

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
«ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АВТОМАТИЗИРОВАННОГО
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

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.

Создание SQL- запроса для работы с внешней реляционной базой данных

1.Цель работы

Разработать SQL-запросы для записи во внешнюю базу данных MS Access значений каналов AI_from_WinCon и AO_from_WinCon в , а также каналов Ain#1...Ain#8.

2. Порядок выполнения работы

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

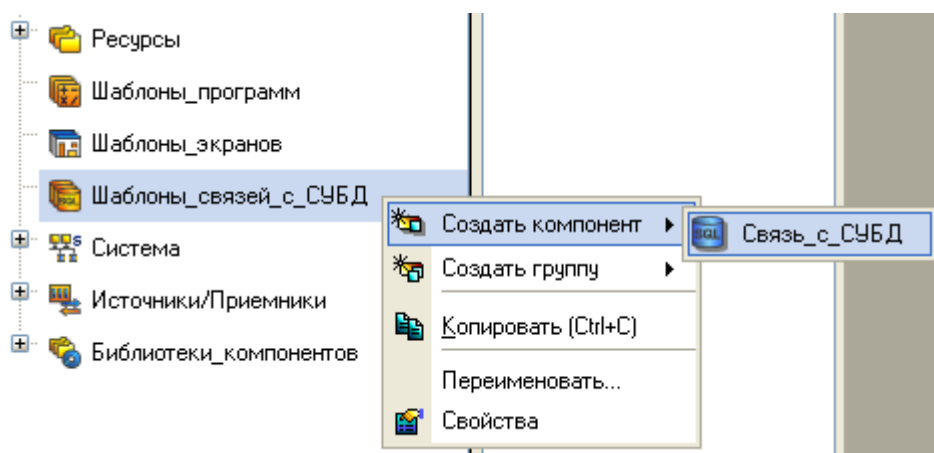
3. Образец выполнения работы

Разработанные SQL-запросы будем записывать во внешнюю базу данных MS Access. Реальные значения каналов AI_from_WinCon и AO_from_WinCon будем заносить в таблицу Tab1, каналов Ain#1...Ain#8 в таблицу Tab2. Причем, в целях оптимизации количества транзакций запись в Tab1 БД будет осуществляться по существенным изменениям реальных значений каналов, в свою очередь в Tab2 – в фоновом режиме в каждом цикле пересчета базы каналов.

Исходным материалом для проектирования служит проект, созданный на ранее изучаемом предмете “Интерфейсы АСОИиУ” и сохраненный с именем **regul.prj**.

Создание SQL-запросов

Разработку начнем с запуска интегрированной среды разработки TRACE MODE 6. Для создания запроса к СУБД выделим ЛК слой **Шаблоны_связей_с_СУБД** и по щелчку ПК создадим новый компонент – **База_данных#1**:



Щелчком ПК на вновь созданном компоненте вызовем его свойства и, перейдя к вкладке **Аргументы**, приступим к созданию аргументов и их привязке к атрибутам каналов.

Создание аргументов

При создании аргументов будем учитывать то, что тип аргументов для записи в СУБД должен быть определен как **IN**, а для автоматического формирования имен аргументов по именам соответствующих каналов при осуществлении привязки во вкладке **Редактор аргументов** настроек ИС установим флажок в пункте **Переименовывать аргументы при привязке**.

В Навигаторе проекта откроем слой **Система**, в ней - узел **RTM_2** и выделим ЛК группу компонентов **Ai_От_WinCon**. Удерживая ЛК нажатой, перетащим канал **Ai_from_WinCon1** в окно табличного редактора аргументов, отпустим ЛК. Те же операции проделаем с каналом **Ao_from_WinCon1**, что расположен в группе **Ao_От_WinCon**. В результате выполненных действий будем иметь:

Информация				
Аргументы				
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Ai_from_WinCon1_R	IN	REAL		Ai_from_WinCon1:Реальное значение (Система.RTM_2Ai_0т_WinCon)
AO_from_WinCon1_R	IN	REAL		AO_from_WinCon1:Реальное значение (Система.RTM_2AO_0т_WinCon)

Далее, в Навигаторе проекта выделим группу компонентов **I7017#1** и также перетащим ее в окно табличного редактора аргументов шаблона **База_данных#1**. Получим:

Информация				
Аргументы				
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Ai_from_WinCon1_R	IN	REAL		Ai_from_WinCon1:Реальное значение (Система.RTM_2Ai_0т_WinCon)
AO_from_WinCon1_R	IN	REAL		AO_from_WinCon1:Реальное значение (Система.RTM_2AO_0т_WinCon)
Aln_1_R	IN	REAL		Aln#1:Реальное значение (Система.RTM_2I7017#1)
Aln_2_R	IN	REAL		Aln#2:Реальное значение (Система.RTM_2I7017#1)
Aln_3_R	IN	REAL		Aln#3:Реальное значение (Система.RTM_2I7017#1)
Aln_4_R	IN	REAL		Aln#4:Реальное значение (Система.RTM_2I7017#1)
Aln_5_R	IN	REAL		Aln#5:Реальное значение (Система.RTM_2I7017#1)
Aln_6_R	IN	REAL		Aln#6:Реальное значение (Система.RTM_2I7017#1)
Aln_7_R	IN	REAL		Aln#7:Реальное значение (Система.RTM_2I7017#1)
Aln_8_R	IN	REAL		Aln#8:Реальное значение (Система.RTM_2I7017#1)
TESTING_35_R	IN	DINT		TESTING:35:Реальное значение (Система.RTM_2I7017#1)

Последний из созданных аргументов **TESTING_25_R** выделим ЛК и удалим с помощью иконки  панели инструментов табличного редактора аргументов или нажатием клавиши **Del**.

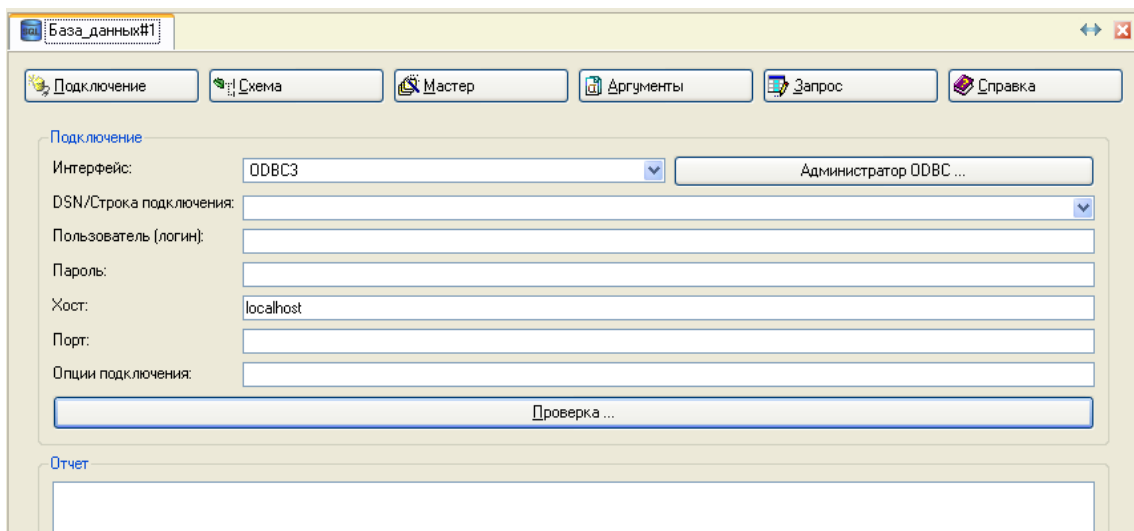
Создадим с помощью иконки  еще один аргумент для передачи в СУБД временной отметки, изменим его имя на **Time_Mark**:

Информация				
Аргументы				
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Ai_from_WinCon1_R	IN	REAL		Ai_from_WinCon1:Реальное значение (Система.RTM_2Ai_0т_WinCon)
AO_from_WinCon1_R	IN	REAL		AO_from_WinCon1:Реальное значение (Система.RTM_2AO_0т_WinCon)
Aln_1_R	IN	REAL		Aln#1:Реальное значение (Система.RTM_2I7017#1)
Aln_2_R	IN	REAL		Aln#2:Реальное значение (Система.RTM_2I7017#1)
Aln_3_R	IN	REAL		Aln#3:Реальное значение (Система.RTM_2I7017#1)
Aln_4_R	IN	REAL		Aln#4:Реальное значение (Система.RTM_2I7017#1)
Aln_5_R	IN	REAL		Aln#5:Реальное значение (Система.RTM_2I7017#1)
Aln_6_R	IN	REAL		Aln#6:Реальное значение (Система.RTM_2I7017#1)
Aln_7_R	IN	REAL		Aln#7:Реальное значение (Система.RTM_2I7017#1)
Aln_8_R	IN	REAL		Aln#8:Реальное значение (Система.RTM_2I7017#1)
Time_Mark	IN	REAL		

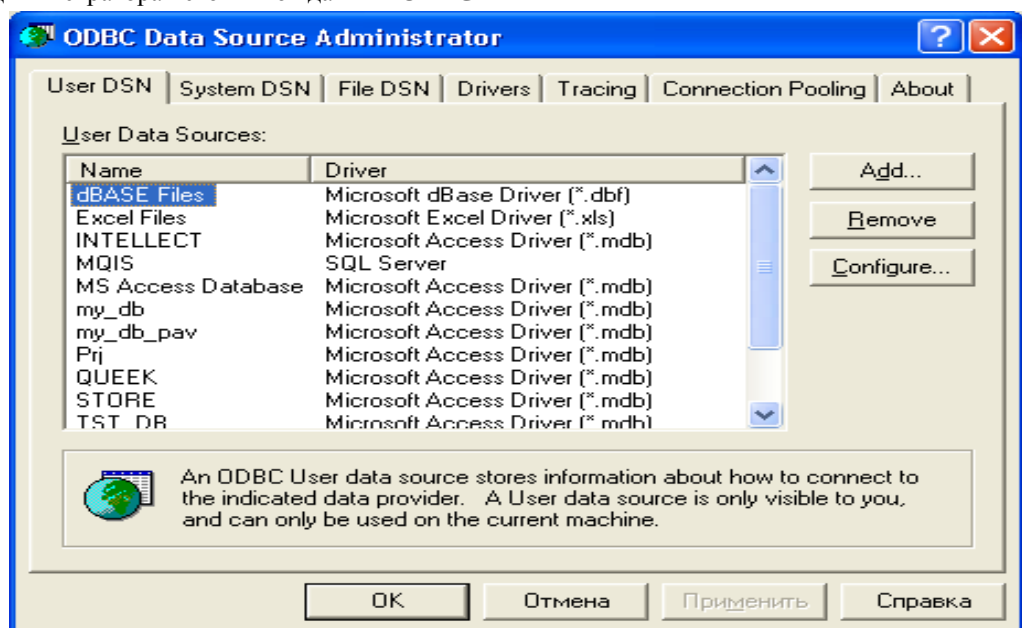
Привязку для данного аргумента выполним позднее. Закроем окно табличного редактора аргументов.

Создание файла базы данных и пользовательского DSN

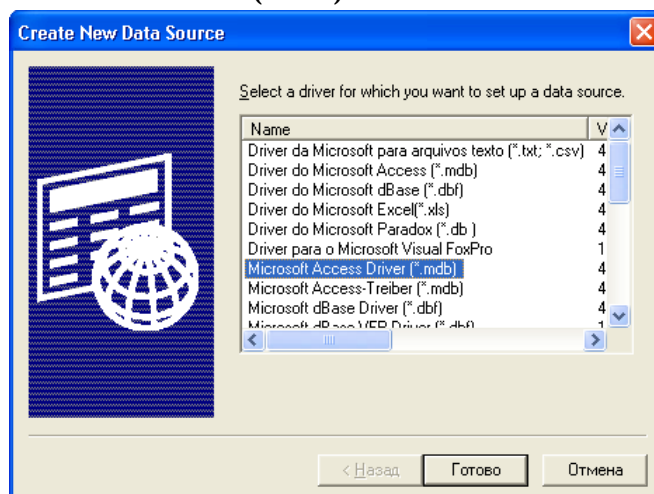
Двойным щелчком ЛК откроем на редактирование шаблон **База_данных#1**. В бланке **Подключение** редактора шаблона



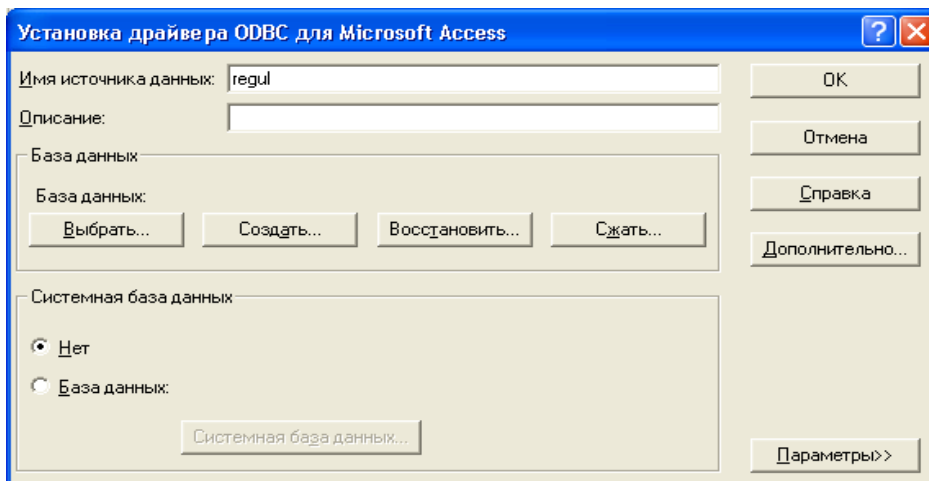
выберем ЛК экранную кнопку **Администратор ODBC...** В бланке **User DSN** открывшегося окна Администратора источников данных ODBC



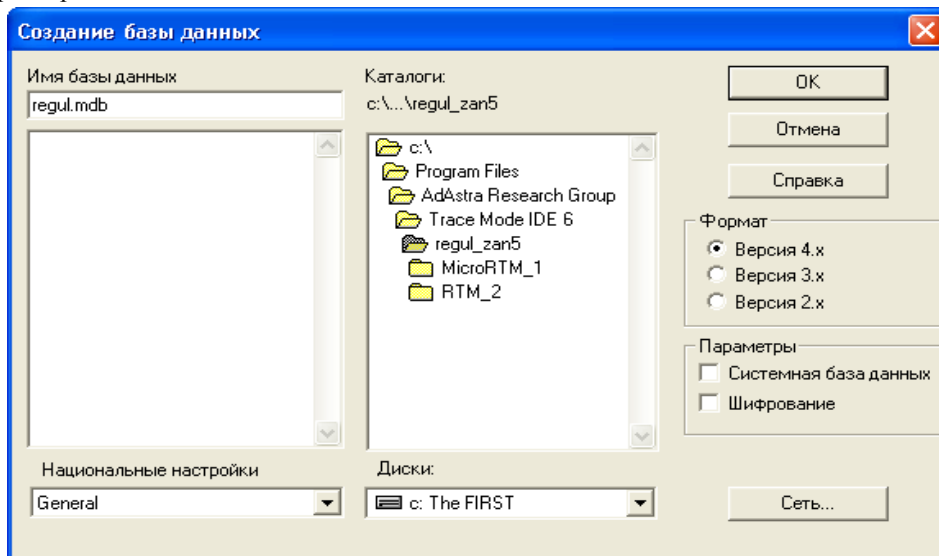
нажмем экранную кнопку **Add...** для создания нового источника данных. В нем выберем драйвер **Microsoft Access Driver (*.mdb)**:



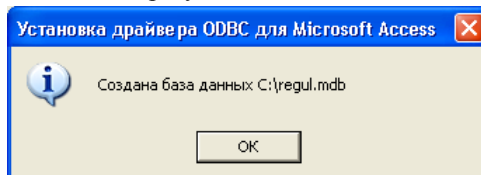
и нажмем экранную кнопку **Готово**. В окне **Установка драйвера ODBC для Microsoft Access** укажем имя источника данных – **regul** и в разделе **База данных** нажмем экранную кнопку **Создать...**



В диалоге задания имени и расположения создаваемого файла укажем **regul.mdb** и корневую директорию диска C.



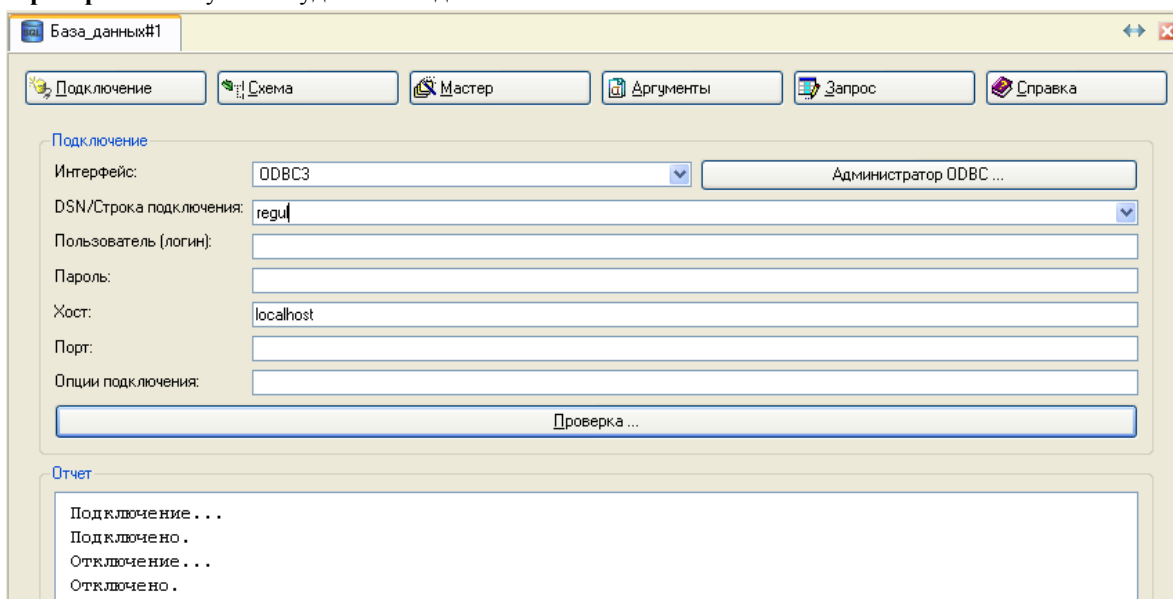
Нажмем **ОК**. В результате выполненного действия файл будет создан:



В списке пользовательских **DSN** будет добавлен новый источник данных:



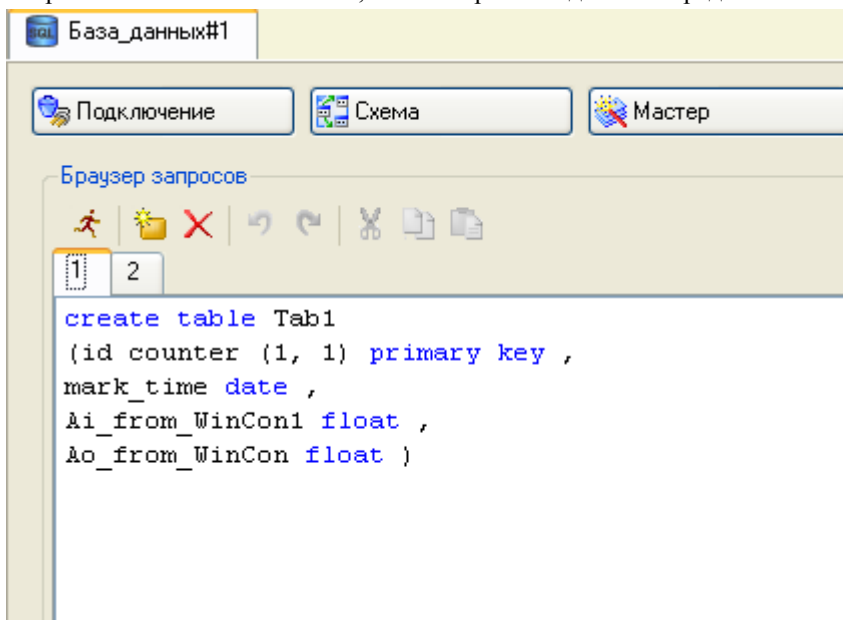
Нажмем **ОК** и вернемся в редактор шаблона **База_данных#1**. В строке **DSN/Строка подключения** из выпадающего списка укажем **regul**. Оставив остальные строки бланка заданными по умолчанию, произведем проверку подключения к вновь созданному источнику, нажав экранную кнопку **Проверка....** Результат будем наблюдать в окне **Отчет** бланка:



Создание таблиц

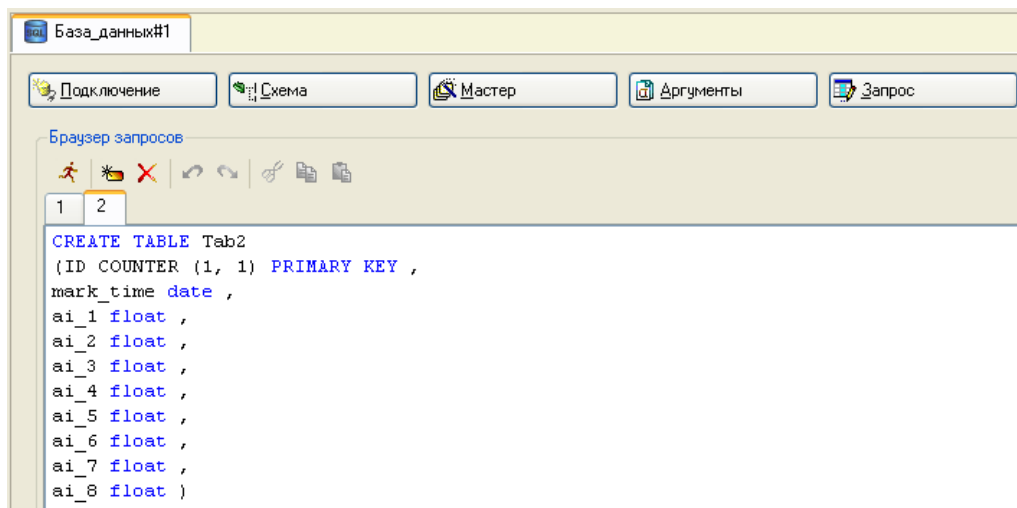
Создадим в файле **regul.mdb** две таблицы для записи реальных значений каналов и метки времени.

Для этого откроем бланк **Запрос** редактора шаблона и с помощью иконки  создадим новый запрос типа **CREATE TABLE**, текст запроса введем непосредственно в поле редактора:



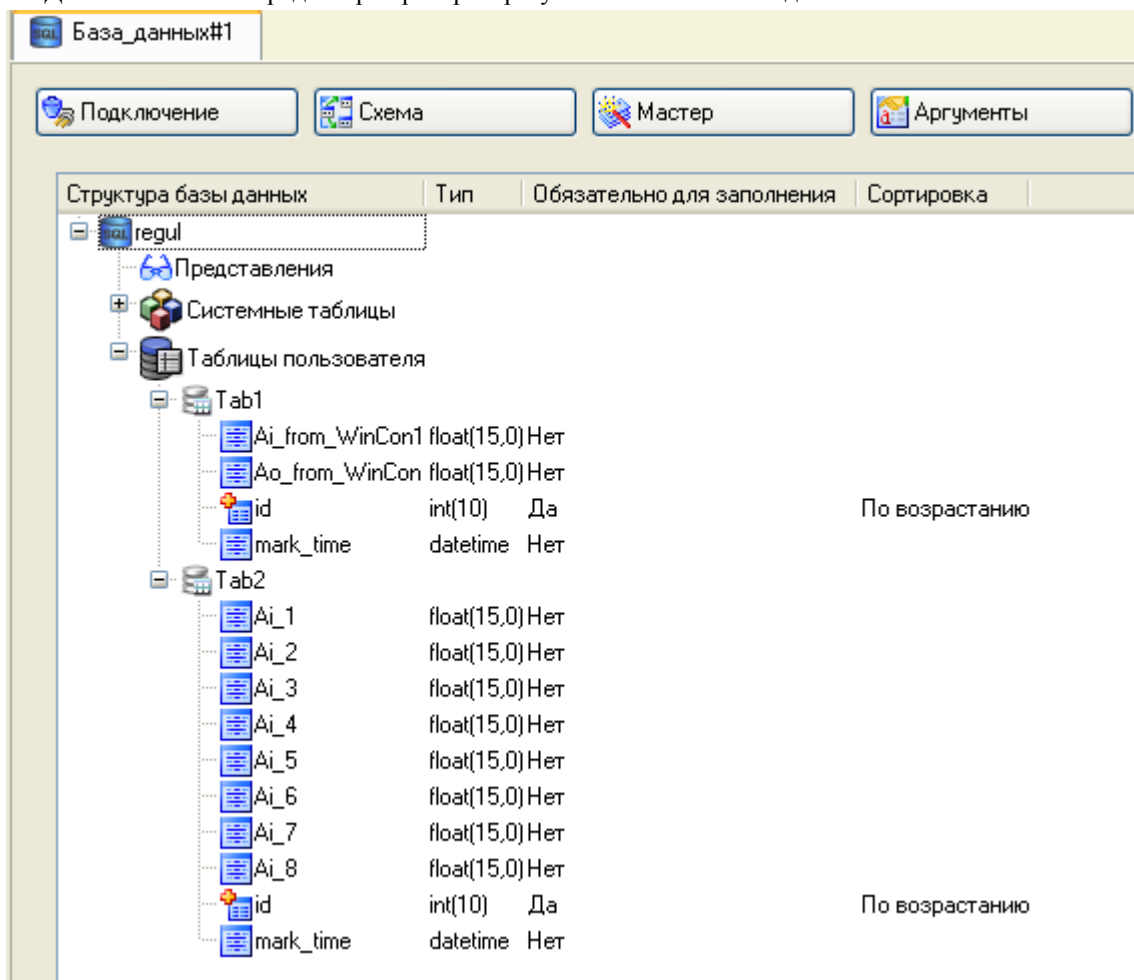
Выполним запрос с помощью иконки  панели инструментов.

Создадим второй запрос, используя возможности редактирования – копирования и вставки:



и также выполним его из интегрированной среды разработки.

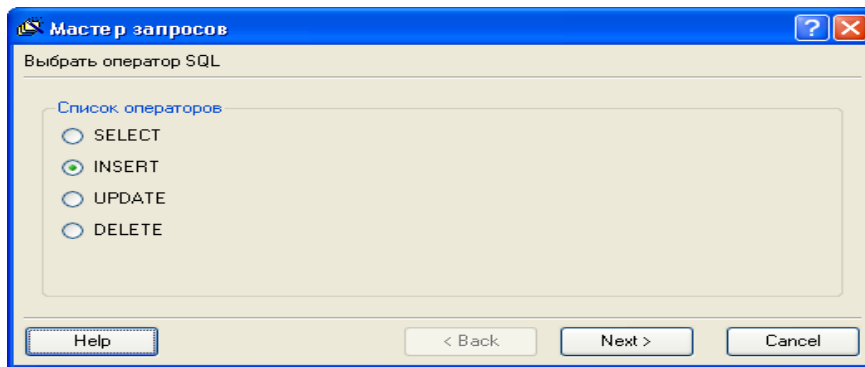
Далее закроем шаблон запроса и откроем вновь, поскольку необходимо выполнить переподключение к БД. В бланке **Схема** редактора проверим результат выполненных действий:



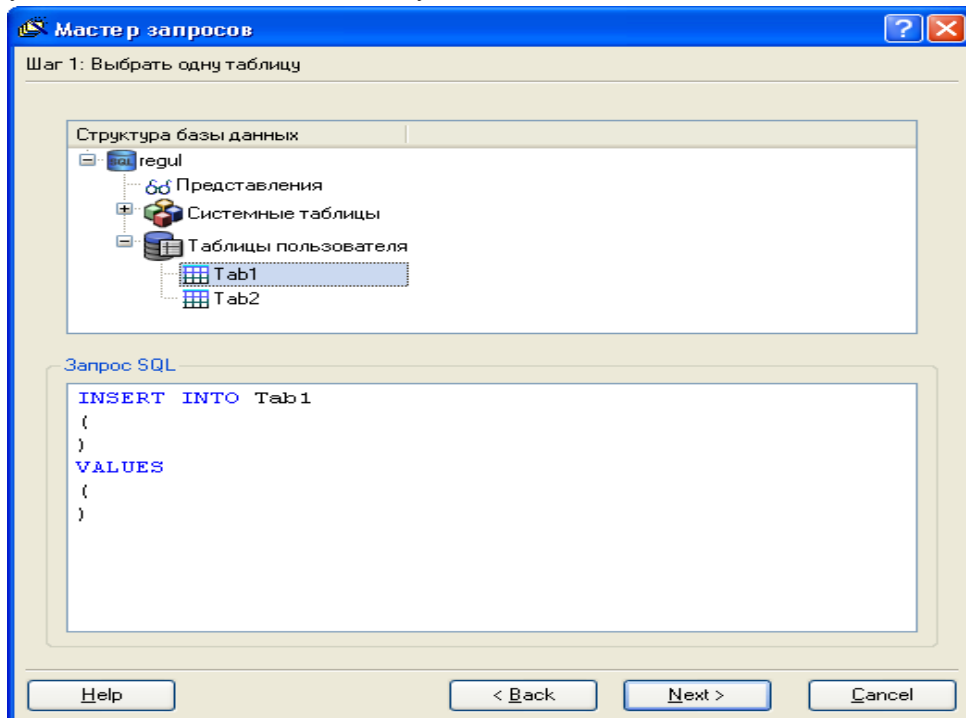
Таким образом, подготовительные работы выполнены и переходим к созданию запросов, помещающих реальные значения каналов в соответствующие поля таблиц базы данных. Разработку запросов будем производить с помощью **Мастера запросов**.

SQL-запрос для заполнения таблицы Tab1

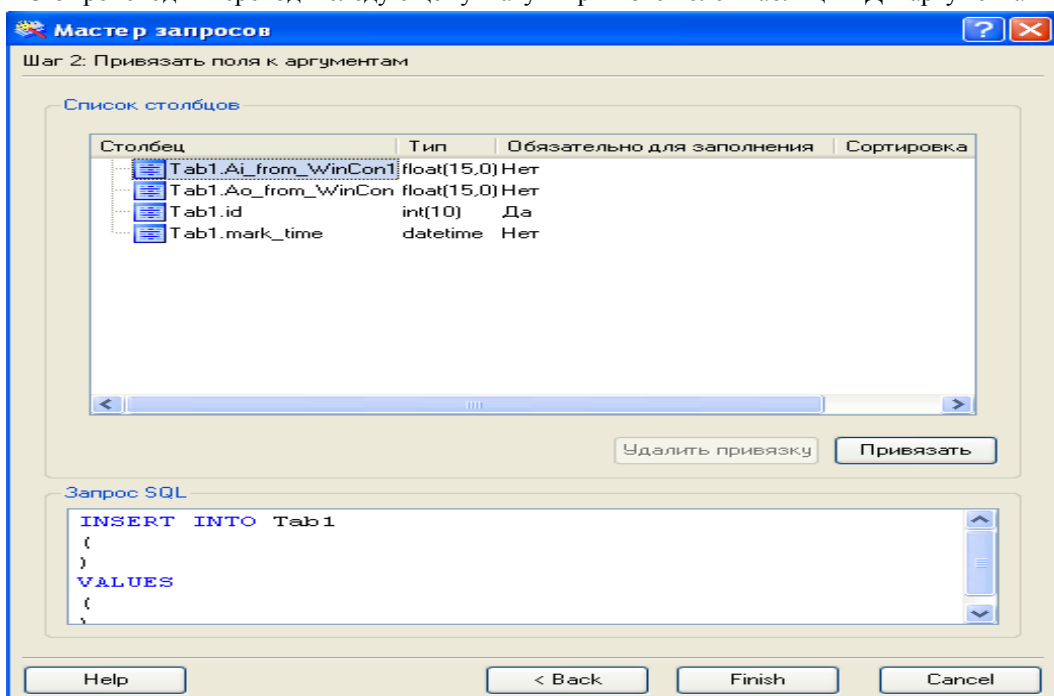
Откроем бланк **Мастер** и выберем из списка операторов SQL **INSERT**:



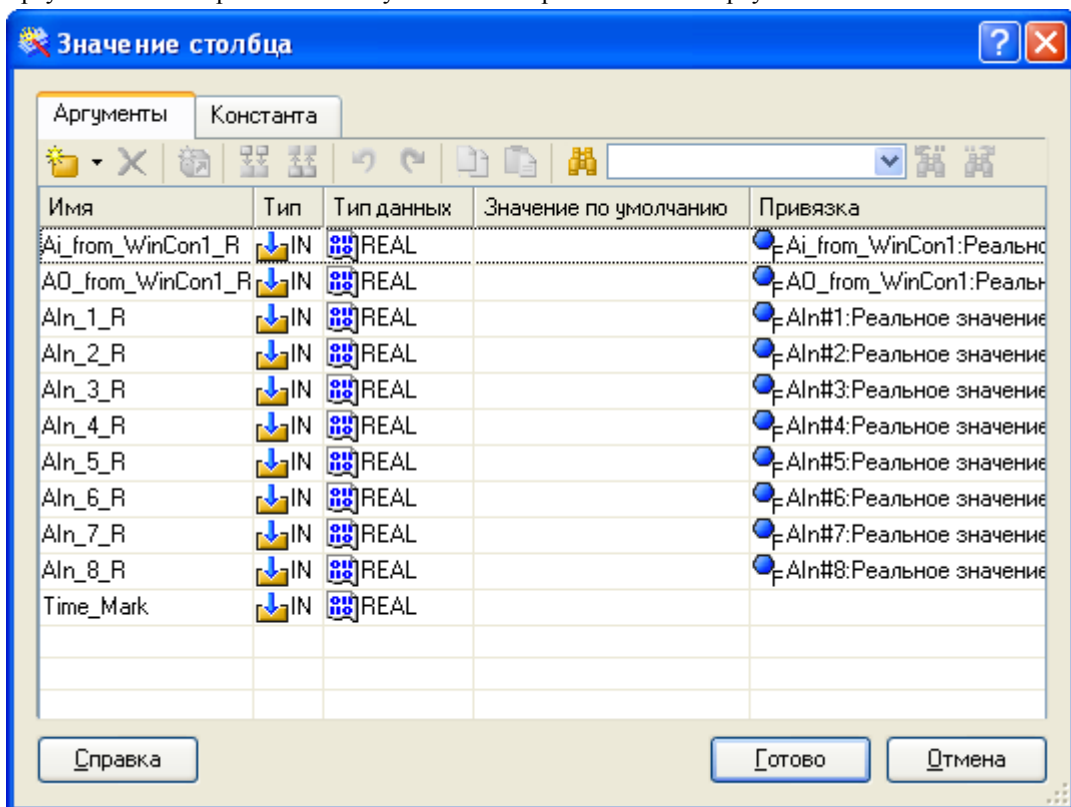
и нажмем клавишу **Next**. В следующем диалоговом окне необходимо указать таблицу БД, в которую будут помещаться данные, в данном случае **Tab1**:



при этом в нижнем окне **Запрос SQL** будет выводиться создаваемый запрос. После нажатия клавиши **Next** происходит переход к следующему шагу – привязке полей таблицы БД к аргументам шаблона.

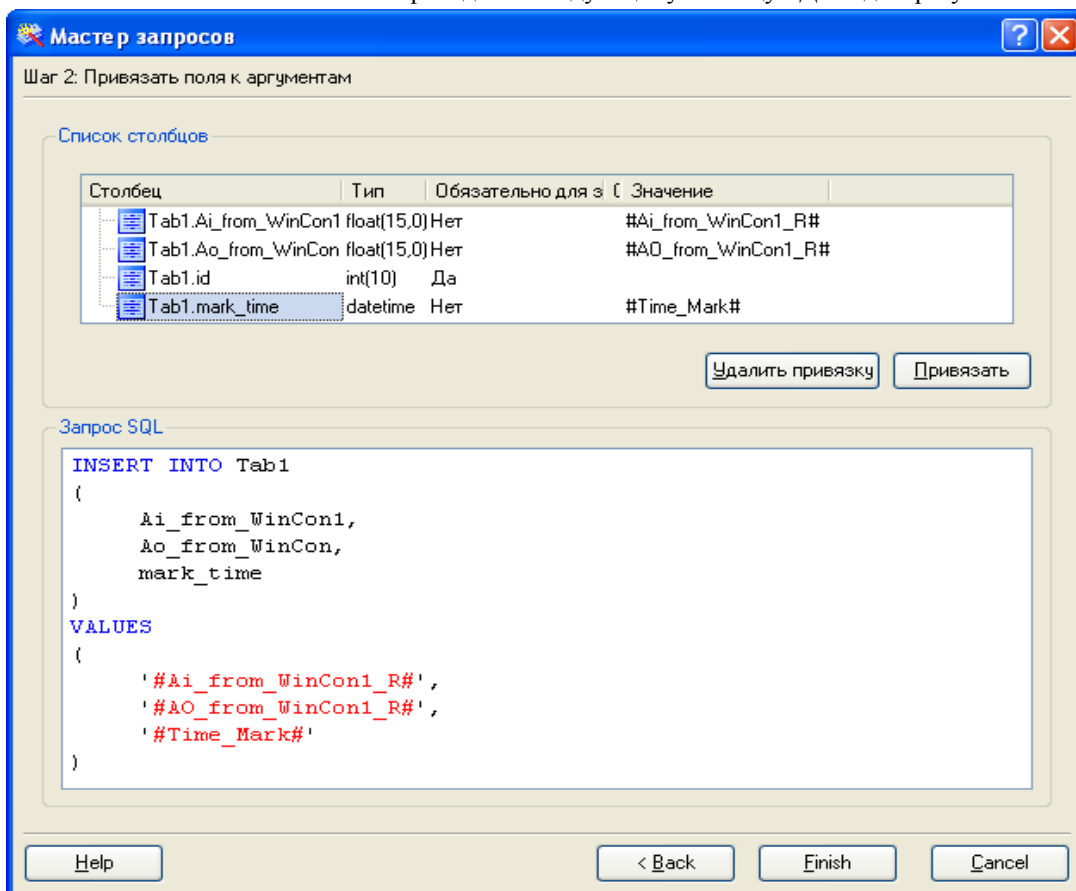


Выделяя ЛК столбец БД по нажатию клавиши **Привязать** открывается окно табличного редактора аргументов в котором также ЛК указывается привязываемый аргумент.



Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Ai_from_WinCon1_R	IN	REAL		#Ai_from_WinCon1:Реальное значение
AO_from_WinCon1_R	IN	REAL		#AO_from_WinCon1:Реальное значение
Aln_1_R	IN	REAL		#Aln#1:Реальное значение
Aln_2_R	IN	REAL		#Aln#2:Реальное значение
Aln_3_R	IN	REAL		#Aln#3:Реальное значение
Aln_4_R	IN	REAL		#Aln#4:Реальное значение
Aln_5_R	IN	REAL		#Aln#5:Реальное значение
Aln_6_R	IN	REAL		#Aln#6:Реальное значение
Aln_7_R	IN	REAL		#Aln#7:Реальное значение
Aln_8_R	IN	REAL		#Aln#8:Реальное значение
Time_Mark	IN	REAL		

После нажатия клавиши **Готово** переходим к следующему столбцу БД и т.д. В результате получим:

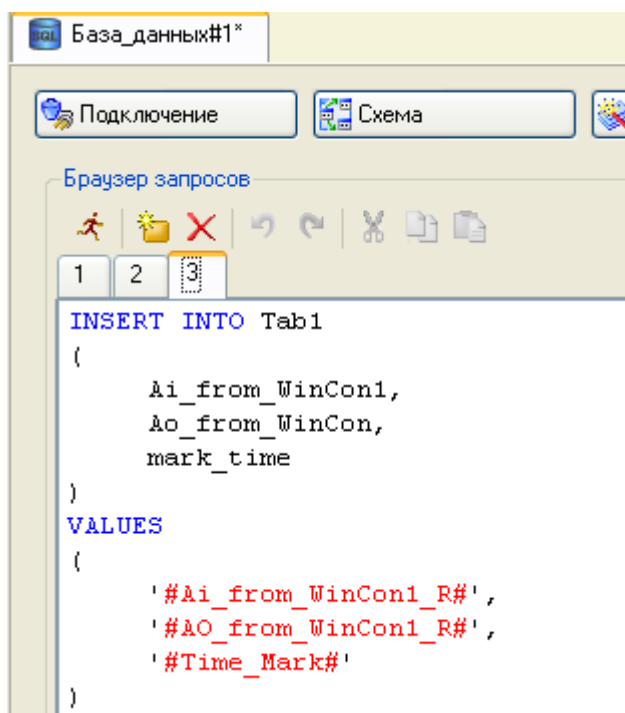


Столбец	Тип	Обязательно для з	Значение
Tab1.Ai_from_WinCon1	float(15,0)	Нет	#Ai_from_WinCon1_R#
Tab1.Ao_from_WinCon1	float(15,0)	Нет	#AO_from_WinCon1_R#
Tab1.id	int(10)	Да	
Tab1.mark_time	datetime	Нет	#Time_Mark#

```

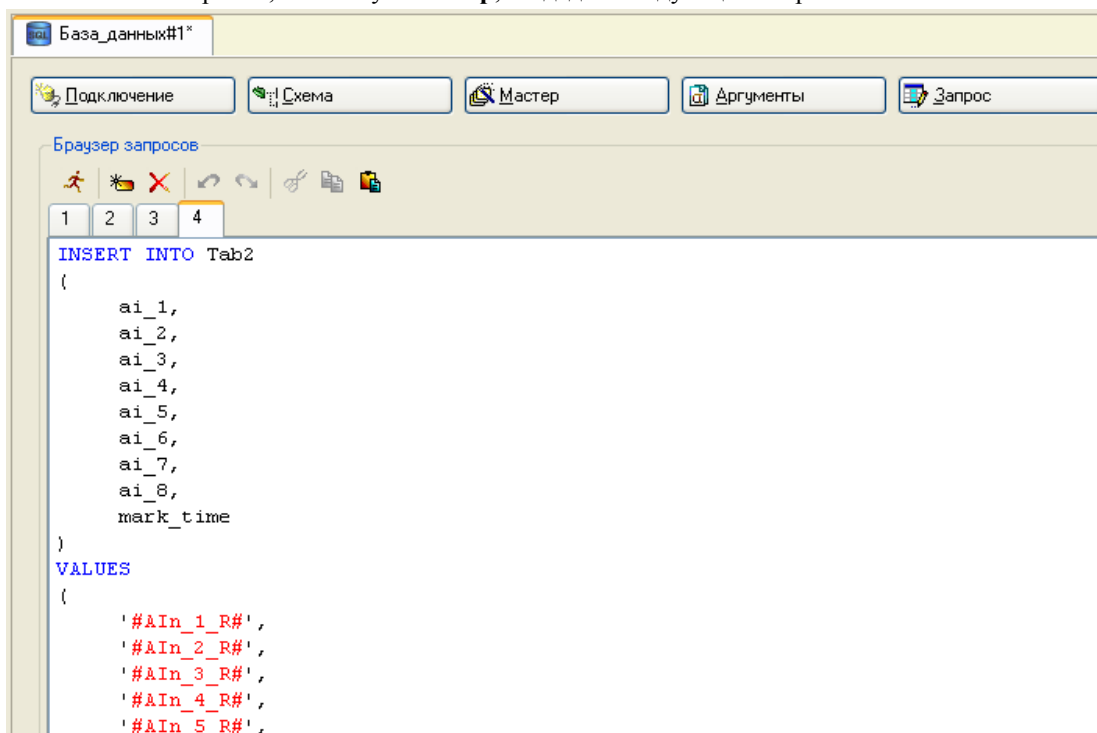
INSERT INTO Tab1
(
    Ai_from_WinCon1,
    Ao_from_WinCon1,
    mark_time
)
VALUES
(
    '#Ai_from_WinCon1_R#',
    '#AO_from_WinCon1_R#',
    '#Time_Mark#'
)
    
```

Завершив создание запроса нажатием клавиши **Finish**. Перейдя в бланк **Запрос** редактора шаблона, наблюдаем, что создан запрос с номером 3:



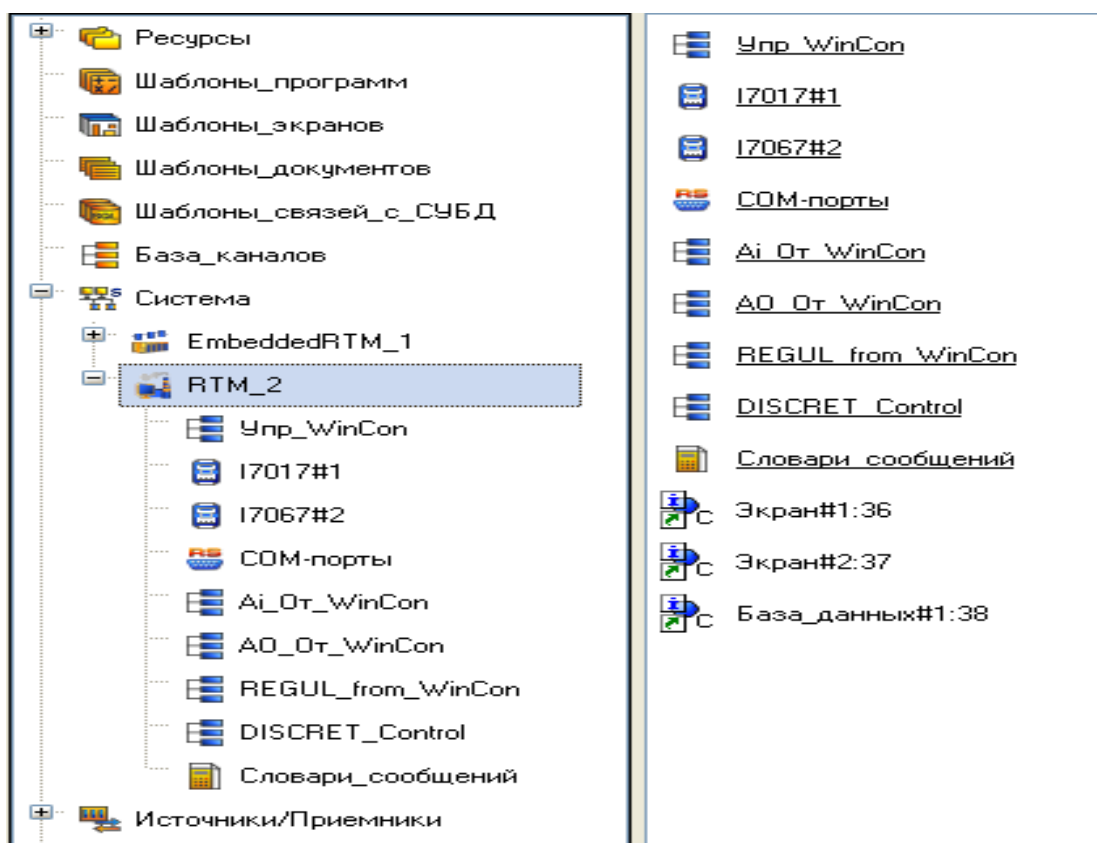
SQL-запрос для заполнения таблицы Tab2

Аналогичным образом, используя **Мастер**, создадим следующий запрос:

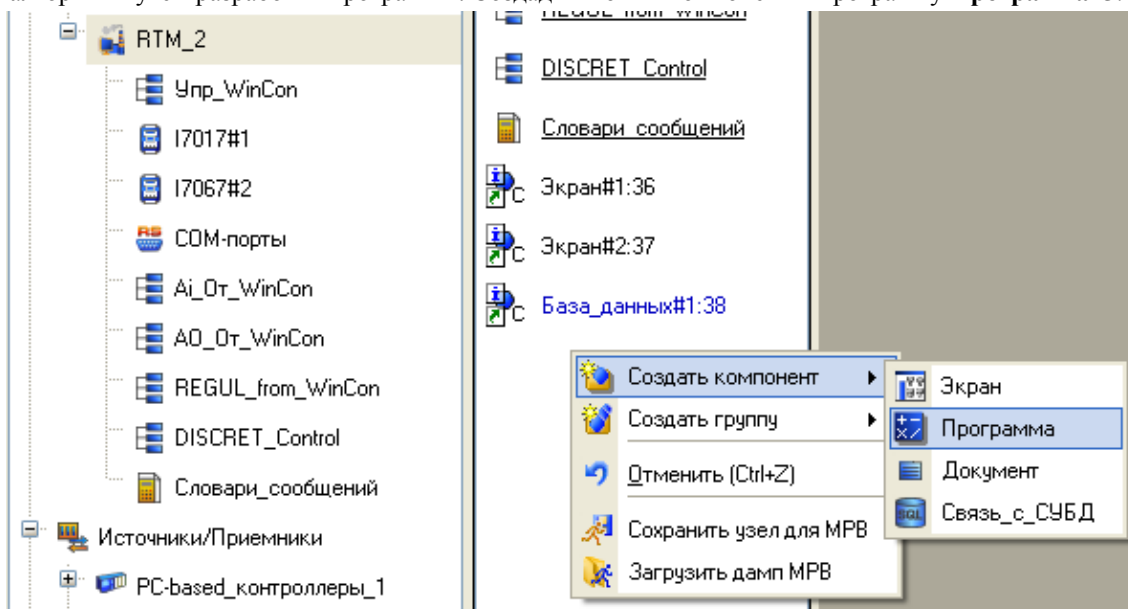


Организация вызова SQL-запросов из узла проекта

Закроем редактор шаблона запроса к СУБД. Выделим ЛК шаблон **База_данных#1** и перетащим его в узел **RTM_2**.



Для выполнения запроса необходимо в атрибут **Входное значение** канала класса **Вызов База_данных#1** подавать номер запроса. Для заполнения таблицы БД **Tab1** – 3, для **Tab2** – 4. Исходя из постановки задачи, определим в качестве существенного изменения для реального значения канала **AI_from_WinCon1** величину в 5% полной шкалы, а для **AO_from_WinCon1** – 15%. Реализуем алгоритм путем разработки программы. Создадим новый компонент – программу **Программа#5**:



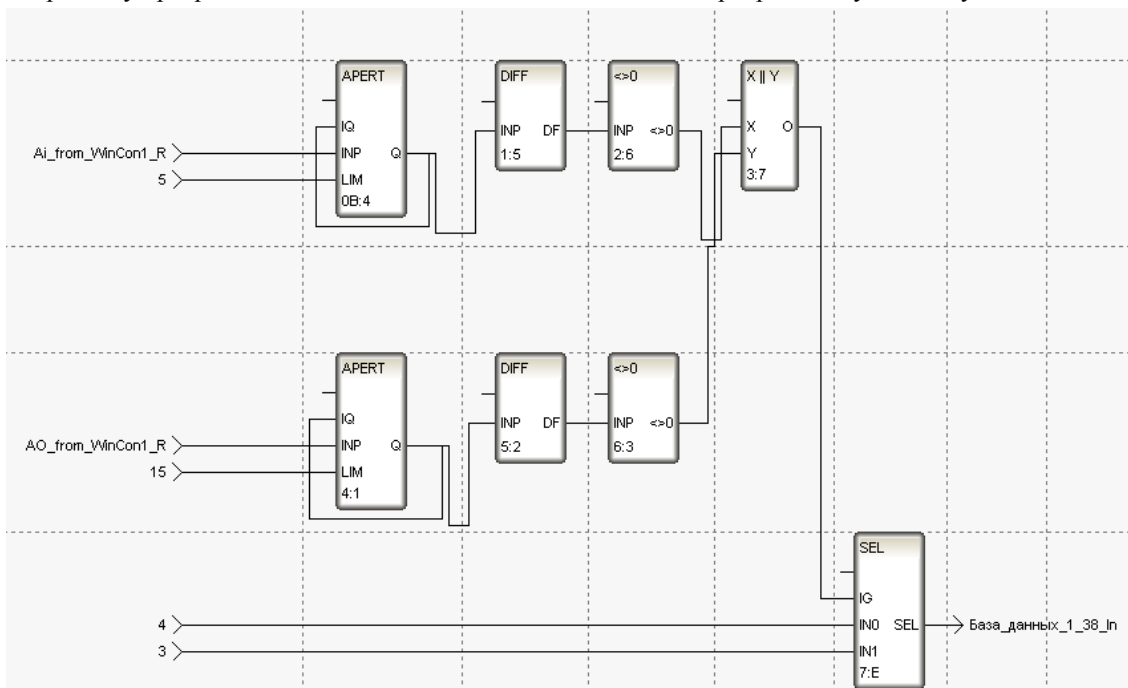
Откроем компонент **Программа#5** на редактирование и перейдем к строке **Аргументы** в структуре программы. Перетащим в окно табличного редактора аргументов каналы **AI_from_WinCon1** и **AO_from_WinCon1**, в результате будут созданы два аргумента программы, привязанные к атрибутам каналов:

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Ai_from_WinCon1_R	↓ IN	REAL		Ai_from_WinCon1:Реальное значение (Система.RTM_2.Ai_От_WinCon)
AO_from_WinCon1_R	↓ IN	REAL		AO_from_WinCon1:Реальное значение (Система.RTM_2.AO_От_WinCon)

С помощью иконки  создадим новый аргумент, изменим его **Тип** на **OUT**, **Тип данных** на **INT**. Выделим ЛК строку, содержащую данный аргумент и перетащим на нее канал класса **Вызов База_данных#1**. Таким образом, список аргументов программы будет:

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Ai_from_WinCon1_R	IN	REAL		Ai_from_WinCon1:Реальное значение (Система.RTM_2.Ai_От_WinCon)
AO_from_WinCon1_R	IN	REAL		AO_from_WinCon1:Реальное значение (Система.RTM_2.AO_От_WinCon)
База_данных_1_38_In	OUT	INT		База_данных#1:38:Входное значение (Система.RTM_2)

Разработку программы выполним на языке **Техно FBD**. Вид программы будет следующим:



Апертура для каналов **AI_from_WinCon1** и **AO_from_WinCon1** реализуется в программе с помощью функциональных блоков **APERT**, изменение значений больше установленного порога фиксируется блоками **DIFF**, а далее с помощью блоков **<=>0** и блока логического сложения **X||Y** производится выбор номера запроса в блоке **SEL**. Таким образом, если изменения значения каналов **AI_from_WinCon1** и **AO_from_WinCon1** не превышают установленных для них порогов, в каждом цикле пересчета базы каналов выполняется запрос **4**, если же значение хоть одного из каналов или обоих одновременно существенно меняются – выполняется запрос **3**.

И, наконец, откроем вкладку **Аргументы** канала класса **Вызов База_данных#1** и выполним привязку аргумента **Time_Mark** к атрибуту **Время изменения** самого же канала **База_данных#1**.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Ai_from_WinCon1_R	IN	REAL		Ai_from_WinCon1:Реальное значение (Система.RTM_2.Ai_От_WinCon)	
AO_from_WinCon1_R	IN	REAL		AO_from_WinCon1:Реальное значение (Система.RTM_2.AO_От_WinCon)	
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)	
База_данных_1_38_T	IN	REAL		База_данных#1:38:Время изменения (Система.RTM_2)	

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

Лабораторная работа № 2.

Создание шаблона документа с заданными параметрами

1. Цель работы

Создать шаблон документа, включающий дату/время, логотип TRACE MODE, таблицу с мгновенными значениями, тренд и таблицу с выводом архивных значений каналов, принимающих данные от контроллера и модуля удаленного ввода.

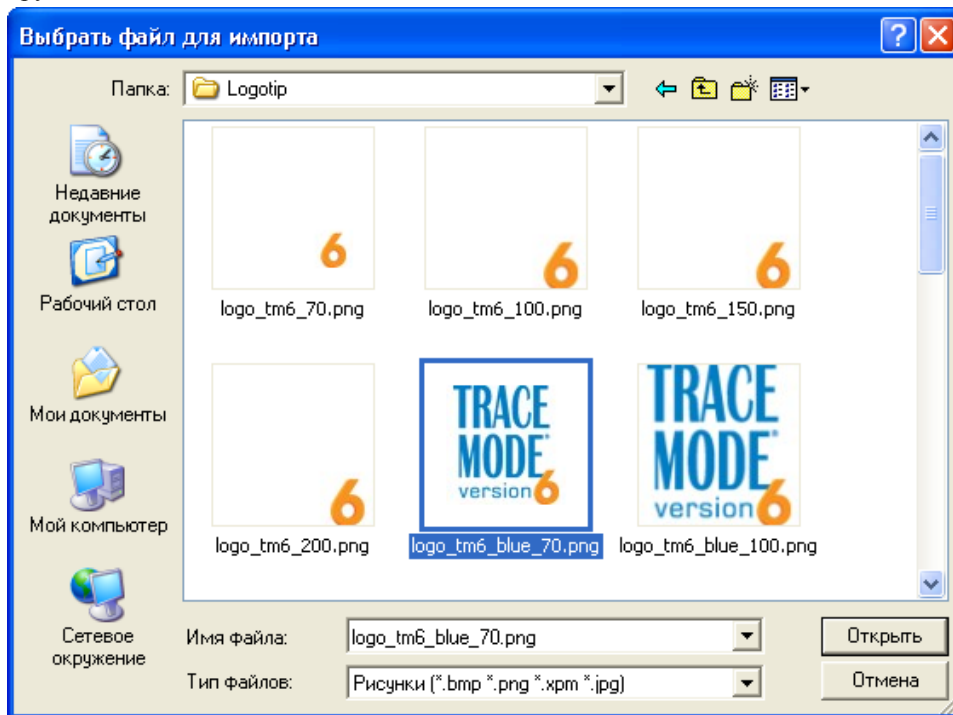
2. Порядок выполнения работы

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

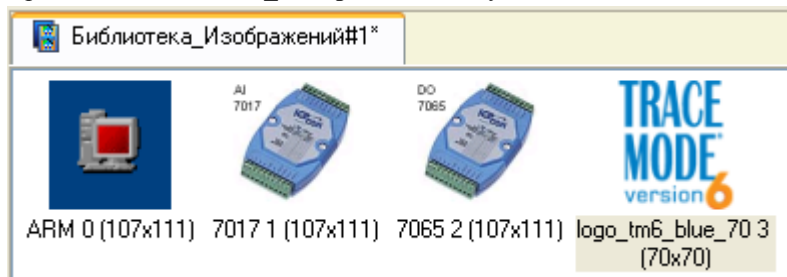
3. Образец выполнения работы

Создание шаблона документа

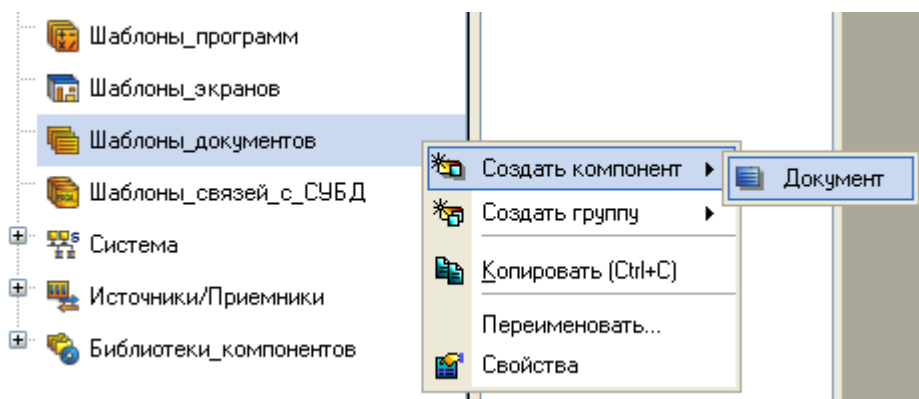
Дополним созданную на Занятии 4 ресурсную библиотеку **Библиотека_Изображений#1** логотипом TRACE MODE из bmp-файлов, размещенных в папке Lib в директории Logotip. Для этого откроем библиотеку и выполним операцию импорта с помощью иконки , размещенной на панели инструментов:



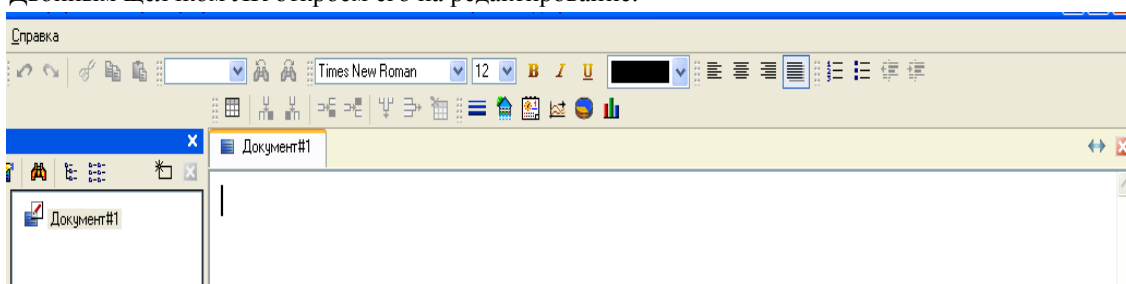
Содержимое **Библиотека_Изображений#1** будет:



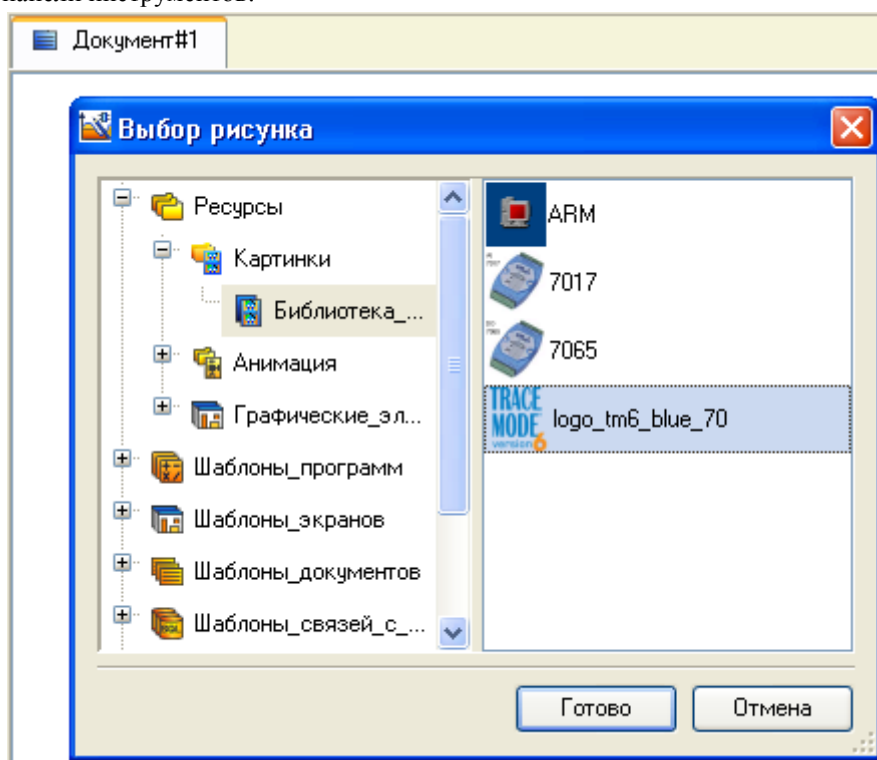
Выберем в дереве проекта слой **Шаблоны_документов** и создадим новый компонент – Документ#1:



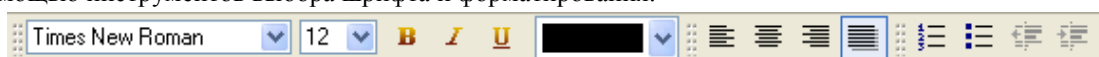
Двойным щелчком ЛК откроем его на редактирование:



Разместим в левом верхнем углу документа логотип TRACE MODE, воспользовавшись иконкой  на панели инструментов:



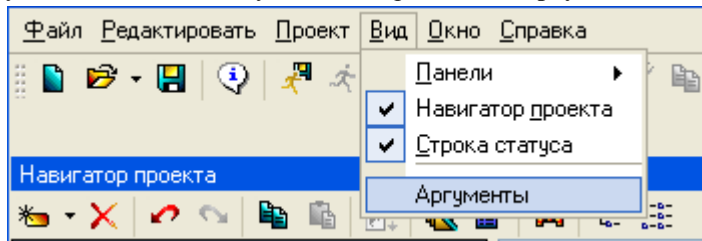
Нажмем Enter и в следующей строке введем надпись «Проект REGUL – сводка», отформатировав ее с помощью инструментов выбора шрифта и форматирования:



получив в итоге:

Проект REGUL - сводка

Выведем в табличной форме мгновенные значения (на момент генерации документа) параметров регулятора, работающего в узле контроллера - Кп, Ки, Кд, Задание и реакция объекта AI_from_WinCon1 из групп компонентов REGUL_from_WinCon и От_WinCon узла RTM_2, а также напряжений, подаваемых на входы удаленного модуля аналогового ввода I-7017 – AIn#1...Ain#8 из группы компонентов I7017#1 узла RTM_2 (всего 13 аргументов). Флаг в настройках ИС **Переименовывать аргументы при привязке** должен быть установлен. Для формирования списка аргументов шаблона Документ#1 откроем окно **Аргументы** из пункта **Вид** основного меню ИС:



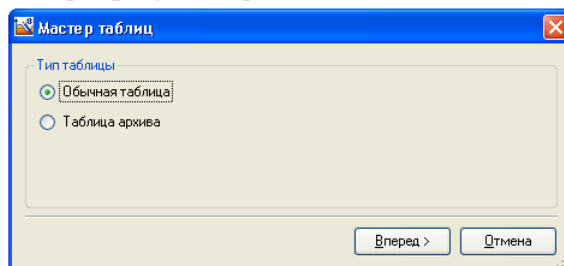
Воспользовавшись механизмом drag'n'drop, создадим необходимый список аргументов как показано:

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Задание_R	IN	REAL		Задание:Реальное значение (Система.RTM_2.REGUL_from_WinCon)
Кп_R	IN	REAL		Кп:Реальное значение (Система.RTM_2.REGUL_from_WinCon)
Ки_R	IN	REAL		Ки:Реальное значение (Система.RTM_2.REGUL_from_WinCon)
Кд_R	IN	REAL		Кд:Реальное значение (Система.RTM_2.REGUL_from_WinCon)
Ai_from_WinCon1_R	IN	REAL		Ai_from_WinCon1:Реальное значение (Система.RTM_2.Ai_От_WinCon)
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)

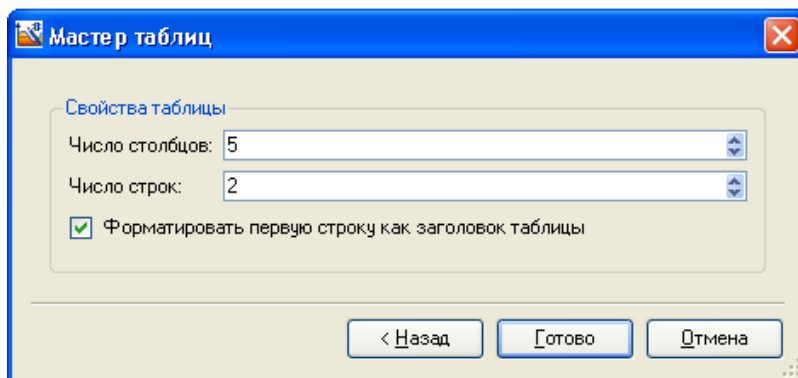
Выполним надпись в документе «Параметры регулятора:», нажмем **Enter** и с помощью иконки  на панели инструментов вставим в документ таблицу, выбрав в открывшемся диалоге **Обычная таблица**:

Проект REGUL - сводка

Параметры регулятора:



Зададим для таблицы следующие параметры:



Мастер таблиц

Свойства таблицы

Число столбцов: 5

Число строк: 2

☒ Форматировать первую строку как заголовок таблицы

< Назад Готово Отмена

и нажмем экранную кнопку **Готово**.

В первой строке таблицы выполним надписи – Кп, Ки, Кд, Задание и Реакция объекта. С помощью механизма drag'n'drop перетащим во вторую строку таблицы соответствующие аргументы из открытого окна **Аргументы**:



Проект REGUL - сводка

Параметры регулятора:

Кп	Ки	Кд	Задание	Реакция объекта
@Кп_R	@Ки_R	@Кд_R	@Задание_R	@Ai_from_WinCon1_R

Выполним надпись «Напряжения на входах модуля I-7017, В» и вставим таблицу с 8 столбцами и двумя строками, выполнив надписи и привязав к ячейкам таблицы аргументы шаблона документа, как показано ниже:

Напряжение на входах модуля I-7017, В:

Вх.1	Вх.2	Вх.3	Вх.4	Вх.5	Вх.6	Вх.7	Вх.8
@AIn_1_R	@AIn_2_R	@AIn_3_R	@AIn_4_R	@AIn_5_R	@AIn_6_R	@AIn_7_R	@AIn_8_R

Для разделения секций мгновенных значений от архивных, воспользуемся горизонтальной линией при помощи иконки  на панели инструментов и нажмем **Enter**.

Для определения временного интервала при организации выборки архивных данных, помещаемых в документ, воспользуемся свойствами ГЭ Тренд, расположенного на графическом экране Экран#1. Вызовем на редактирование Экран#1 и откроем свойства ГЭ Тренд. Двойным щелчком ЛК откроем в основном бланке строку **Ось времени** и свяжем атрибуты Левая граница и Правая граница с вновь созданными аргументами экрана L_Mark и R_Mark, имеющих тип **IN/OUT** и тип данных **UDINT**:

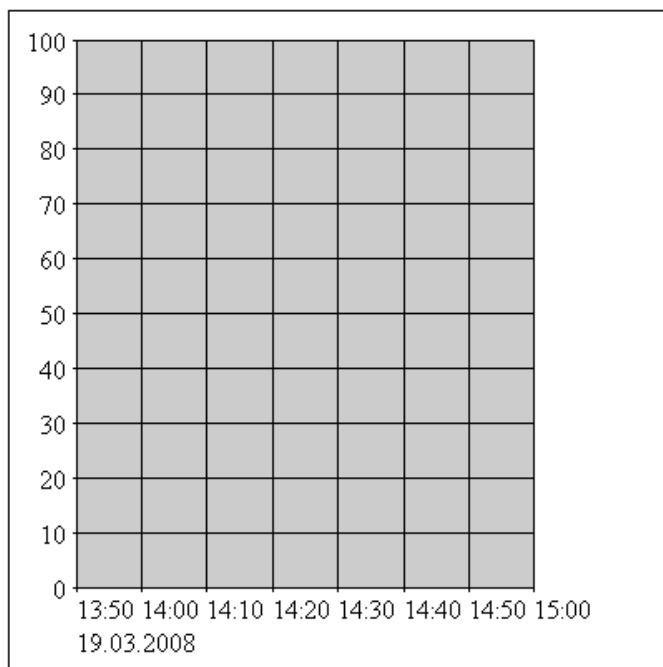
UDINT и в поле **Привязка** свяжем их с аргументами шаблона экрана Экран#1 L_Mark и R_Mark:

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Задание_R		REAL		Задание:Реальное значение (Система.RTM_2.REGUL_from_WinCon)
Kn_R		REAL		Kn:Реальное значение (Система.RTM_2.REGUL_from_WinCon)
Ki_R		REAL		Ki:Реальное значение (Система.RTM_2.REGUL_from_WinCon)
Kd_R		REAL		Kd:Реальное значение (Система.RTM_2.REGUL_from_WinCon)
Ai_from_WinCon1_R		REAL		Ai_from_WinCon1:Реальное значение (Система.RTM_2.Ai_От_WinCon)
Aln_1_R		REAL		Aln#1:Реальное значение (Система.RTM_2.I7017#1)
Aln_2_R		REAL		Aln#2:Реальное значение (Система.RTM_2.I7017#1)
Aln_3_R		REAL		Aln#3:Реальное значение (Система.RTM_2.I7017#1)
Aln_4_R		REAL		Aln#4:Реальное значение (Система.RTM_2.I7017#1)
Aln_5_R		REAL		Aln#5:Реальное значение (Система.RTM_2.I7017#1)
Aln_6_R		REAL		Aln#6:Реальное значение (Система.RTM_2.I7017#1)
Aln_7_R		REAL		Aln#7:Реальное значение (Система.RTM_2.I7017#1)
Aln_8_R		REAL		Aln#8:Реальное значение (Система.RTM_2.I7017#1)
From		UDINT		Экран#1:36:L_Mark (Система.RTM_2)
To		UDINT		Экран#1:36:R_Mark (Система.RTM_2)

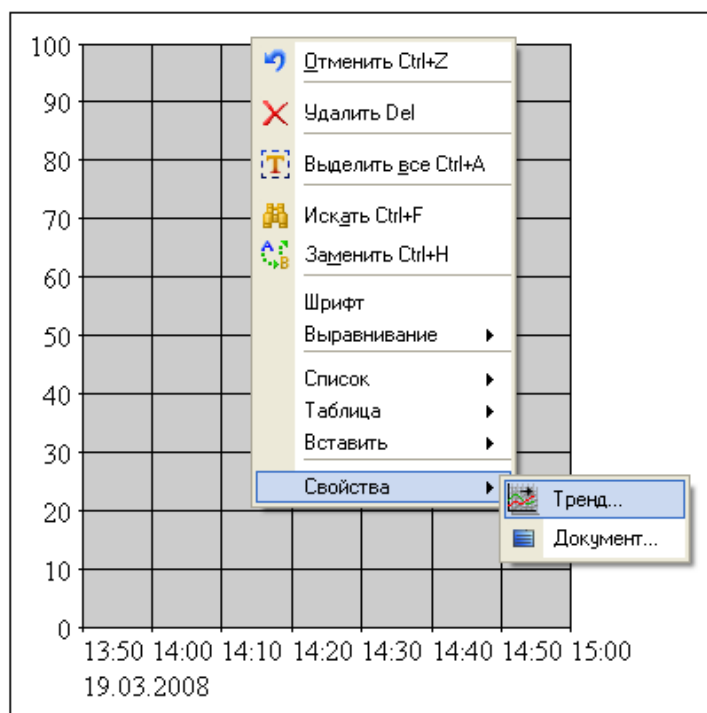
Через аргументы **From** и **To** будет передаваться в шаблон документа Документ#1 временной интервал выборки из архива, соответствующий интервалу, отображающемуся на тренде Экрана#1.

Вставим в документ архивный тренд с помощью иконки на панели инструментов:

<i>Bx.1</i>	<i>Bx.2</i>	<i>Bx.3</i>	<i>Bx.4</i>
@Aln_1_R	@Aln_2_R	@Aln_3_R	@Aln_4_R



и через контекстное меню, вызываемое по щелчку ПК, перейдем к свойствам тренда.



В бланке **Аргументы** с помощью экранной кнопки **Добавить** определим отображение трех кривых для аргументов, связанных с каналами Задание, AI_from_WinCon и Ain#6, как показано:

Свойства тренда

Аргументы Архив Вид Оси Колонтитулы Легенда

Имя	Значение	Фильтр	Цвет
@Ai_from_WinCon1_R.B_NAME	@Ai_from_WinCon1_R.R		Blue
@AIn_6_R.B_NAME	@AIn_6_R.R		Green
@Задание_R.B_NAME	@Задание_R.R		Red

Добавить Удалить Вверх Вниз

Имя
Аргумент: @Задание_R Атрибут: B_NAME

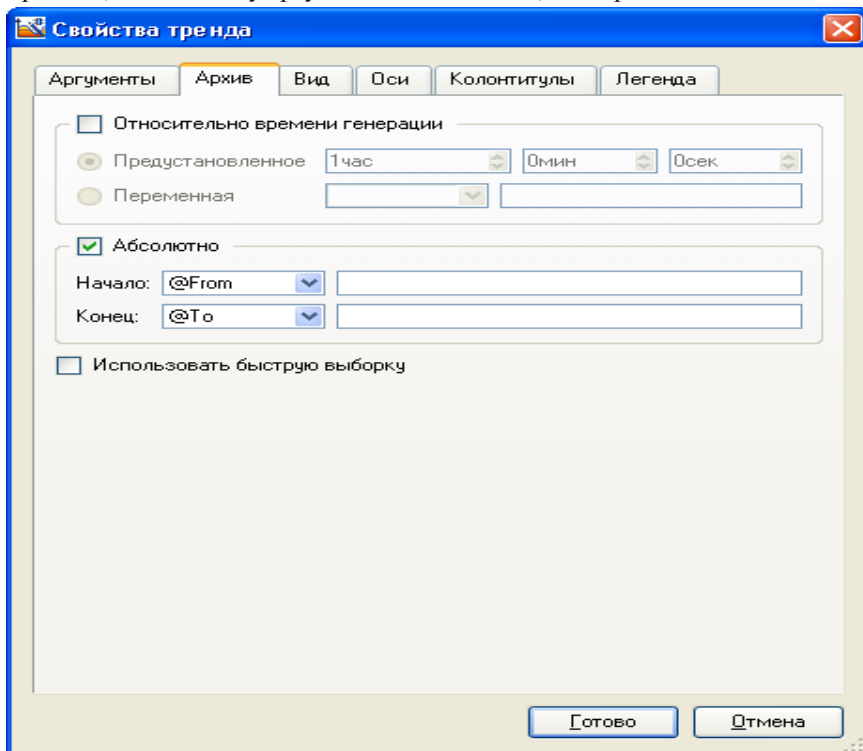
Значение
Аргумент: @Задание_R Атрибут: R

Фильтр
Аргумент: Атрибут: X

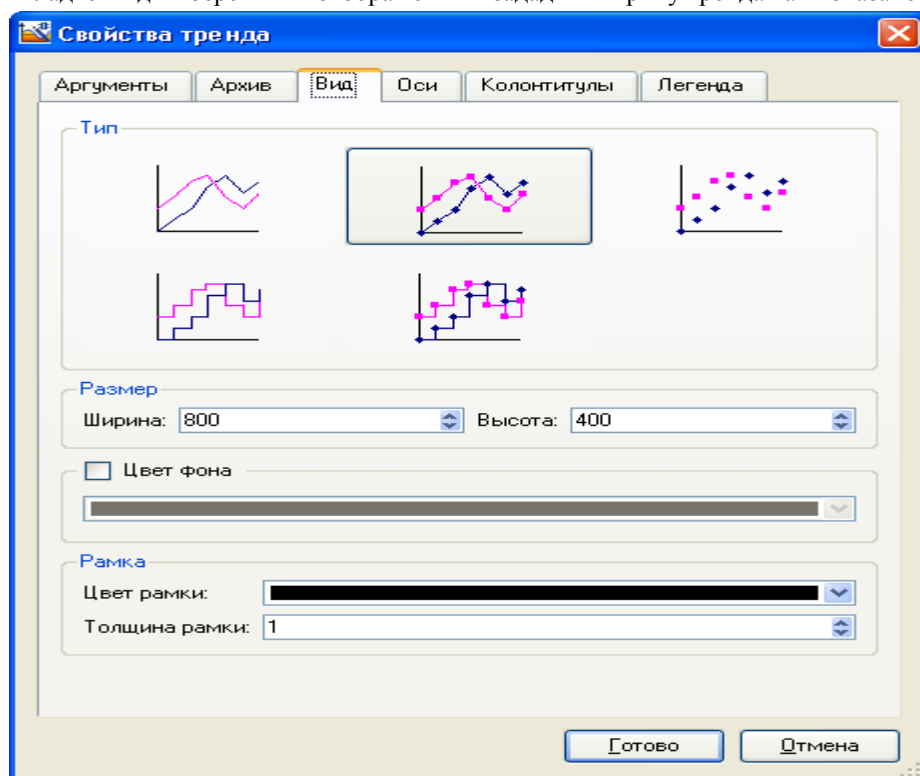
Цвет
[Color Selection Box]

Готово Отмена

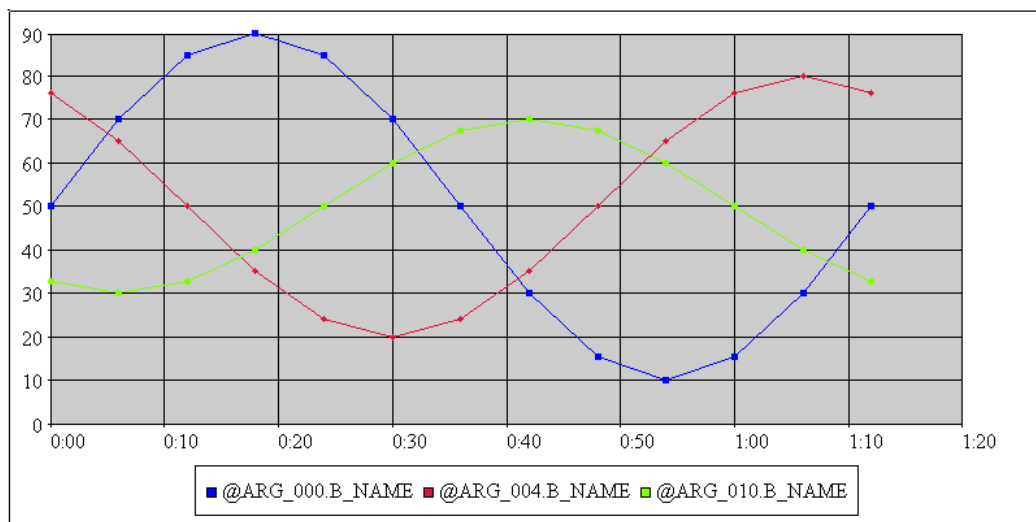
Выберем в свойствах тренда вкладку **Архив** и выполним привязку интервала выборки к абсолютному времени, задаваемому аргументами From и To, сняв флаг **Использовать быструю выборку**:



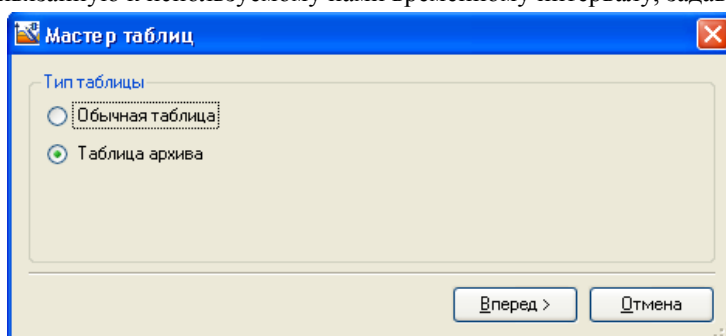
Во вкладке **Вид** выберем тип отображения и зададим ширину тренда как показано ниже:



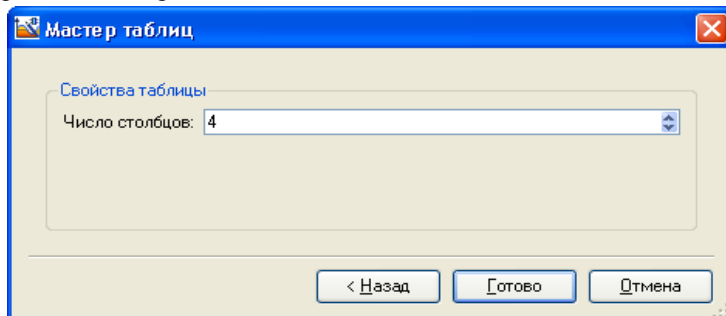
Конфигурацию тренда, задаваемую во вкладках **Оси**, **Колонтитулы** и **Легенда** оставим заданными по умолчанию и нажмем экранную кнопку **Готово**. В шаблоне документа Документ#1 будем иметь:



Для численного представления данных, отображаемых на тренде, ниже поместим архивную таблицу, привязанную к используемому нами временному интервалу, задаваемому аргументами **From** и **To**:



Определим четыре столбца в таблице:



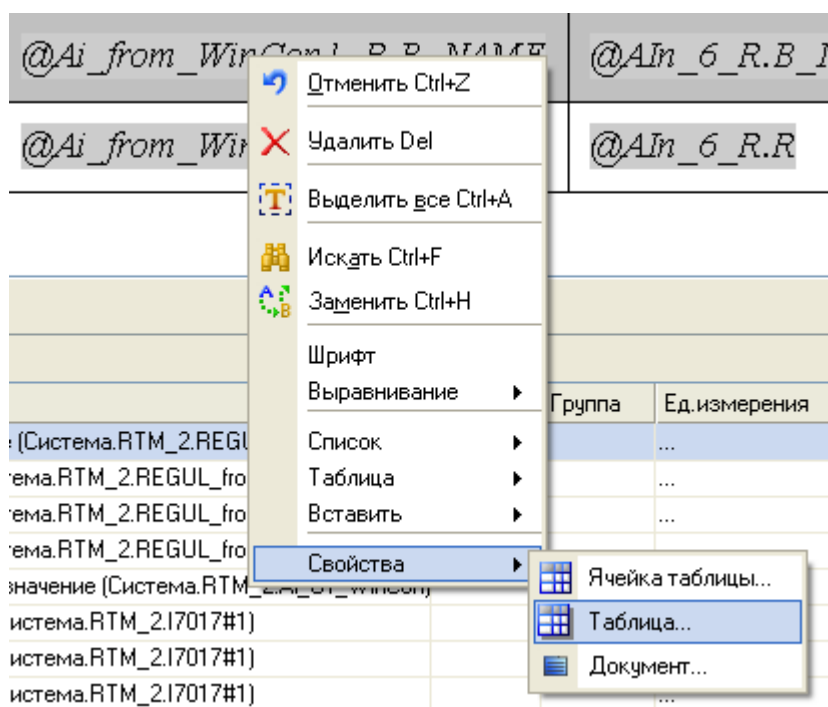
и нажмем кнопку **Готово**, после чего будем иметь следующее:

Время			
0:00:00			

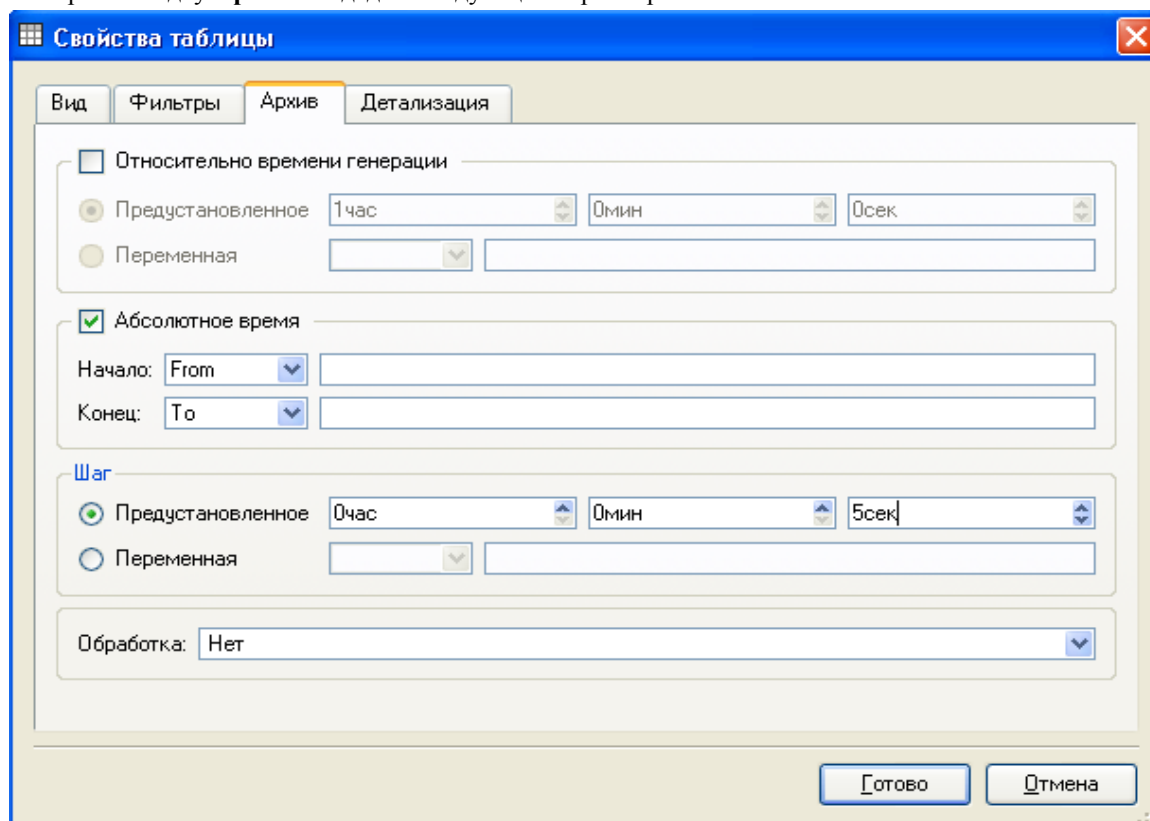
С помощью drag'n'drop перетащим в ячейки таблицы аргументы *Ai_from_WinCon_R*, *Ain_6_R* и *Задание_R* и через разделитель "." укажем в **явном** виде атрибуты каналов – **Базовое имя** и **Реальное значение**, выводимых из архива как показано ниже:

Время	@Ai_from_WinCon1_R.B_NAME	@Ain_6_R.B_NAME	@Задание_R.B_NAME
0:00:00	@Ai_from_WinCon1_R.R	@Ain_6_R.R	@Задание_R.R

Через контекстное меню откроем свойства архивной таблицы:



выберем вкладку **Архив** и зададим следующие параметры:



и нажмем экранную кнопку **Готово**.

Лабораторная работа № 3.

Генерация документов

1. Цель работы

Дополнить проект, созданный в лабораторной работе № 2, новым экраном, включающим ГЭ для ручной генерации документа и средств его просмотра.

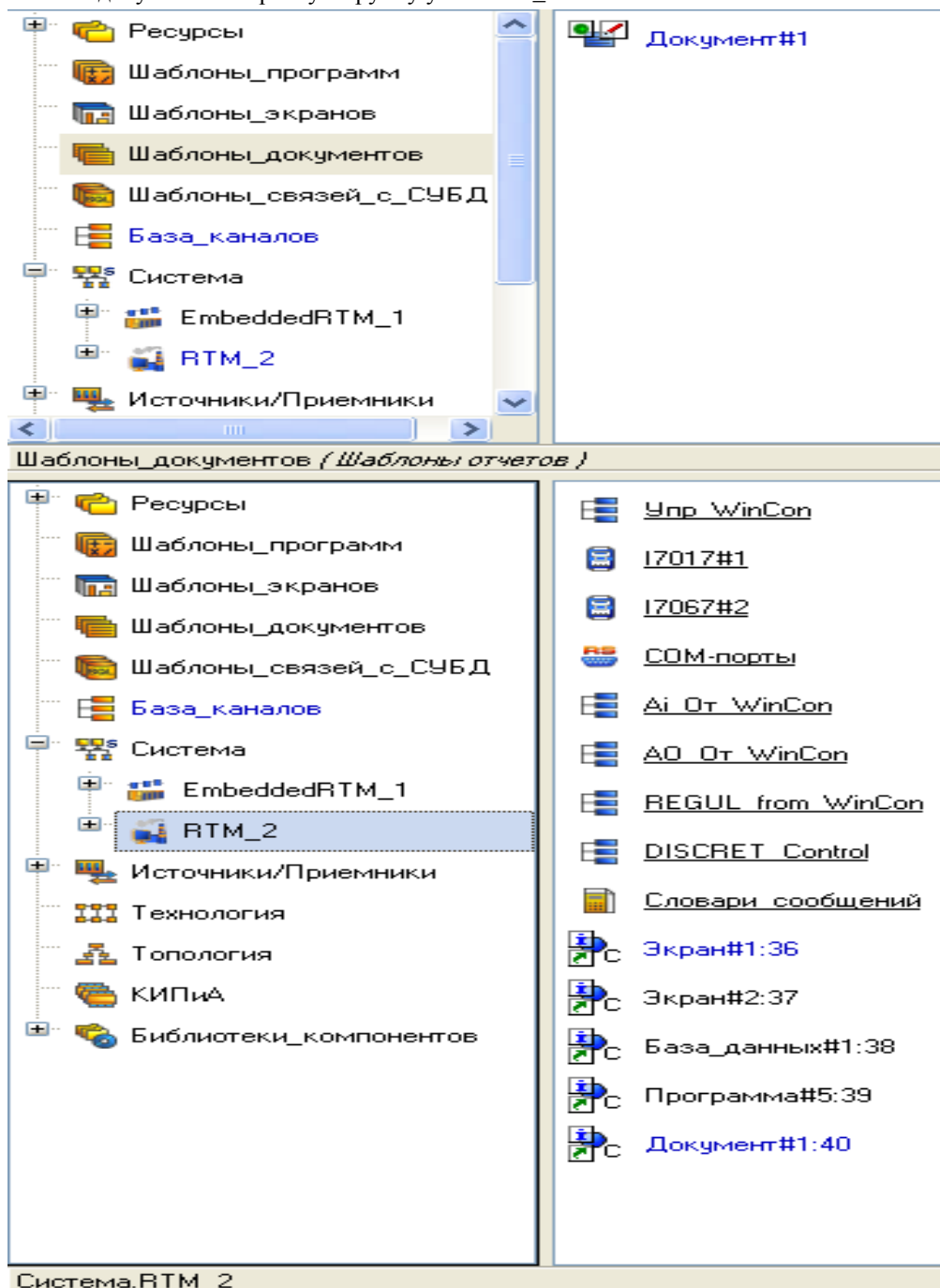
2. Порядок выполнения работы

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

3. Образец выполнения работы

Создание канала класса Вызов в узле RTM_2

Для выполнения генерации подготовленного шаблона документа Документ#1 в узле RTM_2 выполним следующие действия – откроем дополнительное окно Навигатора проекта и перетащим шаблон документа в корневую группу узла RTM_2:

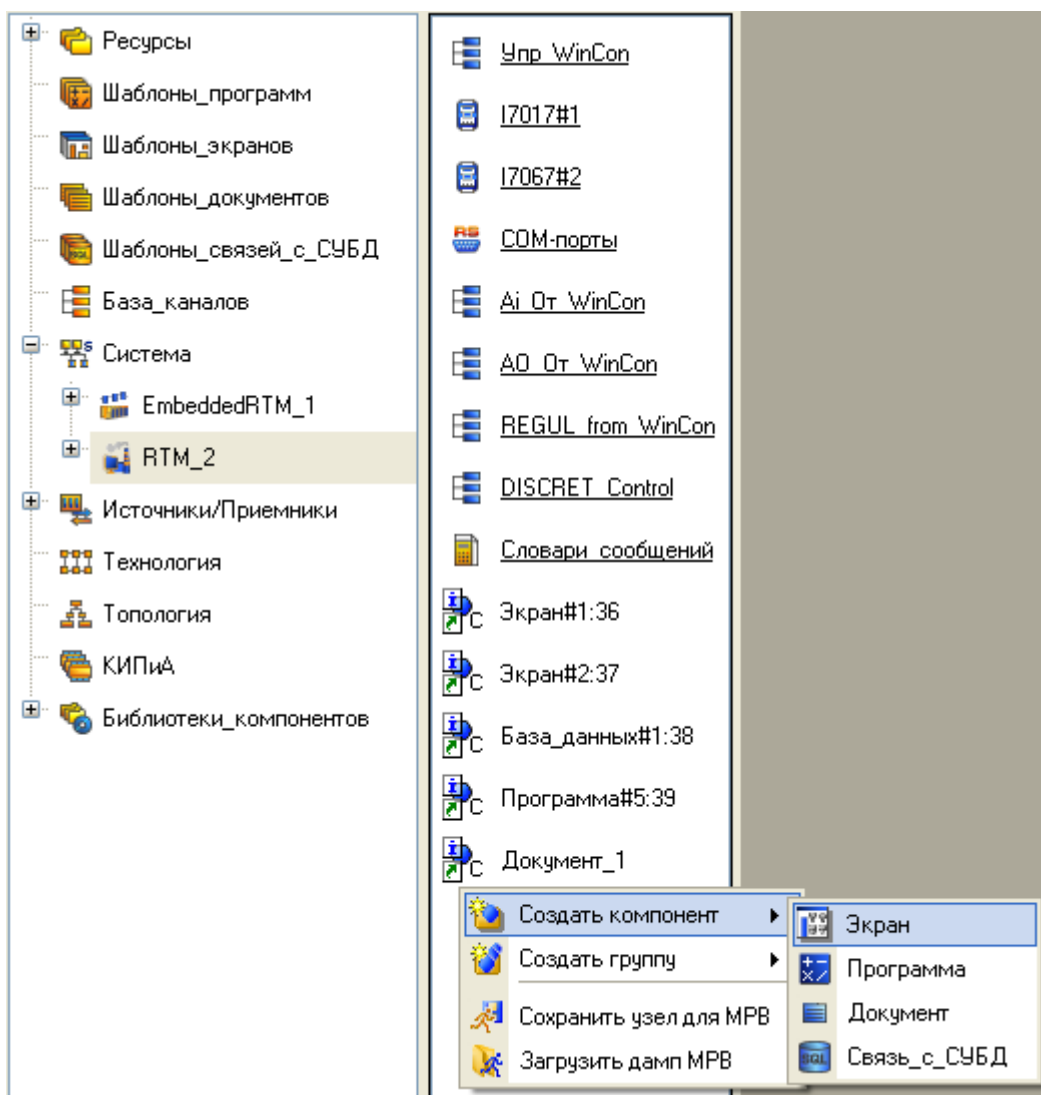


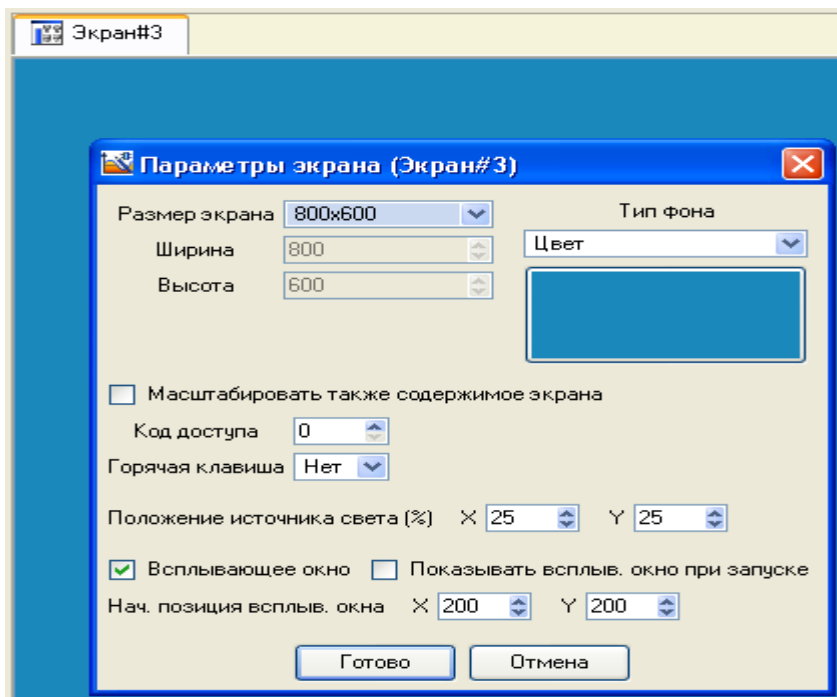
Изменим имя компонента - вновь созданного канала класса Вызов Документ#1:40 на **Документ_1**, затем откроем через контекстное меню его на редактирование и выставим **Параметр** равным 1:

The screenshot shows a configuration window for a component named 'Документ_1'. It has a 'Имя' (Name) field with 'Документ_1' and an empty 'Комментарий' (Comment) field. Under the 'Параметры' (Parameters) section, 'Тип вызова' (Call type) is set to 'Document(Report)', 'Параметр' (Parameter) is set to '1', and 'Глубина выборки' (Sampling depth) is set to 'текущий час' (current hour).

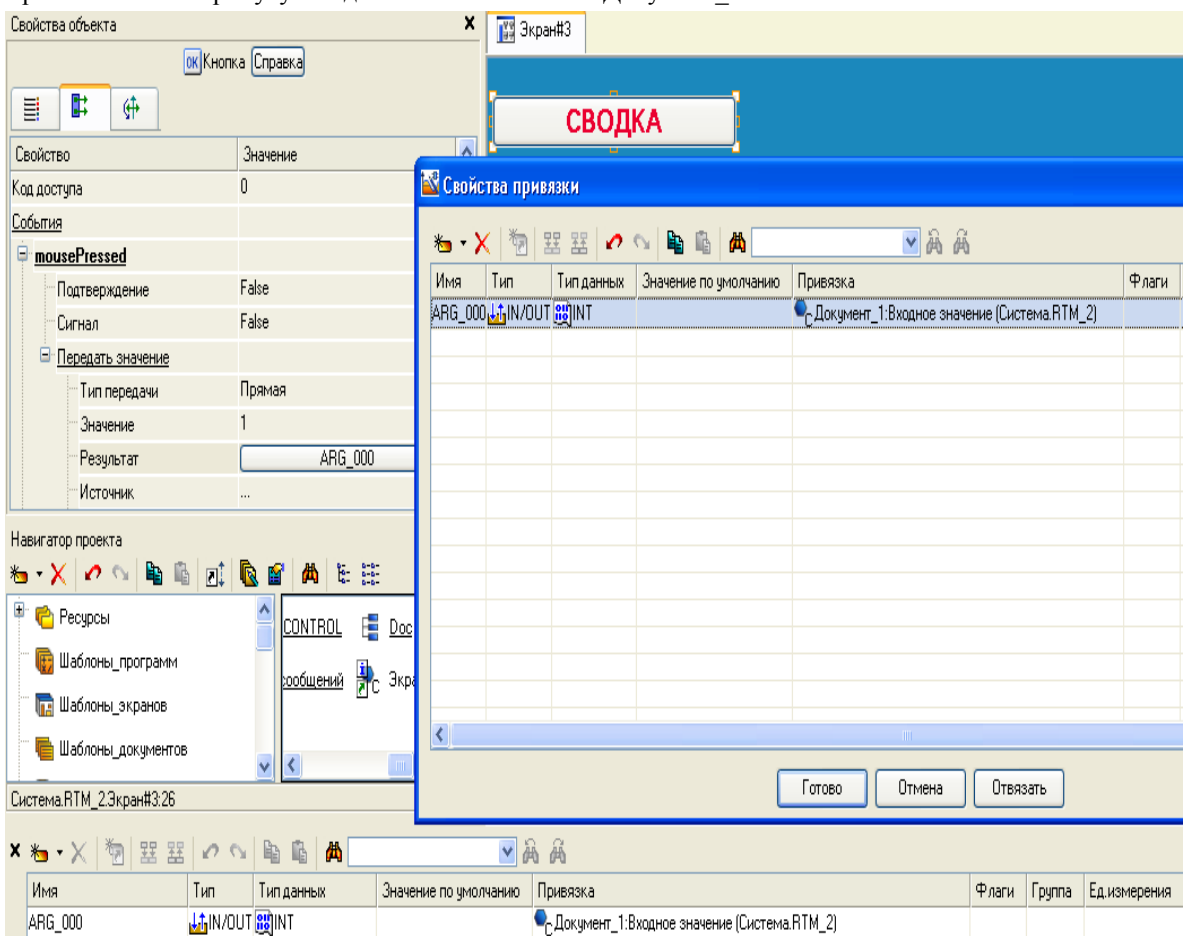
В этом случае генерируемый документ будет записываться в файл в формате **html** в папку узла с именем **Документ_1**, причем вновь сгенерированный документ будет дописываться в конец этого файла в виде секции, а ссылка на данный файл будет содержаться в созданном файле **index.html**. Для выполнения генерации документа необходимо подавать в атрибут **Входное значение** канала класса Вызов Документ_1 значение 1. Данное значение будем формировать с помощью ГЭ Кнопка на вновь разработанном графическом экране, а для просмотра содержимого сводки применим ГЭ Текст из файла. Причем, для обновления содержимого файла на ГЭ Текст из файла применим механизм перепривязки данного ГЭ с помощью ГЭ Переключатель каналов.

Создадим в корневой группе узла RTM_2 новый компонент – Экран#3 и определим его как всплывающий экран, который будем вызывать с экрана Экран#1 с помощью ГЭ Кнопка:

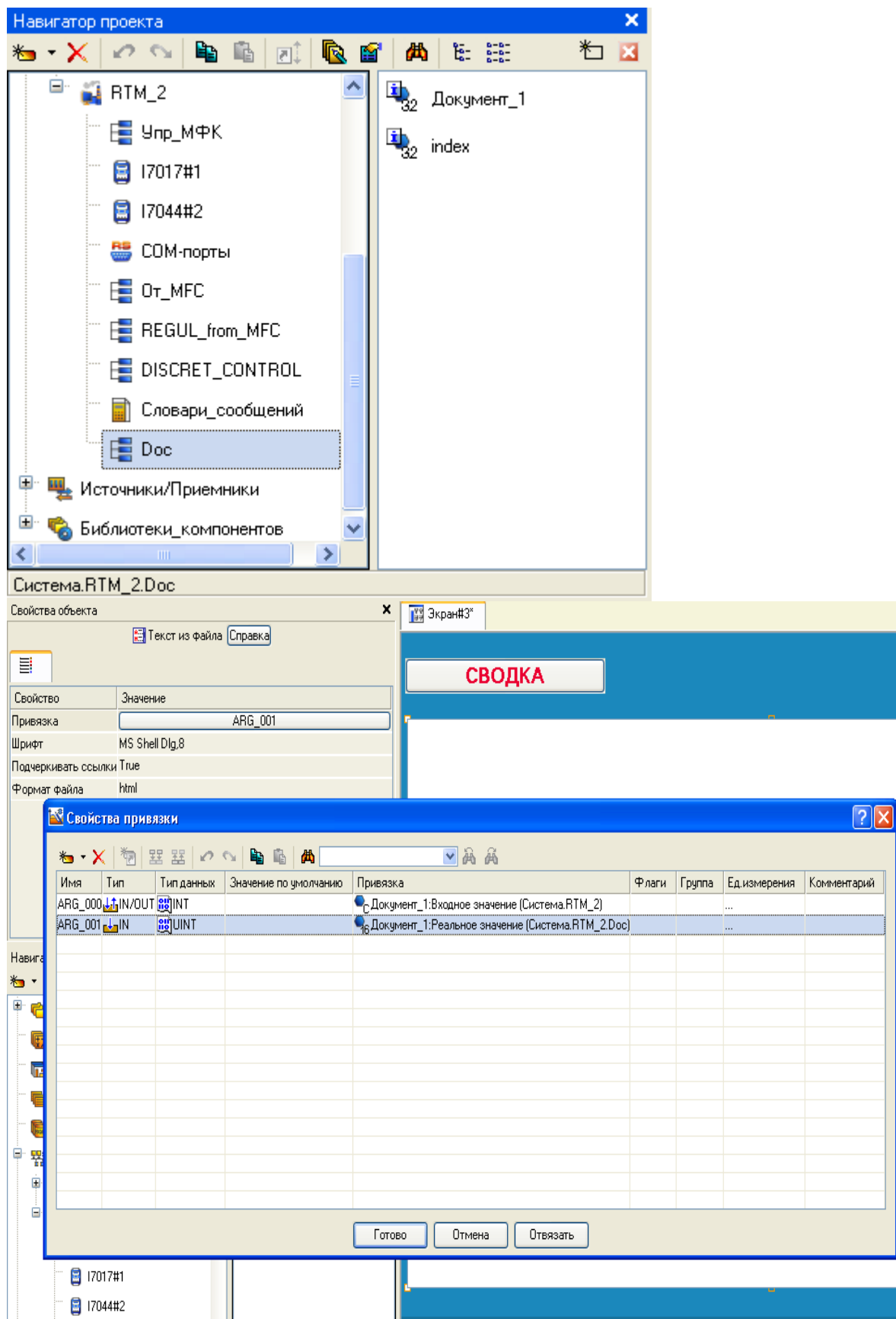




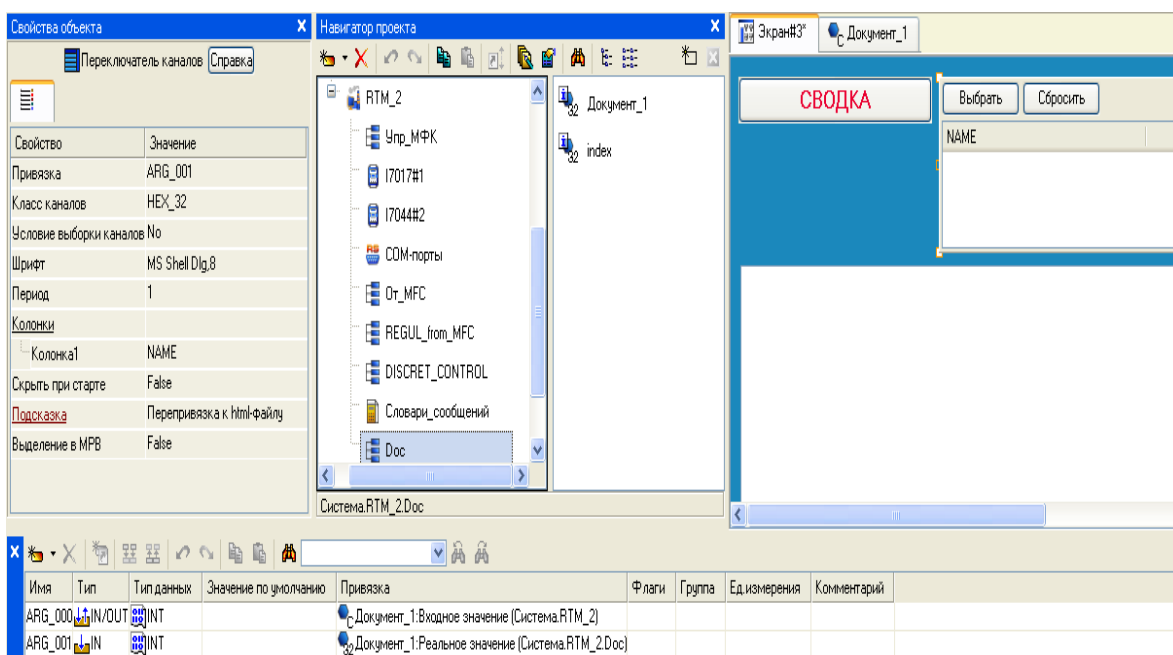
Разместим в левом верхнем углу экрана ГЭ Кнопка с надписью СВОДКА и определим для нее прямую посылку значения 1 во вновь созданный аргумент экрана типа **IN/OUT** с типом данных **INT**, привязанного к атрибуту **Входное значение** канала Документ_1:



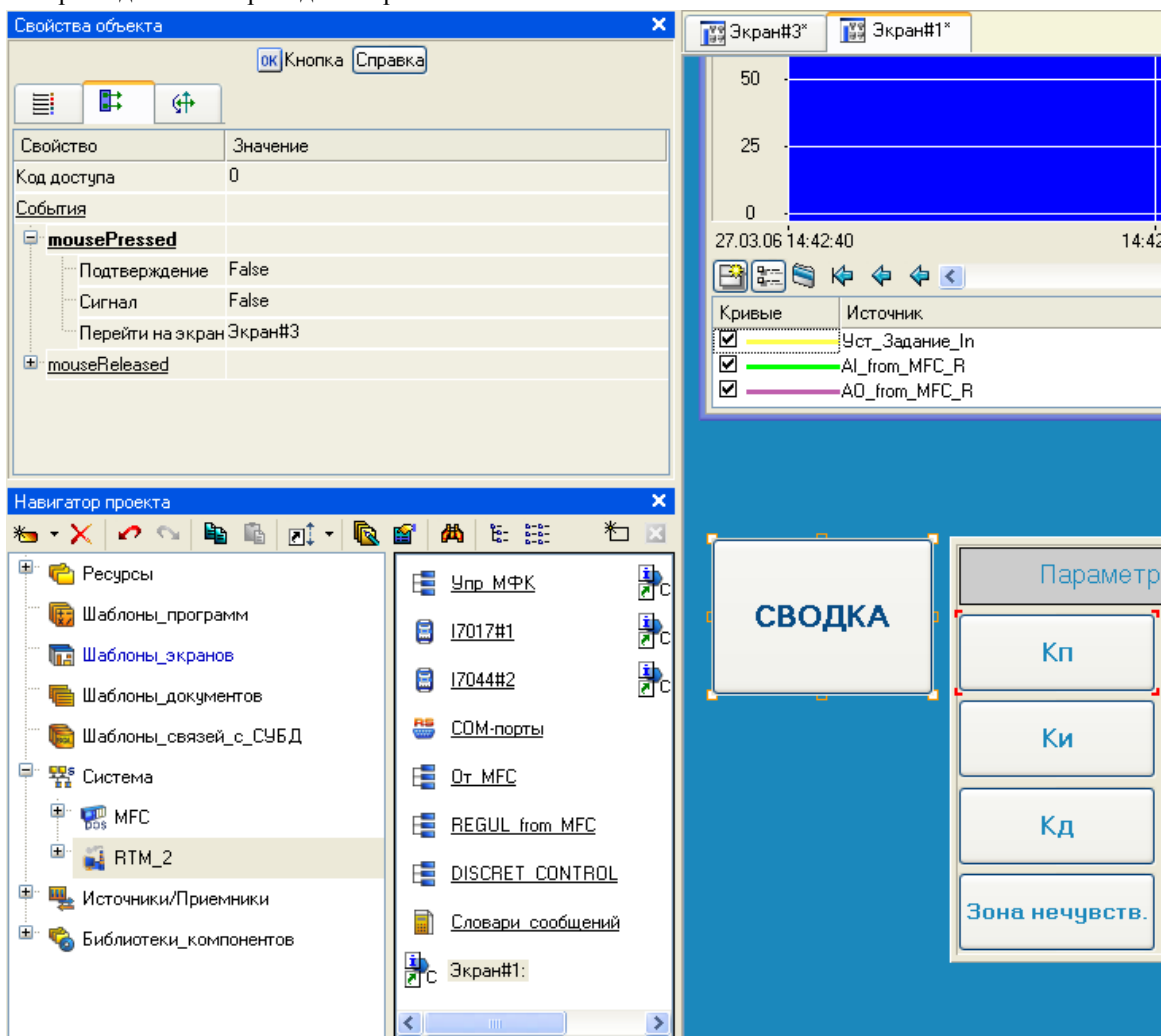
Нижнюю часть экрана займем ГЭ Текст из файла, вызываемым с помощью иконки  с панели инструментов графического редактора, привязав его ко второму аргументу экрана Экран#3 ARG_001, связанного с атрибутом **Реальное значение** канала класса **HEX32** с именем Документ_1, созданного в новой группе **Doc** узла **RTM_2** (там же создадим и второй канал класса **HEX32** с именем **index**):



Для реализации обновления содержимого файла Документ_1.html разместим в правом верхнем углу экрана ГЭ Переключатель каналов с помощью иконки  и настроим его следующим образом:



Для перехода на всплывающий экран Экран#3 с Экрана#1, разместим на *последнем* ГЭ Кнопка, настроив для него переход на Экран#3:



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

Лабораторная работа № 4.

Создание проекта распределенной системы

1. Цель работы

Создать многоузловую распределенную систему, содержащую одиннадцать узлов АРМ (автоматизированное рабочее место), с использованием метода «от **Топологии**». При отладке проекта использовать утилиту сетевой **Шпион**.

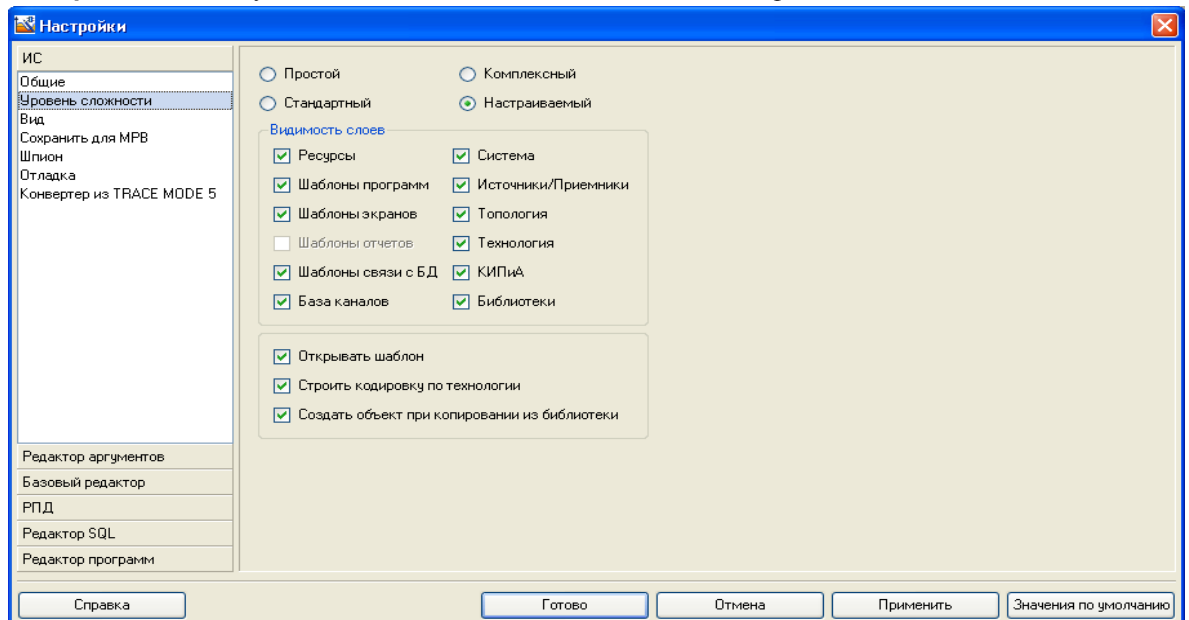
2. Порядок выполнения работы

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

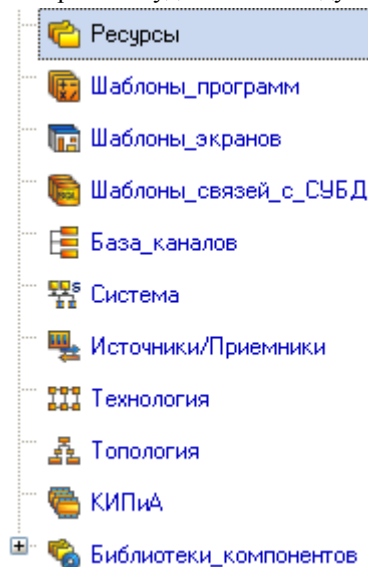
3. Образец выполнения работы

Создание структуры проекта

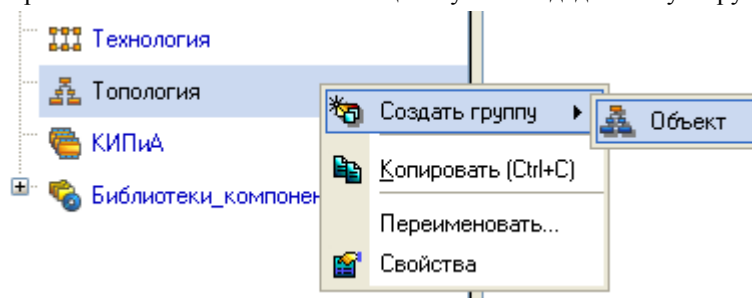
Произведем запуск интегрированной среды разработки TRACE MODE 6 и, выбрав в основном меню пункт **Файл**, откроем настройки ИС. В разделе **Уровень сложности** выберем ЛК **Настраиваемый** и установим видимость всех слоев с помощью флажков:



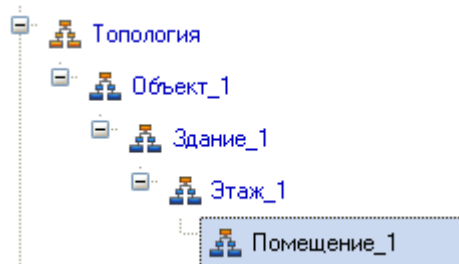
Нажмем ЛК экранную кнопку **Готово**. Создадим новый проект с помощью иконки . В результате дерево проекта будет иметь следующий вид:



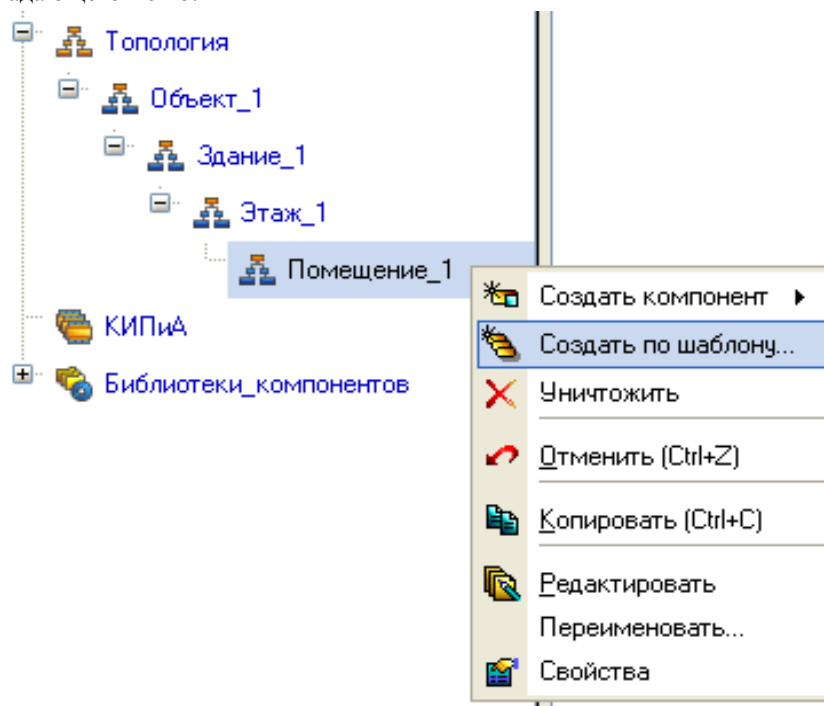
Выберем ЛК слой **Топология** и по щелчку ПК создадим новую группу - **Объект**:



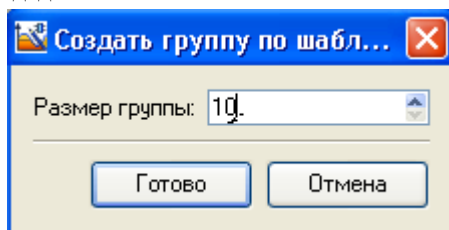
Выполняя последовательно подобную процедуру, получим следующую структуру:



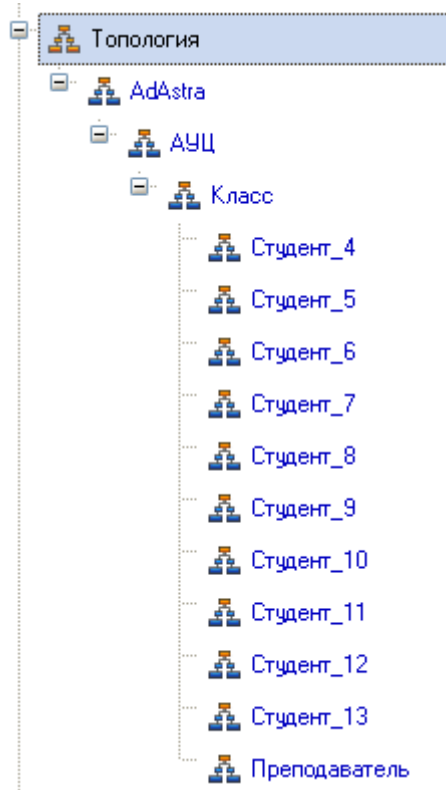
Для создания прообразов будущих АРМ воспользуемся механизмом создания компонентов по шаблону. По щелчку ПК на компоненте **Помещение_1** выберем соответствующий пункт выпадающего меню:



и зададим количество:



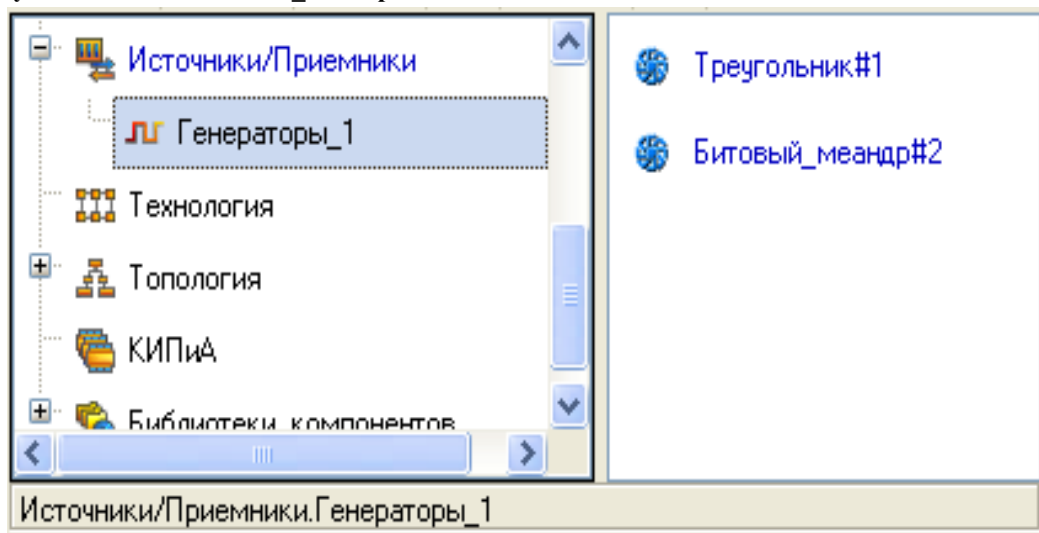
По нажатию экранной клавиши **Готово** получим требуемую структуру проекта, которая после переименования компонентов будет иметь следующий вид:



Создание генераторов тестовых сигналов

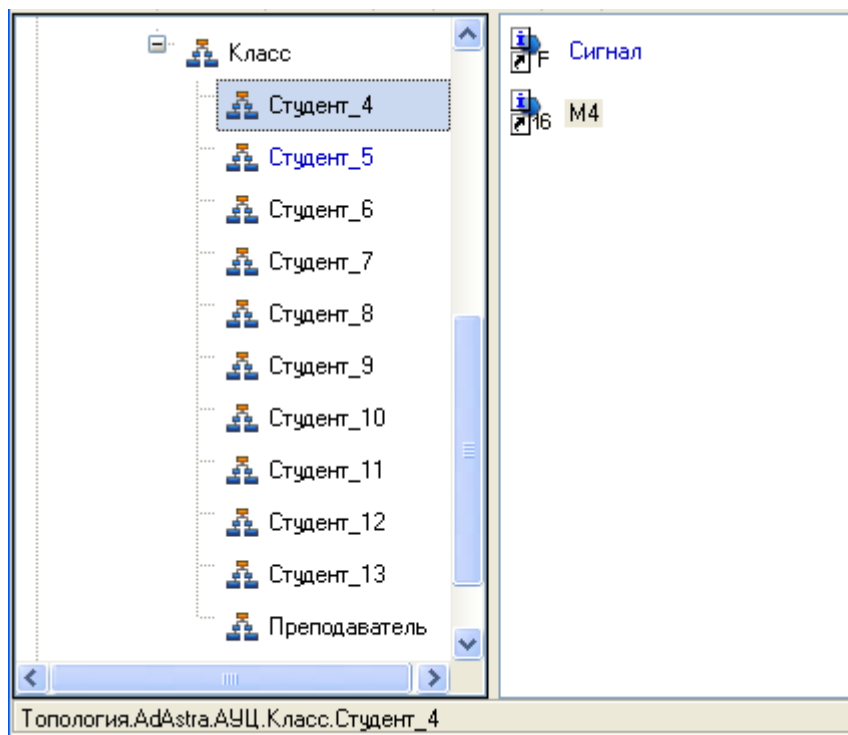
Поскольку в создаваемом проекте мы будем использовать два метода сетевого межкомпонентного взаимодействия – запрос/ответ и автопосылки, то определим для АРМ эстафетную передачу значений генератора **Треугольник** от узла АРМ **Студент_4** к **Студент_5** и т.д. до АРМ **Преподаватель** и автопосылок от каждого узла с использованием генератора **Битовый_меандр**. Прием автопосылок будем осуществлять на узле **Преподаватель**.

Создадим в слое **Источники/приемники** группу **Генераторы_1**, в ней два генератора – **Треугольник#1** и **Битовый_меандр#2**.

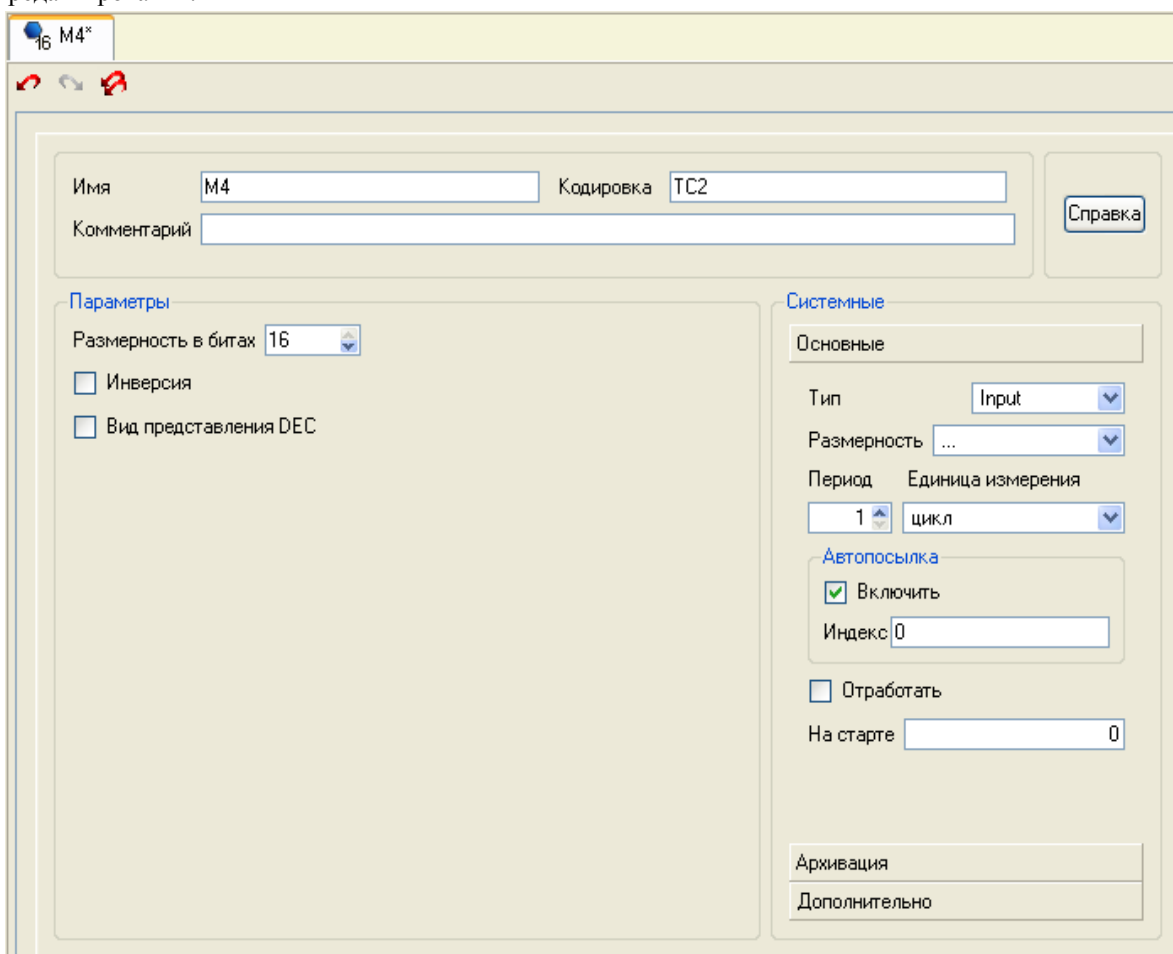


Создание каналов

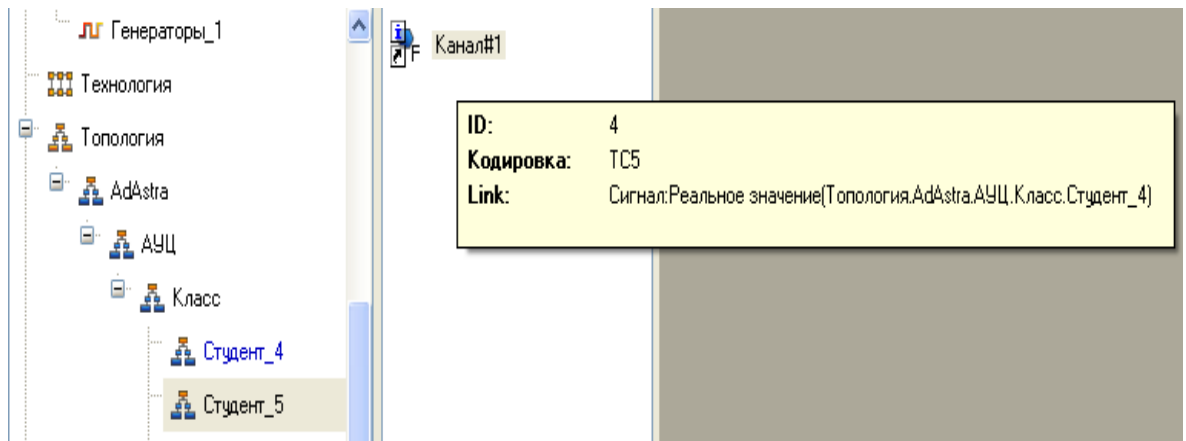
Перетаскиваем созданные генераторы – **Треугольник#1** и **Битовый_меандр#2** из слоя **Источники/приемники** в группу **Студент_4** слоя **Топология** и переименуем полученные каналы соответственно в **Сигнал** и **М4**:



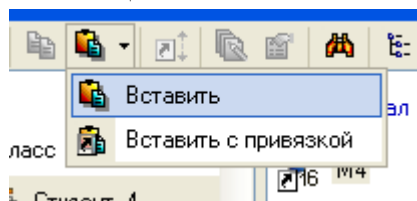
Откроем на редактирование канал **M4** и определим для него автопосылку в основном бланке редактирования:



Откроем дополнительную панель навигатора проекта на группе **Студент_5**. Создадим в ней новый компонент – канал класса **Float Канал#1**. Захватим ЛК в верхней панели канал **Сигнал** и перетащим его на канал **Канал#1**, отпустим ЛК. Тем самым мы зададим связь между данными каналами.



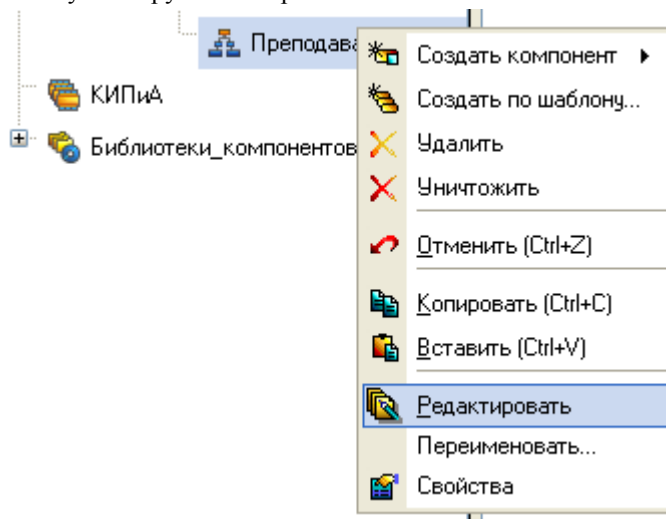
Захватим ЛК канал **М4** в группе **Студент_4** и скопируем его в буфер обмена с помощью иконки  или комбинации клавиш **CTRL+C**. Выберем ЛК группу **Студент_5** и воспользовавшись иконкой  или комбинацией **CTRL+V** извлечем канал из буфера обмена. При этом применим простую вставку:



Полученный в результате канал **Канал#2** переименуем в **М5**.

Далее, оставив открытой панель навигатора проекта на группе **Студент_5**, во второй откроем группу **Студент_6**. Создадим новый компонент – канал класса **Float** и перетащим на него канал **Канал#1** из группы **Студент_5**. Извлечем из буфера обмена с помощью простой вставки канал, переименуем вновь созданный канал в **М6**. Проведем аналогичные операции вплоть до группы **Преподаватель**.

В группе **Преподаватель** также создадим канал класса **Float** для приема сигнала от генератора **Треугольник#1**, передаваемого по эстафете, и перетащим на него канал **Канал#17** из группы **Студент_13**. Переименуем его в **Прием сигнала**. Для приема автопосылок от каналов **М4 – М13** прообразов узлов **Студент_4 – Студент_13**, создадим в группе **Преподаватель** десять каналов класса **HEX16** с помощью процедуры создания компонентов по шаблону. Для переименования каналов воспользуемся групповой правкой:



Перейдем к содержимому группы:

Преподаватель

Содержимое группы

Атрибуты

Основные

Имя: Преподаватель

Кодировка: А

Комментарий:

Кодировка: АА зона

В открывшейся таблице компонентов выделим ЛК вновь созданные каналы. Удерживая нажатой клавишу **CTRL**, нажмем ПК в поле **Базовое имя**. В выпадающем меню выберем пункт **Задать строку**. В окошке базового редактора введем:

Базовый редактор

Шаблон:

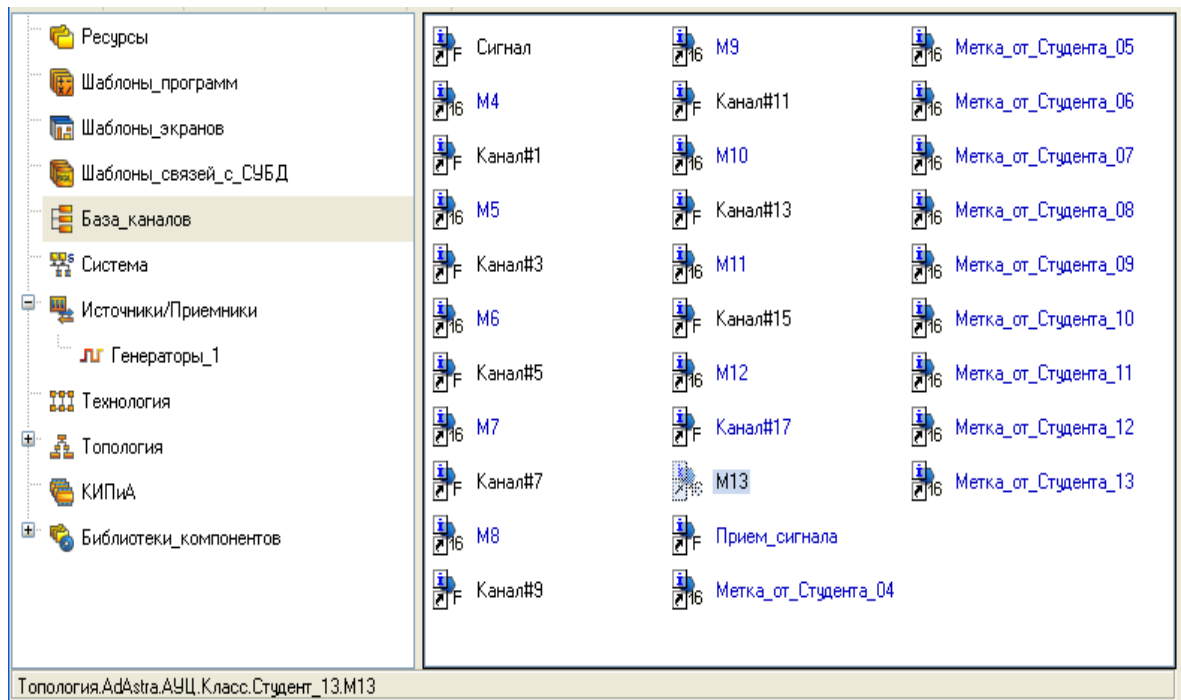
Метка_от_Студента_%2d(4,1)

Справка Готово Отмена

и нажмем **Готово**. В результате выполненных действий атрибуты **Базовое имя** выделенных каналов станут:

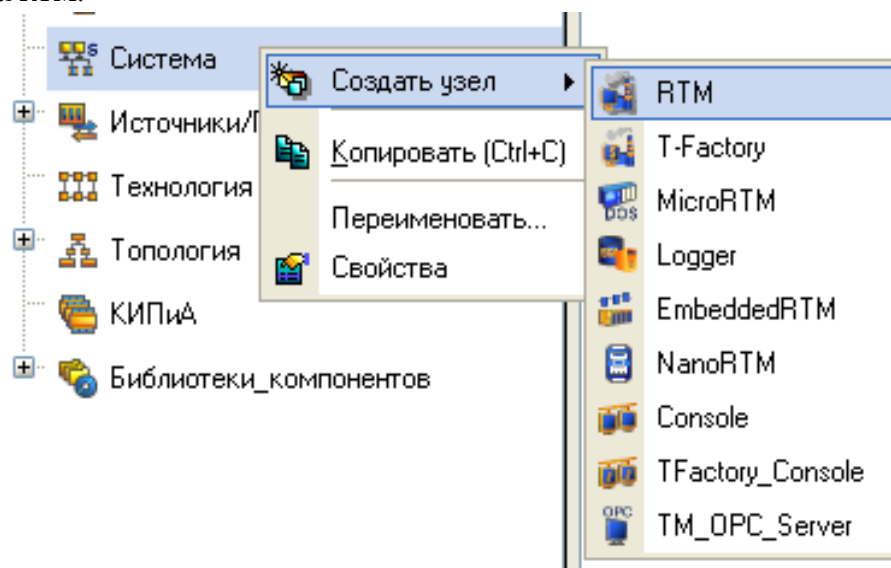
Базовое имя	Кодировка
Прием_сигнала	ТС5
Метка_от_Студента_04	ТС2
Метка_от_Студента_05	ТС2
Метка_от_Студента_06	ТС2
Метка_от_Студента_07	ТС2
Метка_от_Студента_08	ТС2
Метка_от_Студента_09	ТС2
Метка_от_Студента_10	ТС2
Метка_от_Студента_11	ТС2
Метка_от_Студента_12	ТС2
Метка_от_Студента_13	ТС2

Для установления привязок воспользуемся процедурой редактирования в слое **База_каналов**, выделив его для этого ЛК. В открывшемся окне перетащим канал **М4** на канал **Метка_от_Студента_4**, **М5** на **Метка_от_Студента_5** ..., **М13** на **Метка_от_Студента_13**.



Создание узлов проекта

При создании узлов проекта поступим следующим образом, создадим один узел, настроим для него параметры сетевого взаимодействия, а остальные создадим по шаблону. В слое Система создадим узел RTM:



Откроем узел на редактирование в бланке Основные его свойств изменим имя узла и определим для его системного сетевого адаптера прием и посылку:

СТУД_4*

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

Имя:

Атрибуты

Пароль:

Характеристика:

Пересчет

Период:

Разрешение:

Сеть

Имя компьютера / IP адрес:

Адаптеры

Системный	Первый	Второй	Третий
☒ Прием	☐ Прием	☐ Прием	☐ Прием
☒ Посылка	☐ Посылка	☐ Посылка	☐ Посылка

Мост

☐ Назначить

Номер:

Номер узла в проекте

Индивидуальный: Групповой: Код проекта:

Модемы

Телефон №1	Телефон №2	Строка инициализации

Регистратор

При старте: Статус:

Системные

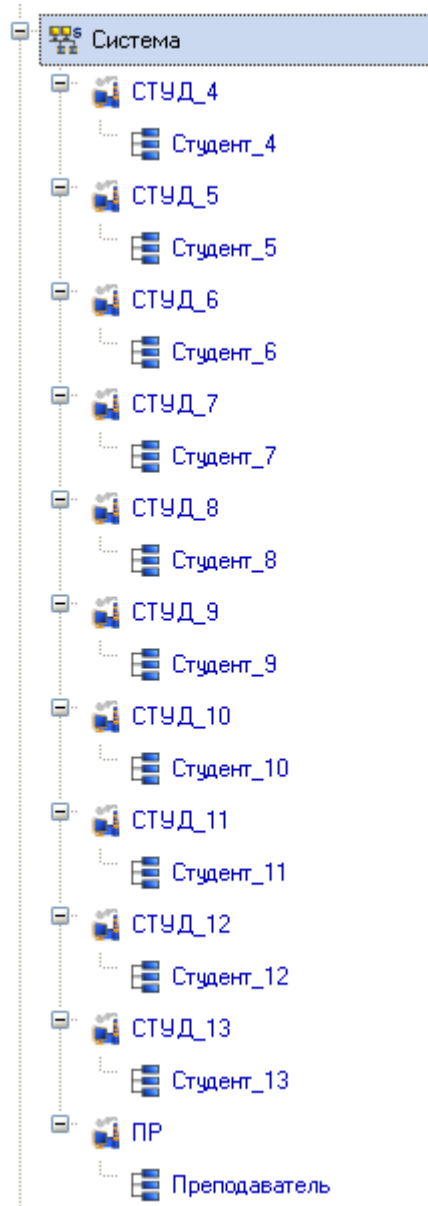
Сторожевой таймер: Тип дисплея: Тип клавиатуры:

После создадим по шаблону десять подобных узлов. Изменим имена узлов:



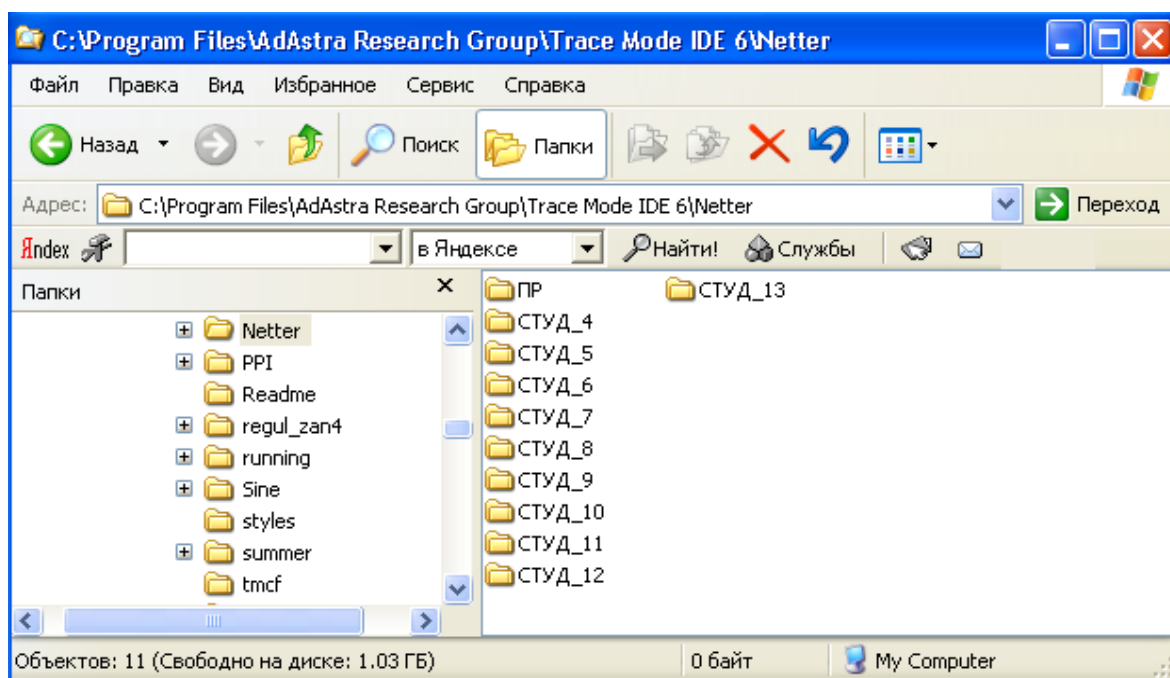
Формирование баз каналов узлов

Во втором окне навигатора проекта откроем слой **Топология**. Последовательно перетащим группу **Студент_4** в узел **СТУД_4** ... **Преподаватель** в узел **ПР**. Создаваемые для узлов по умолчанию группы **Каналы** можно уничтожить.



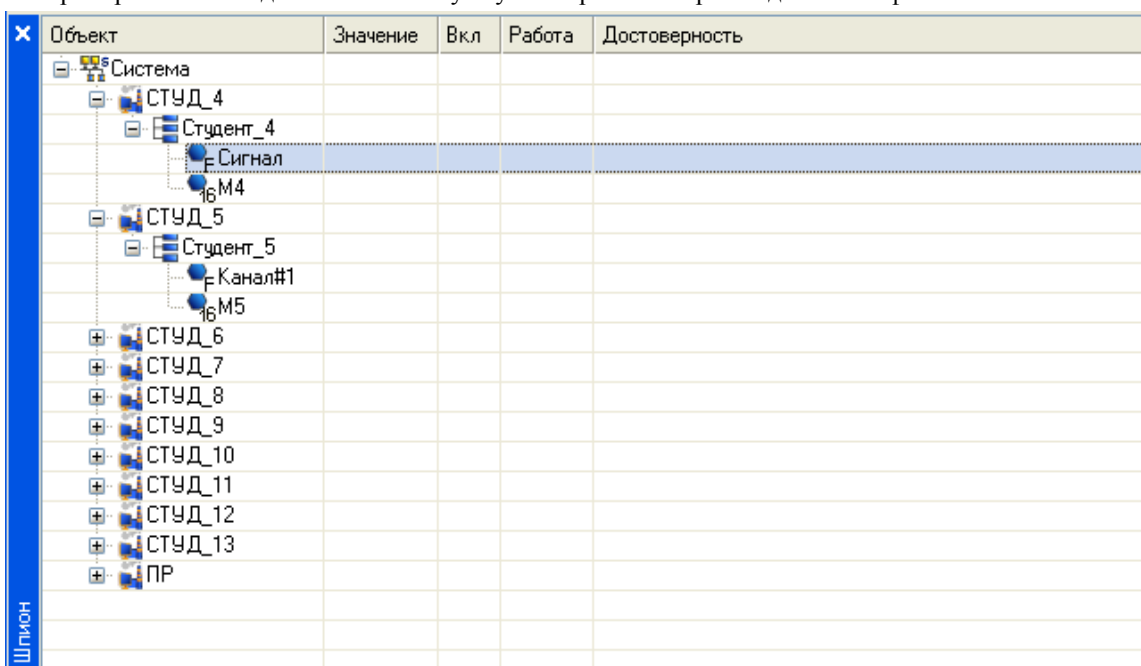
Сохранить выполненный проект как **Netter** с помощью иконки . Скопировать полученный файл **Netter.prj** на все компьютеры, после чего на каждом выполнить запуск ИС TRACE MODE 6, открыть

проект Netter, на панели инструментов выбрать ЛК иконку  и скомпилировать тем самым модуль для запуска проекта в реальном времени. В директории ИС будет создана поддиректория Netter, содержащая папки, предназначенные для запуска узлов проекта:



Запуск проекта

На компьютере преподавателя запустить отладочную утилиту **ШПИОН**, с помощью которой контролировать последовательный запуск узлов проекта и прохождение эстафетного сигнала.



Переходя в окне **ШПИОН** к атрибутам каналов того или иного узла, можно их изменять, например, атрибут **Состояние**, а результаты фиксировать как непосредственно в окне **ШПИОН**, так и на узлах

проекта. После запуска узла **СТУД_13** закрыть утилиту **ШПИОН**, выбрать иконку  на панели инструментов и запустить на исполнение узел **ПР** - наблюдать прохождение как эстафетного сигнала, так и автопосылок от остальных узлов проекта. Отключая/включая тот или иной узел, можно прерывать/восстанавливать обмен данными.

Лабораторная работа № 5.

Взаимодействие с технологической БД

1. Цель работы

Создать базу каналов проекта системы автоматизации методом «от Технологии» с использованием подготовленной базы данных, содержащих описание точек контроля технологического объекта.

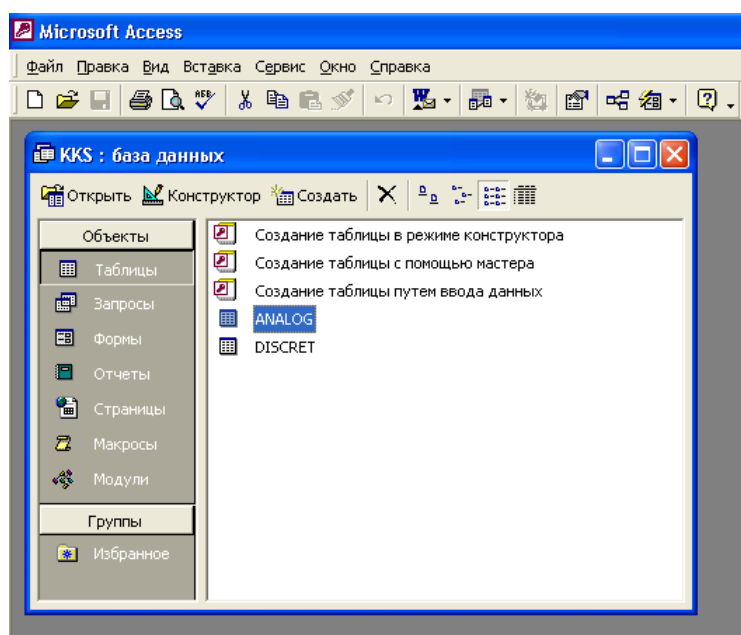
2. Порядок выполнения работы

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

3. Образец выполнения работы

Структура БД

Описание точек контроля содержится в файле БД MS Access **C:\KKS.mdb**. Файл содержит две таблицы в которых сгруппированы параметры аналоговые и дискретные:



Структура таблицы ANALOG:

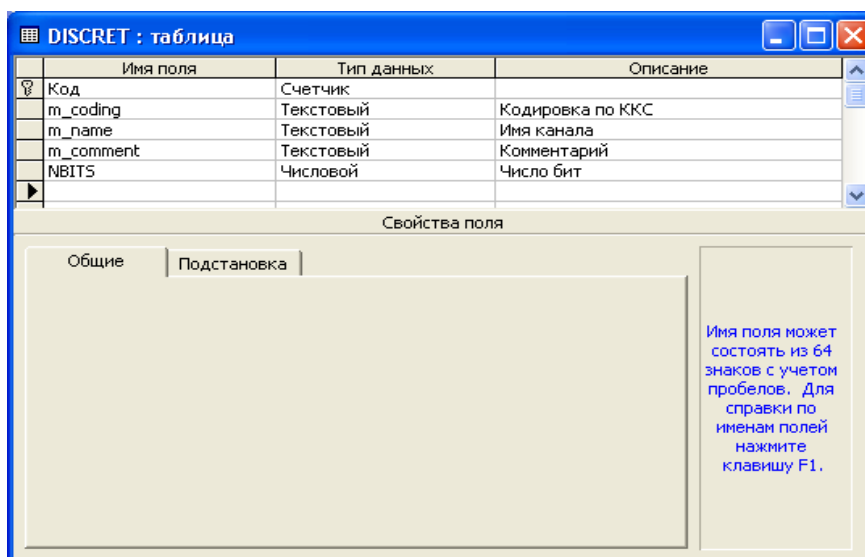
ANALOG : таблица			
	Имя поля	Тип данных	Описание
Код		Счетчик	
m_coding		Текстовый	Кодировка по ККС
m_name		Текстовый	Имя канала
m_comment		Текстовый	Комментарий
HL		Числовой	Верхний предел
LL		Числовой	Нижний предел
HA		Числовой	Верхняя аварийная уставка
LA		Числовой	Нижняя аварийная уставка
HW		Числовой	Верхняя предупредительная уставка
LW		Числовой	Нижняя предупредительная уставка
SC_F		Логический	Контроль границ
BNDR		Логический	Использование границ
HYST		Числовой	Гистерезис
KX		Числовой	Множитель
AP		Числовой	Апертура
PRS		Логический	Обработка в канале
TYPE		Логический	Тип канала (0-INPUT, 1-OUTPUT)

Свойства поля

Общие | Подстановка

Имя поля может состоять из 64 знаков с учетом пробелов. Для справки по именам полей нажмите клавишу F1.

Структура таблицы **DISCRET**:



Имя поля	Тип данных	Описание
Код	Счетчик	
m_coding	Текстовый	Кодировка по ККС
m_name	Текстовый	Имя канала
m_comment	Текстовый	Комментарий
NBITS	Числовой	Число бит

Свойства поля

Общие Подстановка

Имя поля может состоять из 64 знаков с учетом пробелов. Для справки по именам полей нажмите клавишу F1.

Поля **m_coding** в обеих таблицах содержат технологическую кодировку точек контроля по системе обозначения **KKS** широко применяемой, например, в энергетике. Построение кода:

N степени	0	1	2	3
Название	Объект	Функция	Агрегат	Элемент оборудования
Тип данных	(A) или (N)	(N) A A A N N	A A N N N (A)	A A N N

A – буква, N – цифра, () – позиция может отсутствовать.

Нулевая ступень – **Объект** указывает, например, на номер блока электростанции. Первая ступень – **Функция** указывает на номер однотипных частей энергоблока, символьная часть определяет соответствующую функциональность системы, например, HCA – установка воздушного вентилирования. Вторая ступень – **Агрегат** символьная часть указывает на аппараты и агрегаты, устройства электротехники и АСУТП, а также контуры измерения и регулирования; цифровые позиции используются для нумерации элементов, определяемых символьными позициями; последний символ классифицирует отдельные части целостных агрегатов. Третья ступень – **Элемент** введена для различия, например, типа приборов, преобразователей, также эта ступень различает сигналы по источнику и приемнику.

Например, **1JAJ01CF001QP01** - расход теплоносителя по нитке 1 бака первого блока ядерного реактора, измеряемый с помощью прибора с порядковым номером 1, установленного в помещении блочного щита управления.

Содержимое таблицы **ANALOG** следующее:



Код	m_coding	m_name	m comment	HL	LL	HA	LA	HW	LW	SC	F	BNDR	HYST	KX	AP	PRS
1	JAJ01CT001QU01	Temp_001	Температура теплоносителя в баке реактора поз.1	100	0	90	20	75	25	✓	✓	✓	5	0.02442	3.5	✓
2	JAJ01CT001QU02	Temp_002	Температура теплоносителя в баке реактора поз.2	100	0	90	20	75	25	✓	✓	✓	5	0.02442	3.5	✓
3	JGA01CT001QU01	Temp_003	Температура теплоносителя 2-го контура	100	0	90	20	75	25	✓	✓	✓	5	0.02442	3.5	✓
4	JEAO1CT001QU01	Temp_004	Температура теплоносителя 1-го контура	100	0	90	20	75	25	✓	✓	✓	5	0.02442	3.5	✓
5	JAJ01CF001QP01	Flow_001	Расход по нитке 1 бака реактора	2500	0	2000	800	1800	1000	✓	✓	✓	50	0.610501	20	✓
6	JAJ01CF001QP02	Flow_002	Расход по нитке 2 бака реактора	3300	0	3000	1200	2500	1500	✓	✓	✓	50	0.805861	25	✓
7	JAJ01CF001QP03	Flow_003	Расход по нитке 3 бака реактора	2500	0	2150	800	1750	1100	✓	✓	✓	50	0.610501	20	✓
8	JAJ01CF001QP04	Flow_004	Расход по нитке 4 бака реактора	5000	0	4800	1500	3800	2000	✓	✓	✓	75	1.221	50	✓
9	JEAO1CF001QP01	Flow_005	Расход по 1-му контуру реактора	10000	0	9500	2200	7500	3000	✓	✓	✓	100	2.442	75	✓
10	JEAO1CP001QP01	Press_001	Давление в 1-м контуре реактора	100	0	75	10	65	15	✓	✓	✓		0.02442	5	✓
(Счетчик)				0	0	0	0	0	0	✓	✓	✓	0	0	0	✓

Запись: 11 из 11

и включает в себя десять технологических параметров по трем системам технологического объекта – баку реактора, первому и второму контурам.

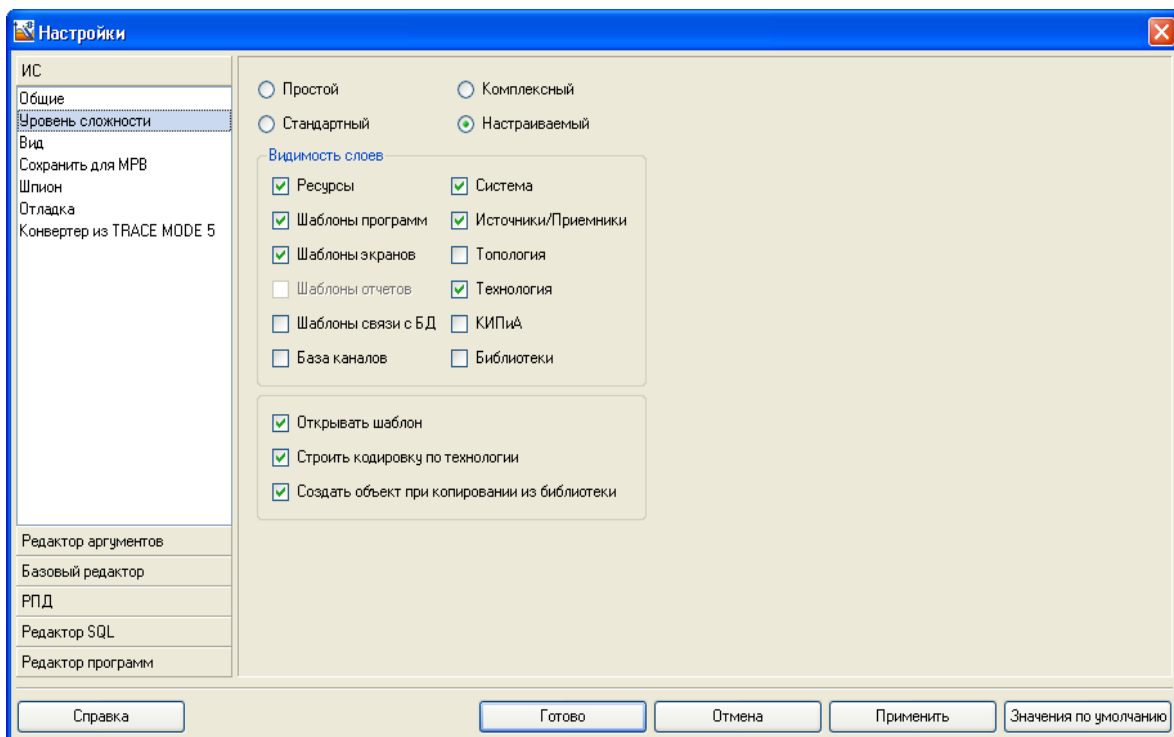
Содержимое таблицы **DISCRET**:

DISCRET : таблица					
Код	m_coding	m_name	m_comment	NBITS	
1	JAJO1CL002QU01	Level_001_D	Сигнализатор уровня в баке реактора Поз.1	1	
2	JGAO1CL002QU01	Level_002_D	Сигнализатор уровня в байпасе 2-го контура Поз.15	1	
3	JEAO1CL002QU01	Level_003_D	Сигнализатор уровня в байпасе 1-го контура Поз.37	1	
4	JAJO1CT002QP01	Temp_001_D	Сигнализатор превышения температуры (бак реактора)	1	
5	JEAO1CP002QP02	Press_001_D	Сигнализатор превышения давления (1ый контур)	1	
(Счетчик)				0	
Запись: 6 из 6					

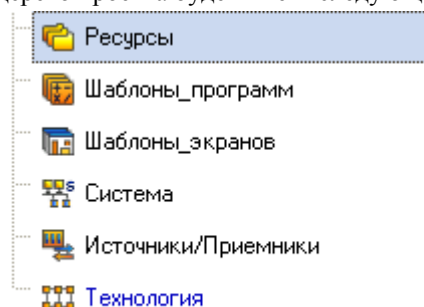
включает пять сигнализаторов уровня и температуры также по перечисленным выше системам объекта.

Создание структуры проекта

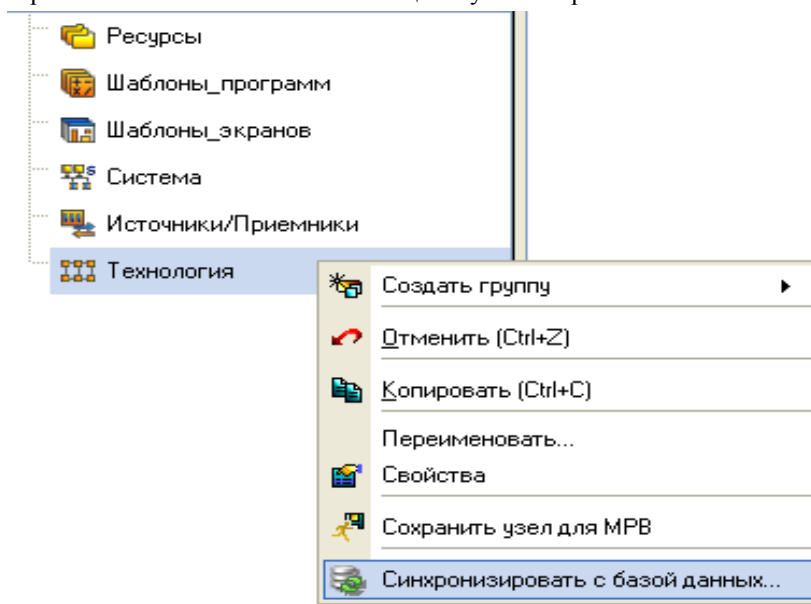
Произведем запуск интегрированной среды разработки TRACE MODE 6 и, выбрав в основном меню пункт **Файл**, откроем **настройки ИС**. В разделе **Уровень сложности** выберем ЛК **Настраиваемый** и установим видимость слоев с помощью флажков:



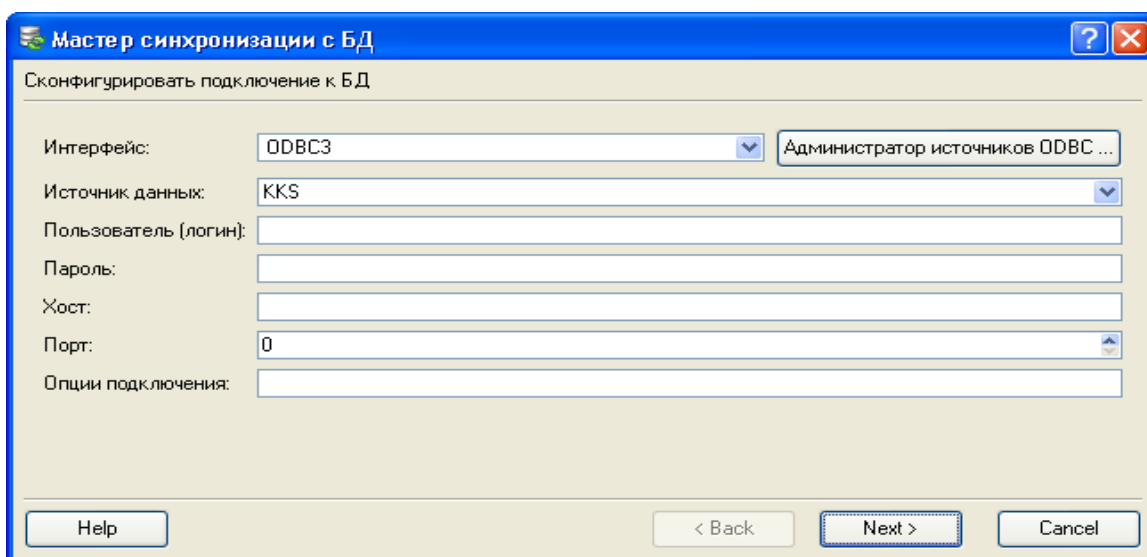
Нажмем ЛК экранную кнопку **Готово**. Создадим новый проект с помощью иконки . В результате дерево проекта будет иметь следующий вид:



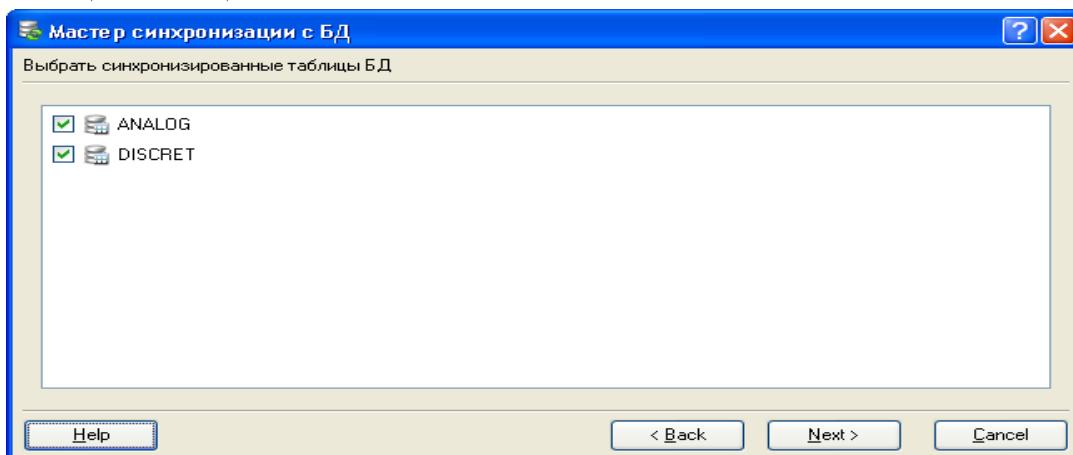
Выберем ЛК слой **Технология** и по щелчку ПК откроем контекстное меню:



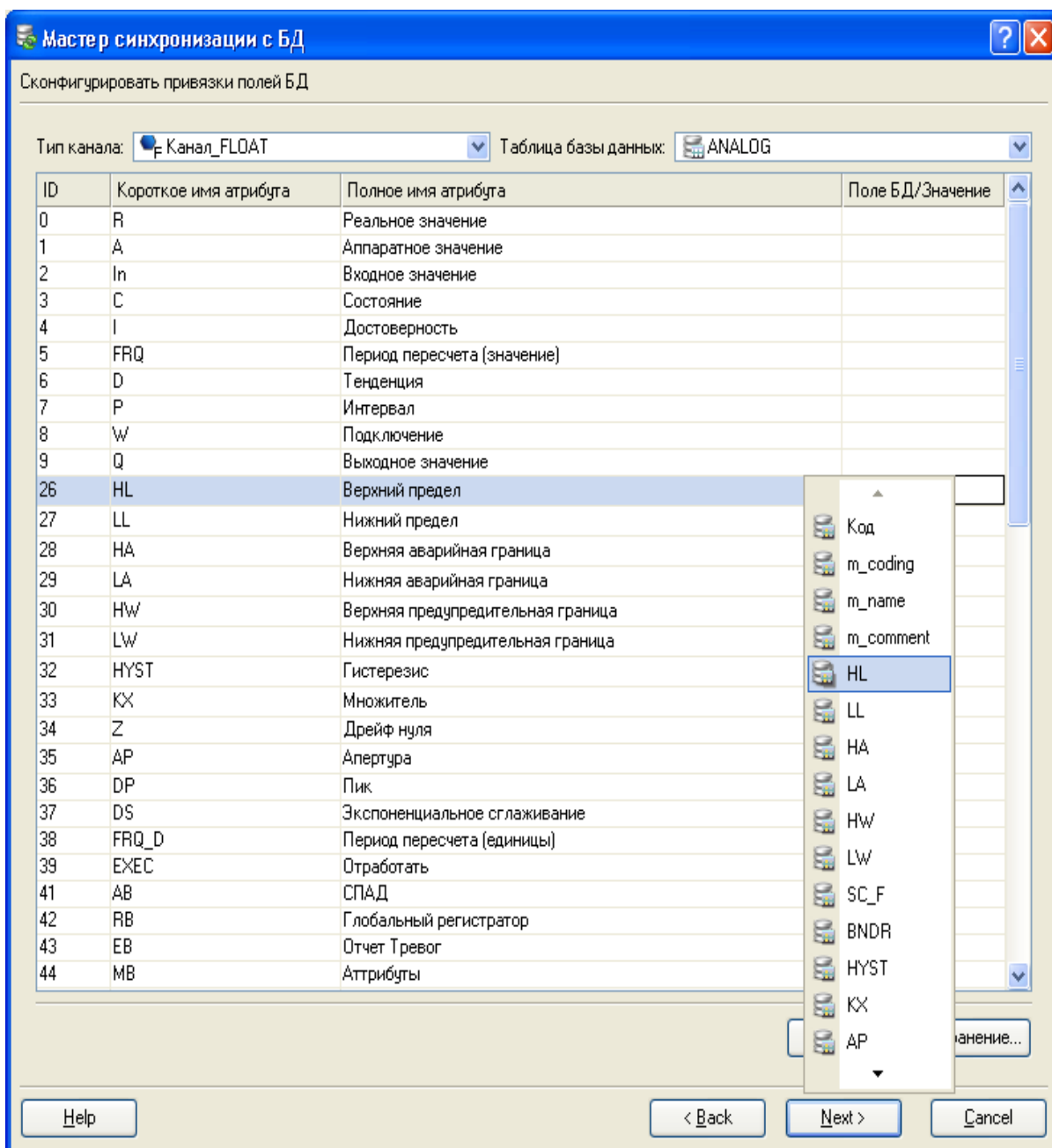
В нем выберем ЛК пункт **Синхронизировать с базой данных...** В открывшемся диалоговом окне с помощью кнопки **Администратор источников ODBC...** создадим пользовательский **DSN** для файла БД **KKS.mdb** с именем **KKS** и укажем его в строке **Источник данных**:



По нажатии экранной кнопки **Next** откроется окно выбора таблиц БД, в котором отметим ЛК обе имеющиеся таблицы – **ANALOG** и **DISCRET**:



После нажмем **Next** и перейдем к окну привязки полей таблицы БД к атрибутам выбираемого класса канала. Двойной щелчок ЛК в строке атрибута вызывает открытие контекстного меню для выбора поля БД. Например, для привязки поля **HL** таблицы **ANALOG** БД к атрибуту **Верхний предел** канала класса **Float**:



Выполним привязки для каналов класса **Float** следующим образом:

Мастер синхронизации с БД

Сконфигурировать привязки полей БД

Тип канала: Канал_FLOAT Таблица базы данных: ANALOG

ID	Короткое имя атрибута	Полное имя атрибута	Поле БД/Значение
0	R	Реальное значение	
1	A	Аппаратное значение	
2	In	Входное значение	
3	C	Состояние	
4	I	Достоверность	
5	FRQ	Период пересчета (значение)	
6	D	Тенденция	
7	P	Интервал	
8	W	Подключение	
9	Q	Выходное значение	
26	HL	Верхний предел	HL
27	LL	Нижний предел	LL
28	HA	Верхняя аварийная граница	HA
29	LA	Нижняя аварийная граница	LA
30	HW	Верхняя предупредительная граница	HW
31	LW	Нижняя предупредительная граница	LW
32	HYST	Гистерезис	HYST
33	KX	Множитель	KX
34	Z	Дрейф нуля	
35	AP	Апертура	AP
36	DP	Пик	
37	DS	Экспоненциальное сглаживание	
38	FRQ_D	Период пересчета (единицы)	
39	EXEC	Отработать	
41	AB	СПАД	
42	RB	Глобальный регистратор	
43	EB	Отчет Тревог	
44	MB	Аттрибуты	
45	T	Время изменения	
46	QE	Защелка	
47	IDSTR	MF	
48	CH_CLASS	Класс канала	Канал_FLOAT
49	DEBG	Отладка	
50	PRS	Обработка	PRS
51	CHG_IO	Смена типа Вх/Вых	
52	FS	Достоверность программная	
53	SC_F	Контроль границ	SC_F
57	ALR_MSG_I	Индекс аварийного словаря	
58	DW	Дамп	
60	T_NET	В сеть	
61	N_SND	Индекс автопосылки	
78	FORMAT	Формат	
79	CODE	Кодировка	m_coding
80	CMNT	Комментарий	m_comment
81	TYPE	Тип	TYPE
82	DIM	Размерность	
83	SGNL	Тип сигнала	
85	BNDR	Использовать границы	BNDR

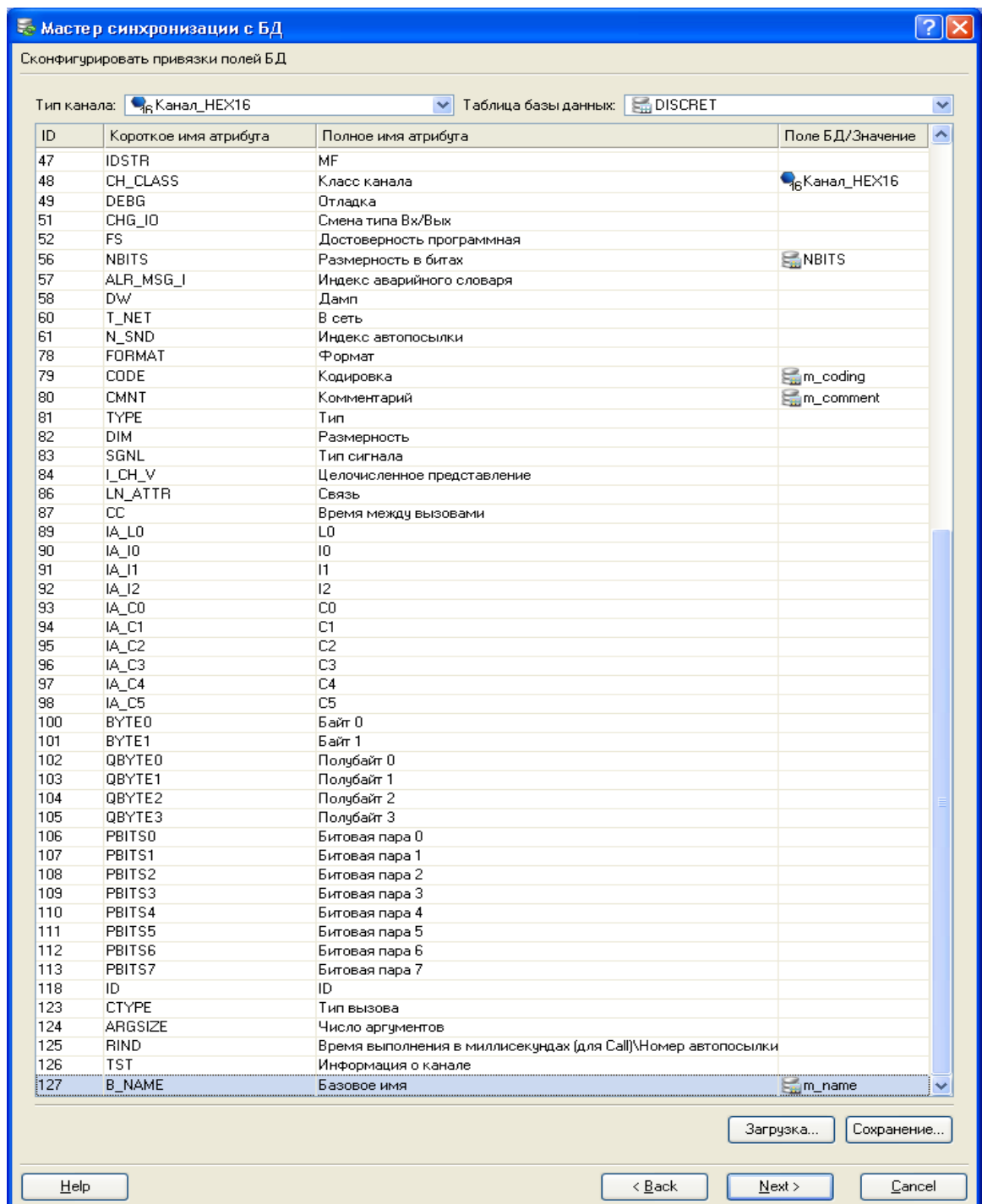
Загрузка... Сохранение...

Help < Back Next > Cancel

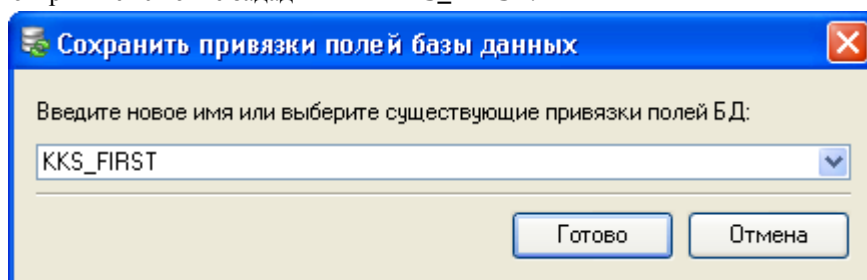
Не уместившаяся в окне привязка поля **m_name** к атрибуту **Базовое имя**:

97	IA_C4	C4	
98	IA_C5	C5	
118	ID	ID	
123	CTYPE	Тип вызова	
124	ARGSIZE	Число аргументов	
125	RIND	Время выполнения в миллисекундах (для Call)\Номер автопосылки	
126	TST	Информация о канале	
127	B_NAME	Базовое имя	m_name

Далее выбираем в верхней части окна конфигурирования класс канала **HEX16** и указываем таблицу **DISCRET**. Выполняем привязки как:

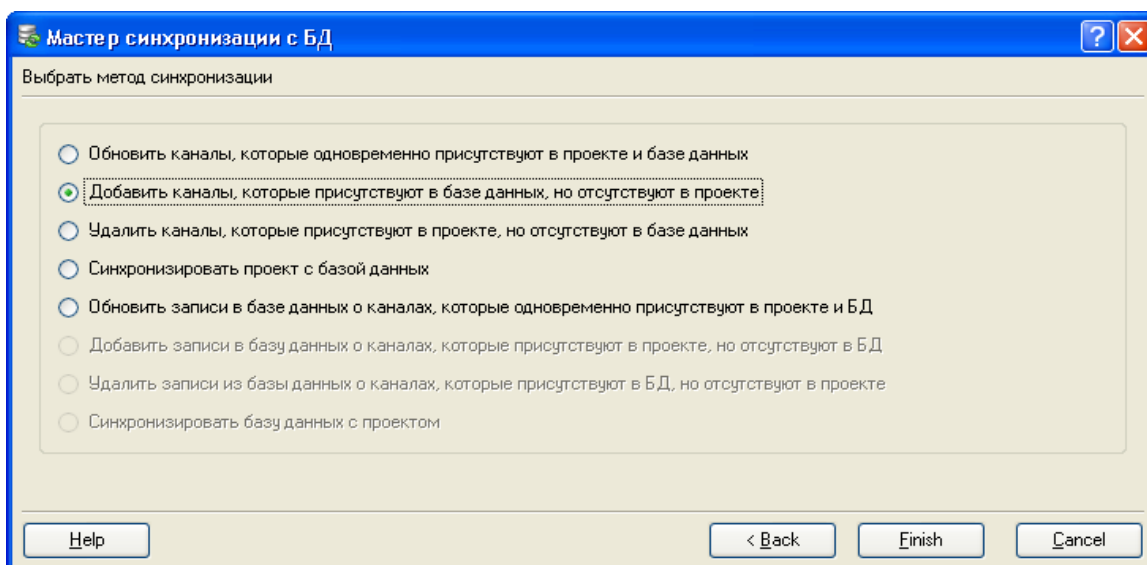


Для сохранения заданных привязок в файл выберем ЛК экранную кнопку **Сохранение...**, в открывшемся окне зададим имя **KKS_FIRST**.

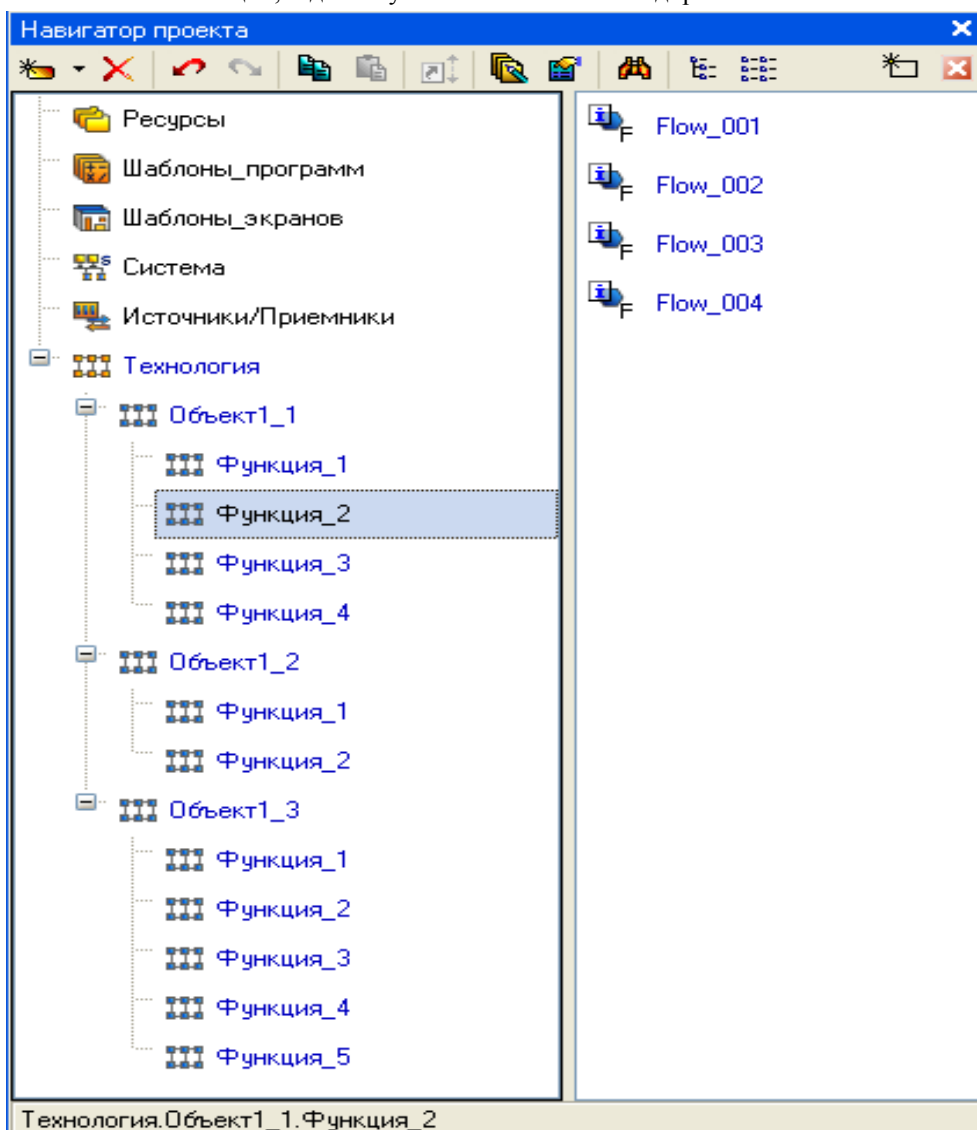


В дальнейшем при необходимости использовать примененную схему привязок будем пользоваться экранной кнопкой **Загрузка...**

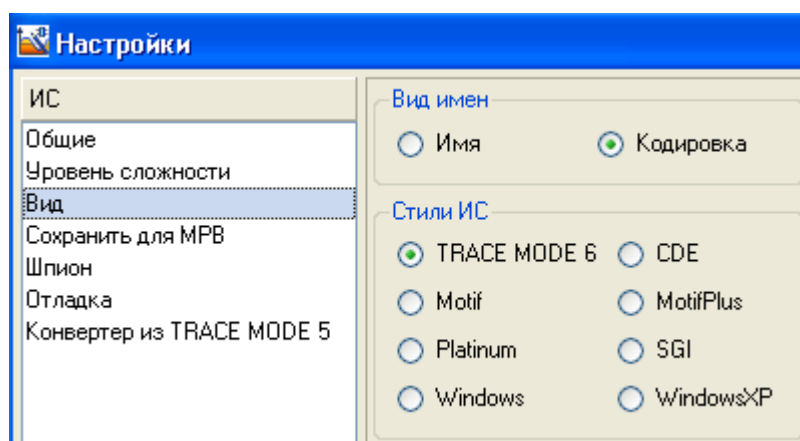
По нажатию кнопки **Next** будет открыто окно выбора метода синхронизации:



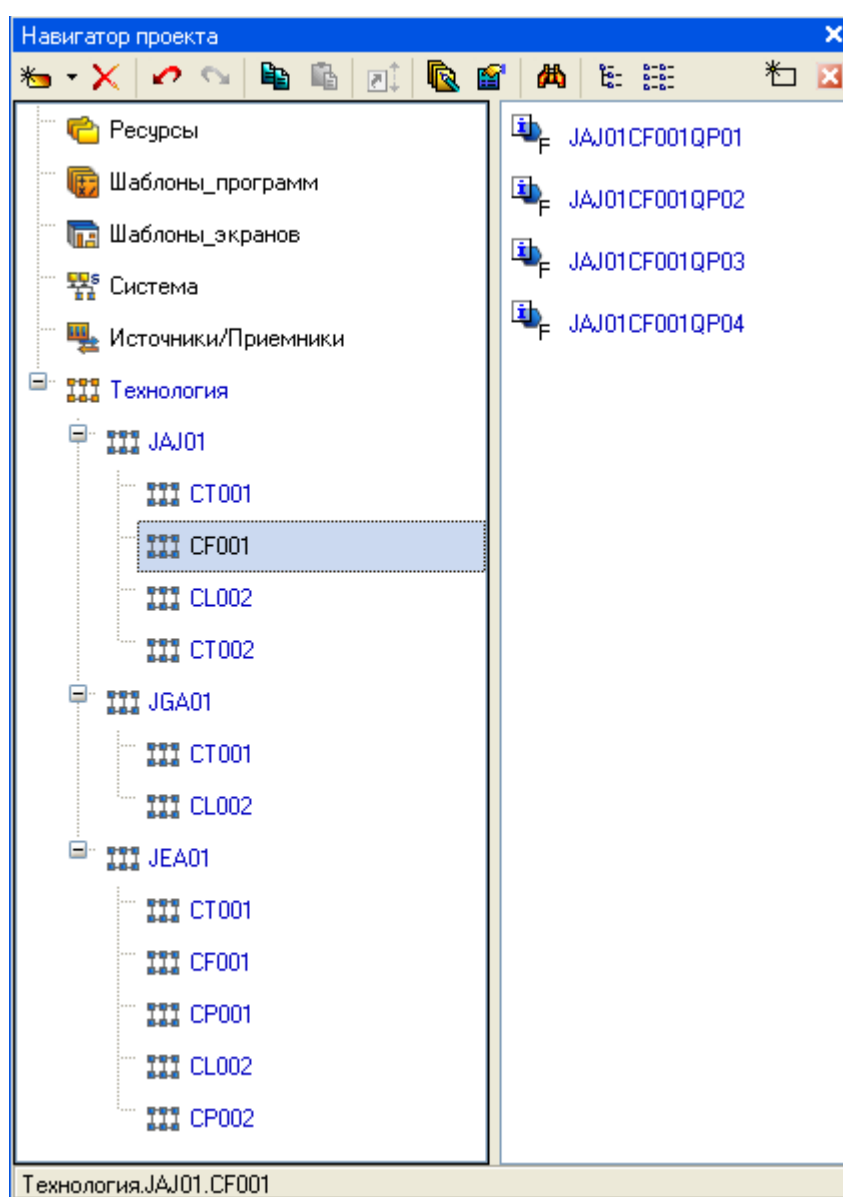
В нашем случае будем применять указываемый по умолчанию пункт – **Добавить каналы, которые присутствуют в базе данных, но отсутствуют в проекте**. Нажмем кнопку **Finish**. В результате выполненных действий в слое Технология будет создано дерево проекта, отображающее иерархию объекта автоматизации, задаваемую технологической кодировкой:



Изменим отображения вида имен каналов в **Настройках ИС** следующим образом:



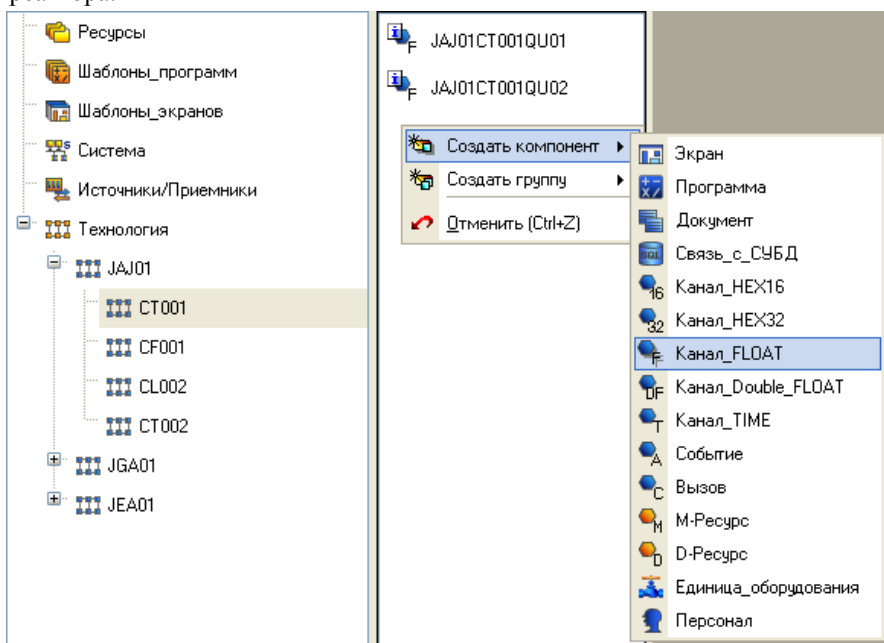
В результате получим отображение всех объектов в слое **Технология** в соответствии с их технологической кодировкой:



Далее можно приступать к настройке межкомпонентного взаимодействия, разработке шаблонов программ, экранов, запросов к СУБД, связыванию каналов с источниками/приемниками, отладке, созданию узлов проекта, распределения по ним каналов и проч. Если в ходе работы, например, технологов с БД происходят какие-либо изменения – добавились новые точки контроля или наоборот

какие-либо исключаются, изменяются пределы, уставки и т.д., то с помощью механизма синхронизации можно оперативно вносить изменения в структуру проекта. В том числе возможен и обратный процесс – добавление в БД записей о каналах, появившихся в проекте, но отсутствовавших в БД.

Например, создадим новый канал в функции **СТ001** объекта **JAJ01 - JAJ01CT001QP05** (кодировка) с именем **Temp_005**, т.е. добавим еще одну измерительную позицию для температуры в системе бака реактора:



Откроем созданный канал на редактирование и зададим для него атрибуты в основном бланке:

JAJ01CT001QP05*

Имя: Temp_005 Кодировка: JAJ01CT001QP05 Справка

Комментарий: Температура теплоносителя на выходе из бака реактора

Границы

☒ Использовать

ВП: 250

ВА: 225

ВГ: 200

НГ: 100

НА: 75

НП: 0

Гистерезис: 3

☒ Контроль границ

Обработка

☒ Использовать

Апертура: 1.5

Пик: 0

Сглаж.: 0

Множитель: 0.0610501

Смещение: 0

☒ Масштабирование

Масштабирование

In	Множитель	Смещение	A
Max			Max
4095	× 0.0610501	+ 0	250
0			0
Min			Min

Рассчитать

Системные

Основные

Тип: Input

Размерность: ...

Период: 1 Единица измерения: цикл

Автопосылка

☐ Включить

Индекс:

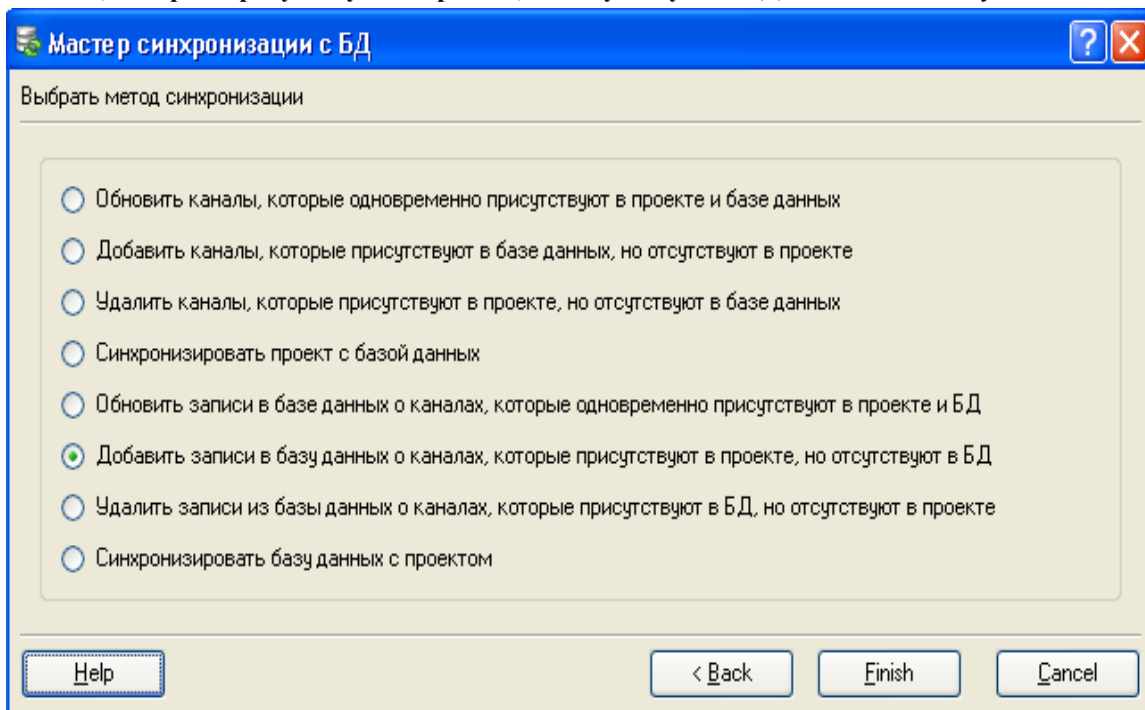
☐ Отработать

На старте: 0

Архивация

Дополнительно

Укажем ЛК слой **Технология** и по нажатию ПК перейдем в контекстное меню, выбрав пункт **Синхронизировать с базой данных**. В окне **Выбрать синхронизированные таблицы БД** укажем таблицу **ANALOG**. Далее в окне **Выбрать метод синхронизации** – пункт **Добавить записи в базу данных, которые присутствуют в проекте, но отсутствуют в БД** и нажать клавишу **Finish**.



Открыв таблицу **ANALOG**, увидим, что в нее была добавлена новая запись,

Код	m_coding	m_name	m_comment	HL	LL	HA	LA	HW	LW	SC_F	BNDR	HYST	KX	AP	PRS	TYPE
11	JAJ01CT001QP05	Temp_005	Температура теплоносителя на выходе из бака реактора	250	0	225	75	200	100	Да	Да	3	6.105006E-02	1.5	Да	Нет

соответствующая атрибутам введенного нами в проект канала.

Лабораторная работа № 6.

Резервирование узла АРМ

1. Цель работы

Произвести реконфигурацию проекта regul.prj и выполнить резервирование узла операторской станции RTM_2.

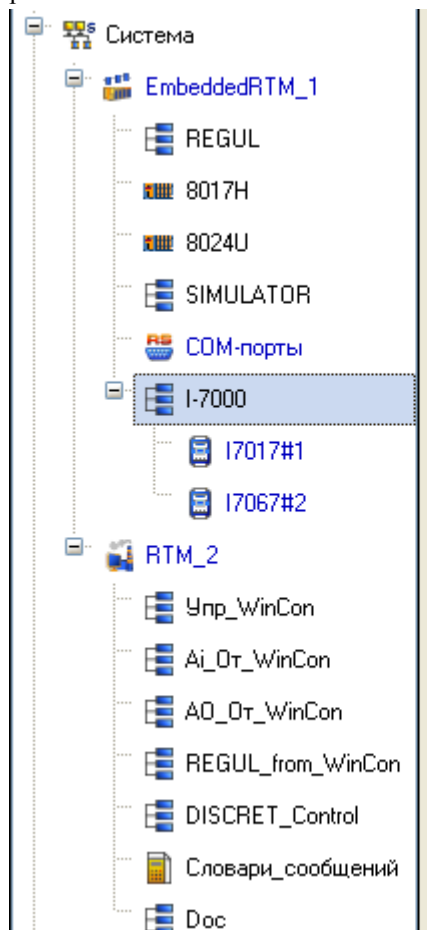
2. Порядок выполнения работы

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

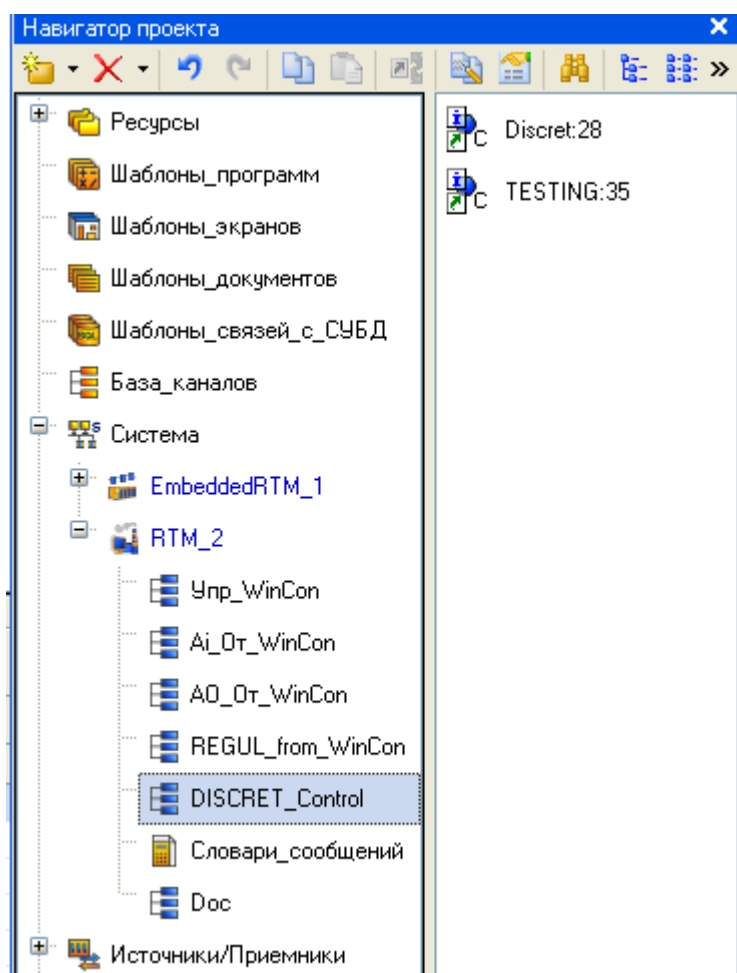
3. Образец выполнения работы

Реконфигурация проекта

Выполним подключение модулей удаленного ввода/вывода I-7017 и I-7067 к последовательному порту COM1 контроллера WinCon. Для этого перенесем группы COM-порты, I7017#1 и I7067#2 из узла RTM_2 в узел EmbeddedRTM_1, оставив канал класса Вызов TESTING, производящий инверсию атрибутов Достоверность канала Ain#1. Перенос осуществляется с помощью механизма drag'n'drop с удерживаемой в нажатом состоянии клавишей Shift.



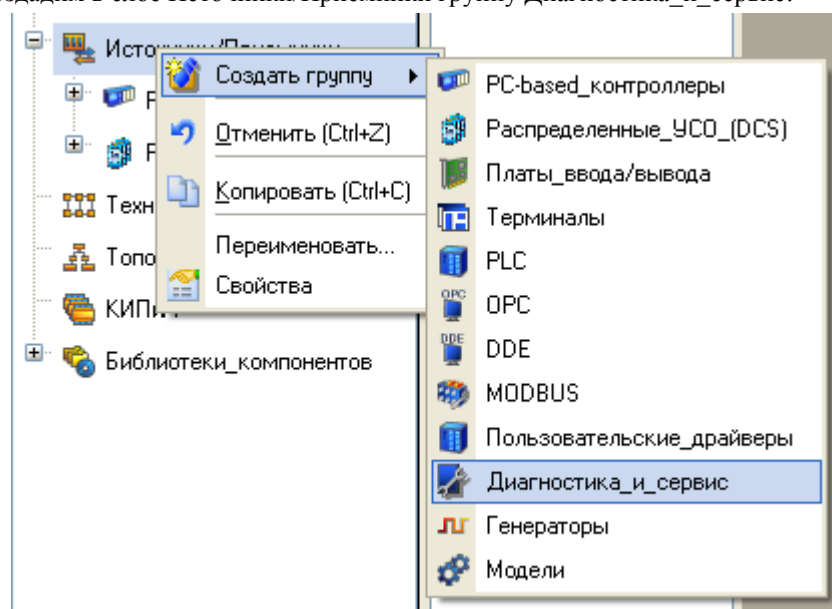
Каналы класса Вызов TESTING переместим в группу компонентов DISCRET_CONTROL узла RTM_2:



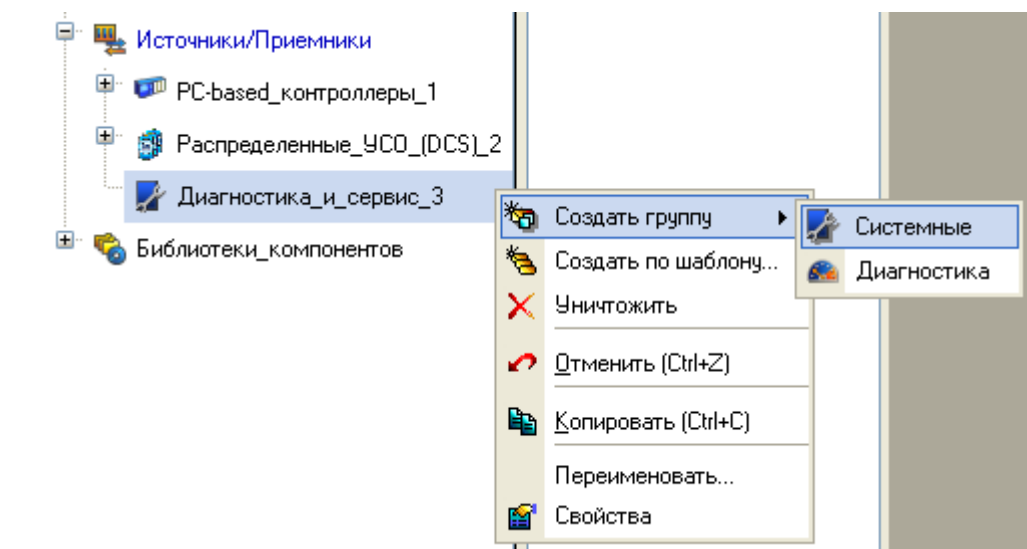
Таким образом, вся аппаратура ввода/вывода подключена к контроллеру WinCon.

Создание и конфигурирование системных переменных

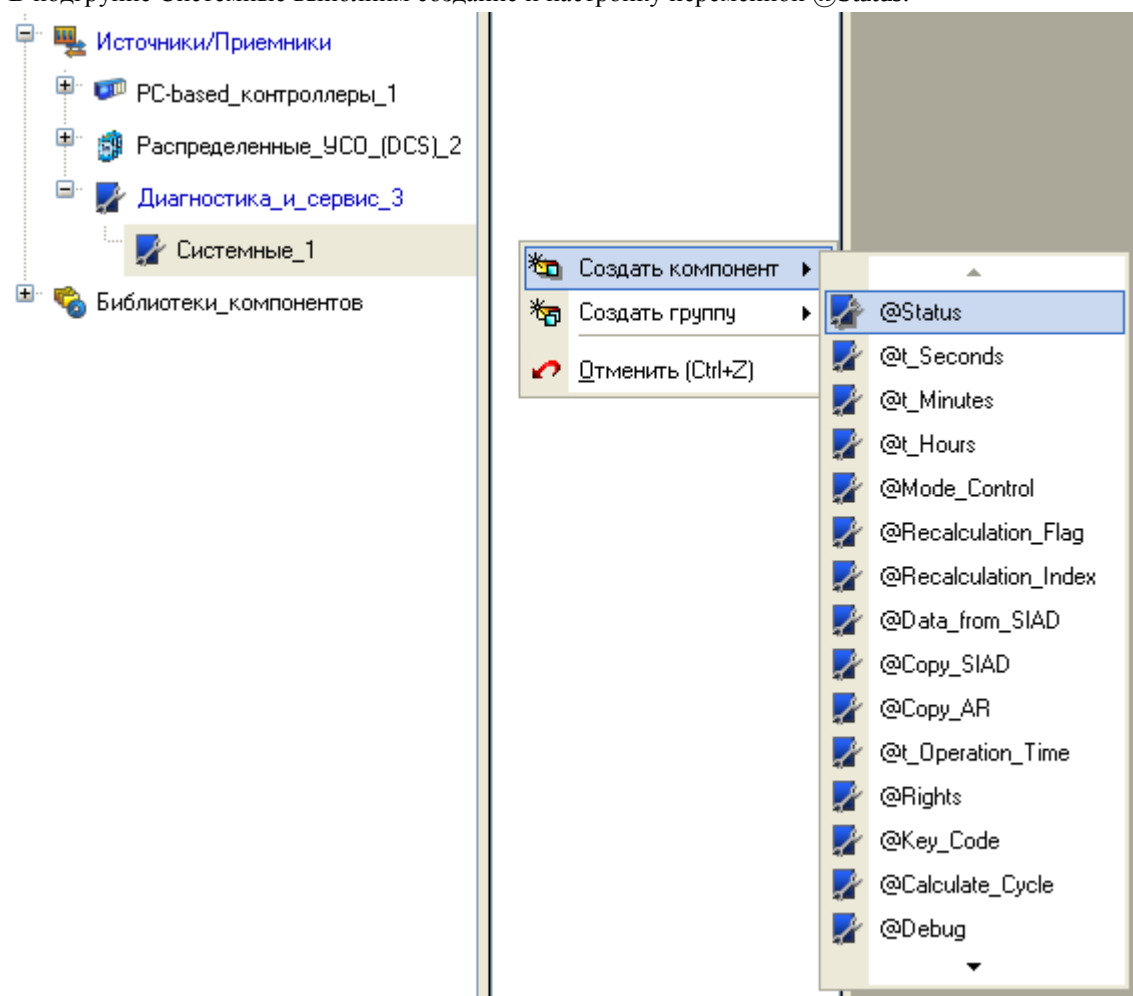
Создадим в слое Источники/Приемники группу Диагностика_и_сервис:



в ней – подгруппу Системные:



В подгруппе Системные выполним создание и настройку переменной @Status:



Двойным щелчком ЛК откроем ее на редактирование и изменим имя на Status_Contr:

Скриншот окна конфигурации 'Status_Contr*'. Вкладка 'Основные' содержит поля: 'Имя' (Status_Contr), 'Кодировка' (TW0), 'Комментарий' (пустое поле). Вкладка 'Дополнительно' содержит: 'Параметр' (STATUS), 'Номер узла' (0), 'Тип' (Input).

Номер узла 0 указывает на узел MFC, а тип переменной Input – на то, что она будет отображать текущее состояние данного узла: 0 – не работает (NO), 1 – запускается (START), 2 – работает в нормальном режиме (WORK), 3 – в подчиненном (TRACE), 4 – в состоянии ожидания (WAIT), 5 – останавливается (STOP).

Если указать в качестве номера узла 256, то связанный с данной системной переменной канал будет индексировать статус того узла, в базе каналов которого он размещен.

Далее создадим аналогичную переменную с именем Status_MAIN для узла операторской станции RTM_2:

Скриншот окна конфигурации 'Status_MAIN*'. Вкладка 'Основные' содержит поля: 'Имя' (Status_MAIN), 'Кодировка' (TW0), 'Комментарий' (пустое поле). Вкладка 'Дополнительно' содержит: 'Параметр' (STATUS), 'Номер узла' (1), 'Тип' (Input).

и такую же переменную для будущего резервного узла для операторской станции с именем Status_HOT, настроив ее на номер узла 2:

Скриншот окна конфигурации 'Status_HOT*'. Вкладка 'Основные' содержит поля: 'Имя' (Status_HOT), 'Кодировка' (TW0), 'Комментарий' (пустое поле). Вкладка 'Дополнительно' содержит: 'Параметр' (STATUS), 'Номер узла' (2), 'Тип' (Input).

Для узлов 1 и 2 создадим переменные, индицирующие IP-адреса сетевых адаптеров с именами IP_MAIN и IP_HOT:

Status_Contr* Status_MAIN* Status_HOT* IP_MAIN*

Основные
 Имя: IP_MAIN Справка
 Кодировка: Tw0
 Комментарий:

Дополнительно
 Параметр: IP_Adress Номер узла: 1 Тип: Input

и

Status_Contr* Status_MAIN* Status_HOT* IP_MAIN* IP_HOT*

Основные
 Имя: IP_HOT Справка
 Кодировка: Tw0
 Комментарий:

Дополнительно
 Параметр: IP_Adress Номер узла: 2 Тип: Input

Для реализации возможности принудительного переключения статуса узла создадим для узлов 1 и 2 системные переменные типа Output - CONTROL_MAIN и CONTROL_HOT:

Status_Contr* Status_MAIN* Status_HOT* IP_MAIN* IP_HOT* CONTROL_M

Основные
 Имя: CONTROL_MAIN Справка
 Кодировка: Tw0
 Комментарий:

Дополнительно
 Параметр: STATUS Номер узла: 1 Тип: Output

и

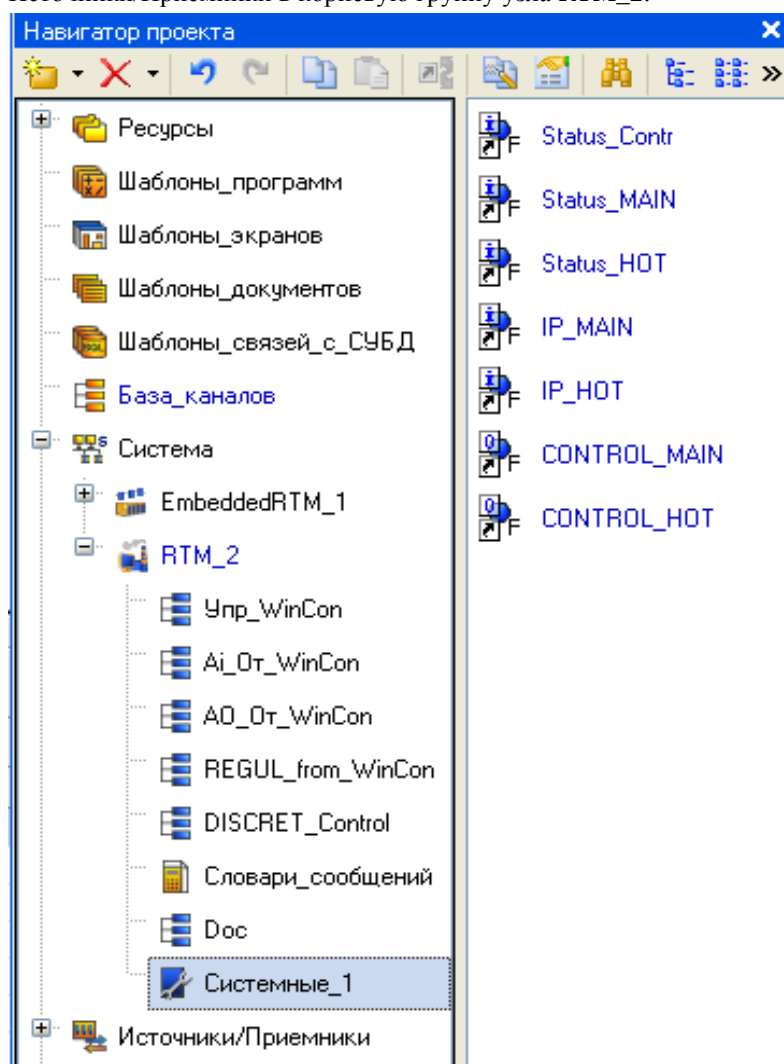
Status_Contr* Status_MAIN* Status_HOT* IP_MAIN* IP_HOT* CONTROL_M

Основные
 Имя: CONTROL_HOT Справка
 Кодировка: Tw0
 Комментарий:

Дополнительно
 Параметр: STATUS Номер узла: 2 Тип: Output

Создание системных каналов в узле RTM_2

Воспользуемся механизмом drag'n'drop для автопостроения каналов, связанных с созданными системными переменными в узле RTM_2, перенеся группу Системные_1 из слоя Источники/Приемники в корневую группу узла RTM_2:

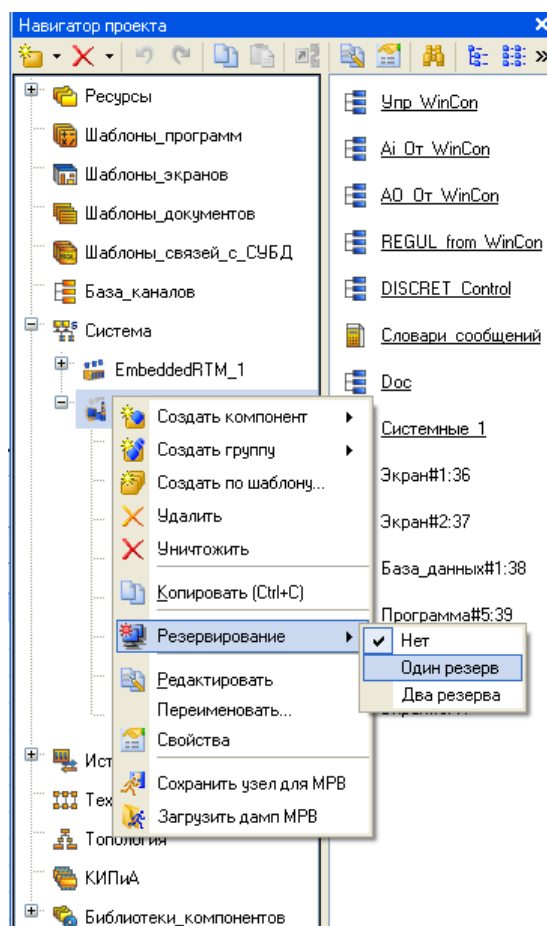


Можно изменить класс созданных каналов с Float на HEX16 путем выставления флага HEX в свойствах каналов группы Системные_1.

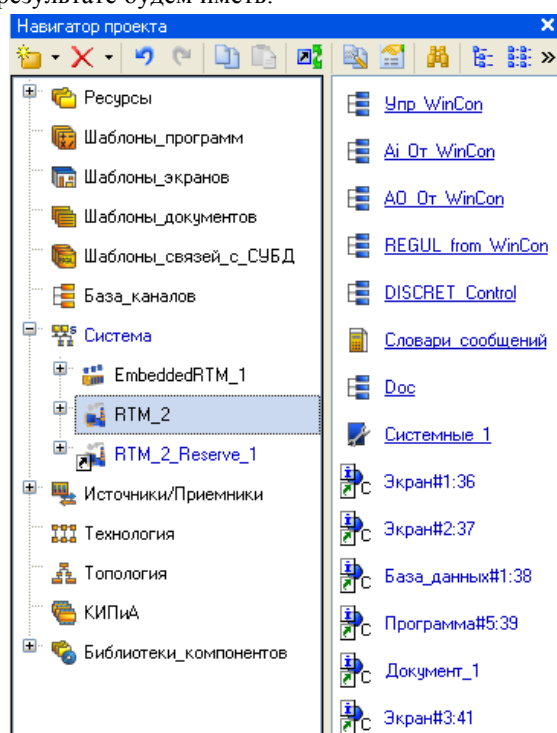
На этом подготовительная работа завершена.

Создание резервного узла

Для создания резервного узла для операторской станции RTM_2 выберем ее в дереве проекта и с помощью контекстного меню выполним данную операцию:



В результате будем иметь:



Запуск узлов необходимо производить под управлением мониторов реального времени Double Force RTM.

Откроем на редактирование созданный резервный узел и настроим для него сетевой адаптер и, если это необходимо, период пересчета, IP-адрес и код проекта:

Во вкладке **Архивы** выполним следующие настройки:

Во вкладке **Отчет тревог/Дамп/Параметры**:

Остальные настройки для узла RTM_2_Reserve_1 оставим заданными по умолчанию, т.е. переключение статуса резервного узла из состояния TRACE в состояние WORK будет осуществляться автоматически в случае отсутствия в сети основного партнера по резерву через 20 секунд. В свою очередь, узел операторской станции (RTM_2 или RTM_2_Reserve_1), стартующий первым будет основным партнером по резерву, стартующий вторым узел – горячим резервом.

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

Лабораторная работа № 7.

Использование технологии GSM SMS

1. Цель работы

Создать систему оповещения персонала на базе GSM сети с использованием службы коротких сообщений (SMS).

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

Создаваемая система оповещения должна включать различные виды взаимодействия с мобильным пользователем:

- отсылку произвольных сообщений (SMS) на телефон мобильного пользователя с АРМ оператора;
- отсылку сообщения, сформированного средствами языка ST и содержащего информацию о текущем состоянии технологического объекта;
- обработку входящих SMS средствами языка ST, отсылка ответного сообщения SMS по запросу мобильного пользователя;
- рассылку сообщений из отчета тревог на мобильные телефоны.

Настройка узла RTM

Создайте в новом проекте TRACE MODE узел RTM. В настройках узла укажите следующие параметры:

номер телефона SIM-карты, установленной в GSM-модеме – в поле «Телефон №1»;

PIN-код и номер SMS-центра – в поле «Строка инициализации», как это показано на рисунке:

Настройка мобильного пользователя ТМ

В том же узле RTM_1 создайте группу «Пользователи_ТМ» и в ней одного пользователя. В его настройках обязательно укажите:

- логин;
- пароль;
- номер мобильного телефона (можете ввести свой личный).

Если Вы хотите, чтобы данный пользователь принимал SMS-рассылки из отчета тревог данного узла, установите флаг «Рассылка».

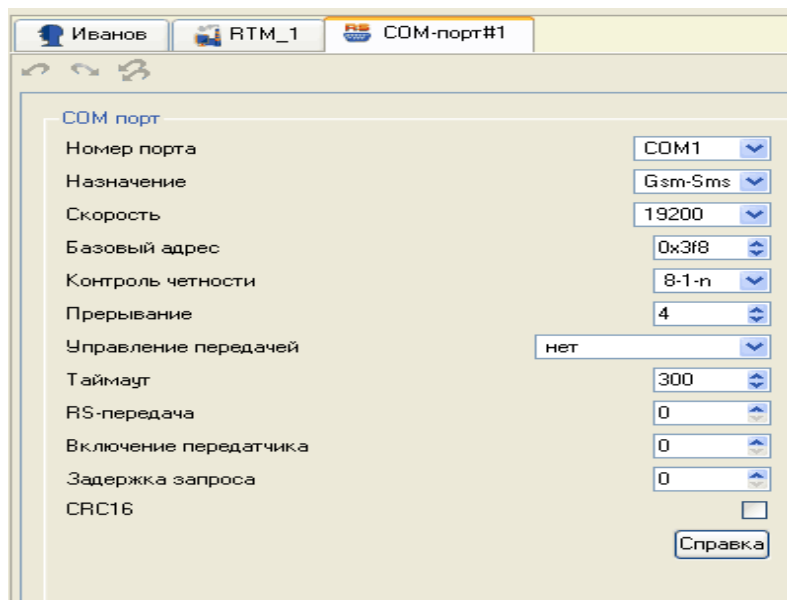
The screenshot shows the configuration window for a user named 'Иванов' (Ivanov) in the 'RTM_1' node. The interface is divided into several sections:

- General Information:** Name (Иванов), Code (TM2), Comment (оператор). A 'Справка' (Help) button is present.
- Авторизация (Authorization):** Login (operator), Password (password).
- Система (System):** Checkboxes for Запуск, Останов, Выход, Изменение, and Квотирование, all of which are checked.
- Доступ (Access):** Grids for 'Экраны' (Screens) and 'Формы' (Forms) showing access levels for 16 screens and 9 forms, with checkboxes for '8' and '1'.
- Логин (Login):** Checkboxes for 'Рассылка' (SMS) and 'Авто-logout', both checked.
- Пользователи (Users):** Checkboxes for Редактирование, Добавление, and Удаление, all unchecked.
- SCADA:** Checkboxes for Редактирование, Добавление, and Удаление, all checked.
- T-Factory:** Checkboxes for Редактирование, Добавление, and Удаление, all unchecked.
- Системные (System):** Basic settings including Type (Input), Size, Period (1), Unit (цикл), Auto-send (unchecked), Index, Process (unchecked), and Start (0).
- Архивация (Archiving):** Additional settings.
- Персональные (Personal):** Phone (+79030059283) and Email fields.

Настройка последовательного (COM) порта.

В том же узле RTM_1 создайте группу «COM-порты», при этом в ней автоматически создастся первый COM-порт. В его настройках укажите:

- номер порта (COM1);
- назначение (**Gsm-Sms**);
- скорость передачи данных (в данном случае 19200);
- базовый адрес COM-порта (определяется автоматически, достаточно установить его отличным от нуля);
- контроль четности (в данном случае 8-1-n).

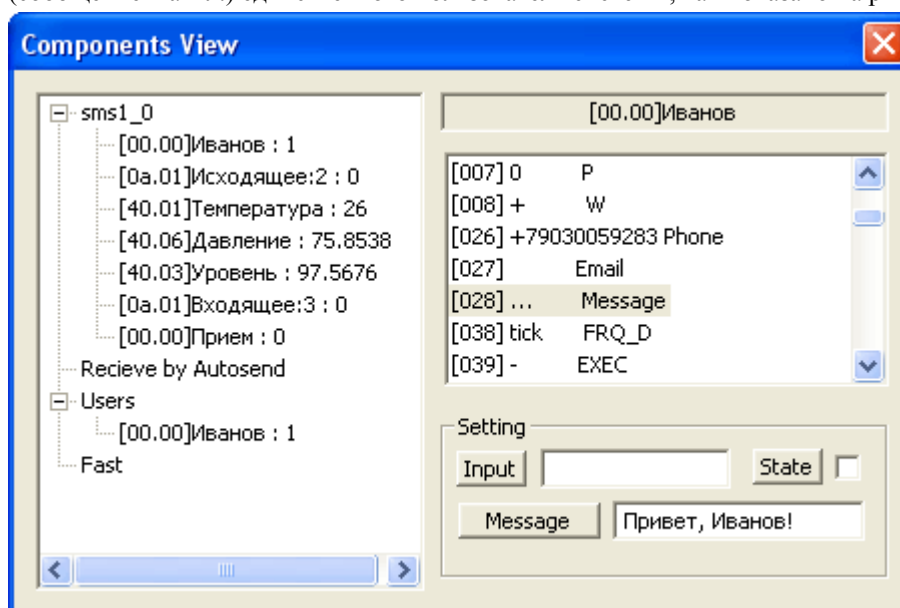


Отправка произвольного сообщения с АРМ на мобильный телефон.

Отсылка произвольных сообщений на мобильный телефон осуществляется через запись строки сообщения в атрибут №28 **Сообщение** компонента **Пользователь**.

Подключите GSM-модем к указанному последовательному порту Вашего ПК.

Сохраните Ваш проект для MPB и запустите его с помощью профайлера (для этого потребуется ввести логин и пароль, которые Вы задали в настройках пользователя). Из меню профайлера вызовите просмотр компонентов (Вид -> Компоненты). В появившемся диалоге найдите атрибут **Message** (сообщение – англ.) единственного пользователя системы, как показано на рисунке:

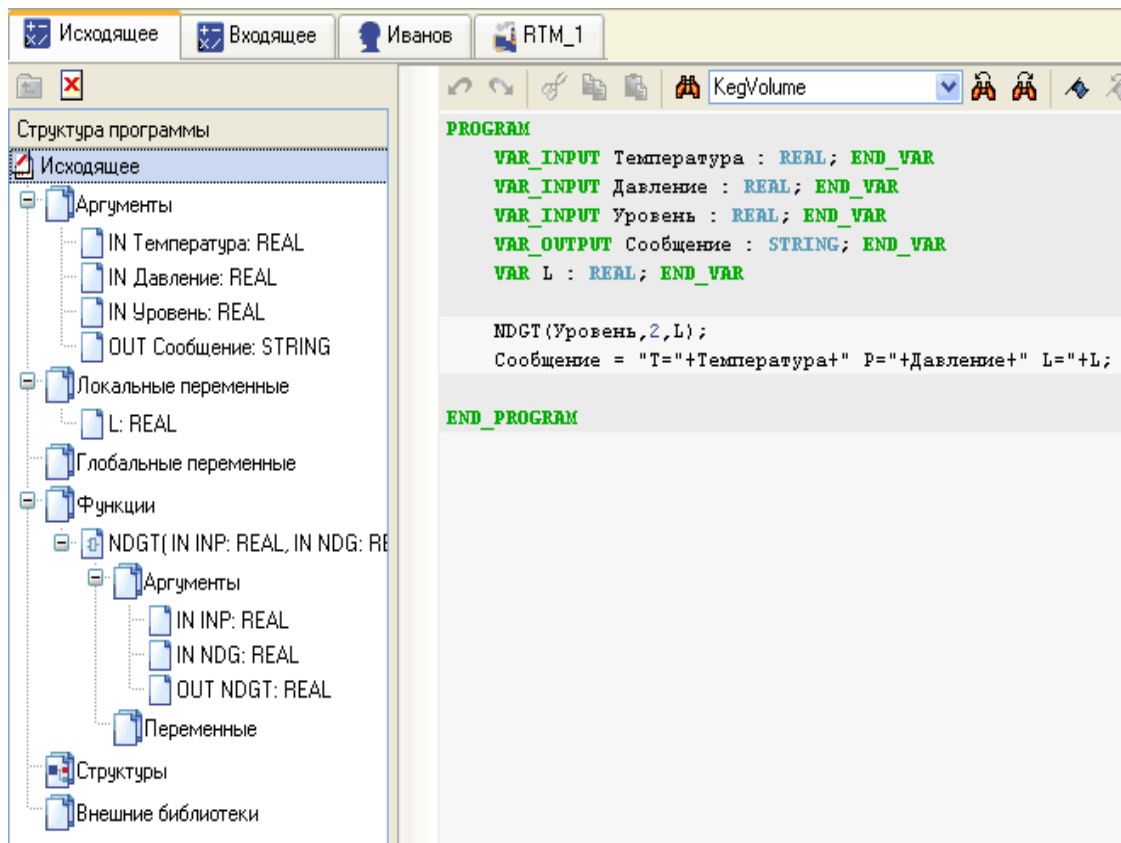


Попробуйте отправить сообщение!

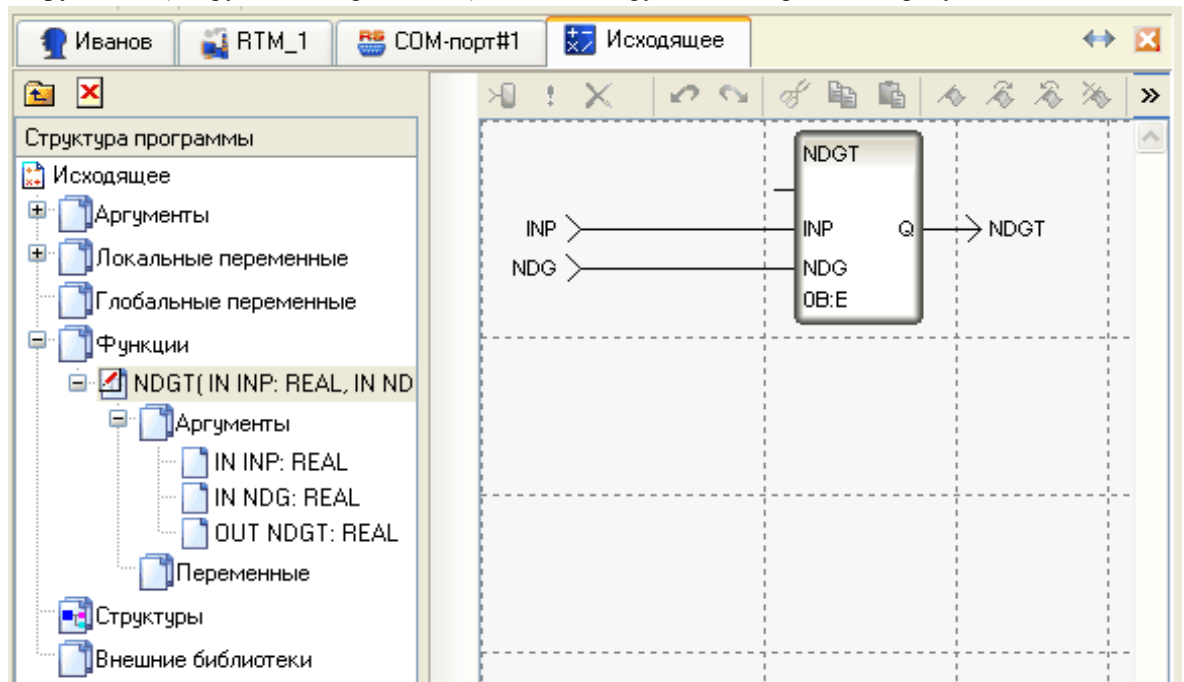
**Самостоятельная работа:* добавьте в проект экран, на котором отобразите значение атрибута "Message" и поместите кнопку с функцией «Ввод и посылка» в этот атрибут. Таким образом, оператор АРМ получит удобное средство для ручного оповещения мобильного сотрудника.

Формирование SMS сообщения средствами языка ST.

Для формирования сообщения о состоянии технологического процесса удобнее всего использовать язык ST. Создайте новую программу под названием **Исходящее** как это показано на рисунке:



В данной программе формируется сообщение (выходной аргумент типа STRING) о трех параметрах: Температуре, Давлении, и Уровне. Чтобы не перегружать SMS сообщение, в программе используется вложенная функция округления NDGT, которая написана на FBD и состоит всего из одного блока **Округление** (из группы *Алгебраические*). Вложенная функция отображена на рисунке:



Откомпилируйте программу **Исходящее** и создайте для нее вызов в узле RTM_1.

ВНИМАНИЕ! В настройках вызова программы **Исходящее** обязательно измените тип канала на **Output**. Иначе данная программа будет вызываться на каждом такте пересчета узла RTM_1 и на каждом же такте посылать новое SMS сообщение.

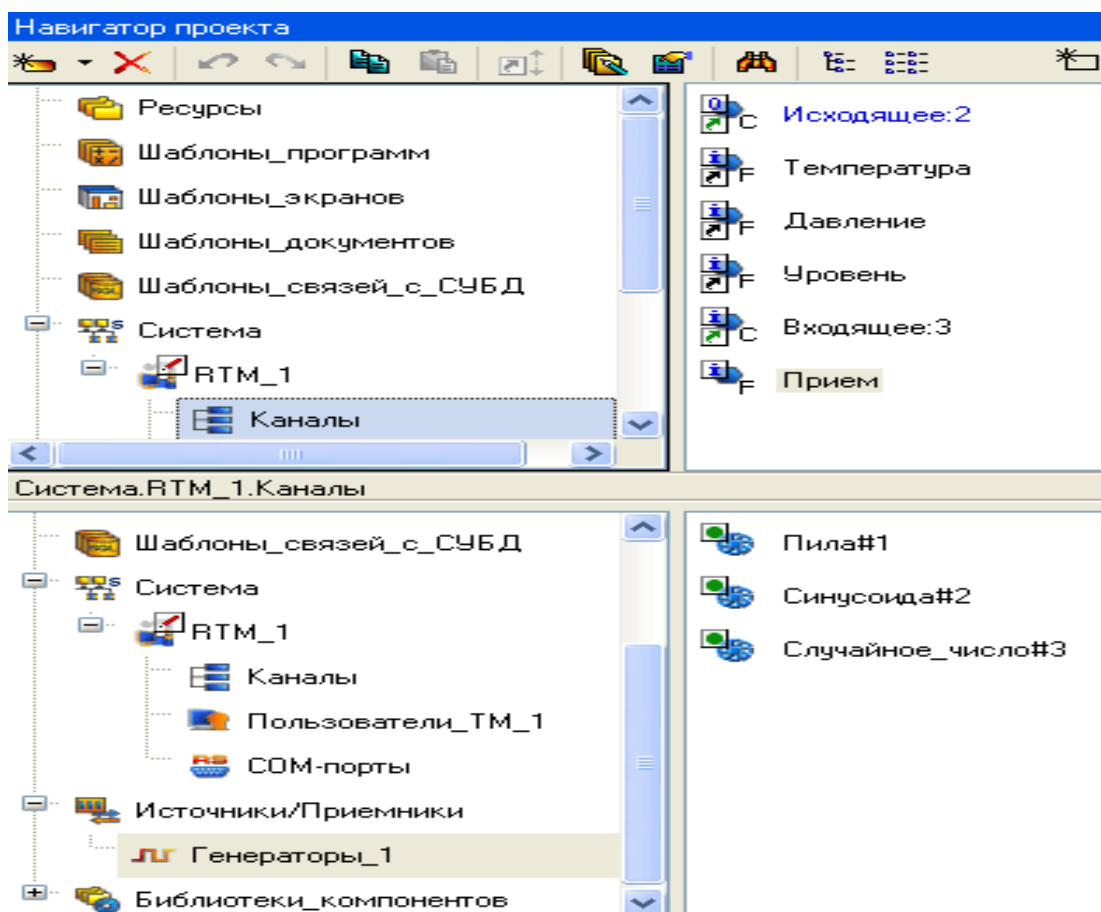
Выполните привязки вызова программы **Исходящее**:

свяжите выходной аргумент **Сообщение** с атрибутом №28 (Сообщение) компонента **Пользователь** как показано на рисунке.

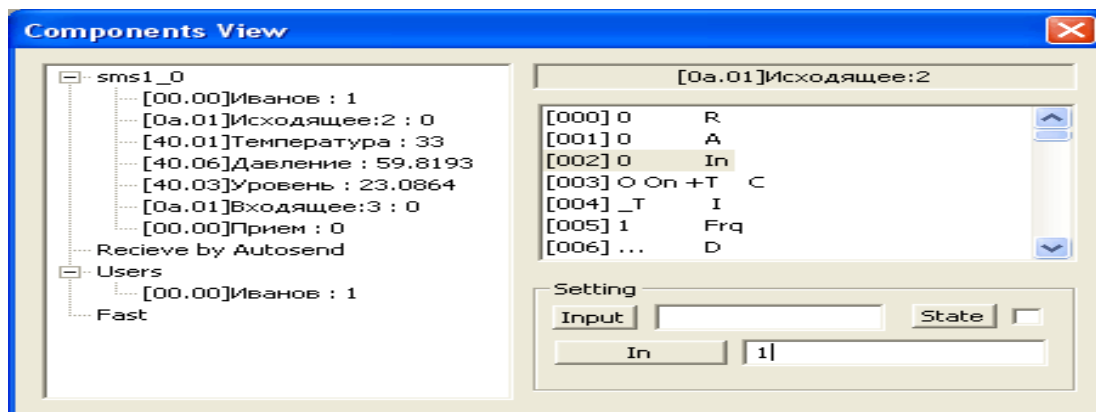
для входных аргументов **Температура**, **Давление**, **Уровень** выполните автопостроение каналов по аргументам.

Система.RTM_1.Каналы				
Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Температура	IN	REAL		Температура:Реальное значение (Система.RTM_1.Каналы)
Давление	IN	REAL		Давление:Реальное значение (Система.RTM_1.Каналы)
Уровень	IN	REAL		Уровень:Реальное значение (Система.RTM_1.Каналы)
Сообщение	OUT	STRING		Иванов:Сообщение (Система.RTM_1.Пользователи_TM_1)

Чтобы пример был нагляднее, создайте три генератора сигналов в слое Источники/Приемники, например, **Пила**, **Синусоида**, **Случайное число** и свяжите их с каналами **Температура**, **Давление**, **Уровень** соответственно.



Сохраните проект для MPV и запустите в профайлере. С помощью диалога просмотра компонентов убедитесь, что генераторы сигналов передают информацию в каналы технологических параметров. Запишите во входной атрибут канала вызова программы **Исходящее** единицу:



Если все было сделано правильно на Ваш телефон (т.е. на телефон указанный в настройках компонента Пользователь) поступит SMS сообщение следующего вида: “**T=36 P=72.0908 L=88.56**”. Такие сообщения будут передаваться при каждом изменении входного атрибута вызова программы **Исходящее**.

**Самостоятельная работа: добавьте в проект экран, на котором отобразите значение технологических параметров и поместите кнопку «Послать отчет» с функцией «передать значение (НЕ-ИЛИ)» константы 1 в вызов программы.. (Исключающее ИЛИ в данном случае используется для того, чтобы значение входного атрибута менялась при каждом нажатии кнопки.)*

Обработка входящих SMS

По умолчанию обрабатываются все входящие SMS, если их отправитель описан как пользователь данного узла. Если пользователи в системе не заданы, обрабатываются все входящие SMS с любых телефонов.

Формат SMS-сообщения для задания значения атрибута канала:

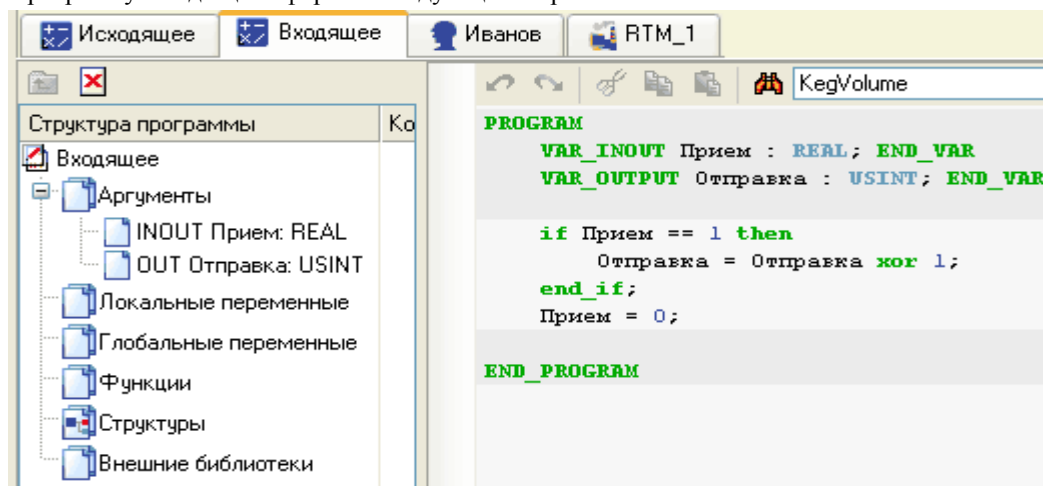
<имя канала>.<номер атрибута>=<строка>

В качестве параметра **<строка>** может использоваться строковое выражение или число. Указание номера атрибута **In** в сообщении необязательно.

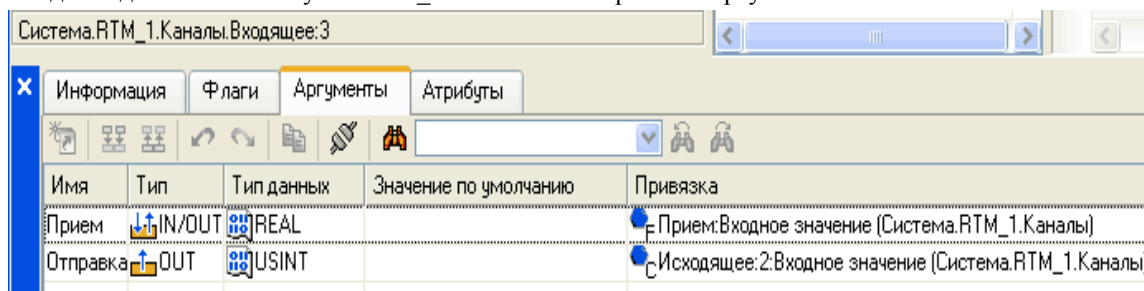
Для более детальной настройки алгоритма обработки SMS сообщений можно использовать системный канал @RTM_Parameter.

Итак, чтобы обработать входящее SMS сообщение, создадим канал **Прием** и программу **Входящее**. Поставим задачу обработки сообщения следующим образом: если в канал **Прием** записано значение 1, то выполнить программу **Исходящее**. Таким образом, отослав сообщение «Прием=1» на номер GSM-модема APM, мы должны в ответ получить сообщение о состоянии технологических параметров.

Программу **Входящее** оформим следующим образом:



Создайте для нее вызов в узле RTM_1 и выполните привязки аргументов:



Программа **Входящее** должна постоянно следить за каналом **Прием**, а значит вызываться на каждом такте пересчета узла, поэтому тип ее вызова следует оставить **Input**.

Аргумент **Отправка** свяжем с входным атрибутом вызова программы **Исходящее**, а аргумент Прием – с одноименным каналом, который можно автопостроить прямо из свойств вызова.

Сохраните проект для MPB и запустите профайлер. Откройте диалог просмотра компонентов, чтобы следить за работой программ. Пошлите SMS сообщение «Прием=1» на номер GSM-модема узла RTM_1. В ответ Вы должны получить еще одно сообщение вида “**T=36 P=72.0908 L=88.56**”.

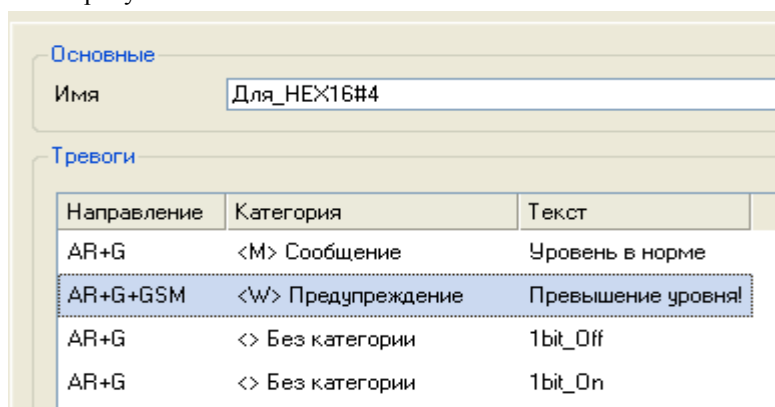
*Самостоятельная работа: попробуйте сделать еще одну программу формирования исходящего сообщения **Исходящее2**. В программе Входящее добавьте обработку еще одного запроса, например, «Прием=2», по которому отправляйте новое сообщение. Таким образом, мобильный пользователь получает возможность запрашивать различные данные по своему желанию.

Отсылка SMS сообщений из отчета тревог.

Вы можете назначить рассылку любого сообщения из отчета тревог на телефоны мобильных пользователей. Функцию отправки SMS легко добавить в любой существующий проект.

Для этого необходимо:

- описать настройку GSM-модема в свойствах узла APM, на котором формируется отчет тревог;
- назначить COM-порт для работы с GSM SMS;
- в настройках каждого пользователя указать номер телефона и установить флаг **Рассылка**, если данный пользователь должен получать сообщения отчета тревог;
- в словаре сообщений необходимо указать направление вывода **GSM**, как это показано на рисунке:



ВНИМАНИЕ! К каждому сообщению отчета тревог добавляется информация о времени и дате возникновения события, а также имя канала. Поэтому длина SMS сообщения в словаре весьма ограничена по сравнению с максимальной длиной произвольного SMS сообщения.

**Для справки:* максимальная длина сообщения в английской кодировке составляет 160 символов. При использовании русских символов в тексте SMS, его максимальная длина сокращается до 70. Таким образом, конкретное ограничение на длину SMS-сообщения в словаре отчета тревог зависит от длины имени канала и выбранной кодировки.

Темы индивидуальных лабораторных заданий “ Теоретические основы автоматизированного управления”

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; учет медикаментов и услуг; список сотрудников с расчётом прибыли; предупреждения о запланированных приёмах; база поставщиков с балансом; различные отчёты

10. АСО и У учёта почтовых отправлений

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 «Теоретические основы автоматизированного управления»
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.	
Создание SQL-запроса для работы с внешней реляционной БД.....	4
ЛАБОРАТОРНАЯ РАБОТА № 2.	
Создание шаблона документа с заданными параметрами.....	15
ЛАБОРАТОРНАЯ РАБОТА № 3.	
Генерация документов.....	25
ЛАБОРАТОРНАЯ РАБОТА № 4.	
Создание проекта распределенной системы.....	30
ЛАБОРАТОРНАЯ РАБОТА № 5.	
Взаимодействие с технологической базой данных.....	40
ЛАБОРАТОРНАЯ РАБОТА № 6.	
Резервирование узла АРМ.....	51
ЛАБОРАТОРНАЯ РАБОТА № 7.	
Использование технологий GSM SMS.....	59
ТЕМЫ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ.....	66
Требования к отчету.....	67
СПИСОК ЛИТЕРАТУРЫ.....	68
СОДЕРЖАНИЕ.....	69

Учебно-методическое пособие

Изосимова Татьяна Анатольевна

Игошкина Наталия Геннадьевна

Максимова Алина Петровна

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ

Лабораторный практикум

Компьютерная верстка: Т.А. Изосимова, Н.Г. Игошкина

Подписано в печать . Формат

Бумага писчая. Печать оперативная. Усл. печ.л.9,25 .

Тираж 100 экз. Заказ № . Цена свободная

Волжский филиал ФГБОУ ВПО «Московский автомобильно-дорожный государственный
технический университет (МАДИ)

428024, г. Чебоксары, пр. Тракторостроителей, 101, корпус 30