перевод из процентов (f%)
Управление	Зона нечувствительности (DZONE)
Арифметические функции	Вычитание (-)

размещая их на рабочем поле и связывая их входы и выходы составим следующую программу, реализующую ПИД-регулятор с возможностью переключения режима работы с автоматического на ручной и обратно:

Откомпилируем составленную программу нажатием функциональной клавиши **F7**.



Откроем дополнительную панель Навигатора проекта, на верхней панели оставим слой шаблонов программ, на нижней - группу Каналы узла контроллера. Изменим имя группы с Каналы на REGUL. С помощью мыши перетащим шаблон программы PID из слоя шаблонов в группу REGUL узла EmbeddedRTM_1. Выделим ЛК созданный канал класса Вызов PID и с помощью ПК откроем окно его свойств:



В открывшемся бланке перейдем к закладке Аргументы. Для аргументов Входной_параметр и ВЫХОД в столбце Флаги установим значения NP. При процедуре автопостроения каналов по аргументам шаблонов каналы для указанных аргументов созданы не будут, поскольку они уже были созданы на предыдущем занятии:

Система MicroRTM_1.REGULPID:3						
Информация Флаги Аргументы Атрибуты						
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги	
Входной_параметр	IN	REAL			NP	
Задание	IN	REAL				
DLT	IN	REAL				
Кп	IN	REAL				
Ки	IN	REAL				
Кд	IN	REAL				
Режим_управления	IN	USINT				
Ручное_управление	IN	REAL				
ВЫХОД	OUT	REAL			NP	

Для запуска процедуры автопостроения произведем щелчок ЛК по иконке , в результате будут созданы каналы в группе REGUL и их атрибуты будут связаны с аргументами программы PID:

Система

- EmbeddedRTM_1
 - 8017H
 - 8024U
 - REGUL**
- Источники/Приемники
- Библиотеки_компонентов

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Входной_параметр	IN	REAL			NP
Задание	IN	REAL		Задание:Реальное значение (Система.EmbeddedRTM_1.REGUL)	
DLT	IN	REAL		DLT:Реальное значение (Система.EmbeddedRTM_1.REGUL)	
Кп	IN	REAL		Кп:Реальное значение (Система.EmbeddedRTM_1.REGUL)	
Ки	IN	REAL		Ки:Реальное значение (Система.EmbeddedRTM_1.REGUL)	
Кд	IN	REAL		Кд:Реальное значение (Система.EmbeddedRTM_1.REGUL)	
Режим_управления	IN	USINT		Режим_управления:Реальное значение (Система.EmbeddedRTM_1.REGUL)	
Ручное_управление	IN	REAL		Ручное_управление:Реальное значение (Система.EmbeddedRTM_1.REGUL)	
ВЫХОД	OUT	REAL			NP

Свяжем аргументы Входной_параметр и ВЫХОД с атрибутами каналов вручную, используя ЛК в столбце Привязка данных аргументов:

Система

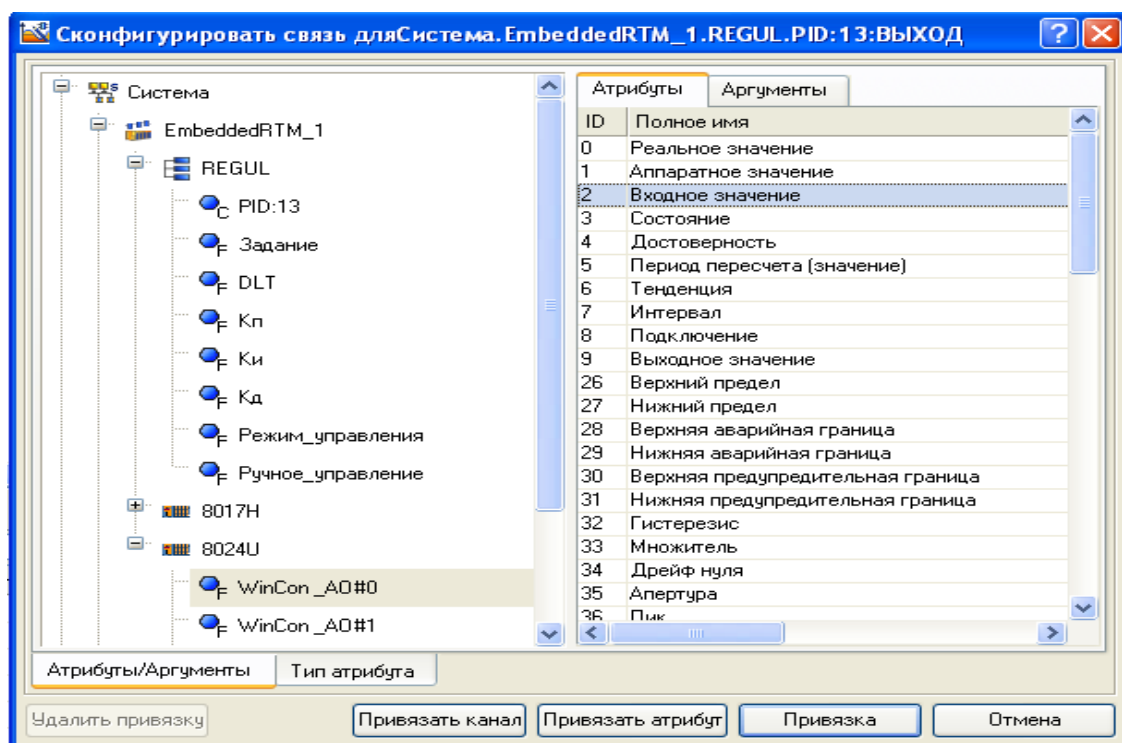
- EmbeddedRTM_1
 - REGUL
 - PID:13
 - Задание
 - DLT
 - Кп
 - Ки
 - Кд
 - Режим_управления
 - Ручное_управление
 - 8017H
 - WinCon_AI#0
 - WinCon_AI#1
 - WinCon_AI#2

ID	Полное имя
0	Реальное значение
1	Аппаратное значение
2	Входное значение
3	Состояние
4	Достоверность
5	Период пересчета (значение)
6	Тенденция
7	Интервал
8	Подключение
9	Выходное значение
26	Верхний предел
27	Нижний предел
28	Верхняя аварийная граница
29	Нижняя аварийная граница
30	Верхняя предупредительная граница
31	Нижняя предупредительная граница
32	Гистерезис
33	Множитель
34	Дрейф нуля
35	Апертура
36	Пик

Атрибуты/Аргументы Тип атрибута

Удалить привязку Привязать канал Привязать атрибут Привязка Отмена

и



Таким образом, аргументы канала класса Вызов PID после всех манипуляций будут следующими:

Система.EmbeddedRTM_1.REGUL.PID:13						
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги	Группа
Входной_параметр	IN	REAL		WinCon_AI#0:Реальное значение (Система.EmbeddedRTM_1.8017H)	NP	
Задание	IN	REAL		Задание:Реальное значение (Система.EmbeddedRTM_1.REGUL)		
DLT	IN	REAL		DLT:Реальное значение (Система.EmbeddedRTM_1.REGUL)		
Кп	IN	REAL		Кп:Реальное значение (Система.EmbeddedRTM_1.REGUL)		
Ки	IN	REAL		Ки:Реальное значение (Система.EmbeddedRTM_1.REGUL)		
Кд	IN	REAL		Кд:Реальное значение (Система.EmbeddedRTM_1.REGUL)		
Режим_управления	IN	USINT		Режим_управления:Реальное значение (Система.EmbeddedRTM_1.REGUL)		
Ручное_управление	IN	REAL		Ручное_управление:Реальное значение (Система.EmbeddedRTM_1.REGUL)		
ВЫХОД	OUT	REAL		WinCon_AO#0:Входное значение (Система.EmbeddedRTM_1.8024U)	NP	

- Для того, чтобы при привязке атрибутов каналов к аргументам шаблона не происходило переименование аргументов, необходимо в разделе **Редактор аргументов Настроек ИС** снять флажок с пункта **Переименовывать аргументы при привязке**.

Задание начальных значений каналов

С помощью процедуры группового редактирования группы компонентов REGUL узла контроллера зададим начальные значения:

REGUL		
Содержимое группы		
Базовое имя	Входное значение	Отработать
PID:13	0	1
Задание	25	1
DLT	0.1	1
Кп	1	1
Ки	0.1	1
Кд	0.001	1
Режим_управления	0	1
Ручное_управление	25	1

На этом закончим работу с узлом контроллера EmbeddedRTM_1 и сосредоточимся на формировании базы каналов узла APM RTM_2.

Формирование базы каналов APM

С помощью дополнительно открытой панели Навигатора проекта перетащим из узла контроллера в узел APM группу компонентов REGUL. Переименуем вновь созданную в узле APM группу в REGUL_from_WinCon, кроме того, удалим из этой группы компонент – канал класса Вызов. Созданным канал зададим соответствующее имена Результирующий состав группы каналов, получающих соответствующие данные от одноименных каналов узла контроллера.

Задание
 DLT
 Кп
 Ки
 Кд
 Режим_управления
 Ручное_управление

Кд > _____
 Ки > _____
 100 > _____
 0 > _____
 0 > _____
 10 > _____
 Режим_управления > _____
 Ручное_управление > _____

ID: 74
 Кодировка: TC5
 Узел: Система.RTM_2
 Link: Ручное_управление:Реальное значение (Система.EmbeddedRTM_1.REGUL)

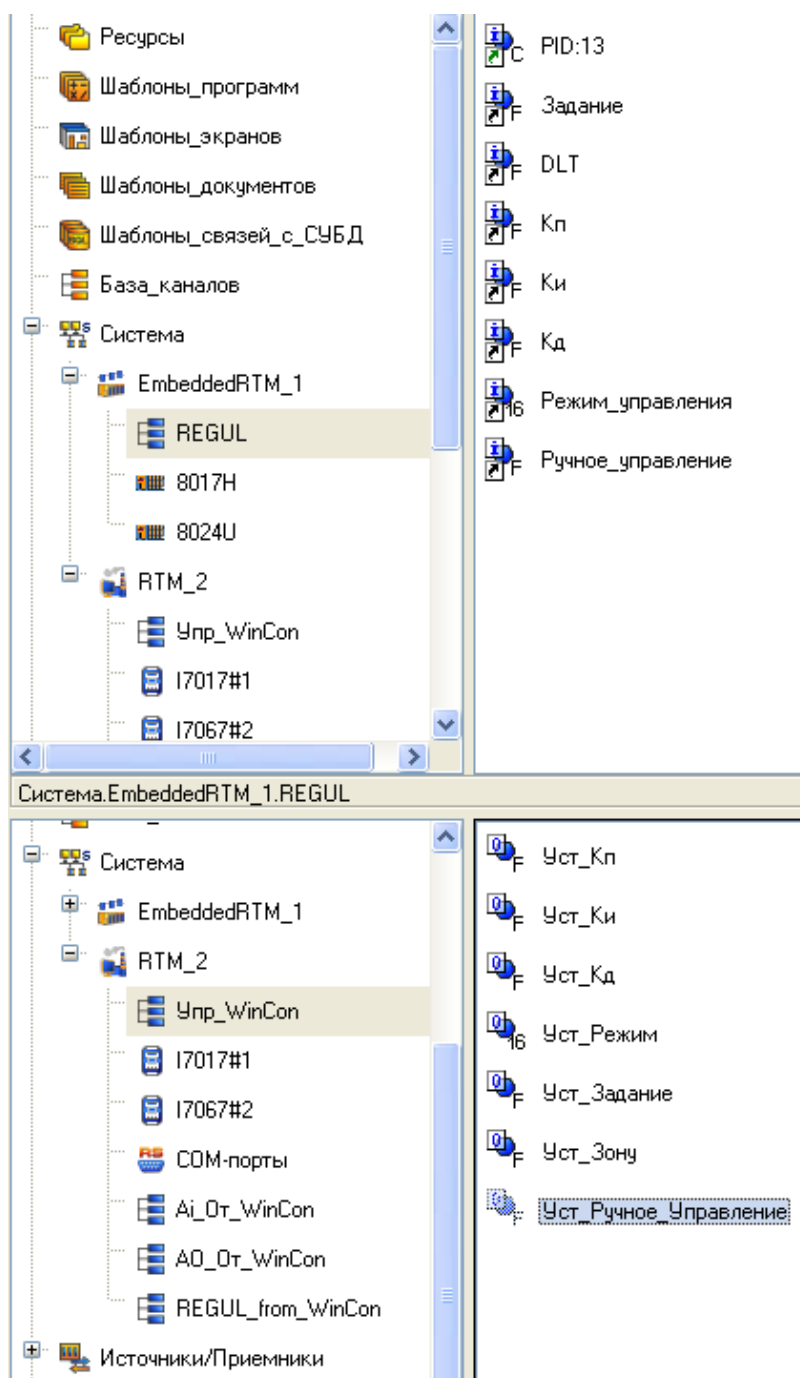
Переименуем группу Каналы узла APM в Упр_WinCon . Для того чтобы иметь возможность формировать значения параметров регулятора в контроллере от APM, создадим в группе Упр_WinCon каналы типа Output:

- Уст_Кп
- Уст_Ки
- Уст_Кд
- Уст_Режим
- Уст_Задание
- Уст_Зону
- Уст_Ручное_управление

Вызовем группу на редактирование и зададим каналам начальные значения:

REGUL		Упр_WinCon	
Содержимое группы			
   			
Базовое имя	Входное значение	Отработать	
Уст_Кп	1	1	
Уст_Ки	0.1	1	
Уст_Кд	0.001	1	
Уст_Режим	0	1	
Уст_Задание	25	1	
Уст_Зону	0.1	1	
Уст_Ручное_управление	25	1	

С помощью дополнительной панели Навигатора проекта установим связи между каналами узлов, последовательно накладывая с помощью ЛК мыши источник на приемник данных:



Лабораторная работа № 8.

ПО Программа для дискретного управления

1. Цель работы

Разработать программу дискретного управления.

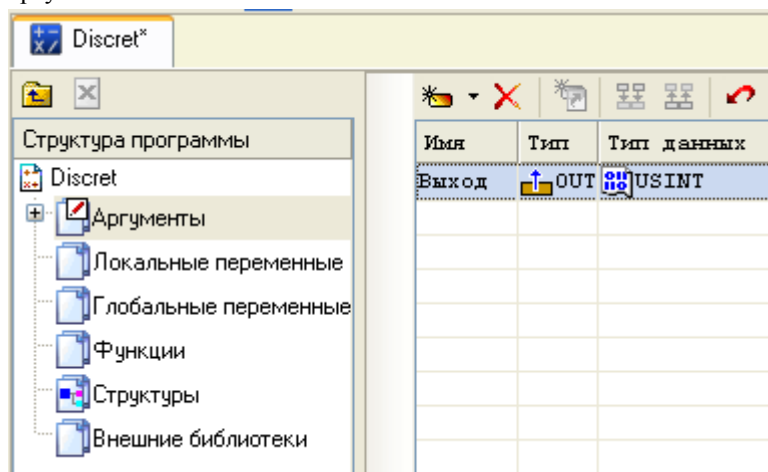
2. Порядок выполнения работы

- 1). Изучите теоретическую часть. Выполните задание и продемонстрируйте его преподавателю
- 2). Оформите отчет по лабораторной работе.

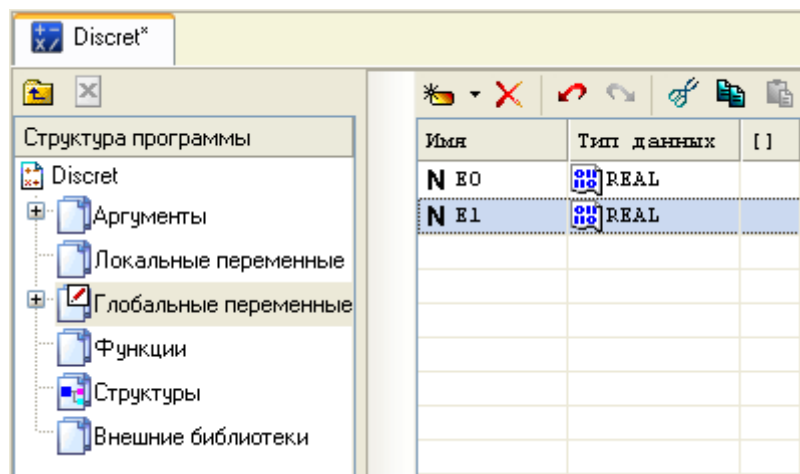
3. Образец выполнения работы

Для автоматического формирования значения дискретного выхода модуля удаленного ввода/вывода I-7067 создадим специальную программу, последовательно устанавливающую в состояние «1» выходные разряды, а затем также последовательно сбрасывающую их в состояние «0». В слое

шаблонов программ создадим компонент Программа#2, изменим ее имя на Discret и зададим аргументы:



Для реализации программы нам потребуется две статических переменных, значения которых будут сохраняться в промежутках между вызовами программы – E0 и E1.



В качестве языка программирования выберем Техно-ST. Введем текст программы в следующем виде:

```

PROGRAM
  VAR_OUTPUT Выход : USINT; END_VAR

  IF E1 == 0 THEN
    E0=E0 * 2 +1;
  ELSE
    E0=E0 / 2;
    FUN(E0,E0);
  END_IF;

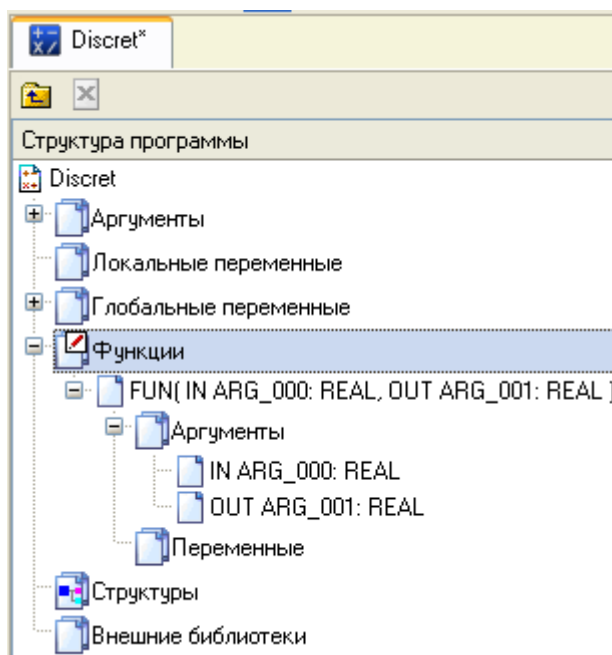
  IF E0 > 255 THEN
    E1 = 1;
    E0 = 255;
  ELSE
    IF E0 == 0 THEN
      E1 = 0;
    END_IF; END_IF;

    Выход = E0;
  END_PROGRAM

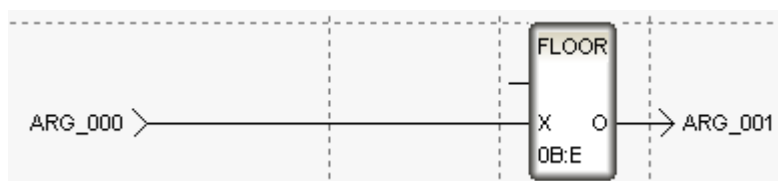
```

Примененная в тексте программы функция FUN() предназначена для выделения целой части числа

имеет один входной и один выходной аргумент типа REAL и оформляется как:



Непосредственно программа, реализующая функции разработана на языке Техно-FBD и содержит один функциональный блок из библиотеки арифметических операций:



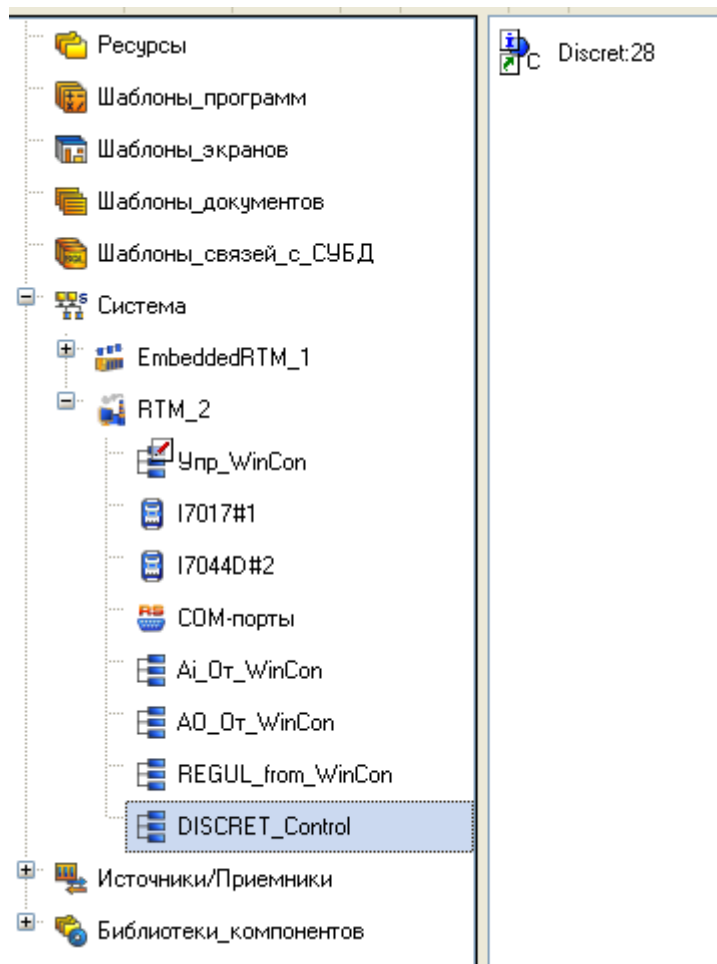
Не забудем откомпилировать по клавише F7 функцию и саму программу Discret. Откроем окно переменных и запустив программу посмотрим ход ее выполнения:

Переменные	Имя	Тип данных	Десятичное	Шестнадцатеричное	Двоичное
	Выход	USINT	127	16#7F	2#01111111
Аргументы Локальные Глобальные Текущая структура Просмотр					

Убедившись в правильности работы программы, остановим ее выполнение и закроем редактор шаблонов программ.

Привязка аргументов программы к атрибутам каналов

В узле APM создадим новую группу, переименуем ее в DISCRET_CONTROL и перетащим в нее шаблон программы Discret.



Откроем свойства созданного канала класса вызовов Discret и во вкладке Аргументы вручную настроим связь единственного аргумента с атрибутом канала:

Информация						
Флаги						
Аргументы						
Атрибуты						
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги	
Выход	OUT	USINT		DO#1:Реальное значение (Система.RTM_2.I7067#2)		

Редактирование базы каналов APM

Подготовка групп компонентов к просмотру в профайлере

Для удобства наблюдения атрибутов каналов узлов при работе в профайлерах все созданные группы компонентов в узлах EmbeddedRTM_1 и RTM_2 пометим как Загружаемые, отредактировав последовательно их свойства во вкладке Флаги, например, для группы REGUL узла контроллера:

Система EmbeddedRTM_1.8017H

Информация | **Флаги**

Флаги

☐ Выключить
 ☐ Отладка
 ☐ Регистратор
 ☐ Синхронизировать

☐ Отключить от источника
 ☐ Отчет тревог
 ☐ Атрибуты
 ☐ Дамп
 ☐ Запрос времени значения

☒ Синхронизировать флаги

Архив

Номер SIAD :

Автопосылка

☐ Разрешить

Загрузить группу

☒ Загрузить

Настройка границ

Для приведения данных, получаемых от контроллера к относительным величинам, отредактируем каналы группы REGUL_from_WinCon, Ai_от_WinCon и AO_от_WinCon, настроив в основном бланке разделы Граница и Обработка:

Имя Кодировка

Комментарий

Границы

☒ Использовать

ВП
 ВА
 ВГ
 НГ
 НА
 НП
 Гистерезис
☒ Контроль границ

Обработка

☒ Использовать

Апертура
 Пик
 Сглаж.
 Множитель
 Смещение
☒ Масштабирование

Масштабирование

In	Множитель	Смещение	A
Max 10	× 10	+ 0	Max 100
Min 0			Min 0

Системные

Основные

Тип
 Размерность
 Период Единица измерения
Автопосылка
☐ Включить
 Индекс
☐ Отработать
 На старте

Архивация

Дополнительно

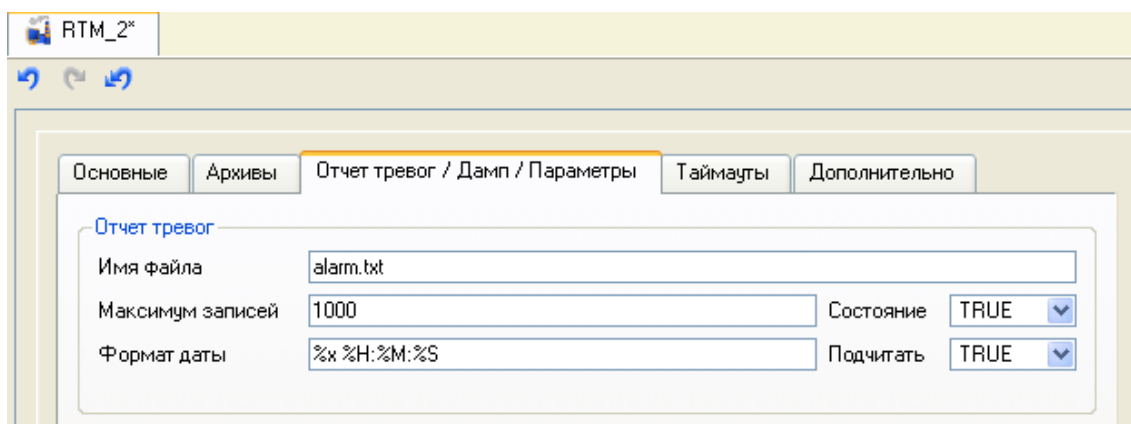
и

Для каналов, находящихся в группе компонентов I7017#1, зададим групповым образом границы, соответствующие входному диапазону, установленного для модуля – (0 ÷ 10)В.

Настройка архива и отчета тревог

Определим для узла АРМ файлы архива и отчета тревог. Выделим ЛК в слое Система узел RTM_2 и ПК вызовем его на редактирование. Перейдя во вкладку Архивы зададим параметры для архивации данных в архив СПАД 1:

Параметры отчета тревог зададим в соответствующей вкладке:

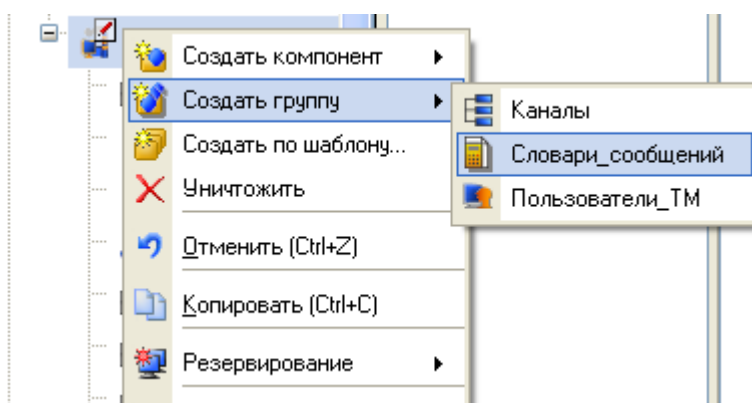


Выберем для архивирования каналы узла АРМ, получающие данные от аппаратуры ввода/вывода, т.е. размещенные в группах REGUL_from_WinCon, Ai_от_WinCon, AO_от_WinCon, I7017#1 и I7067#2. Для этого воспользуемся процедурой группового редактирования, указав для атрибута СПАД уровень архивации (для нас СПАД 1):

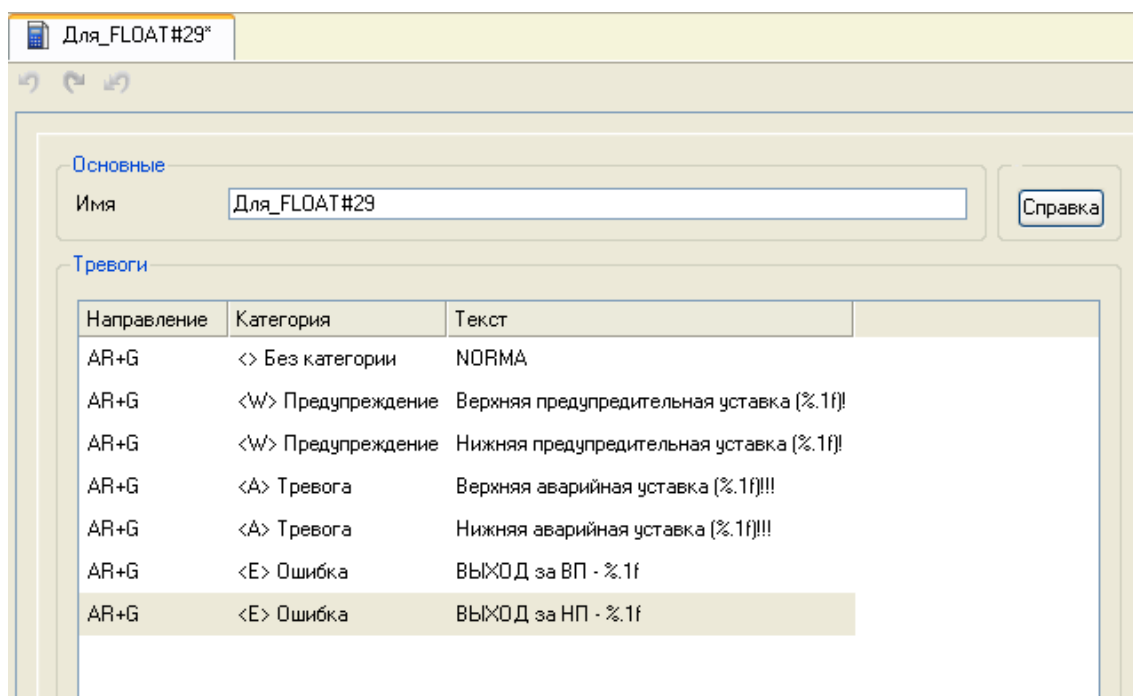
Содержимое группы			
Базовое имя	СПАД	Контроль границ	И
Канал#9	1	1	1
Канал#10	1	1	1
Канал#11	1	1	1
Канал#12	1	1	1

Подключение и настройка словарей сообщений

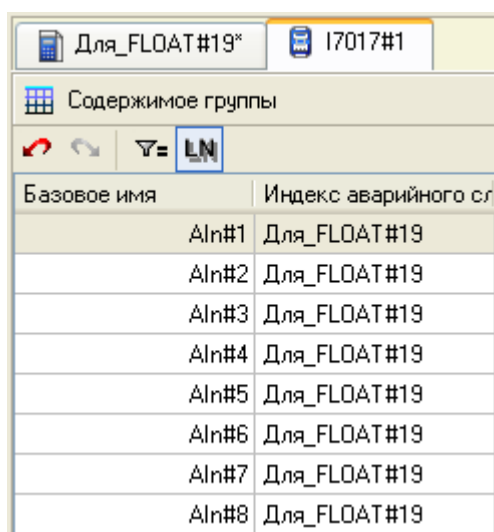
В отчет тревог будем помещать сообщения по каналам группы I7017#1, поскольку к входам модуля I-7017 будем подключать датчик сигнала и, таким образом, изменять входной сигнал в широком диапазоне. Воспользуемся стандартным словарем сообщений в отчет тревог для канала класса Float. Для реализации этого создадим для узла АРМ новую группу – Словари_сообщений:



В этой группе создадим компонент – словарь Для_FLOAT#19. Откроем его на редактирование, дополнив строки вывода в отчет тревог численными значениями, и присвоим классы для формируемых сообщений:



Подключение отредактированного словаря сообщений к каналам произведем с помощью процедуры группового редактирования атрибутов каналов группы I7017#1, указав для выделенных ЛК каналов в столбце Индекс аварийного словаря при нажатой клавиши Ctrl и двойном щелчке ЛК словарь Для_FLOAT#19:

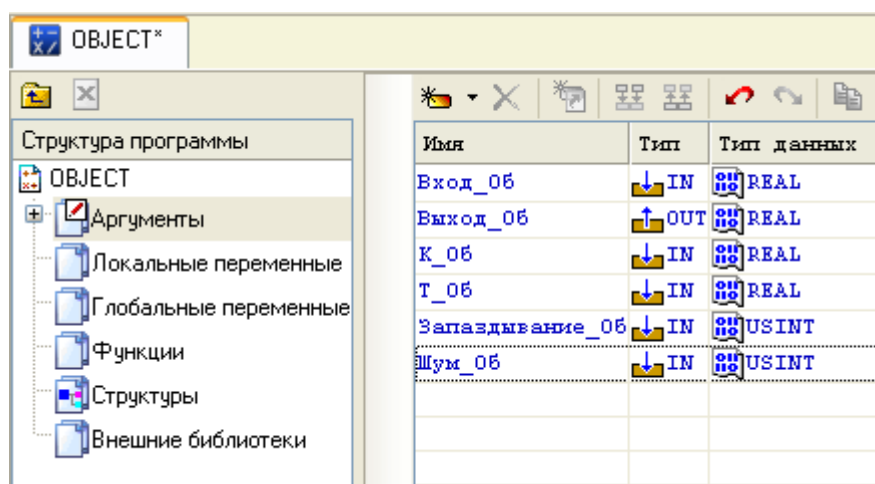


Сохраним выполненный проект с помощью иконки .

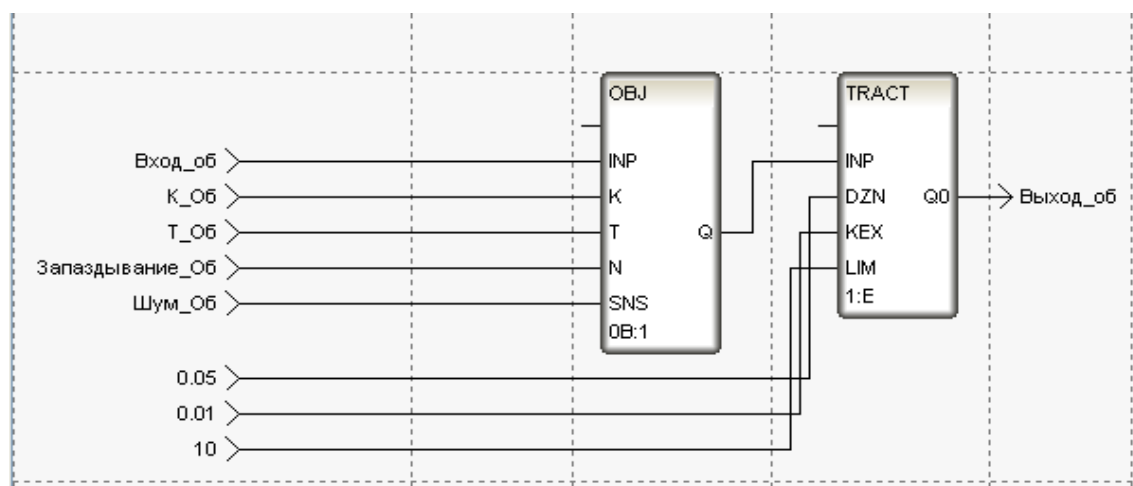
5. **ВНИМАНИЕ!** Дальнейшие пункты выполняются в случае замены контроллера его имитатором – обычным ПК.

Разработка программы имитатора объекта управления

Создадим и разместим в узле контроллера MicroRTM_1 имитатор объекта управления. В слое шаблонов программ создадим новый компонент – Программа#3, переименуем ее в ОБЪЕКТ. В качестве аргументов зададим:



В качестве математической модели объекта воспользуемся обычным аperiodическим звеном первого порядка с запаздыванием. Разработку программы будем вести на языке Техно-FBD. Вид программы следующий:



Откомпилируем программу с помощью клавиши F7 и проверим ее в режиме эмуляции, задавая в окне переменных различные значения, как входного аргумента, так и внутренних переменных. Убедившись в правильности работы программы, остановим эмуляцию и закроем редактор программ.

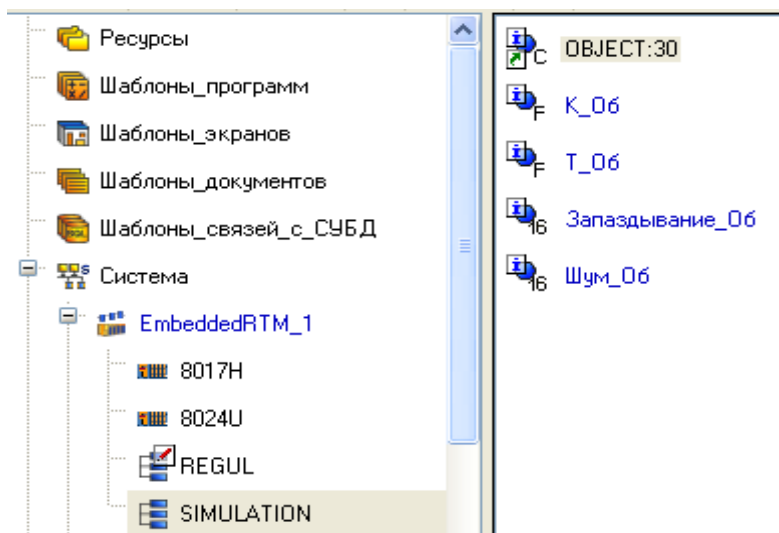
Привязка аргументов программы к атрибутам каналов в контроллере

В узле контроллера EmbeddedRTM_1 создадим новую группу каналов SIMULATOR. С помощью мыши перетащим созданный шаблон программы OBJECT в группу SIMULATOR узла контроллера EmbeddedRTM_1. Откроем свойства канала класса Вызов OBJECT и во вкладке Аргументы выставим флаг NP аргументам Вход_Об и Выход_Об, выполним операцию автопостроения и

автопривязки к атрибутам каналов для оставшихся аргументов с помощью иконки . Уберем привязки каналов WinCon_AI#0 и WinCon_AO#0 к компонентам источникам/приемникам, для этого последовательно откроем свойства каналов и в поле Привязка по двойному щелчку ЛК в открывшемся окне диалога выберем Удалить привязку. Тем самым, связь данных каналов с аппаратурой (платой ввода/вывода) будет снята. Далее, аргументы Вход_Об и Выход_Об привяжем вручную по двойному щелчку ЛК в поле Привязка. Результат будет:

Информация					
Аргументы					
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Вход_об	IN	REAL		WinCon_A0#0:Реальное значение (Система.EmbeddedRTM_1.8024U)	NP
Выход_об	OUT	REAL		WinCon_A1#0:Входное значение (Система.EmbeddedRTM_1.8017H)	NP
K_Об	IN	REAL		F_K_Об:Реальное значение (Система.EmbeddedRTM_1.SIMULATION)	
T_Об	IN	REAL		F_T_Об:Реальное значение (Система.EmbeddedRTM_1.SIMULATION)	
Запаздывание_Об	IN	USINT		F_Запаздывание_Об:Реальное значение (Система.EmbeddedRTM_1.SIMULATION)	
Шум_Об	IN	USINT		F_Шум_Об:Реальное значение (Система.EmbeddedRTM_1.SIMULATION)	

Содержимое группы SIMULATOR в результате будет:

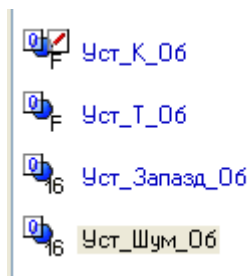


Групповым образом зададим каналам начальные значения и взведем флаг Отработать.

Базовое имя	Входное значение	Отработать
OBJECT:30	0	0
K_Об	2.5	1
T_Об	10	1
Запаздывание_Об	1	1
Шум_Об	5	1

Подготовка базы каналов APM для формирования параметров имитатора объекта

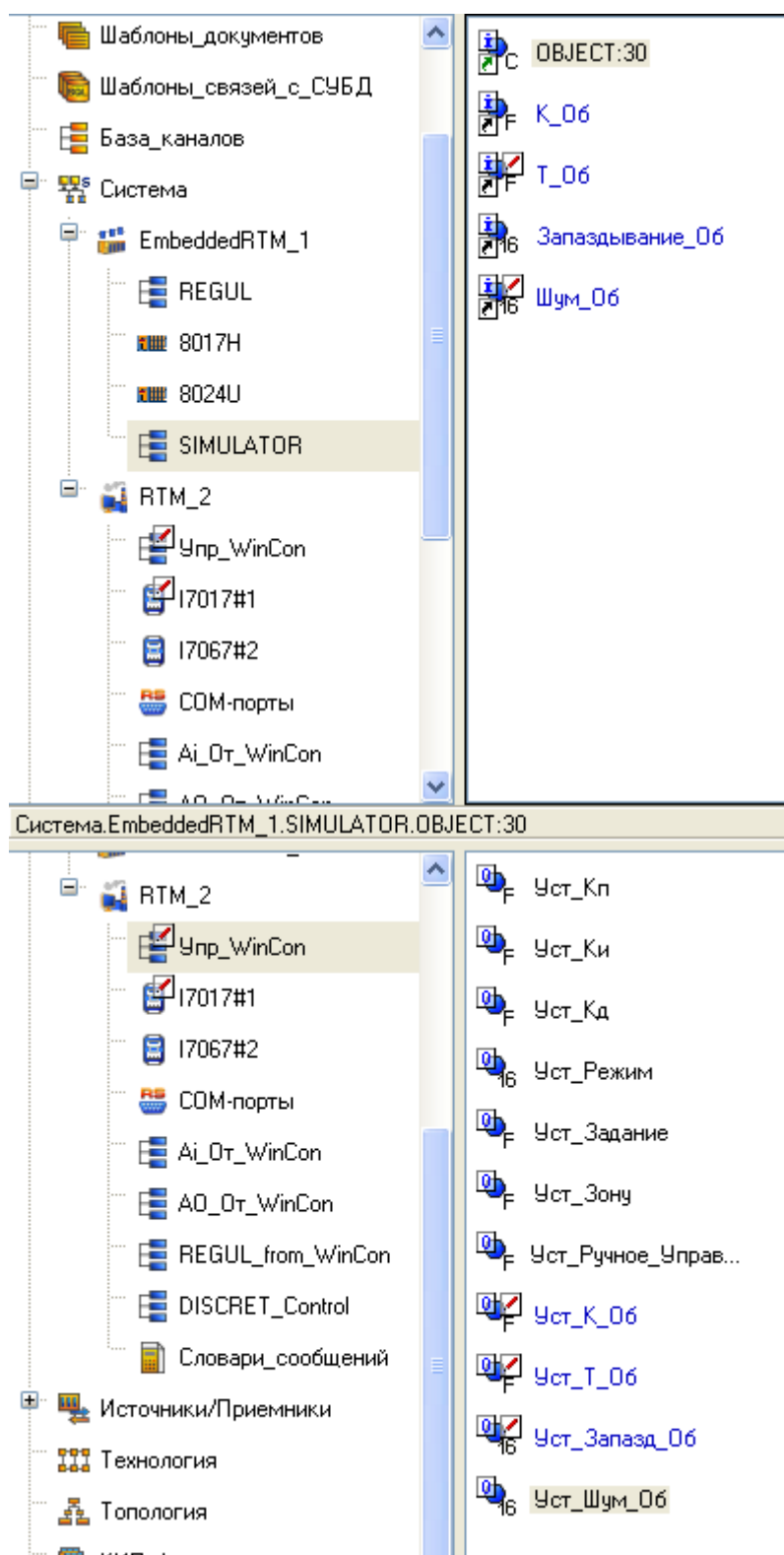
Теперь для того чтобы задавать параметры объекта с узла APM, дополним группу Упр_WinCon каналами типа Output:



Групповым образом зададим созданным каналам начальные значения и взведем для них флаг Отработать.

REGUL	Упр_WinCon*	Для_FLOAT#29	I7017#1	OBJECT*
Содержимое группы				
Базовое имя	Входное значение	Отработать	Тип	
Уст_Кп	1	1	Output	
Уст_Ки	0.1	1	Output	
Уст_Кд	0.001	1	Output	
Уст_Режим	0	1	Output	
Уст_Задание	25	1	Output	
Уст_Зону	0.1	1	Output	
Уст_Ручное_управление	25	1	Output	
Уст_К_Об	2.5	1	Output	
Уст_Т_Об	10	1	Output	
Уст_Запад_Об	1	1	Output	
Уст_Шум_Об	5	1	Output	

Откроем дополнительную панель Навигатора проекта и определим связь между соответствующими каналами группы Упр_WinCon узла RTM_2 (источники) и каналами группы SIMULATOR узла EmbeddedRTM_1 (приемники):



Сохраним выполненный проект с помощью иконки .



Лабораторная работа № 9.

Создание графического пользовательского интерфейса АРМ

1. Цель работы

Спроектировать пользовательский интерфейс узла АРМ в объеме двух графических экранов, создать графический объект и ресурсные библиотеки.

2. Порядок выполнения работы

- 1). Изучите теоретическую часть. Выполните задание и продемонстрируйте его преподавателю
- 2). Оформите отчет по лабораторной работе.

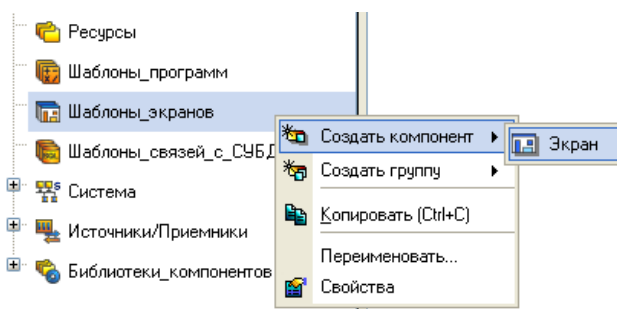
3. Образец выполнения работы

Для проектирования пользовательского интерфейса узла АРМ в объеме двух графических экранов будем использовать ГЭ Текст, Тренд, Кнопка и Ползунок. Создадим один графический объект (ГО) для отображения дискретного выхода модуля удаленного ввода/вывода I-7067 и спроектируем ресурсные библиотеки, содержащие файлы формата bmp и видеоклип.

Исходным материалом для проектирования служат информационная и математическая базы проекта, созданные на предыдущих занятиях, сохраненных с именем **regul.prj**.

Создание графических экранов

Разработку экранов начнем с запуска интегрированной среды разработки TRACE MODE 6. Для создания экрана выделим ЛК слой шаблонов экранов и по щелчку ПК создадим новый компонент – Экран#1:



Щелчком ПК на вновь созданном компоненте вызовем его свойства и, перейдя к вкладке Аргументы, приступим к редактированию. На данном экране будем отображать в графическом виде с помощью ГЭ Тренд величины параметров Уст_Задание, AI_От_WinCon и AO_От_WinCon ; с использованием ГЭ Кнопка вводить параметры ПИД-регулятора и объекта управления, с помощью ГЭ Текст индцировать вводимые параметры; с помощью ГЭ Ползунок формировать величины Задания регулятору и ручного управления; с помощью ГЭ Кнопка выполним переключатель режима управления Автоматический/ручной, а ГЭ Текст используем для индикации текущего режима.

Создание аргументов

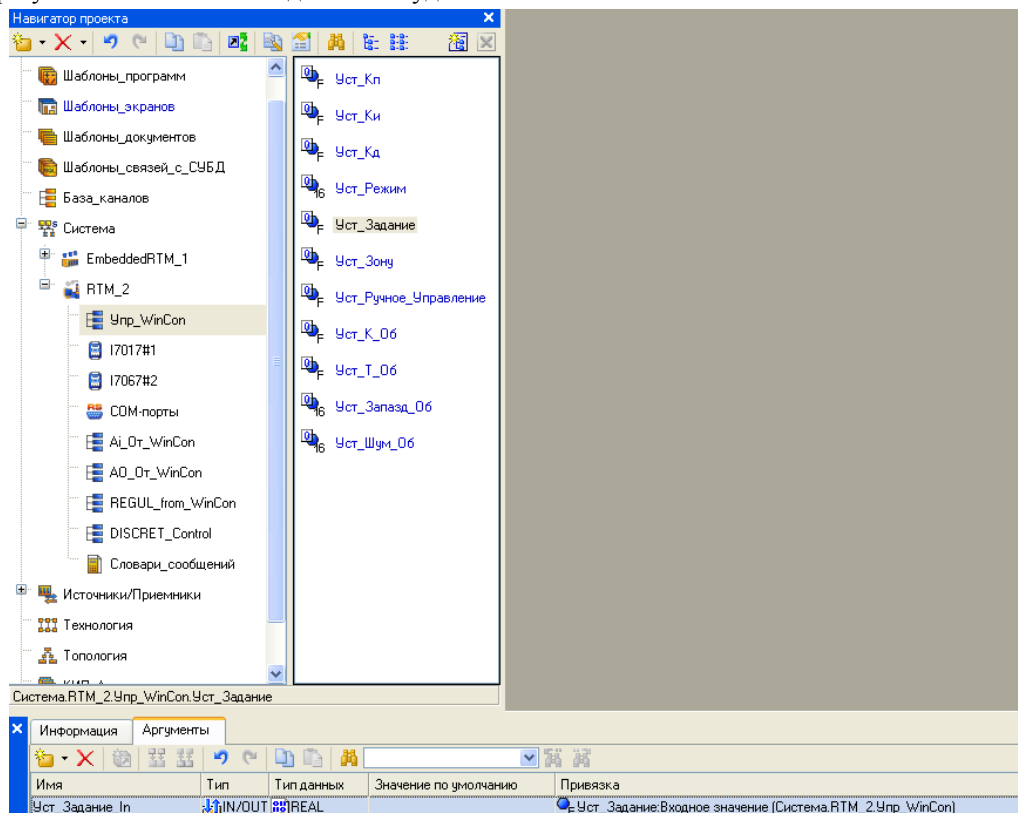
Сформируем аргументы шаблона экрана с автопривязкой к атрибутам каналов. Для этого откроем свойства шаблона и перейдем во вкладку Аргументы. Будем исходить из того, что тип аргументов для отображения с использованием ГЭ определим как IN, а для аргументов, предполагающих их формирование посредством ГЭ – IN/OUT. Для автоматического создания имен аргументов по именам соответствующих каналов во вкладке Редактор аргументов настроек ИС установим флажок в пункте Переименовывать аргументы при привязке и нажмем клавишу Готово.

В окне табличного редактора аргументов с помощью иконки  создадим новый аргумент ARG_000 и изменим его тип на IN/OUT. Для создания аргумента экрана, связанного с каналом Уст_Задание в группе компонентов Упр_WinCon с целью его задания с помощью ГЭ Кнопка и отображения с помощью ГЭ Текст на графическом экране, выполним следующие действия:

- выделим ЛК строку ARG_000;

- откроем группу компонентов Упр_WinCon узла RTM_2;
- захватим ЛК канал Уст_Задание и, удерживая ЛК нажатой, перетащим его на строку, содержащую ARG_000.

В результате выполненных действий будем иметь:



Для отображения на графическом экране реальных значений каналов AI_from_WinCon1 и AO_from_WinCon1 без предварительных операций перетащим с помощью мыши данные каналы в окно табличного редактора аргументов.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги	Группа	Ед.измерения
Уст_Задание_In	IN/OUT	REAL		Уст_Задание:Входное значение (Система.RTM_2.Упр_WinCon)			...
Ai_from_WinCon1_R	IN	REAL		Ai_from_WinCon1:Реальное значение (Система.RTM_2.Ai_Ot_WinCon)			...
AO_from_WinCon1_R	IN	REAL		AO_from_WinCon1:Реальное значение (Система.RTM_2.AO_Ot_WinCon)			...

Создадим с помощью иконки  еще один аргумент, изменим его тип на IN/OUT и создадим по шаблону еще 9 таких аргументов:

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Уст_Задание_In	IN/OUT	REAL		Уст_Задание:Входное значение (Система.RTM_2.Упр_WinCon)	
Ai_from_WinCon1_R	IN	REAL		Ai_from_WinCon1:Реальное значение (Система.RTM_2.Ai_Ot_WinCon)	
AO_from_WinCon1_R	IN	REAL		AO_from_WinCon1:Реальное значение (Система.RTM_2.AO_Ot_WinCon)	
ARG_000	IN/OUT	REAL			

Новая группа по шаблону

Размер группы: 9

Готово

Отмена

Просмотр >>>

Последовательно перетащим на созданные аргументы каналы Уст_Кп, Уст_Ки, Уст_Кд, Уст_Зону, Уст_Ручное_управление, Уст_Режим, Уст_К_Об, Уст_Т_Об, Уст_Запасд_Об и Уст_Шум_Об*.

6. **ВНИМАНИЕ!** Выделенные курсивом аргументы создаются при имитации контроллера обычным ПК.

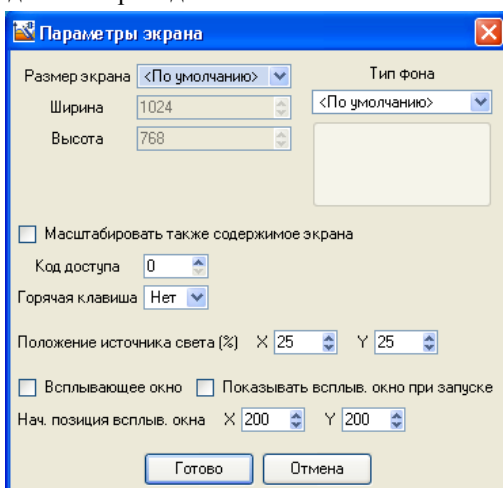
Информация		Аргументы				
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги	Группа
Уст_Задание_In	IN/OUT	REAL		Уст_Задание:Входное значение (Система.RTM_2_Упр_WinCon)		
Ai_from_WinCon1_R	IN	REAL		Аi_from_WinCon1:Реальное значение (Система.RTM_2_Аi_Ot_WinCon)		
AO_from_WinCon1_R	IN	REAL		АO_from_WinCon1:Реальное значение (Система.RTM_2_АO_Ot_WinCon)		
Уст_Kn_In	IN/OUT	REAL		Уст_Kn:Входное значение (Система.RTM_2_Упр_WinCon)		
Уст_Ki_In	IN/OUT	REAL		Уст_Ki:Входное значение (Система.RTM_2_Упр_WinCon)		
Уст_Kd_In	IN/OUT	REAL		Уст_Kd:Входное значение (Система.RTM_2_Упр_WinCon)		
Уст_Режим_In	IN/OUT	REAL		Уст_Режим:Входное значение (Система.RTM_2_Упр_WinCon)		
Уст_Зону_In	IN/OUT	REAL		Уст_Зону:Входное значение (Система.RTM_2_Упр_WinCon)		
Уст_Ручное_управление_In	IN/OUT	REAL		Уст_Ручное_управление:Входное значение (Система.RTM_2_Упр_WinCon)		
Уст_K_06_In	IN/OUT	REAL		Уст_K_06:Входное значение (Система.RTM_2_Упр_WinCon)		
Уст_T_06_In	IN/OUT	REAL		Уст_T_06:Входное значение (Система.RTM_2_Упр_WinCon)		
Уст_Запад_06_In	IN/OUT	REAL		Уст_Запад_06:Входное значение (Система.RTM_2_Упр_WinCon)		
Уст_Шум_06_In	IN/OUT	REAL		Уст_Шум_06:Входное значение (Система.RTM_2_Упр_WinCon)		

Закроем окно свойств шаблона Экран#1.

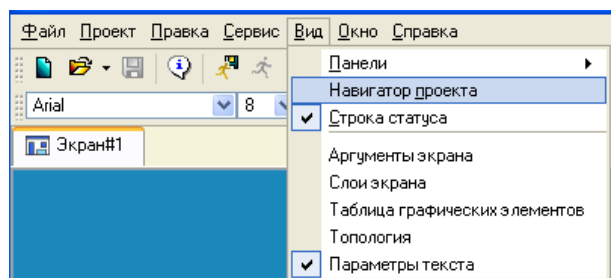
Чтобы приступить непосредственно к созданию графического экрана совершим двойной щелчок ЛК по шаблону Экран#1 и тем самым запустим редактор графики.

Настройка свойств графического экрана

Свойства текущего графического экрана задаются в пункте меню **Сервис/Параметры экрана**, вид диалога приведен ниже:



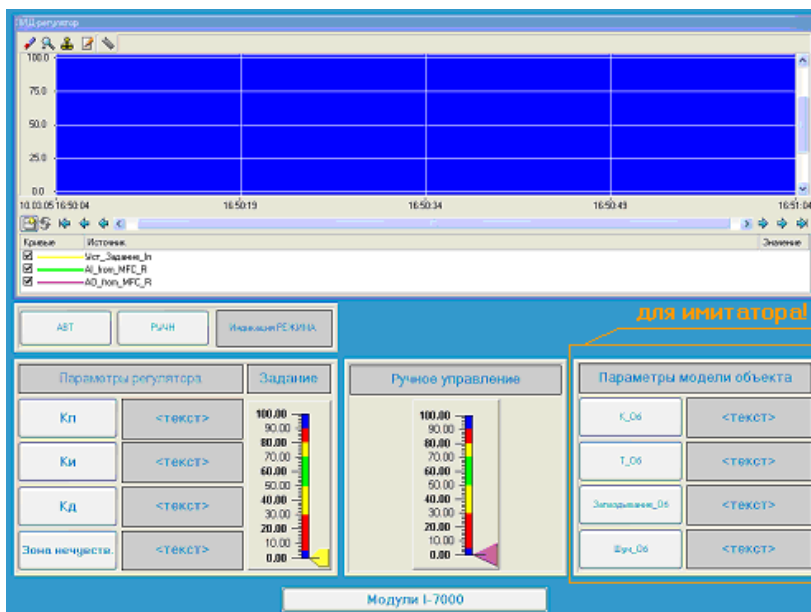
Для удобства работы, разрешение редактируемого графического экрана должно быть несколько ниже, чем установленное разрешение для интегрированной среды разработки, например, если для ИСР установлено разрешение 1280x1024, то для графического экрана желательно установить 1024x768. Это позволит держать в поле видимости основную часть содержимого экрана и лишь в редких случаях пользоваться полосами прокрутки. Открытую панель Навигатора проекта можно закрыть с помощью щелчка ЛК по иконке , вновь открыть – выбрав в основном меню пункт **Вид/Навигатор проекта**.



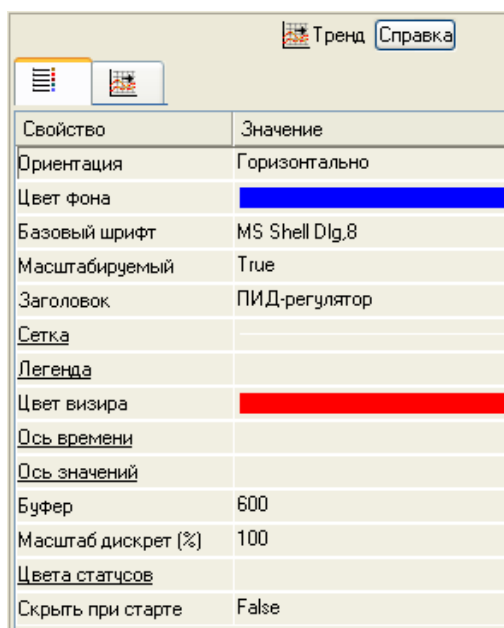
Размещение ГЭ, настройка атрибутов

Создание экрана выполняется в свободном стиле. То есть, выделяя ЛК мыши ГЭ из основного меню, будем их размещать на рабочем поле, перемещать и изменять их размеры. Выделение ГЭ на рабочем поле осуществляется щелчком ЛК, перемещение производится мышью при удерживаемой в нажатом состоянии ЛК, изменение размеров – после выделения ГЭ путем выделения ЛК необходимой стороны

контура ГЭ и удержания ЛК до достижения требуемого размера. Для перехода к свойствам ГЭ необходимо произвести на нем двойной щелчок ЛК. В открывшемся окне свойств ГЭ производятся операции редактирования. Разместим на рабочем поле экрана ГЭ следующим образом:

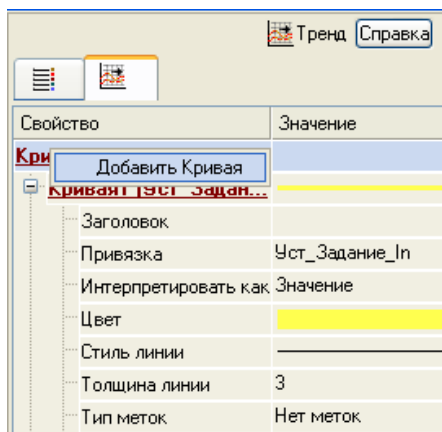


Основные свойства ГЭ Тренд зададим как:

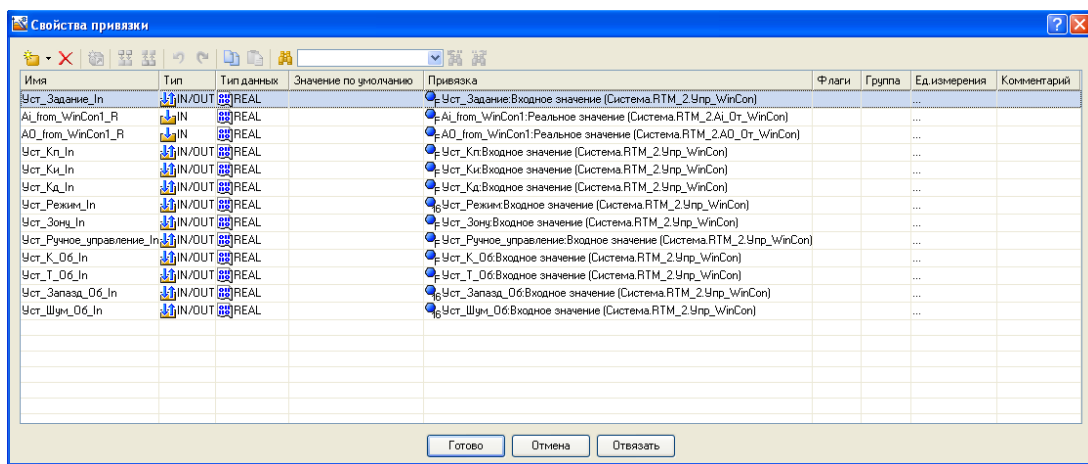


Перейдя во вкладку , определим три кривые для отображения аргументов Уст_Задание_In, AI_from_WinCon1_R и AO_from_WinCon1_R.

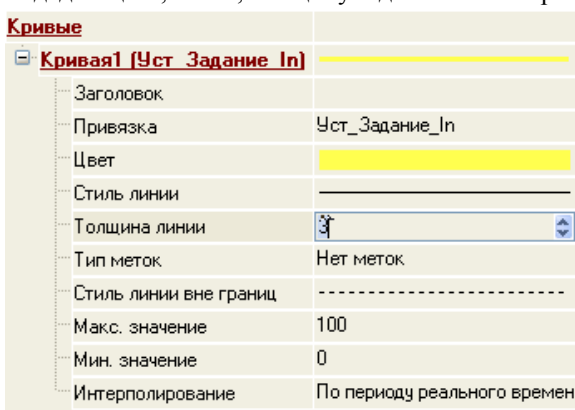
Дополнительные кривые создадим следующим образом: ЛК выделим элемент **Кривые** и нажмем ПК, в появившемся окне щелкнем ЛК на элементе **Добавить кривая**:



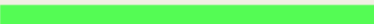
Создадим две дополнительные кривые и затем двойным щелчком ЛК на верхней кривой войдем в редактирование ее свойств. Поле **Привязка** служит для связывания кривой с аргументом шаблона, по щелчку ЛК открывается окно редактора аргументов для привязки аргумента к атрибуту ГЭ, выберем аргумент Уст_Задание_In и нажмем Готово.

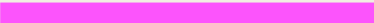


Зададим цвет, стиль, толщину и диапазон отображения для выбранного аргумента:



Подобным образом поступим в отношении двух других кривых:

Кривая [Ai from WinCo...	
Заголовок	
Привязка	Ai_from_WinCon1_R
Интерпретировать как	Значение
Цвет	
Стиль линии	
Толщина линии	3
Тип меток	Нет меток
Формат	%g
Стиль при I < 0 и W = 0	-----
Стиль при I = 0 и W = 1	— — — — —
Стиль при I < 0 и W = 1	-----
Макс. значение	100
Мин. значение	0
Интерполирование	По периоду реального времени

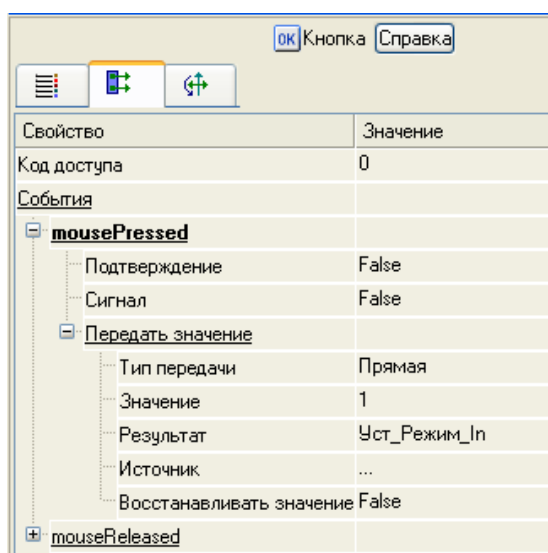
Кривая [AO from WinC...	
Заголовок	
Привязка	AO_from_WinCon1_R
Интерпретировать как	Значение
Цвет	
Стиль линии	
Толщина линии	3
Тип меток	Нет меток
Формат	%g
Стиль при I < 0 и W = 0	-----
Стиль при I = 0 и W = 1	— — — — —
Стиль при I < 0 и W = 1	-----
Макс. значение	100
Мин. значение	0
Интерполирование	По периоду реального времени

По завершению закроем свойства ГЭ щелчком ЛК по иконке  или в свободном пространстве рабочего поля.

Кнопки для переключения режима работы АВТ и РУЧН сконфигурируем для прямой отправки значения в связанный с ними аргумент шаблона по нажатию ЛК на данных ГЭ, для автоматического режима 0, для ручного - 1. Для ГЭ Кнопка, переводящего работу в автоматический режим:

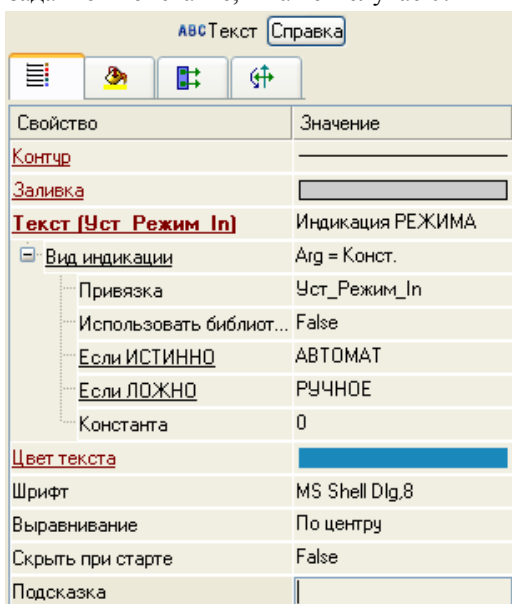
Свойство	Значение
Код доступа	0
События	
mousePressed	
Подтверждение	False
Сигнал	False
Передать значение	
Тип передачи	Прямая
Значение	0
Результат	Уст_Режим_In
Источник	...
Восстанавливать значение	False
mouseReleased	

для ручного режима:

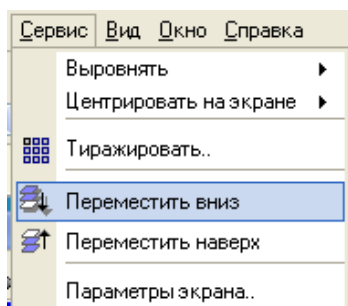


Оба ГЭ привязаны к аргументу шаблона Уст_Режим_In.

ГЭ Текст, индицирующий текущий режим, настраивается как текстовый индикатор, выводящий фиксированное сообщение по анализу на равенство/неравенство связанного с ним аргумента заданной константе, в нашем случае 0:



Все три связанных элемента расположим на ГЭ , настроенном для расположения на заднем плане с помощью выбора пункта меню **Сервис/Переместить вниз**:



Для обратной операции используется помещение ГЭ на передний план с помощью пункта **Сервис/Переместить вверх**.

ГЭ, составляющие блок для задания параметров регулятора и модели объекта управления, настраиваются во многом подобным образом. Для ввода значений в аргумент используется метод

Ввод и посылка, для отображения – форматный вывод. Свойства ГЭ Кнопка для задания параметра регулятора Кп выглядят так:

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

Свойства ГЭ Текст, отображающего данное значение на экране:

Свойство	Значение
Контур	
Заливка	
Текст (Уст_Кп_In)	<текст>
Вид индикации	Значение
Привязка	Уст_Кп_In
Формат	Float
Float	%.3f
Цвет текста	
Шрифт	MS Shell Dlg.12,жирный
Выравнивание	По центру
Скрыть при старте	False
Подсказка	

Свойства ГЭ Ползунок, применяемого для ввода задания регулятору и величины ручного управления, оставлены заданными по умолчанию, привязка к аргументу шаблона установлена для отображения и управления, например, для Уст_Задание:

Свойство	Значение
Скрыть при старте	False
Код доступа	0
Отображаемая величина (Уст_Задание_In)	0
Задаваемая величина (Уст_Задание_In)	
Положение ползунка	Справа
3D-эффекты	False
Ползунок	True
Полоса	True
Шкала	True
Фон	
Рамка	
Основная привязка	Уст_Задание_In

Оставшийся ГЭ Кнопка, предназначенный для организации перехода на второй экран разместим, но к настройке его свойств вернемся позже.

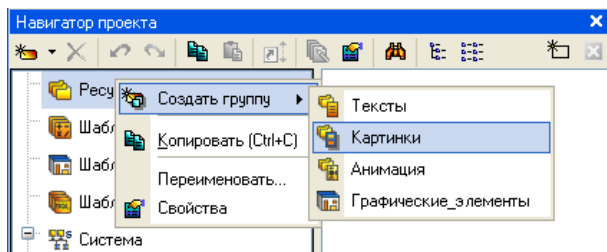
Модули I-7000

Сохраним проект с помощью иконки .

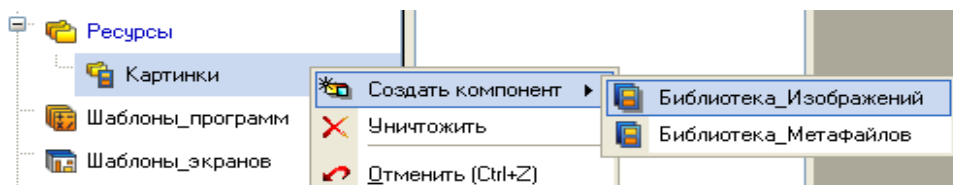
Создадим следующий экран – Экран#2. На этом экране будем выводить данные, получаемые/передаваемые от/на модули удаленного ввода/вывода I-7017 и I-7067 с помощью ГЭ Текст, Линия (с использованием динамического свойства Бегущая дорожка) и Замкнутый многоугольник (динамическое свойство Заполнение).

Использование ресурсных библиотек

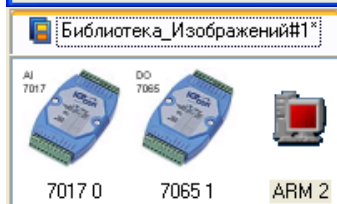
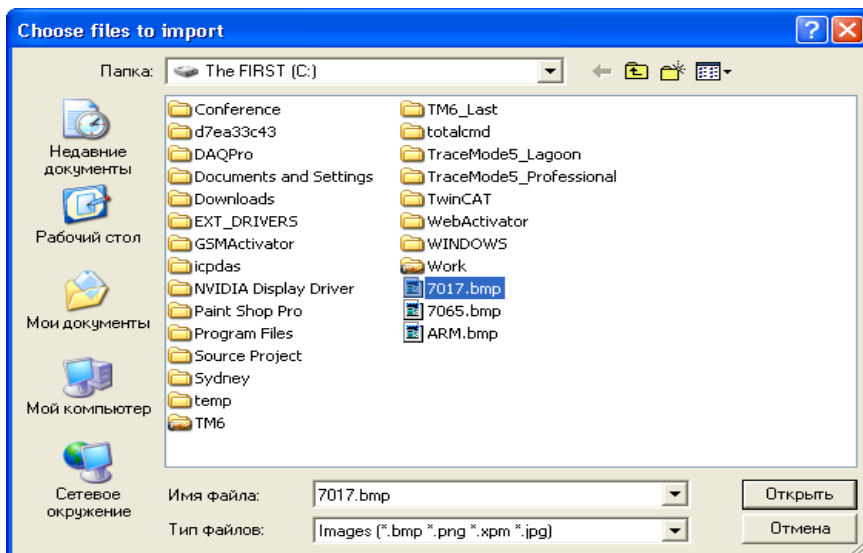
Оформление экрана выполним с помощью создаваемых ресурсных библиотек – графических элементов, картинок и видеоклипа. Сформируем ресурсные библиотеки. Выделим ЛК слой Ресурсы и ПК создадим новую группу – Картинки.



В ней – компонент Библиотека_Изображений#1.

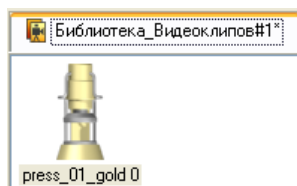


Войдем во вновь созданную библиотеку, для наполнения ее файлами воспользуемся иконкой  на панели инструментов, при выборе которой ЛК открывается диалог задания имени файла, содержащего изображение. В корневом каталоге диска **С** расположены файлы формата **bmp** с изображением модулей I-7000 и АРМ, которые импортируем в текущую библиотеку.

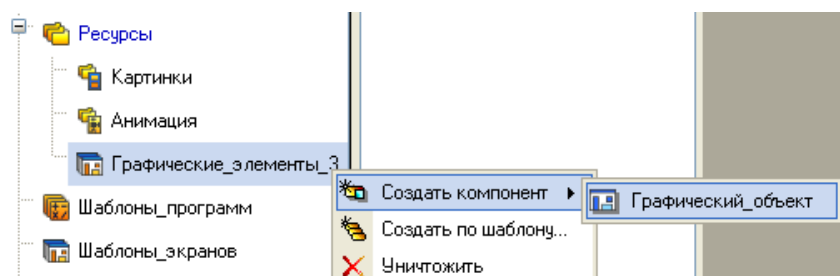


Сохраним изменения с помощью иконки , библиотека изображений подготовлена.

Далее создадим библиотеку анимаций и импортируем в нее видеоклип **press_01_gold.mng** из каталога **с:\каталог_инструментальной_системы\Lib\Animation\Presses**.

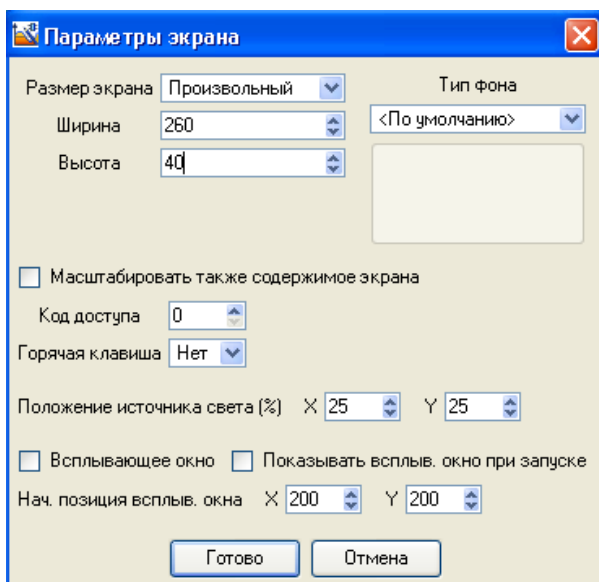


Также сохраним изменения с помощью иконки , библиотека анимаций подготовлена.
В слое Ресурсы создадим группу Графические_элементы, в ней – Графический_объект_1:

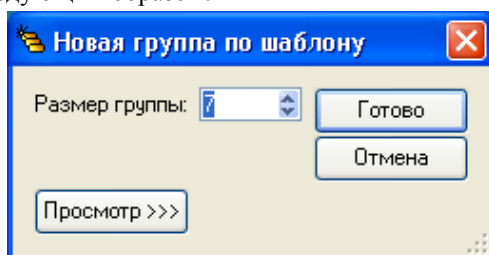


Создание графического объекта

Разработаем графический объект (ГО) для отображения состояния модуля I-7044, имеющего восемь дискретных выходов. Установим размеры ГО в пункте меню **Сервис\Параметры экрана** :



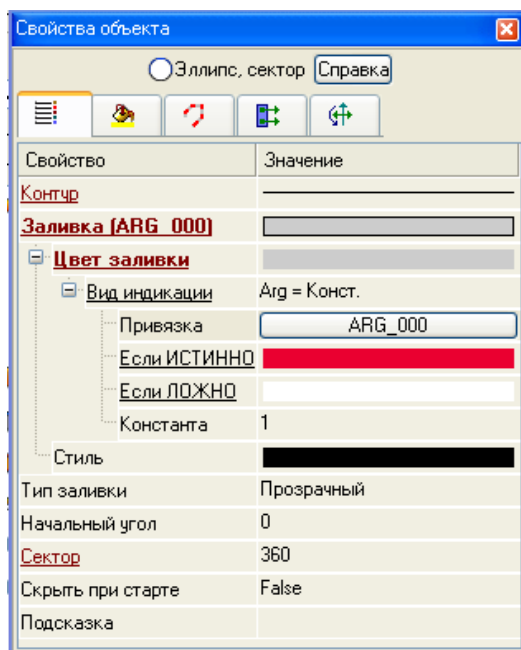
Нажмем Готово. Поскольку нам требуется индифицировать с помощью ГО состояние восьми дискретных выходов модуля, зададим ГО восемь аргументов, вызвав ПК свойства и перейдя к вкладке Аргументы. Определим один входной аргумент типа USINT с именем по умолчанию, последующие семь создадим с помощью иконки  (Создать по образцу) редактора аргументов следующим образом:



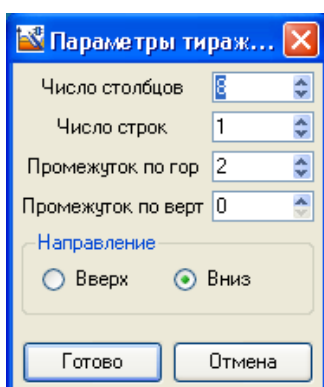
В результате будут созданы необходимые восемь аргументов для представления дискретных выходов:

Информация				
Аргументы				
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
ARG_000	IN	REAL		
ARG_001	IN	REAL		
ARG_002	IN	REAL		
ARG_003	IN	REAL		
ARG_004	IN	REAL		
ARG_005	IN	REAL		
ARG_006	IN	REAL		
ARG_007	IN	REAL		

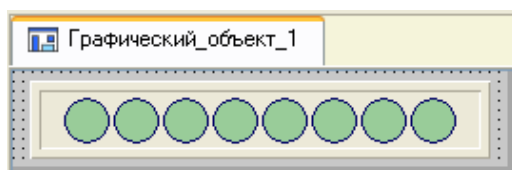
Двойным щелчком ЛК на имени ГО в правом окне Навигатора проекта откроем графический редактор и приступим к процессу редактирования. Используя ГЭ Рамка для создания подложки ГО, разместим на нем ГЭ  Эллипс. В свойствах ГЭ укажем привязку к аргументу ГО и определим метод динамизации, как изменение цвета заполнения по анализу на равенство аргумента 1:



Для создания семи оставшихся ГЭ воспользуемся пунктом меню **Сервис/Тиражировать**, указав в диалоге:



В результате получим восемь ГЭ Эллипс, привязанных к соответствующим аргументам ГО:



Отредактируем привязки созданных ГЭ к аргументам ГО, затем сохраним внесенные изменения с помощью иконки , графический объект подготовлен.

Закроем с помощью иконки  созданные библиотеки и вызовем свойства Экрана#2. Во вкладке Аргументы зададим, используя метод создания по образцу и виртуальные привязки следующий список аргументов:

Информация				
Аргументы				
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
ARG_000	IN	ab STRING		Канал_FLOAT:Базовое имя
ARG_001	IN	ab STRING		Канал_FLOAT:Базовое имя
ARG_002	IN	ab STRING		Канал_FLOAT:Базовое имя
ARG_003	IN	ab STRING		Канал_FLOAT:Базовое имя
ARG_004	IN	ab STRING		Канал_FLOAT:Базовое имя
ARG_005	IN	ab STRING		Канал_FLOAT:Базовое имя
ARG_006	IN	ab STRING		Канал_FLOAT:Базовое имя
ARG_007	IN	ab STRING		Канал_FLOAT:Базовое имя
ARG_008	IN	REAL		
ARG_009	IN	REAL		
ARG_010	IN	REAL		
ARG_011	IN	REAL		
ARG_012	IN	REAL		
ARG_013	IN	REAL		
ARG_014	IN	REAL		
ARG_015	IN	REAL		
ARG_016	IN	ab STRING		Канал_HEX16:Базовое имя
ARG_017	IN	USINT		
ARG_018	IN	USINT		Канал_HEX16:Бит 1
ARG_019	IN	USINT		Канал_HEX16:Бит 2
ARG_020	IN	USINT		Канал_HEX16:Бит 3
ARG_021	IN	USINT		Канал_HEX16:Бит 4
ARG_022	IN	USINT		Канал_HEX16:Бит 5
ARG_023	IN	USINT		Канал_HEX16:Бит 6
ARG_024	IN	USINT		Канал_HEX16:Бит 7
ARG_025	IN	USINT		Канал_HEX16:Бит 8

Первые восемь предназначены для отображения атрибутов Базовое имя каналов модуля I-7017, последующие восемь – реальных значений, следующий предназначен для имени канала модуля I-7067, один – реальное значение, последние восемь – для привязки битов выходного значения, формируемого для дискретного выхода модуля I-7067 к графическому объекту.

Удерживая нажатой клавишу CTRL выделяем ЛК первые восемь аргументов. В окне Навигатора проекта захватываем ЛК группу компонентов I7017#1 и перетаскиваем на выделенные строки табличного редактора компонентов, отпускаем ЛК. В результате имеем:

Информация		Аргументы			
					

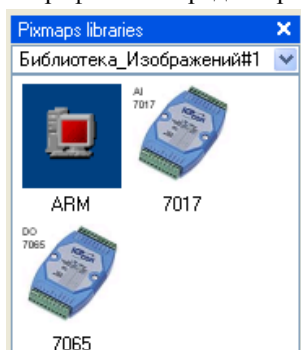
Выделяем следующие восемь аргументов и проделываем аналогичную операцию.

Затем выделяем аргумент ARG_016 и перетаскиваем на него группу компонентов I7067#2. То же самое производим с аргументом ARG_017.

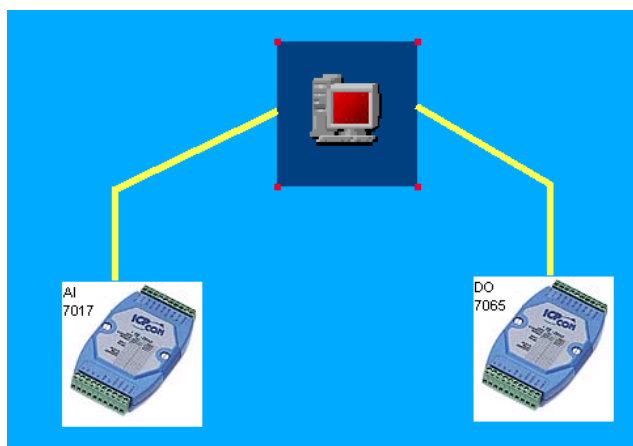
Далее выделяем оставшиеся восемь аргументов ARG_018 - ARG_025 и перетаскиваем на них канал DO#1 из группы компонентов I7067#2. В итоге будем иметь:

Информация		Аргументы			
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	
Aln_1_B_NAME	IN	ab STRING		Aln#1:Базовое имя (Система.RTM_2.17017#1)	
Aln_2_B_NAME	IN	ab STRING		Aln#2:Базовое имя (Система.RTM_2.17017#1)	
Aln_3_B_NAME	IN	ab STRING		Aln#3:Базовое имя (Система.RTM_2.17017#1)	
Aln_4_B_NAME	IN	ab STRING		Aln#4:Базовое имя (Система.RTM_2.17017#1)	
Aln_5_B_NAME	IN	ab STRING		Aln#5:Базовое имя (Система.RTM_2.17017#1)	
Aln_6_B_NAME	IN	ab STRING		Aln#6:Базовое имя (Система.RTM_2.17017#1)	
Aln_7_B_NAME	IN	ab STRING		Aln#7:Базовое имя (Система.RTM_2.17017#1)	
Aln_8_B_NAME	IN	ab STRING		Aln#8:Базовое имя (Система.RTM_2.17017#1)	
Aln_1_R	IN	REAL		Aln#1:Реальное значение (Система.RTM_2.17017#1)	
Aln_2_R	IN	REAL		Aln#2:Реальное значение (Система.RTM_2.17017#1)	
Aln_3_R	IN	REAL		Aln#3:Реальное значение (Система.RTM_2.17017#1)	
Aln_4_R	IN	REAL		Aln#4:Реальное значение (Система.RTM_2.17017#1)	
Aln_5_R	IN	REAL		Aln#5:Реальное значение (Система.RTM_2.17017#1)	
Aln_6_R	IN	REAL		Aln#6:Реальное значение (Система.RTM_2.17017#1)	
Aln_7_R	IN	REAL		Aln#7:Реальное значение (Система.RTM_2.17017#1)	
Aln_8_R	IN	REAL		Aln#8:Реальное значение (Система.RTM_2.17017#1)	
DO_1_B_NAME	IN	ab STRING		DO#1:Базовое имя (Система.RTM_2.17067#2)	
DO_1_R	IN	USINT		DO#1:Реальное значение (Система.RTM_2.17067#2)	
DO_1_b1	IN	USINT		DO#1:Бит 1 (Система.RTM_2.17067#2)	
DO_1_b2	IN	USINT		DO#1:Бит 2 (Система.RTM_2.17067#2)	
DO_1_b3	IN	USINT		DO#1:Бит 3 (Система.RTM_2.17067#2)	
DO_1_b4	IN	USINT		DO#1:Бит 4 (Система.RTM_2.17067#2)	
DO_1_b5	IN	USINT		DO#1:Бит 5 (Система.RTM_2.17067#2)	
DO_1_b6	IN	USINT		DO#1:Бит 6 (Система.RTM_2.17067#2)	
DO_1_b7	IN	USINT		DO#1:Бит 7 (Система.RTM_2.17067#2)	
DO_1_b8	IN	USINT		DO#1:Бит 8 (Система.RTM_2.17067#2)	

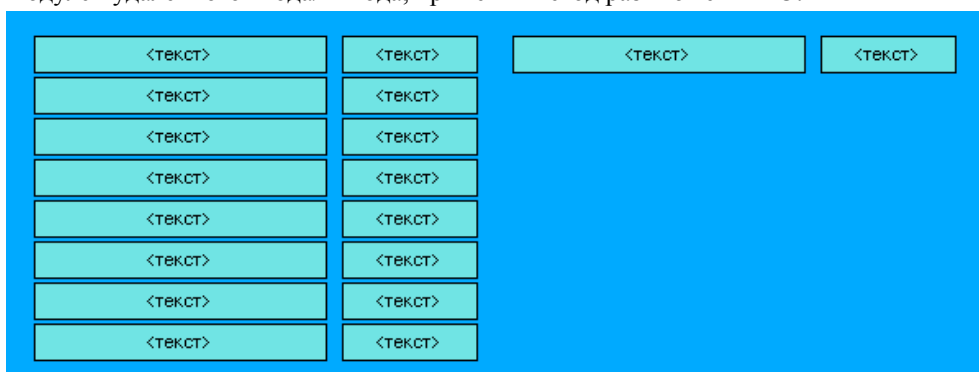
После завершения работы с редактором аргументов шаблона экрана Экран#2, вызовем его на редактирование. Для доступа к содержимому библиотек воспользуемся иконкой  основного меню. В графическом редакторе откроется окно ресурсной библиотеки, содержащее заданные картинки:



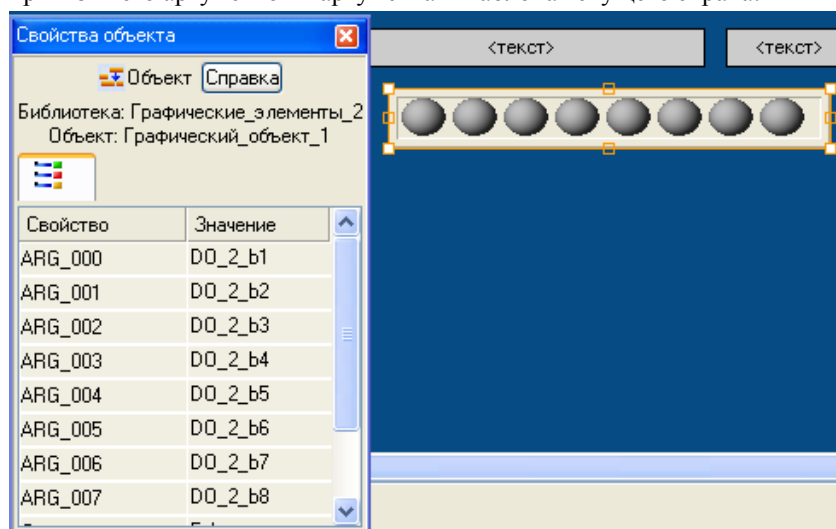
Выбрав ЛК изображение, и, удерживая ЛК в нажатом состоянии, перетащим картинку на рабочее поле для размещения. Таким образом, разместим наши картинки и свяжем их между собой с помощью ГЭ Ломаная линия:



Ниже с помощью ГЭ Текст разместим две группы (по два столбца каждая) для отображения базовых имен каналов и реальных значений, привязанных к соответствующим аргументам экрана обоих модулей удаленного ввода/вывода, применяя метод размножения ГЭ.



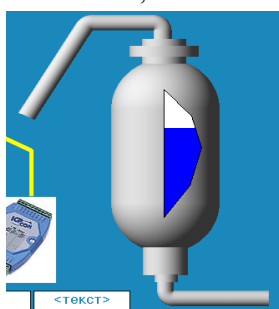
Разместим на рабочем поле созданный ГО, выбрав в основном меню  и перетащив его, удерживая нажатой ЛК. После отпущения ЛК на экран будет выведено окно свойств размещенного ГО для привязки его аргументов к аргументам шаблона текущего экрана:



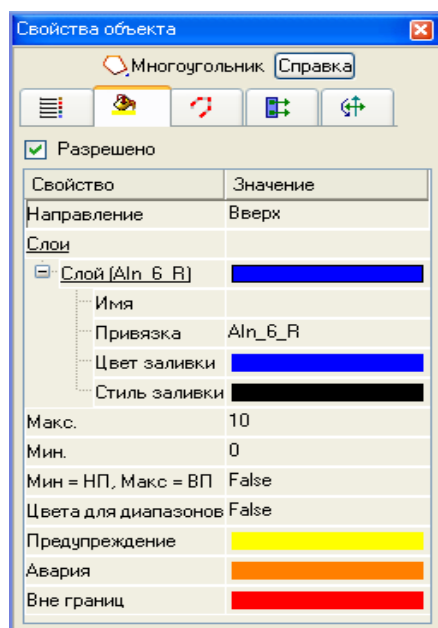
Последовательно осуществим привязку аргумента ARG_000 к DO_2_b1, ARG_001 к DO_2_b2 и т.д. Закроем свойства ГО щелчком ЛК на свободном месте рабочего поля.

Создание гистограммы

В правом верхнем углу экрана с помощью объемных ГЭ ,  и  изобразим некий технологический объект и разместим поверх его ГЭ . Последний будем использовать в качестве гистограммы произвольной формы для отображения технологического параметра - уровня заполнения, связанного с одним из аналоговых входов модуля I-7017.

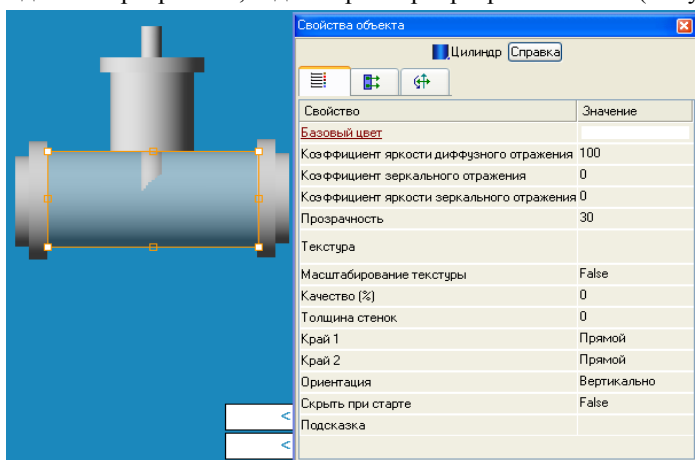


Выделив ЛК ГЭ , перейдем к редактированию его динамических свойств:



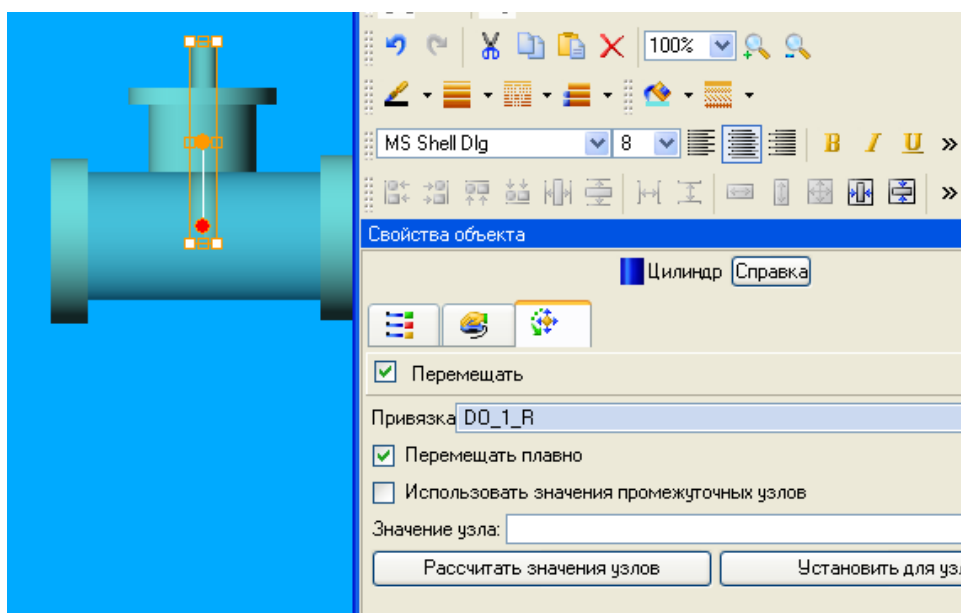
Определенная гистограмма будет отображать уровень заполнения, связанный с шестым входом модуля I-7017 в диапазоне 0 – 10, цвет заполнения – синий. Остальные настройки ГЭ сохранены заданными по умолчанию. Закроем свойства ГЭ.

В левом верхнем углу экрана с помощью объемных ГЭ изобразим клапан. Среднюю часть клапана сделаем прозрачной, задав параметр Прозрачность=30 (полупрозрачный).



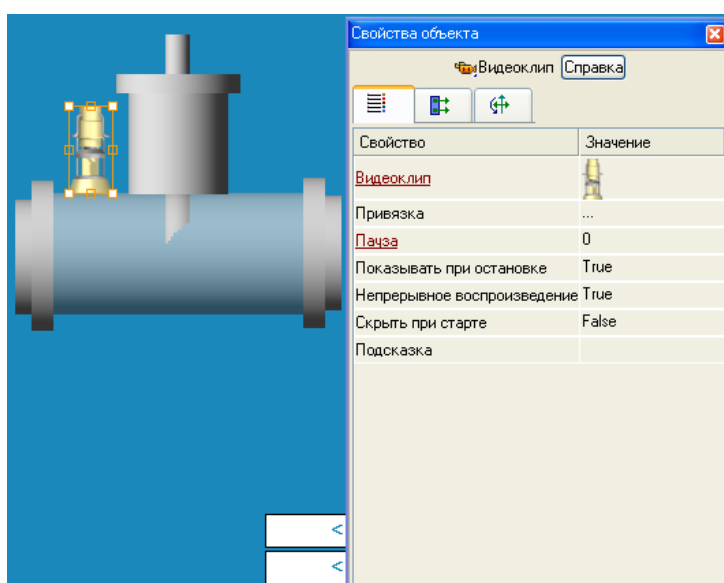
Перемещение ГЭ

Шток клапана выполним перемещающимся, в качестве величины, задающей перемещение, возьмем реальное значение канала дискретного выхода модуля I-7067 – DO_1_R. Определим начальную и конечную точки перемещения ГЭ:

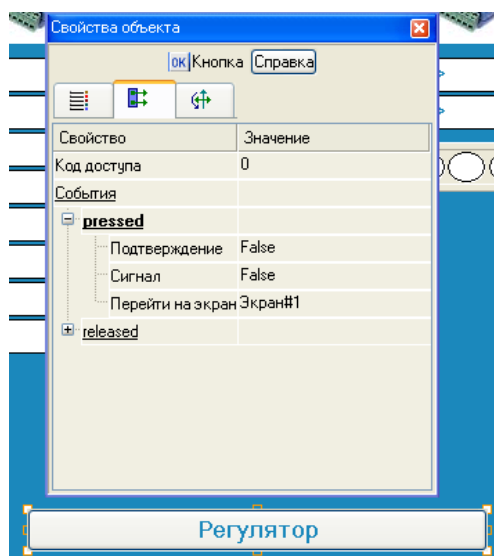


Использование видеоклипов

Здесь же разместим видеоклип из библиотеки анимаций. Воспользуемся иконкой  основного меню, захватим в открывшемся окне клип и, удерживая ЛК нажатой, перетащим его к клапану. Свойства видеоклипа оставим заданными по умолчанию, хотя можно его связать с аргументом DO_1_R шаблона экрана, связанного с дискретным выходом модуля I-7067. В последнем случае, задав атрибут Непрерывное воспроизведение=False, проигрывание клипа будет возможно при ненулевом значении дискретного входа.

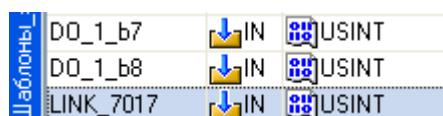


В нижней части экрана разместим ГЭ Кнопка для реализации перехода на Экран#1. В свойствах ГЭ укажем ссылку на шаблон Экран#1:

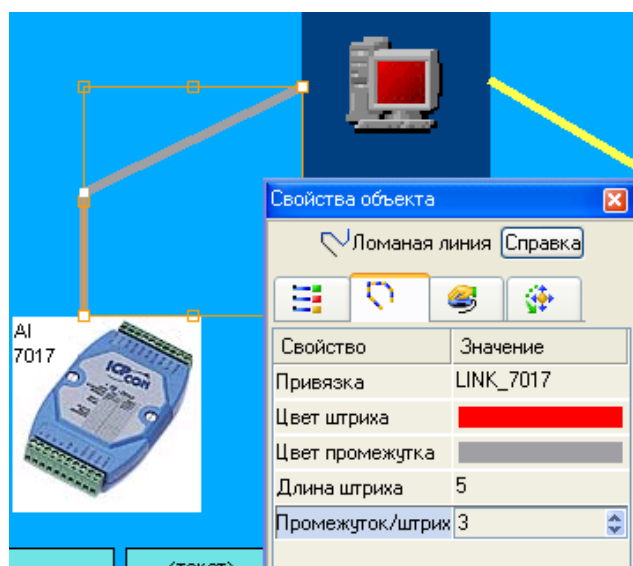


Бегущие дорожки

Реализацию бегущих дорожек сделаем таким образом, чтобы они индицировали состояние обмена с модулями удаленного ввода/вывода. Анализ состояния обмена будем осуществлять по атрибуту Достоверность канала Ain#1 для модуля I-7017. Поскольку атрибут Достоверность принимает значение 1 в случае возникновения проблем со связью по последовательным коммуникациям, то необходимо предусмотреть программу, инвертирующую значение данного атрибута для его применения в целях динамизации ГЭ Линия. Однако, определим в качестве аргументов шаблона экрана Экран#2 LINK_7017, добавив их в редакторе аргументов:

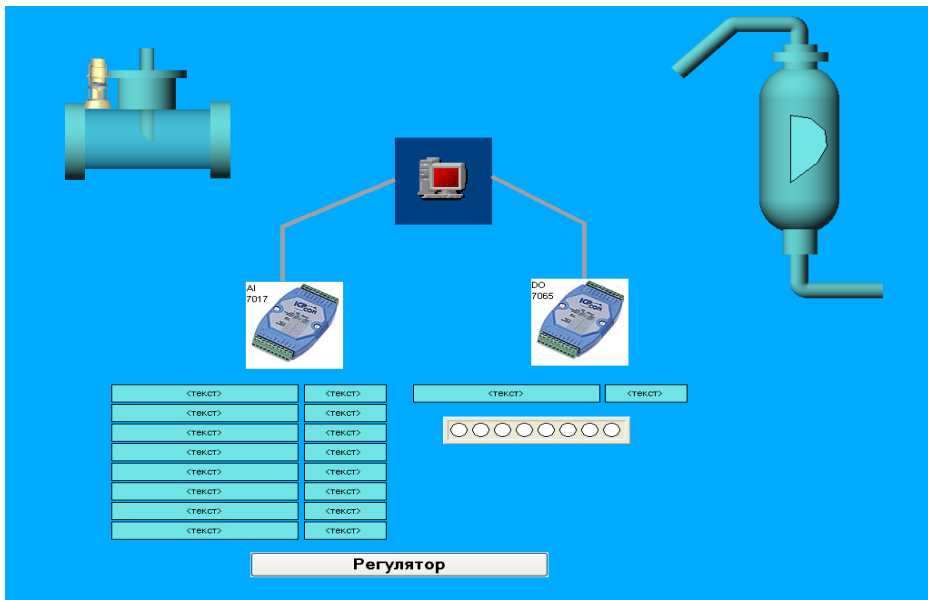


Выделим последовательно ГЭ Линия, связывающие изображения АРМ и модулей ввода/вывода, и задав им динамическое свойство Бегущая дорожка, привяжем их к соответствующим аргументам:

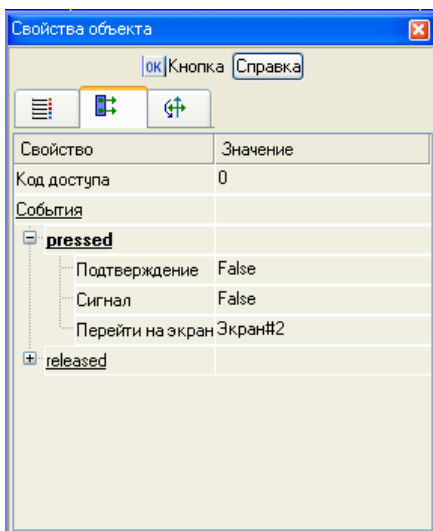


Изменить направление движения бегущей дорожки можно «перевернув» ее в ходе редактирования.

В результате проделанной работы Экран#2 будет иметь вид:



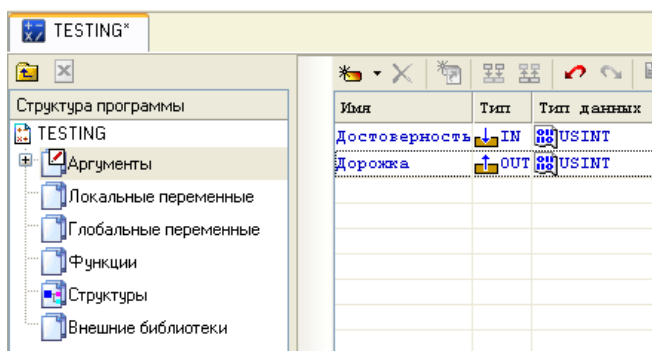
Вернемся к шаблону Экран#1. ГЭ Кнопка настраивается на переход к шаблону экрана Экран#2 как:



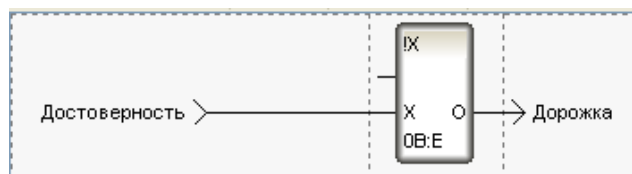
Сохраним внесенные изменения с помощью иконки .

Разработка программы для динамизации Бегущих дорожек

Прежде чем размещать созданные экраны на узле АРМ и осуществлять связывание их аргументов с атрибутами каналов, разработаем программу для инвертирования атрибута Достоверность. В слое шаблонов программ создадим новую программу, переименуем ее в TESTING и зададим аргументы:



Выберем Техно-FBD в качестве языка программирования и создадим следующее:



Откомпилируем программу с помощью клавиши F7. Откроем дополнительную панель Навигатора проекта и перетащим шаблон программы TESTING в группу узла RTM_2 I7017#1. Во вкладке Аргументы свойств канала класса вызовов свяжем входной аргумент, предварительно в настройках ИС сняв флаг **Переименовывать аргумент при привязке**:

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Достоверность	IN	USINT		Aln#1: Достоверность (Система.RTM_2.I7017#1)
Дорожка	OUT	USINT		

Теперь привяжем выходные аргументы каналов класса Вызов TESTING:35 непосредственно с аргументами шаблона экрана Экран#2. Для этого откроем вкладку Аргументы свойств шаблона Экран#2 и двойным щелчком ЛК в поле Привязка аргумента LINK_7017 укажем:

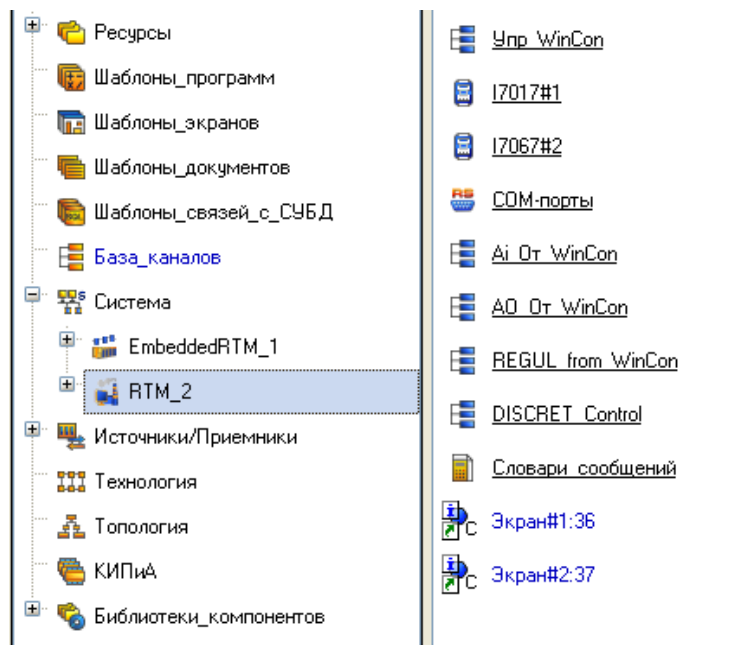
Имя	Тип	Тип данных	Значение по умолчанию
Достоверность	IN	USINT	
Дорожка	OUT	USINT	

Buttons: Удалить привязку, Привязать канал, Привязать аргумент, Привязка, Отмена

и нажмем клавишу Привязка. Подобным образом поступим в случае с привязкой аргумента LINK_7044. Окончательно будем иметь:

Информация		Аргументы		
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Aln_1_B_NAME	IN	ab STRING		Aln#1:Базовое имя (Система.RTM_2.I7017#1)
Aln_2_B_NAME	IN	ab STRING		Aln#2:Базовое имя (Система.RTM_2.I7017#1)
Aln_3_B_NAME	IN	ab STRING		Aln#3:Базовое имя (Система.RTM_2.I7017#1)
Aln_4_B_NAME	IN	ab STRING		Aln#4:Базовое имя (Система.RTM_2.I7017#1)
Aln_5_B_NAME	IN	ab STRING		Aln#5:Базовое имя (Система.RTM_2.I7017#1)
Aln_6_B_NAME	IN	ab STRING		Aln#6:Базовое имя (Система.RTM_2.I7017#1)
Aln_7_B_NAME	IN	ab STRING		Aln#7:Базовое имя (Система.RTM_2.I7017#1)
Aln_8_B_NAME	IN	ab STRING		Aln#8:Базовое имя (Система.RTM_2.I7017#1)
Aln_1_R	IN	REAL		Aln#1:Реальное значение (Система.RTM_2.I7017#1)
Aln_2_R	IN	REAL		Aln#2:Реальное значение (Система.RTM_2.I7017#1)
Aln_3_R	IN	REAL		Aln#3:Реальное значение (Система.RTM_2.I7017#1)
Aln_4_R	IN	REAL		Aln#4:Реальное значение (Система.RTM_2.I7017#1)
Aln_5_R	IN	REAL		Aln#5:Реальное значение (Система.RTM_2.I7017#1)
Aln_6_R	IN	REAL		Aln#6:Реальное значение (Система.RTM_2.I7017#1)
Aln_7_R	IN	REAL		Aln#7:Реальное значение (Система.RTM_2.I7017#1)
Aln_8_R	IN	REAL		Aln#8:Реальное значение (Система.RTM_2.I7017#1)
DO_1_B_NAME	IN	ab STRING		DO#1:Базовое имя (Система.RTM_2.I7067#2)
DO_1_R	IN	USINT		DO#1:Реальное значение (Система.RTM_2.I7067#2)
DO_1_b1	IN	USINT		DO#1:Бит 1 (Система.RTM_2.I7067#2)
DO_1_b2	IN	USINT		DO#1:Бит 2 (Система.RTM_2.I7067#2)
DO_1_b3	IN	USINT		DO#1:Бит 3 (Система.RTM_2.I7067#2)
DO_1_b4	IN	USINT		DO#1:Бит 4 (Система.RTM_2.I7067#2)
DO_1_b5	IN	USINT		DO#1:Бит 5 (Система.RTM_2.I7067#2)
DO_1_b6	IN	USINT		DO#1:Бит 6 (Система.RTM_2.I7067#2)
DO_1_b7	IN	USINT		DO#1:Бит 7 (Система.RTM_2.I7067#2)
DO_1_b8	IN	USINT		DO#1:Бит 8 (Система.RTM_2.I7067#2)
LINK_7017	IN	USINT		TESTING:35:Дорожка (Система.RTM_2.I7017#1)

Перетаскиваем шаблоны экранов Экран#1 и Экран#2 из слоя Шаблоны_экранов в узел RTM_2 слоя Система.



Сохраним внесенные изменения в проект с помощью иконки  и затем выполним процедуру подготовки проекта к запуску в реальном времени – .

Темы индивидуальных лабораторных заданий “Интерфейсы в АСОИиУ”

1. АСО и У по бронированию отелей.

База данных гостиниц, отелей, хостелов и квартир; справочник клиентов; индивидуальные бланки заявок и подтверждений бронирования; учёт брони; расчёт премий для агентов.

2. АСО и У по учёту мероприятий в спортивной организации

База участников соревнований, список матчей и семинаров, начисление баллов, учётные карты.

3. АСО и У электронным журналом для ВУЗов

База данных учащихся, дисциплин, расписание занятий, оценок, преподавателей.

4. АСО и У потоками данных для учебного центра

Ведение базы данных учащихся, курсы и семинары, расписание занятий, баланс по клиентам, контроль оплаты.

5. АСО и У торговых операций малого предприятия

Ведение базы клиентов; список поставщиков; каталог товаров; форма заказов; учёт по складам; операции поступлений, перемещений, списаний и продаж; отчёты по клиентам, товарам и торговому балансу.

6. АСО и У автошколы

Автоматизация автошколы: справочники курсантов и групп; список автотранспорта и преподавателей; документы для зачисления, прохождения обучения, сдачи экзаменов, для ГИБДД и другие.

7. АСО и У автотранспортом на предприятии

База данных транспортных средств и водителей; учёт страховых полисов, ремонта, расходов, ДТП и пр.; доверенности и закрепления за ТС; техническое обслуживание и мероприятия; справочники по персоналу и автосервису.

8. АСО и У выписками медицинских рецептов

Форма рецепта со сроком действия и списком препаратов; база пациентов; справочники медикаментов, групп, категорий и другие; отчёт за период.

9. АСО и У ветеринарной клиники

Ведение базы клиентов и питомцев; учёт медикаментов и услуг; список сотрудников с расчётом прибыли; предупреждения о запланированных приёмах; база поставщиков с балансом; различные отчёты

10. АСО и У учёта почтовых отправок

База данных покупателей и товаров, справочник по индексам, документы для отправления и наклейка со штрих-кодом суммы стоимости и возвратов, отчёт за период.

11. АСО и У сервисного центра

Обработка заказов; поступление и расход комплектующих; справочники моделей, работ, производителей и пр.; состояние склада; баланс по поставщикам; квитанция на приём и гарантийный талон.

Требования к отчету.

1. Отчет выполняется на листах формата А4 с использованием любого текстового процессора и распечатывается на принтере.
2. Титульный лист отчета выполняется по стандартной форме.
3. Рамка на последующих листах отчета необязательна.
4. Листы отчета необходимо скрепить.
5. Отчет должен быть подписан исполнителем.
6. Тексты программ должны содержать комментарии к использованию переменных и работе программы.
7. Тексты программ распечатываются на принтере.

Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)» Волжский филиал Факультет <u>очный</u> Кафедра <u>математики и информатики</u> Направление подготовки <u>Автоматизированные системы обработки информации и управления</u> О Т Ч Е Т к лабораторной работе №1 «Интерфейсы в АСОИиУ» Дата сдачи «__»_____20__г Выполнил: студент гр. ОП-14, Петров И.В. Зачтено «__»_____20__г Преподаватель: Иванов И.И. Чебоксары 2015

Пример оформления титульного листа.

СПИСОК ЛИТЕРАТУРЫ

1. Герасимов А.В., Титовцев А.С. SCADA Trace Mode 6. Учебное пособие – Казань: Изд-во Казан. гос. технол. ун-та, 2011. – 128 с.
2. Сундуков Е.Ю. Интегрированные системы проектирования и управления PDF. Сборник описаний лаб. работ. – Сыктывкар: СЛИ, 2013. – 104с.
3. AdAstra Research Group. Руководство пользователя Trace mode 6. Учебное пособие – Издательство AdAstra Research Group, 2008.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
ЛАБОРАТОРНАЯ РАБОТА № 1.	
Интерфейс ПО “Система мониторинга с узлом АРМ”	4
ЛАБОРАТОРНАЯ РАБОТА № 2.	
Функции управления ПО “Узел АРМ”	11
ЛАБОРАТОРНАЯ РАБОТА № 3.	
Форматирование экранной формы ПО “Узел АРМ”	16
ЛАБОРАТОРНАЯ РАБОТА № 4.	
Связь по протоколу DDE с математическим редактором. Подключение реального внешнего модуля ввода сигналов	20
ЛАБОРАТОРНАЯ РАБОТА № 5.	
Создание информационной базы	26
ЛАБОРАТОРНАЯ РАБОТА № 6.	
Модули подключения технического обеспечения к АРМ	33
ЛАБОРАТОРНАЯ РАБОТА № 7.	
Математическое обеспечение контроллера	39
ЛАБОРАТОРНАЯ РАБОТА № 8.	
ПО программа для дискретного управления	47
ЛАБОРАТОРНАЯ РАБОТА № 9.	
Создание графического пользовательского интерфейса АРМ	57
ТЕМЫ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ	80
Требования к отчету	81
СПИСОК ЛИТЕРАТУРЫ	82
СОДЕРЖАНИЕ	83