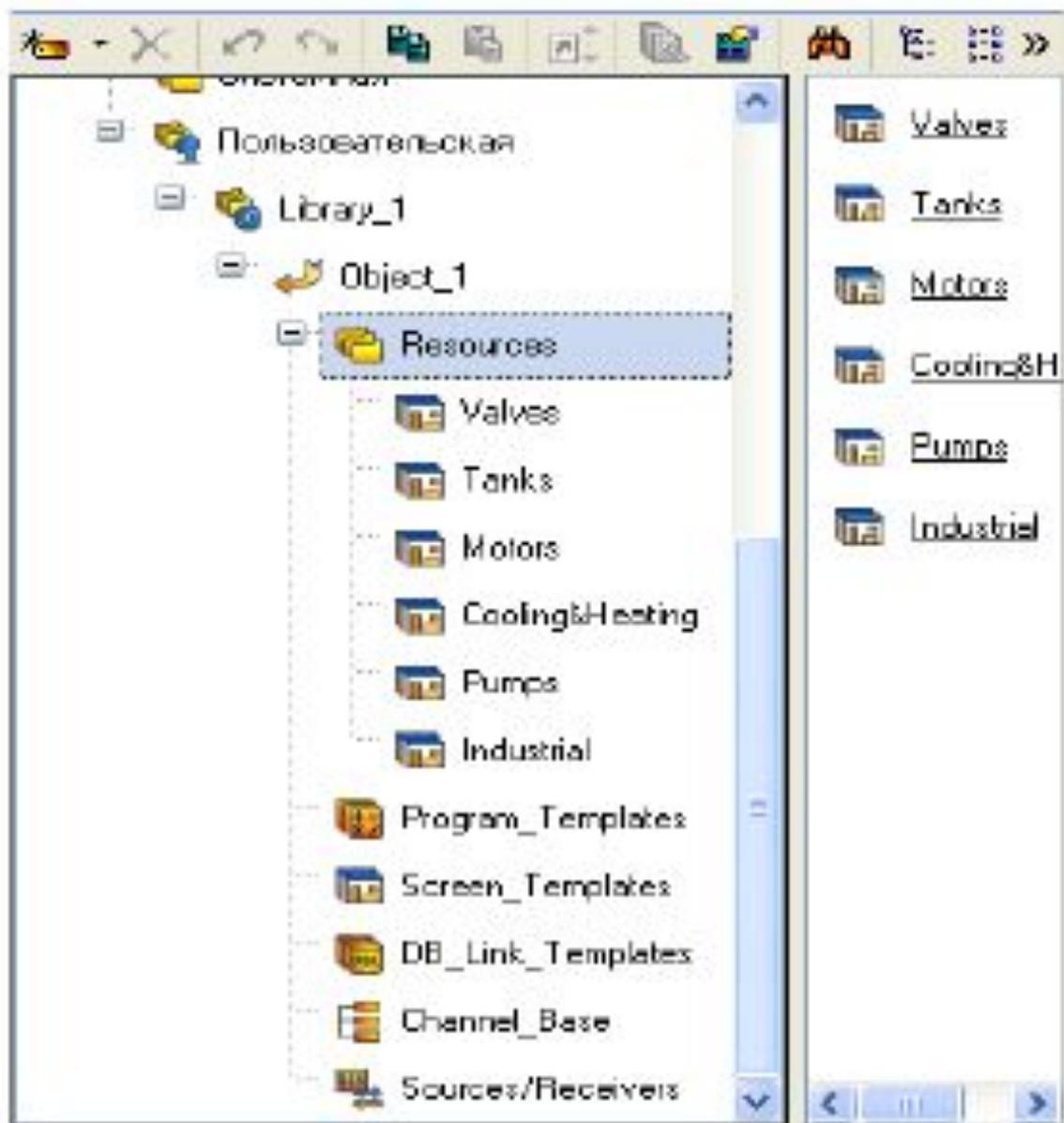


# Создание проекта

## Создание шаблона экранов.

Откроем систему разработки и с помощью нажатия ЛК мыши на иконку создадим новый проект. В качестве стиля разработки выберем **Стандартный**. Пример набора графических объектов показан на рис.



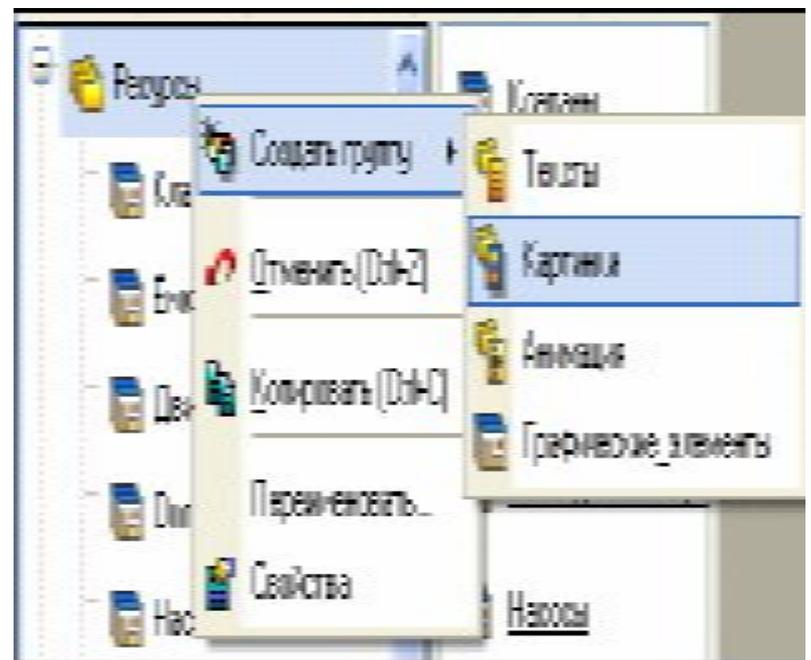
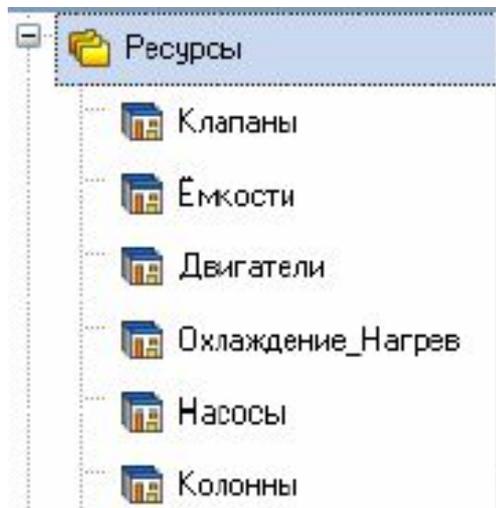


Перейдем в слой “**Библиотеки\_компонентов**”, где в разделе “**Пользовательская**” откроем библиотеку `Library_1`. Сохраненный в данной библиотеке объект `Object_1` содержит в своем слое “**Resources**”

(Ресурсы) необходимый для дальнейшей разработки набор графических объектов: таких как изображения двигателей, емкостей, клапанов и т.д.

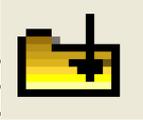
Перенесем группы в слой **Ресурсы** текущего проекта с помощью механизма **drag-and-drop** и переименуем их.

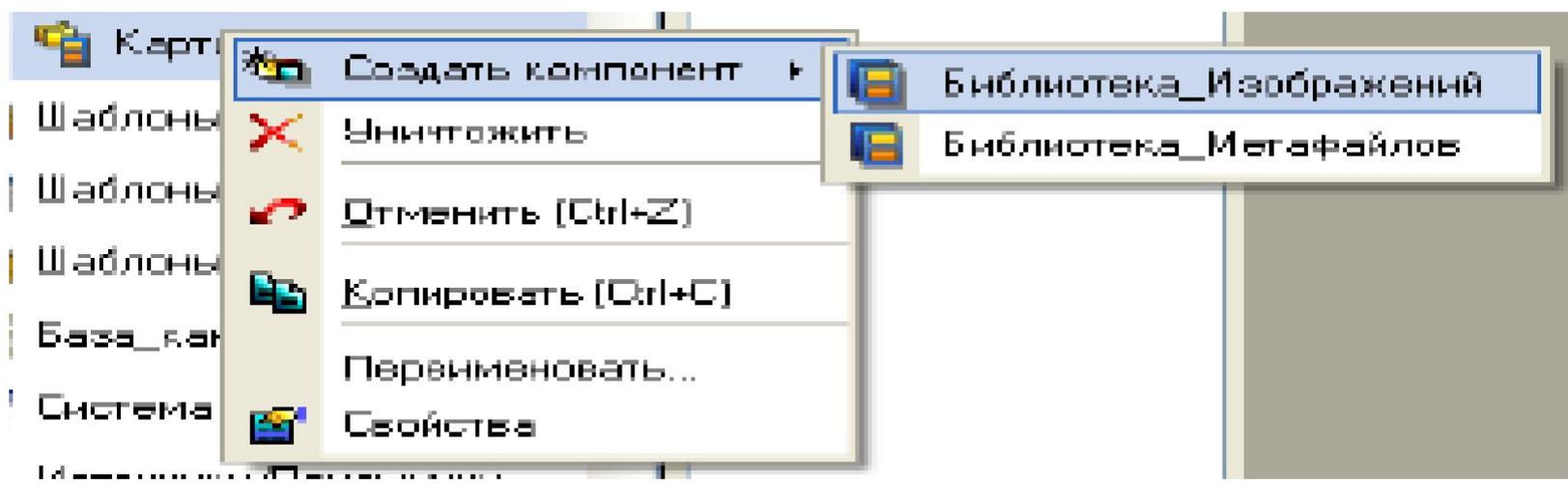
Здесь же в слое **Ресурсы** создадим группу **Картинки** для помещения в нее текстур, которые будут использованы в оформлении создаваемых графических экранов, и группу **Анимация** для использования движущихся объектов (например, **Огонь**, изображающий включенный нагреватель).



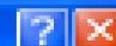
Создадим в группе **Картинки** новый компонент – **Библиотека\_Изображений#1**.

Диалоговое окно примет вид, показанный на рис. Откроем двойным щелчком ЛК вновь созданную библиотеку для редактирования. Для ее заполнения воспользуемся иконкой

на панели инструментов.  Открывшемся диалоге выбора файлов для импорта укажем поддиректорию **Trace Mode IDE 6 Base\Lib\Texture**, выберем все файлы и нажмем кнопку **Открыть**



Выбрать файл для импорта



Папка: Texture



Недавние документы



Рабочий стол



Мои документы



Мой компьютер



Сетевое окружение

- brick\_022
- ice\_016
- metal\_011
- stone\_001
- water\_020

Имя файла: "brick\_022.bmp" "ice\_016.bmp" "metal\_011.b

Тип файлов: Рисунок (\*.bmp;\*.png;\*.xpm;\*.ico)

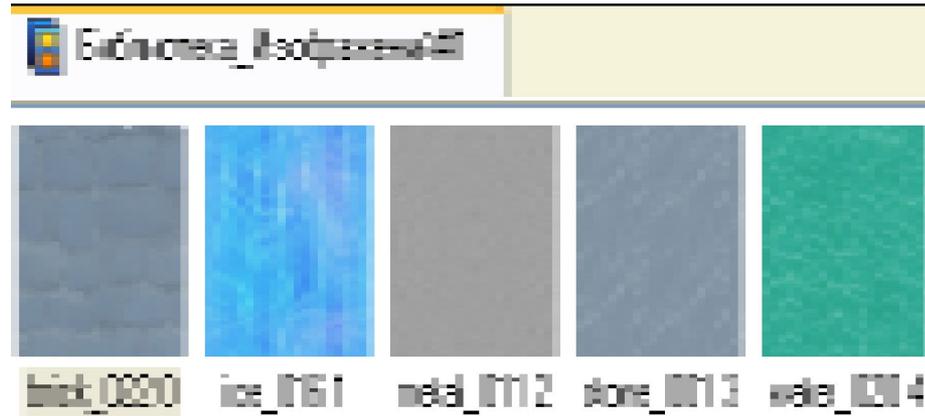
Открыть

Отмена

Содержимое

библиотеки

**Библиотека\_Изображений#1** будет иметь вид,  
показанный на рис.



Проект имеет древовидную структуру. В него входят следующие компоненты, приведенные на рис.



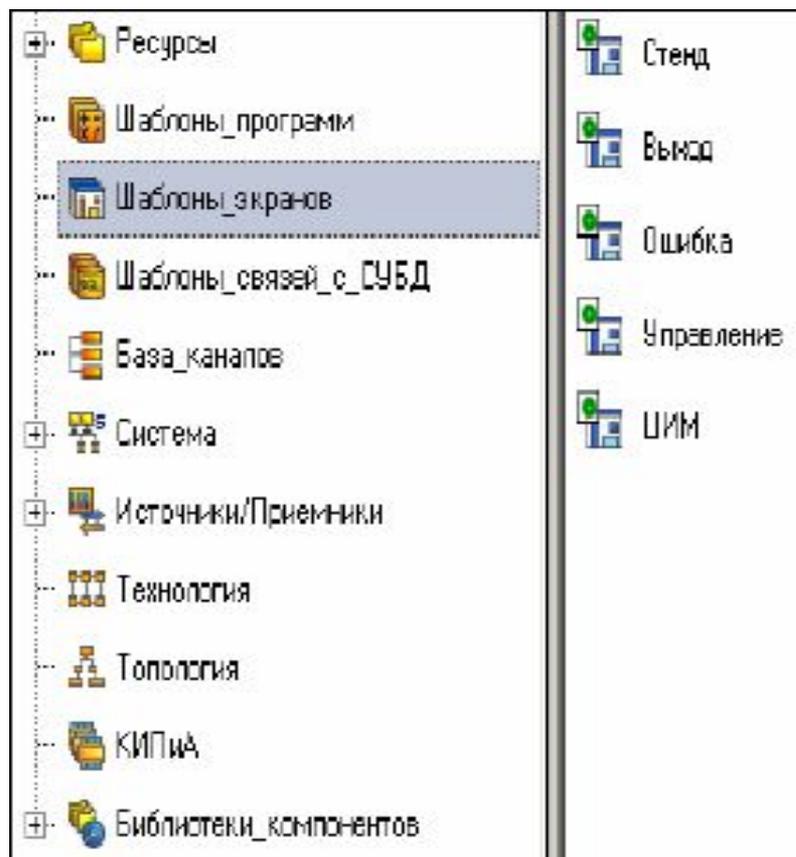
**Ресурсы** – содержат библиотеки картинок, загружаемых из библиотеки компонентов.

**Шаблоны программ** – содержат алгоритмы FVD-программ, используемых в проекте.

**Шаблоны экранов** – содержат статические изображения мнемосхем, которые представляют проект в мониторе реального времени (МРВ). В **базе каналов** находятся все каналы, необходимые для обмена информацией между компонентами проекта. **Система** – отображает все узлы проекта (МРВ, микроМРВ).

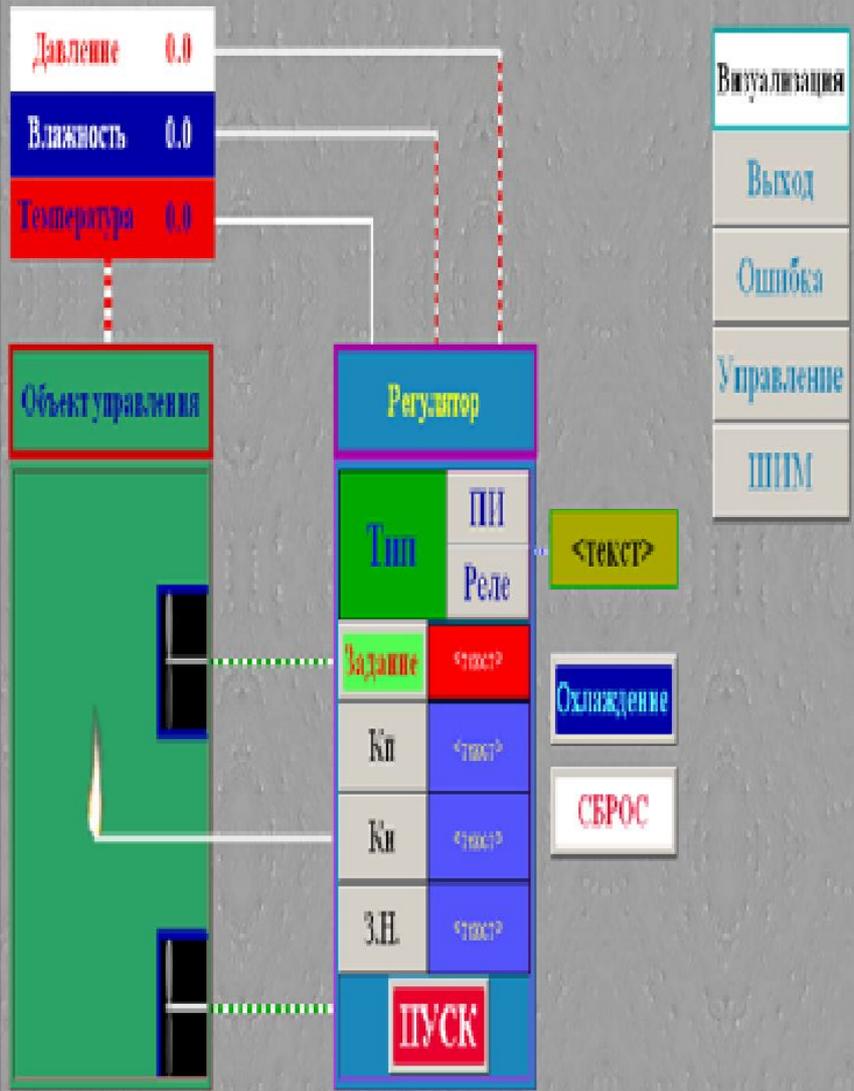
**Источники/приёмники** – предназначены для связи МРВ с объектом (УСО).

В шаблонах экранов созданы пять экранов: стенд, выход, ошибка, управление и ШИМ, как показано на рис.



Опишем назначение каждого из них и приведём параметры элементов.

Экран **стенд**: отображает мнемосхему, представляющую понятную оператору схему процесса управления, позволяющая контролировать процесс и вносить изменения в параметры. Мнемосхема приведена на рис.



Информация		Аргументы			
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Input	IN	REAL			
Zdn	OUT	REAL			
Kp	OUT	REAL			
Ki	OUT	REAL			
Dk	OUT	REAL			
VLAG	IN	REAL			
DAVL	IN	REAL			
Mode	OUT	REAL			
Start	OUT	REAL			
Doroga	IN	REAL			
Nagrev	IN	REAL			
Vent1	IN	REAL			
Vent2	IN	REAL			
Time	IN	DATE_AND_TIME			NP
Oxlajdenie	OUT	REAL			
Reset	OUT	REAL			

Аргументами экрана, приведенного на рис., являются:

- **Input** – показания датчика температуры в объекте управления;
- **Zdn** – задание регулятору;
- **Kp** – коэффициент пропорциональности регулятора;
- **Ki** – коэффициент интегрирования регулятора;
- **Dlt** – величина зоны нечувствительности;
- **VLAG** – показания датчика влажности в объекте управления;

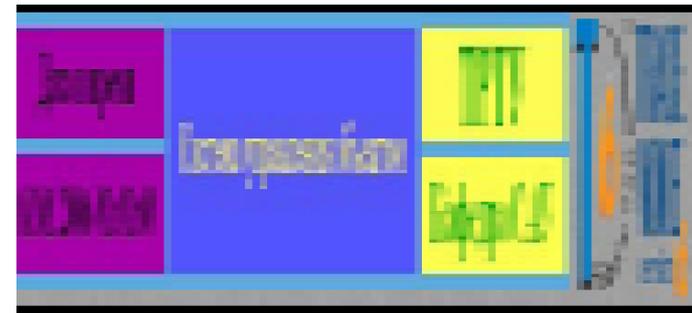
- **DAVL** – показания датчика атмосферного давления в объекте управления;
- **Mode** – выбор регулятора;
- **Start** – разрешает запись в теги;
- **Doroga** – обеспечивает динамическое движение бегущей дорожки на мнемосхеме;
- **Nagrev** – передаёт значение в канал нагревателя;

- **Time** – предназначен для вывода астрономического времени на мнемосхеме;
- **Ochlajdenie** – позволяет снизить температуру в объекте управления после нагревания;
- **Reset** – записывает ноль во все теги.

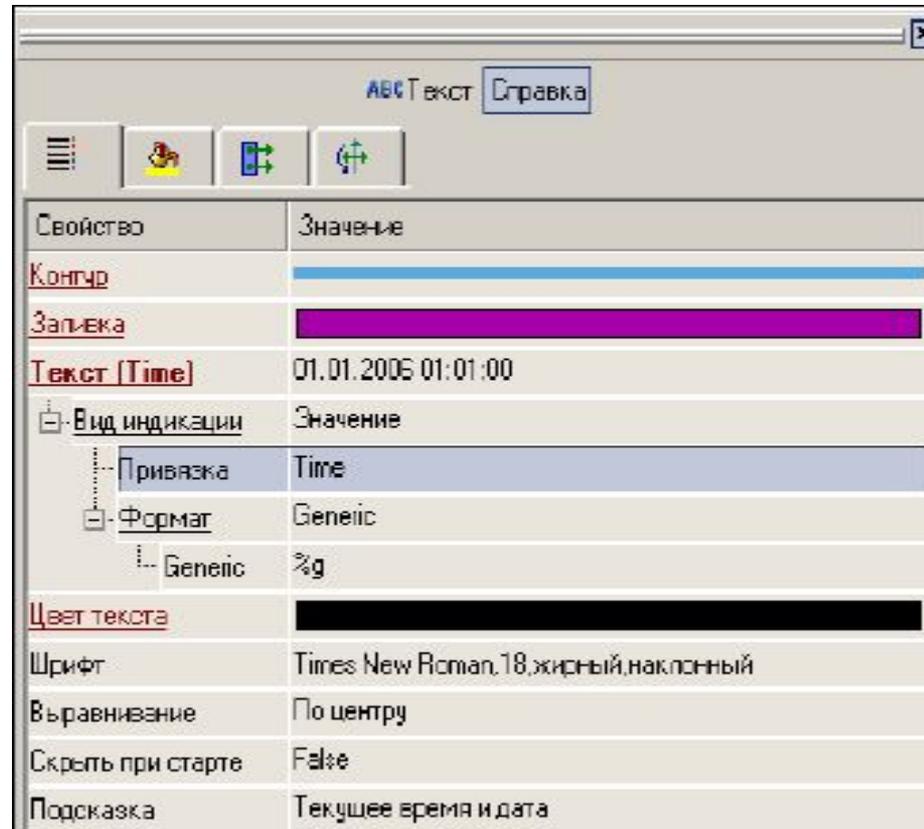
Тип **IN** означает, что данные считываются из канала, а тип **OUT** - записываются в канал.

Тип данных **Real** – реальное значение данных,  
тип данных **DATE AND TIME** – системное  
время, которое отображается на мнемосхеме.  
Флаг **NP** означает, что этот аргумент не  
участвует в автопостроении каналов.

Графические элементы (ГЭ), приведенные на  
рис., являются статическими, кроме элемента  
отображения даты и времени.



Они являются динамическими, привязанные к аргументу Time и имеют настройки, показанные на рис.



# Графичечкие элементы

Давление	0.0
Влажность	0.0
Температура	0.0

отображают

показания датчиков и привязываются так, как это показано на следующих трех рис.

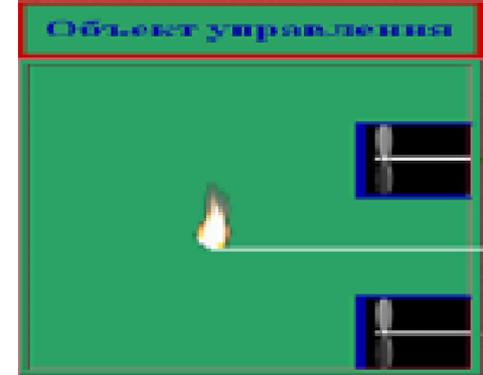
Свойство	Значение
<b>Контур</b>	
<b>Заливка</b>	
<b>Текст (DAVL)</b>	0.0
[-] Вид индикации	Значение
Привязка	DAVL
[-] Формат	Float
Float	%1f
<b>Цвет текста</b>	
Шрифт	Times New Roman, 16, жи
Выравнивание	По центру
Скрыть при старте	False
Подсказка	Давление

Свойство	Значение
<b>Контур</b>	
<b>Заливка</b>	
<b>Текст (Input)</b>	0.0
[-] Вид индикации	Значение
Привязка	Input
[-] Формат	Float
Float	%1f
<b>Цвет текста</b>	
Шрифт	Times New Roman, 16, жирный
Выравнивание	По центру
Скрыть при старте	False
Подсказка	Температура

Свойство	Значение
<b>Контур</b>	
<b>Заливка</b>	
<b>Текст (Input)</b>	0.0
[-] Вид индикации	Значение
Привязка	Input
[-] Формат	Float
Float	%1f
<b>Цвет текста</b>	
Шрифт	Times New Roman, 16, жирный
Выравнивание	По центру
Скрыть при старте	False
Подсказка	Температура

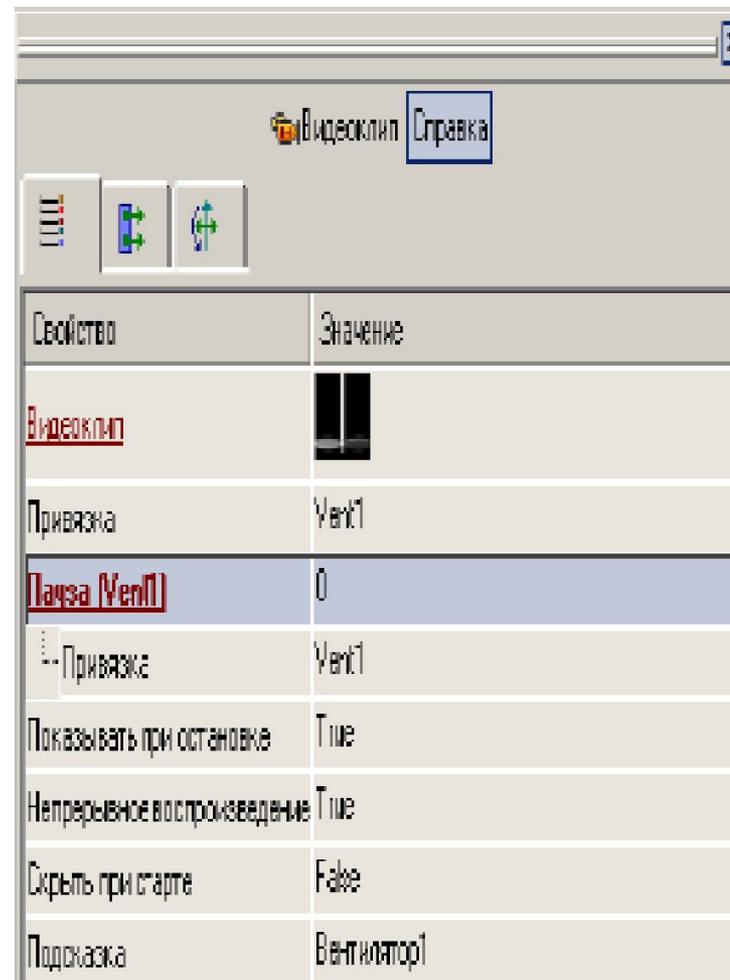
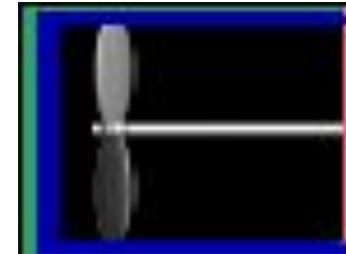
# Графический элемент

содержит три динамических элемента. Они содержатся в ресурсах проекта.



# Графический элемент

имеет настройки,  
показанные на рис.



Свойство	Значение
<u>Видеоклип</u>	
Привязка	Vent1
<u>Палка (Vent1)</u>	0
Привязка	Vent1
Показывать при остановке	True
Непрерывное воспроизведение	True
Скрыть при старте	False
Подсказка	Вентилятор1

# Графические элементы



являются «кнопками».

Графический элемент,

имитирует работу регулятора.

A detailed graphic element for a regulator control panel. It has a blue header bar with the word 'Регулятор' in yellow. Below the header is a table with two columns. The first column has a green header 'Тип' and three rows with labels 'Задание', 'Кп', 'Кп', and 'З.Н.'. The second column has a grey header with 'ПИ' and 'Реле' stacked, and three rows with '<текст>' labels. At the bottom is a red button with the word 'ПУСК' in white.

Регулятор	
Тип	ПИ
	Реле
Задание	<текст>
Кп	<текст>
Кп	<текст>
З.Н.	<текст>
ПУСК	

При выборе кнопок открываются диалоговые окна.  
При выборе кнопки «ПИ» открывается диалоговое окно.

При выборе кнопки «Реле» открывается соответствующее окно.

Привязка кнопки  
показана на рис.



OK Кнопка Справка

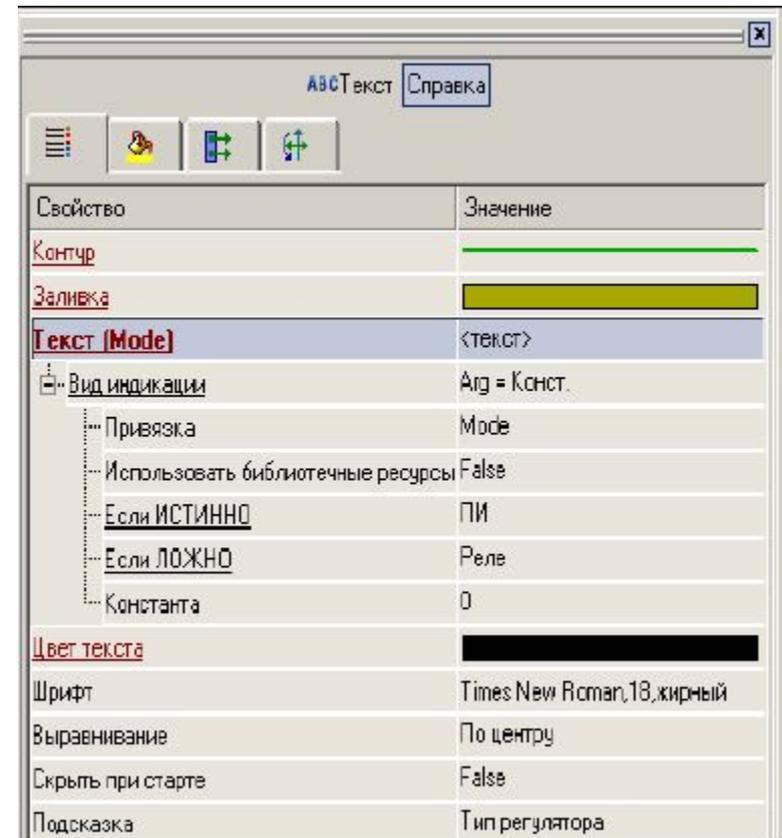
☰ ↔ ↻

Свойство	Значение
Код доступа	0
<b>События</b>	
<b>pressed</b>	
Подтверждение	False
Сигнал	False
<b>Передать значение</b>	
Тип передачи	Прямая
Значение	1
Результат	Start
Источник	...
Восстанавливать значение	False
<b>released</b>	

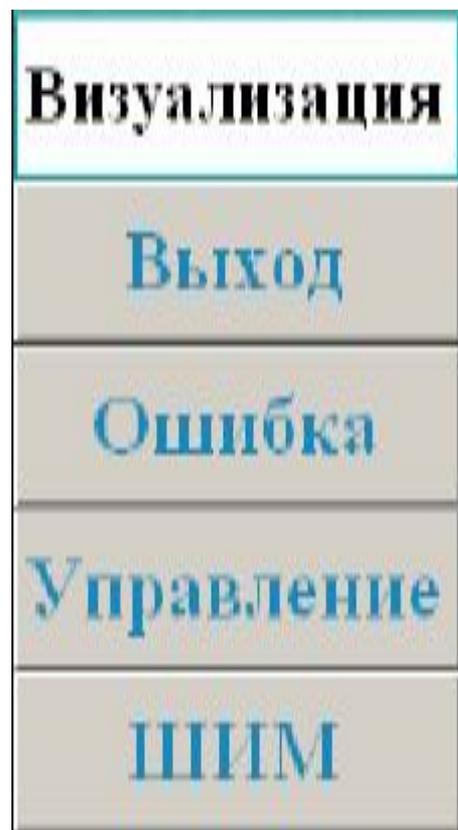
# Графический элемент «Динамический текст»



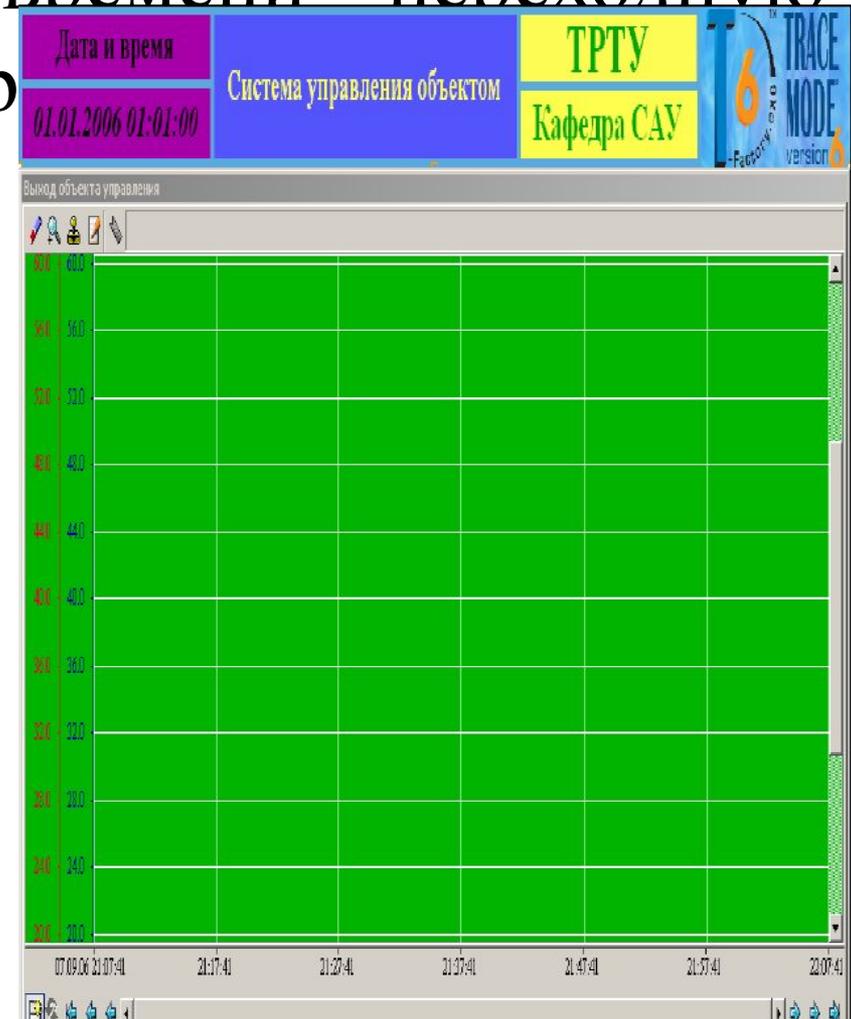
отображает тип регулятора и настраивается в диалоговом окне, вид которого показан на рис.



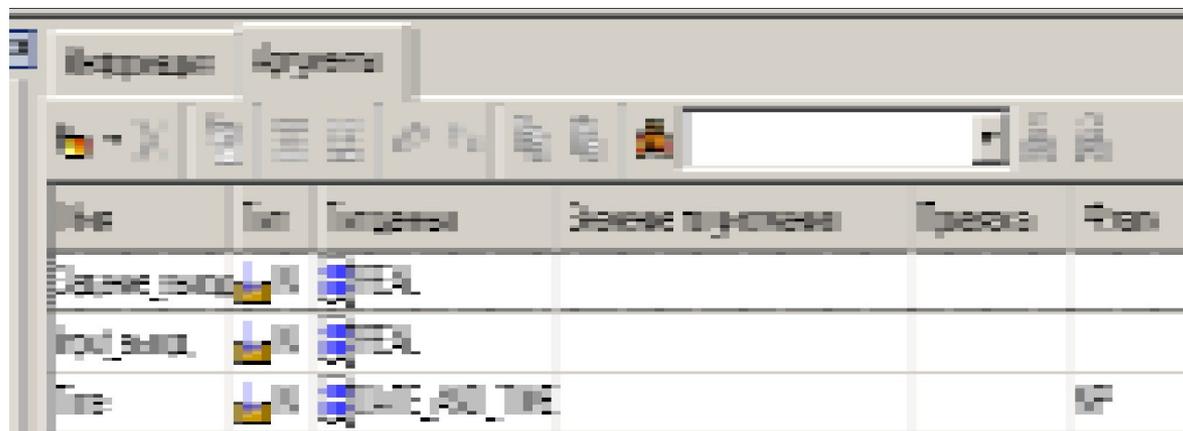
- Поскольку в проекте используются несколько экранов, предусмотрен переход между ними с помощью кнопок, показанных на рис.



Экран выход отображает в режиме реального времени показания датчика температуры и значение задания. С помощью него можно наблюдать в реальном времени переходную характеристику объекта. Пр



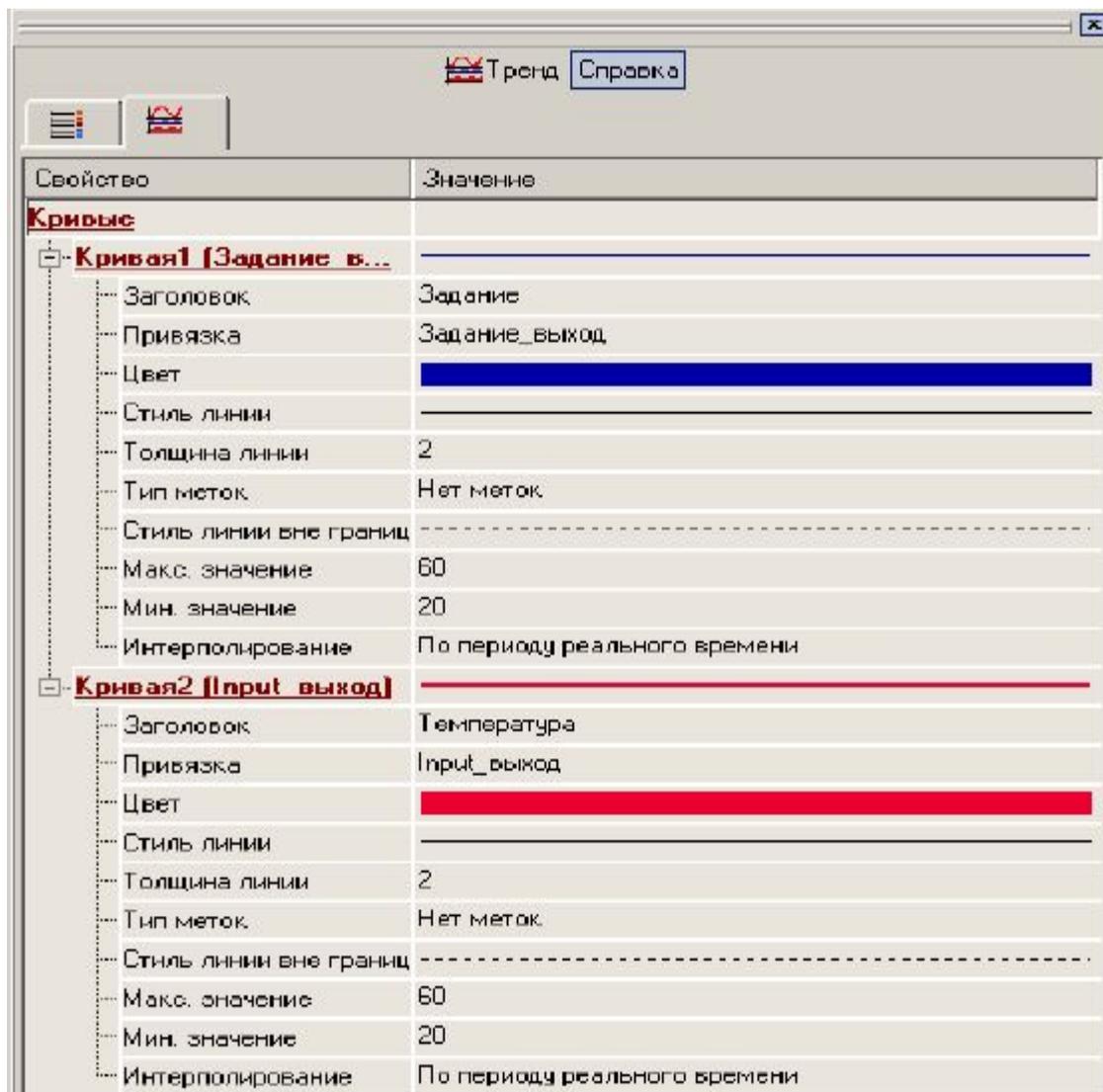
Определение аргументов данного экрана  
показано на рис.

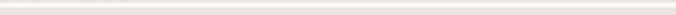


The screenshot shows a software interface with a menu bar containing 'Исходные' and 'Аргументы'. Below the menu bar is a toolbar with various icons. The main area contains a table with the following data:

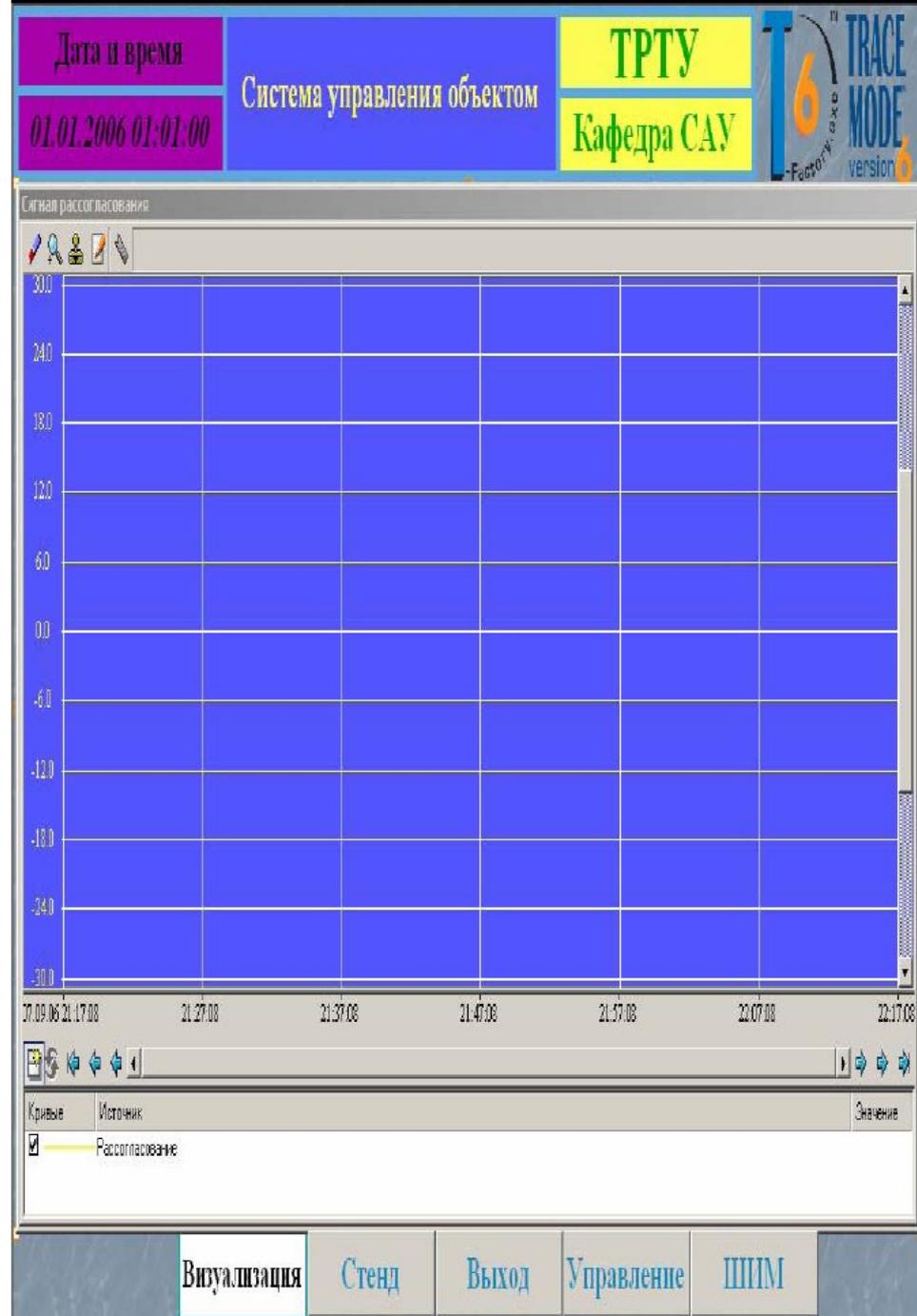
Имя	Тип	Получены	Элементы программы	Проекты	Файлы
Зачем_польз	LN	FEA			
код_зада	LN	FEA			
тип	LN	FEA_FSI_TPE			SP

# Настройки тренда показаны на рис.

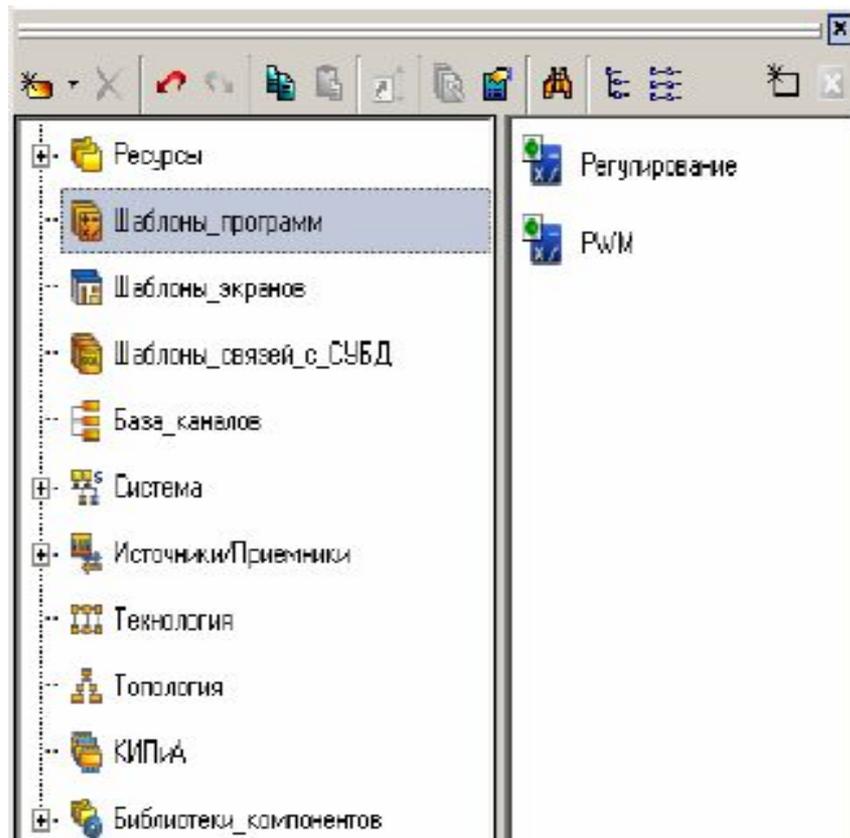


Свойство	Значение
<b>Кривые</b>	
- <b>Кривая1 (Задание в...</b>	
Заголовок	Задание
Привязка	Задание_выход
Цвет	
Стиль линии	
Толщина линии	2
Тип меток	Нет меток
Стиль линии вне границ	
Макс. значение	60
Мин. значение	20
Интерполирование	По периоду реального времени
- <b>Кривая2 (Input выход)</b>	
Заголовок	Температура
Привязка	Input_выход
Цвет	
Стиль линии	
Толщина линии	2
Тип меток	Нет меток
Стиль линии вне границ	
Макс. значение	60
Мин. значение	20
Интерполирование	По периоду реального времени

Экран ошибка  
отображает в  
режиме  
реального  
времени  
значение сигнала  
рассогласования.  
Пример показан  
на рис.



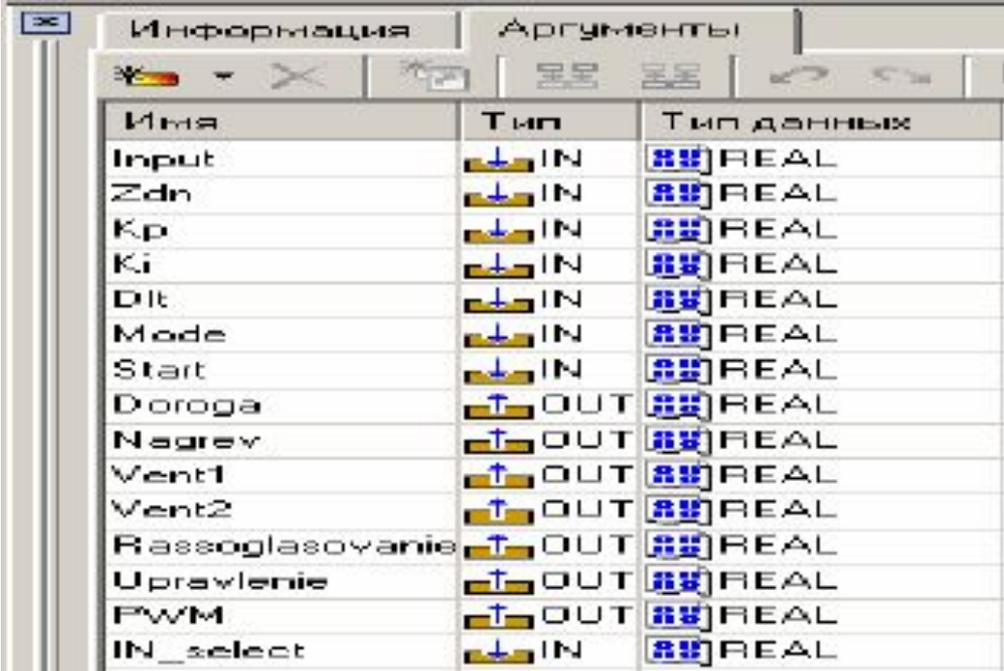
- Экран управление отображает в режиме реального времени значение управляющего воздействия.
- Экран ШИМ отображает в режиме реального работу широтно-импульсного модулятора.
- В шаблонах программ создадим два FBD (Functional Blocks Diagram)-программы, как это показано на рис.



## Создание шаблонов программ.

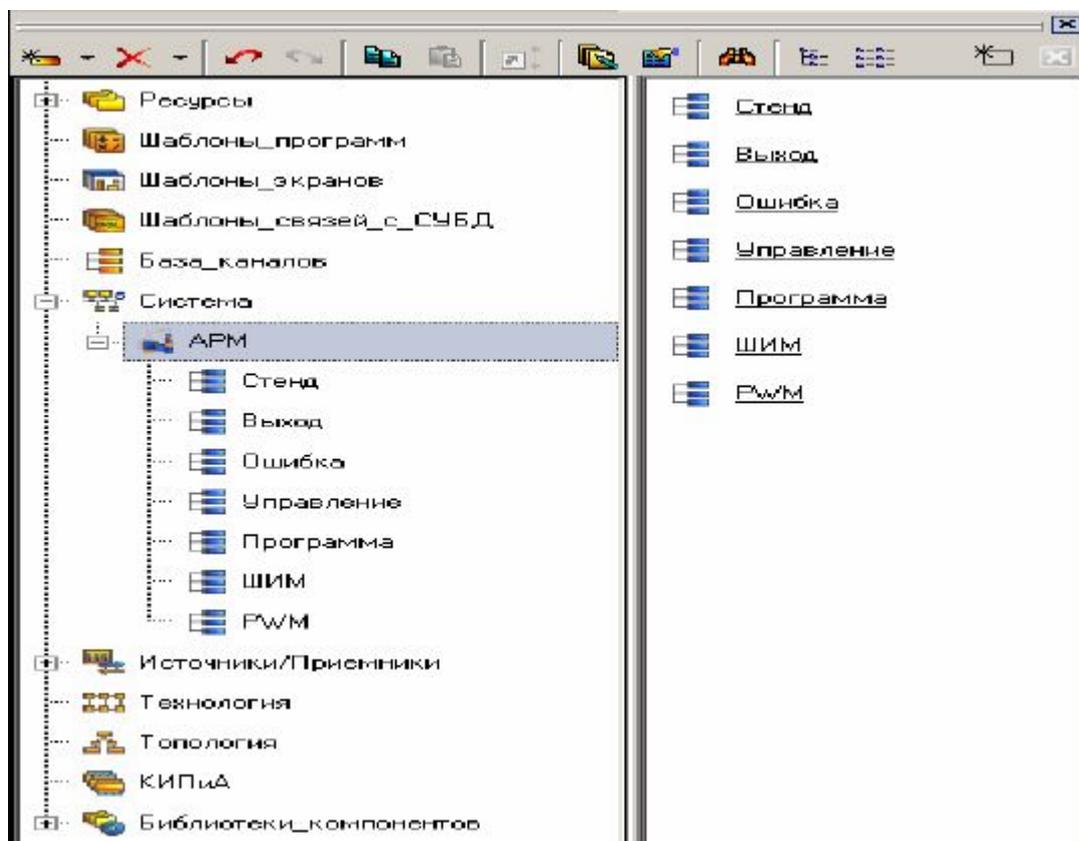
- Программа **Регулирование** реализует алгоритмы ПИ-регулятора, имитирует работу релейного регулятора, осуществляет переключение между ними. Опишем назначение функциональных блоков:
  - $X-Y$  - арифметическое вычитание для вычисления величины рассогласования;
  - DZONE - зона нечувствительности, которая вводится для устранения высокочастотных помех;
  - PID - блок, вырабатывающий управляющее воздействие по ПИД-закону, причем, поскольку в данном проекте реализован ПИ-закон управления, аргумент для  $KD$  входа не создаётся;

Аргументы для этого шаблона FBD-программы будут такими, как показано на рис.

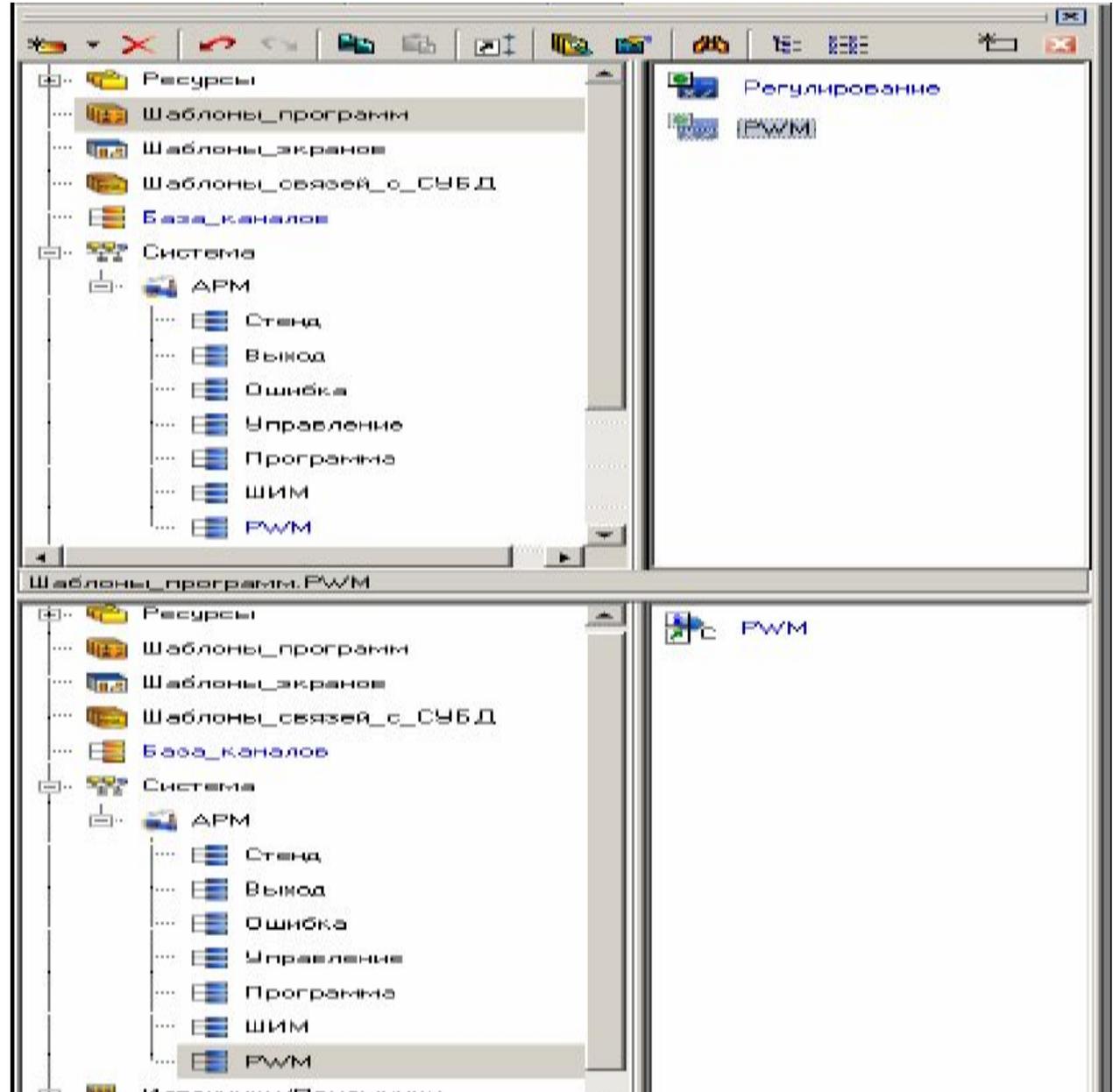


Имя	Тип	Тип данных
Input	↓ IN	REAL
Zdn	↓ IN	REAL
Kp	↓ IN	REAL
Ki	↓ IN	REAL
Dlt	↓ IN	REAL
Mode	↓ IN	REAL
Start	↓ IN	REAL
Doroga	↑ OUT	REAL
Nagrev	↑ OUT	REAL
Vent1	↑ OUT	REAL
Vent2	↑ OUT	REAL
Rassoglasovanie	↑ OUT	REAL
Upravlenie	↑ OUT	REAL
PWM	↑ OUT	REAL
IN_select	↓ IN	REAL

**Автопостроение каналов.** После проведения работ, определенных выше, создадим в системе узел АРМ и в нём 7 объектов, как это показано на рис.



Откроем  
дополнительное  
окно навигатора  
(рис.) и из  
шаблонов экранов и  
программ  
перетащим мышкой  
созданные шаблоны  
в соответствии с  
именами объектов:



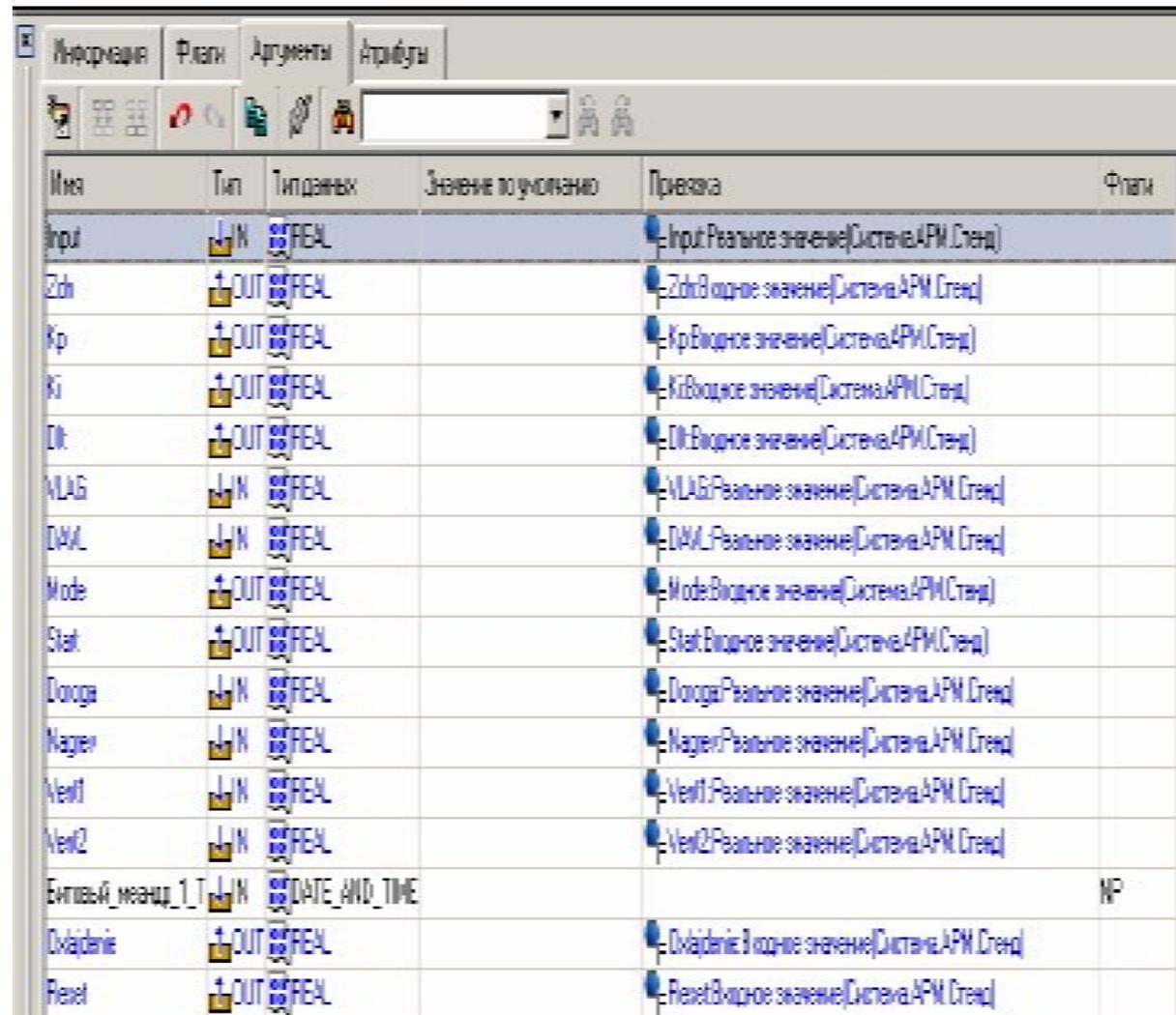
- Аналогично для всех остальных объектов. В результате этого получили канал класса «Вызов».
- Рассмотрим автопостроение каналов. Для этого откроем свойства канала «Вызов». На экран монитора будет выведено диалоговое окно, вид которого приведен на рис.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Input	 IN	 REAL			
Zdn	 DLT	 REAL			
Kp	 DLT	 REAL			
Ki	 DLT	 REAL			
Di	 DLT	 REAL			
PLAG	 IN	 REAL			
DAVL	 IN	 REAL			
Mode	 DLT	 REAL			
Stat	 DLT	 REAL			
Danga	 IN	 REAL			
Nagev	 IN	 REAL			
Vert1	 IN	 REAL			
Vert2	 IN	 REAL			
Битовый_выход_1_Г	 IN	 DATE_AND_TIME			NP
Onajderie	 DLT	 REAL			
Reset	 DLT	 REAL			

Нажатием на иконку

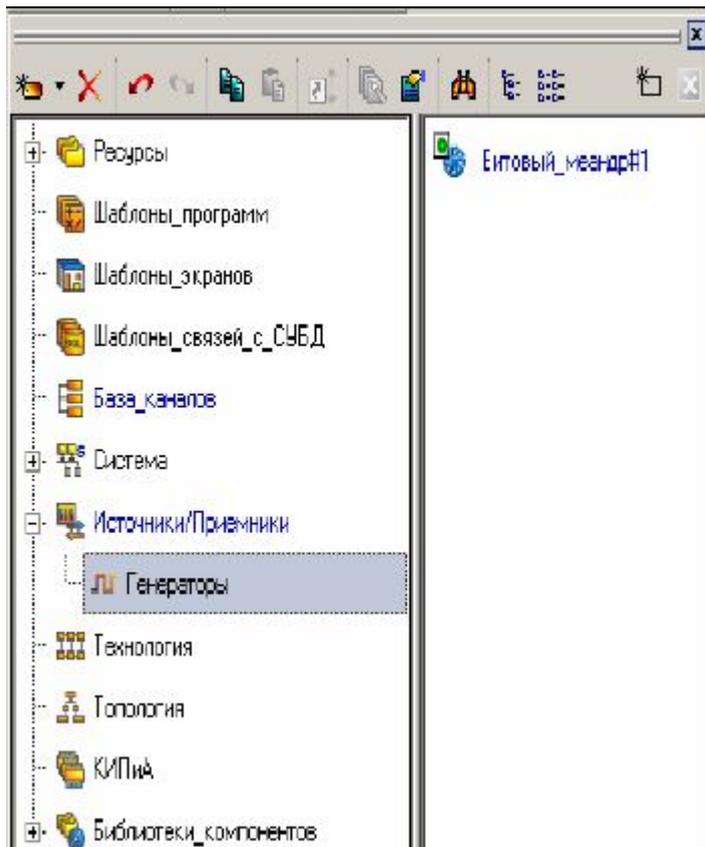


автопостроим каналы. Результат приведен на рис.



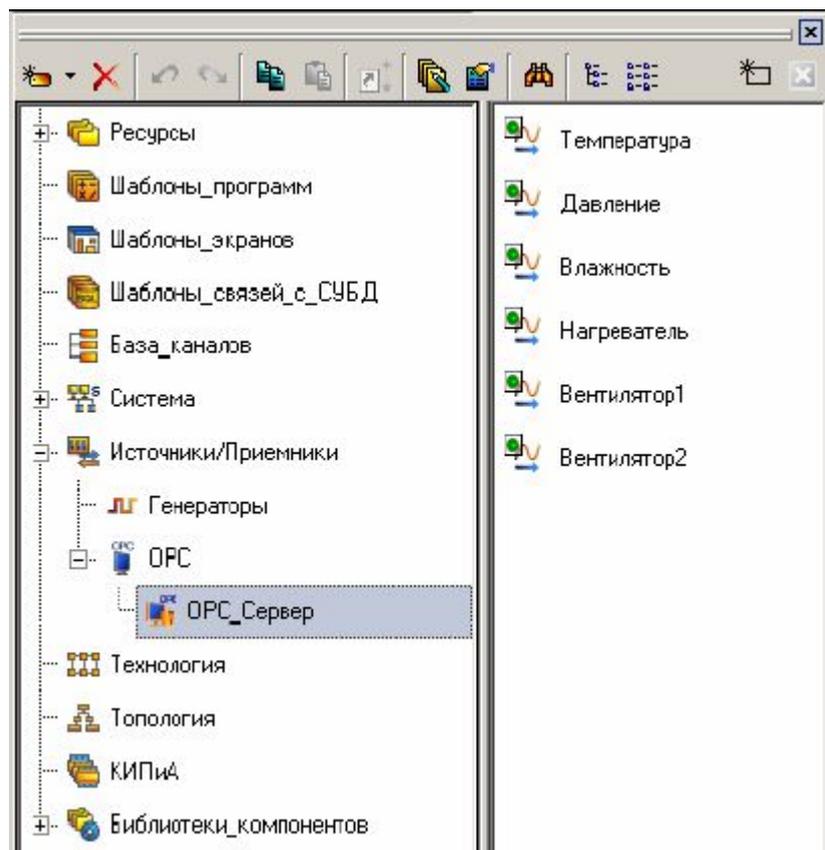
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги
Input	IN	REAL		Input: Реальное значение (Система АРМ Стенд)	
Zdn	OUT	REAL		Zdn: Вводное значение (Система АРМ Стенд)	
Kp	OUT	REAL		Kp: Вводное значение (Система АРМ Стенд)	
Ki	OUT	REAL		Ki: Вводное значение (Система АРМ Стенд)	
Dlt	OUT	REAL		Dlt: Вводное значение (Система АРМ Стенд)	
NLAB	IN	REAL		NLAB: Реальное значение (Система АРМ Стенд)	
DAL	IN	REAL		DAL: Реальное значение (Система АРМ Стенд)	
Mode	OUT	REAL		Mode: Вводное значение (Система АРМ Стенд)	
Stat	OUT	REAL		Stat: Вводное значение (Система АРМ Стенд)	
Doxda	IN	REAL		Doxda: Реальное значение (Система АРМ Стенд)	
Magren	IN	REAL		Magren: Реальное значение (Система АРМ Стенд)	
Verf1	IN	REAL		Verf1: Реальное значение (Система АРМ Стенд)	
Verf2	IN	REAL		Verf2: Реальное значение (Система АРМ Стенд)	
Вспльв_мемд_1_T	IN	DATE_AND_TIME			NP
Dvajdenic	OUT	REAL		Dvajdenic: Вводное значение (Система АРМ Стенд)	
Reset	OUT	REAL		Reset: Вводное значение (Система АРМ Стенд)	

- Аналогичная работа выполняется и для других каналов класса «Вызов».
- В источниках/приёмниках создадим источник «Генераторы», а в нём создадим объект «Битовый меандр», как это показано на рис.



# Создание источников/приемников.

- Создадим компоненты OPC-сервера и шесть тегов. Результат приведен на рис .

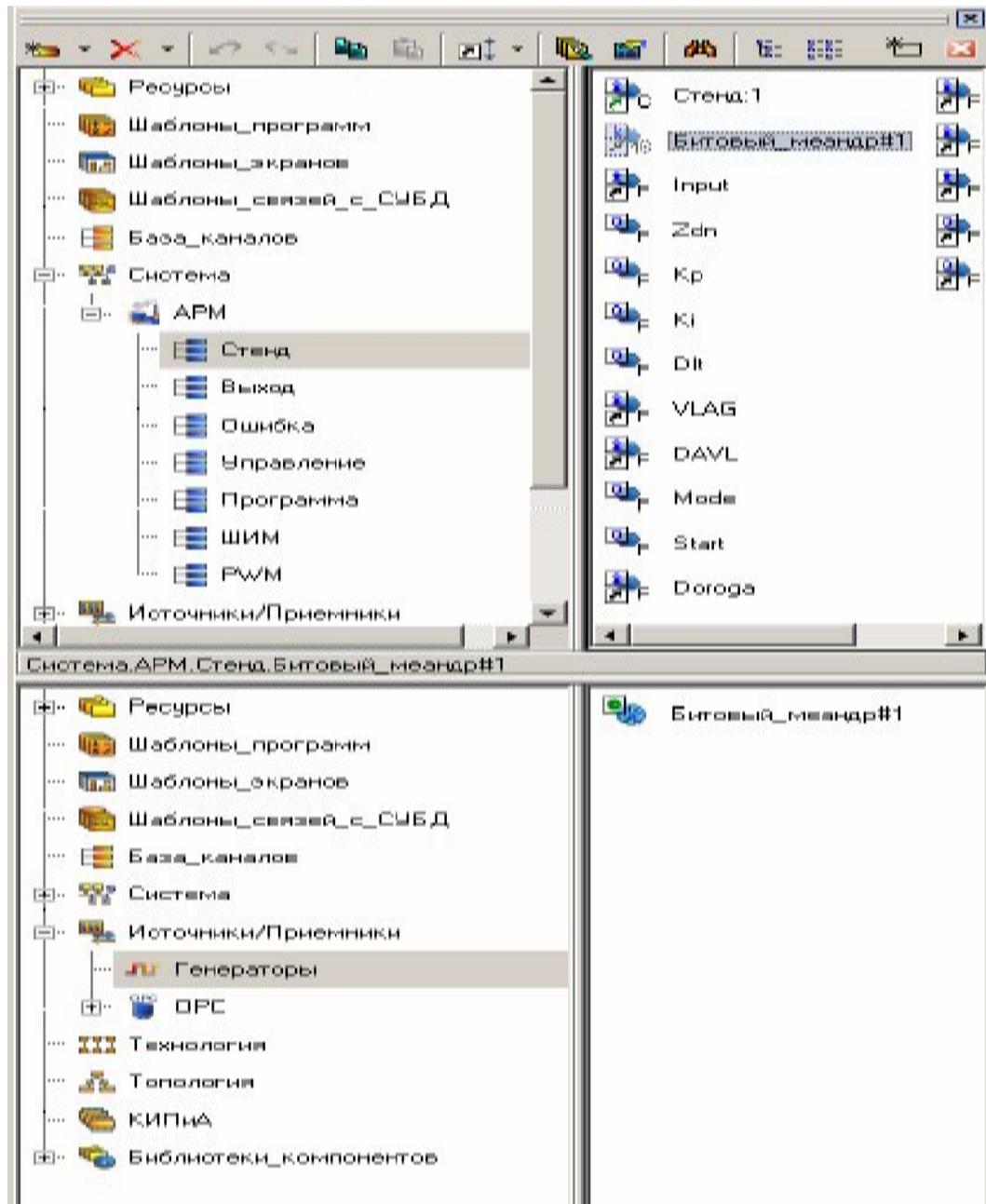


- Привяжем теги к виртуальным обозначениям дискретных выходов модуля ввода-вывода с помощью кнопки «Обзор» в окне редактора тега так, как это показано на след.рис.
- Для остальных тегов выполняется аналогичная работа.



# **Взаимосвязь компонентов проекта**

Во все созданные объекты перетащим «Битовый меандр» из «Генераторов», как это показано на рис.



Затем во всех каналах класса «Вызов»  
вручную привяжем аргумент **Time** к каналу  
«Битовый меандр» в том же объекте и  
установим атрибут «Время изменения», как  
это показано на рис.

Сконфигурировать связь для Система.APM.Стенд.Стенд:1:Битовый\_меандр\_1\_T

Ресурсы

- База\_каналов
- Система
  - APM
    - Стенд
      - Стенд:1
        - Битовый\_меандр#1
        - Input
        - Zdn
        - Kp
        - Ki
        - Dlt
        - VLAG
        - DAVL
        - Mode
        - Start
        - Doroga
        - Nagrev
        - Vent1
        - Vent2

Атрибуты | Аргументы

ID	Полное имя
24	Бит 10
25	Бит 16
38	Период пересчета (единицы)
39	Отработать
40	Инерсия
41	СПАД
42	Глобальный регистратор
43	Отчет Трвог
44	Атрибуты
45	Время изменения
46	Защелка
47	MF
48	Класс канала
49	Отладка
51	Смена типа Вх/Вых
52	Достоверность программная
56	Размерность в битах
57	Индекс аварийного словаря
58	Дамп
60	В сеть
61	Индекс автопосылки
78	Формат
79	Кодировка
80	Комментарий
81	Тип
82	Размерность
83	Тип сигнала
84	Целочисленное представление
86	Связь

Атрибуты/Аргументы | Тип атрибута

Удалить привязку | Привязать канал | Привязать атрибут | Привязка | Отмена

Информация

Имя	Имя	Имя
Input	Битовый_меандр_1_T	IN
Zdn	DATE_AND_TIME	REAL
Kp	Битовый_меандр#1	Время изменения(Система.APM.Стенд) NP
Ki	Oxajderie	↑ OUT REAL
Dlt	Reset	↑ OUT REAL
VLAG		
DAVL		
Mode		
Start		
Doroga		
Nagrev		
Vent1		
Vent2		

Это позволяет правильно отображать текущую дату и время.

Для взаимодействия всех компонентов проекта необходимо сделать привязку каналов в соответствии с назначением проекта.

Привязка каналов осуществляется перетаскиванием мышью канала типа **OUT** на канал типа **IN**.

Нажатием иконки



сохраним проект для МРВ. После сохранения запустим профайлер .



- Запустив проект на исполнение, задавая параметры и режим работы, будем иметь возможность наблюдать процесс управления.

Таблица 10

Имя	Привязка
Стенд	
Стенд:1	
Битовый_меандр#1	Битовый_меандр#1: Значение(Источники/Приемники. Генераторы)
Input	Температура:Значение(Источники/Приемники. OPC.OPC_Сервер)
Zdn	
Kp	
Ki	
Dlt	
VLAG	Влажность:Значение(Источники/ Приемники.OPC.OPC_Сервер)
DAVL	Давление:Значение(Источники/ Приемники.OPC.OPC_Сервер)
Mode	
Start	
Doroga	Doroga:Реальное значение(Система.APM.Программа)
Nagrev	Nagrev:Реальное значение(Система.APM.Программа)
Vent1	Vent1:Реальное значение(Система.APM.Программа)
Vent2	Vent2:Реальное значение(Система.APM.Программа)
Oxladzenie	Вентилятор2:Значение(Источники/Приемники. OPC.OPC_Сервер)
Reset	Вентилятор2:Значение(Источники/Приемники. OPC.OPC_Сервер)
Выход	
Выход:2	

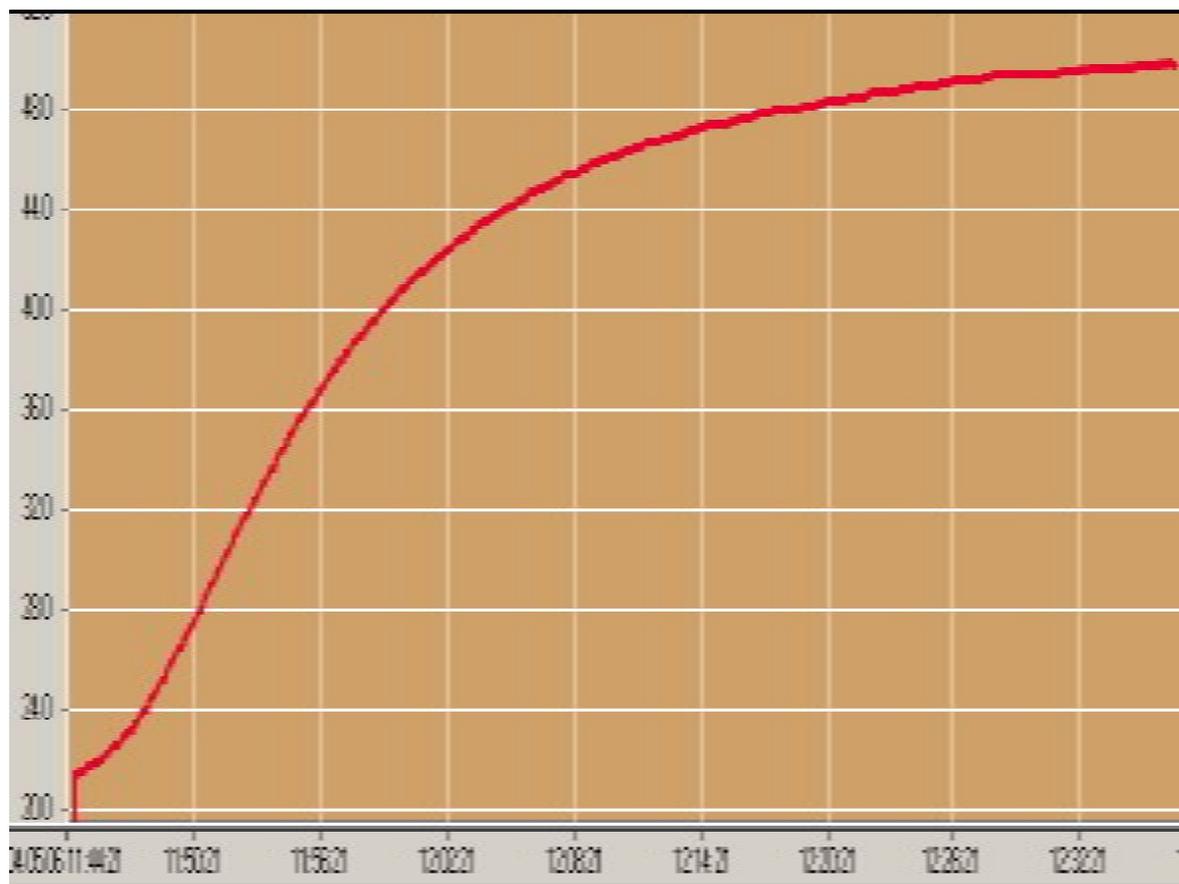
Имя	Привязка
Битовый_меандр#1	Битовый_меандр#1:Значение(Источники/Приемники.Генераторы)
Задание_выход	Zdn:Реальное значение(Система.APM.Стенд)
Input_выход	Температура:Значение(Источники/Приемники.OPC.OPC_Сервер)
Ошибка	
Ошибка:3	
Битовый_меандр#1	Битовый_меандр#1:Значение(Источники/Приемники.Генераторы)
Задание_Ошибка	Zdn:Реальное значение(Система.APM.Стенд)
Рассогласование	Rassoglasovanie:Реальное значение(Система.APM.Программа)
Управление	
Управление:4	
Битовый_меандр#1	Битовый_меандр#1:Значение(Источники/Приемники.Генераторы)
Задание_Управление	Zdn:Реальное значение(Система.APM.Стенд)
Управление	Upravlenie:Реальное значение(Система.APM.Программа)
Программа	
Регулирование:5	
Input	Температура:Значение(Источники/Приемники.OPC.OPC_Сервер)
Zdn	Zdn:Реальное значение(Система.APM.Стенд)
Kp	Kp:Реальное значение(Система.APM.Стенд)
Ki	Ki:Реальное значение(Система.APM.Стенд)
Dlt	Dlt:Реальное значение(Система.APM.Стенд)
Mode	Mode:Реальное значение(Система.APM.Стенд)
Start	Start:Реальное значение(Система.APM.Стенд)
Doroga	
Nagrev	Нагреватель:Значение(Источники/Приемники.OPC.OPC_Сервер)

Окончание табл. 10

Имя	Привязка
Vent1	Вентилятор1:Значение(Источники/Приемники. OPC.OPC_Сервер)
Vent2	Вентилятор2:Значение(Источники/Приемники. OPC.OPC_Сервер)
Rassoglasovanie	
Upravlenie	
PWM	
IN select	Out:Реальное значение(Система.APM.PWM)
ШИМ	
ШИМ:6	
ШИМ	Out:Реальное значение(Система.APM.PWM)
Битовый_меандр#1	Битовый_меандр#1:Значение(Источники/ Приемники.Генераторы)
PWM	
PWM	
Input	Upravlenie:Реальное значение(Система.APM.Программа)
Out	

# Идентификация объекта управления

Для того чтобы правильно выбрать регулятор и его параметры, необходимо знать математическую модель объекта управления (ОУ). С этой целью на ОУ был выполнен эксперимент по получению его разгонной характеристики при включении питания на нагреватель. Пусть имеем указанную характеристику в виде тренда, представленного на рис.



Используя метод идентификации по разгонной характеристике, пришли к выводу, что передаточная функция ОУ будет иметь вид:

$$W_{oy}(p) = \frac{k_0}{T_0 p + 1} \exp(-p T_{зап})$$

со следующими параметрами  **$k_0=12$** ,  
 **$T_0=11$  мин**,  **$T_{зап}=41$  мин**.

# Настройка регулятора

В качестве примера выбрали ПИ-регулятор с параметрами:  $k_{рег}=0,17$ ,  $T_i=14$  мин.

На отобразившемся после запуска профайлера экране можно задать эти параметры регулятора и, для наглядности, нажатием кнопки **Визуализация** переходить к экрану наблюдения за переходными характеристиками.

В качестве вводимых параметров используются следующие:

$$k_p=k_{рег}=0,17 \text{ и } k_i=k_{рег}/T_i=0,01545.$$

# **Снятие экспериментальных характеристик.**

После запуска системы с заданием нужной температуры в 45 С получаем тренд (см.рис.), на котором представлены графики задания, изменения в процессе регулирования температуры и изменения в процессе контроля влажности

