

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФГБОУ ВПО «РОССИЙСКИЙ
ХИМИКО–ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ
им. Д.И. МЕНДЕЛЕЕВА»**

НОВОМОСКОВСКИЙ ИНСТИТУТ (ФИЛИАЛ)

А.Г. ЛОПАТИН, П.А.КИРЕЕВ, С.В. ЛОПАТИНА

**УПРАВЛЯЮЩИЕ ВЫЧИСЛИТЕЛЬНЫЕ КОМПЛЕКСЫ
МЕТОДИЧЕСКИЕ УКАЗАНИЯ К КУРСОВОЙ РАБОТЕ**

**Новомосковск
2013**

УДК 66.011
ББК 35
Л 771

Рецензент:
кандидат технических наук, доцент Маслова Н.В.
(НИ (филиал) ФГБОУ ВПО РХТУ им. Д. И. Менделеева)

Лопатин А. Г., Киреев П.А., Лопатина С.В.

Управляющие вычислительные комплексы. Методические указания к курсовой работе. /ФГБОУ ВПО РХТУ им. Д. И. Менделеева, Новомосковский ин-т (филиал). Новомосковск, 2013.–28 с.

Методические указания предназначены для студентов, очной и заочной формы обучения по направлению подготовки 220700 Автоматизация технологических процессов и производств

УДК 66.011
ББК 35

© Новомосковский институт (филиал)
ФГБОУ ВПО Российского химико–
технологического университета им.
Д.И. Менделеева, 2013

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
МИНИМИЗАЦИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ МЕТОДОМ КВАЙНА МАК- КЛАСКИ	5
РЕАЛИЗАЦИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ ПРИ ПОМОЩИ SCADA– СИСТЕМЫ TRACE MODE	10
ВЫБОР ТЕХНИЧЕСКИХ СРЕДСТВ АВТОМАТИЗАЦИИ	27
ЗАДАНИЯ ДЛЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ	28

ВВЕДЕНИЕ

Целью освоения дисциплины «Управляющие вычислительные комплексы» состоит в формировании у студентов знаний и умений выбора, наладки и эксплуатации программно-технического обеспечения автоматизированных систем управления технологических процессов и производств.

Задачи преподавания дисциплины:

Изучение основных протоколов построения промышленных сетей, иерархии промышленных сетей и SCADA систем.

В методическом пособии описан порядок выполнения курсовой работы по курсу УБК на тему «Построение логических систем управления при помощи SCADA системы TRACE MODE».

Курсовая работа состоит из трех частей, при выполнении первой части студент в соответствии с заданием должен произвести минимизацию логической функции, во второй части реализовать полученную минимизированную функцию при помощи SCADA системы TRACE MODE, а в третьей части выбрать технические средства автоматизации отечественных и зарубежных производителей для реализации данной функции, обосновать свой выбор.

МИНИМИЗАЦИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ МЕТОДОМ КВАЙНА МАК-КЛАСКИ

Основные этапы минимизации логических функций методом Квайна и Мак-Класки.

1. Нахождение простых импликант. Сначала все 0-кубы разделяются на группы с одинаковым числом единиц. Затем попарно сравниваются между собой все 0-кубы соседних групп. Если два 0-куба различаются только по одной координате, то они образуют 1-куб (0-кубы, образующие 1-куб, отмечаются). После построения всех 1-кубов, образующих комплекс K_1 , производится построение 2-кубов и т.д. При построении r -кубов все $(r-1)$ – кубы, образующие r -кубы, также отмечаются. Этап заканчивается, когда следующий $(r + 1)$ – куб не может быть построен. Все не отмеченные кубы комплекса $K(f)$ являются простыми импликантами и образуют покрытие $\Pi(f)$ функции f .

2. Построение таблицы покрытий матрицы Квайна. Строки таблицы покрытий соответствуют простым импликантам, а столбцы – 0-кубам (конституентам единицы) минимизируемой функции. На пересечении строки i и столбца j ставится метка единица, если импликанта i покрывает конституенту j . Импликанта покрывает конституенту, если она отличается от нее только независимыми координатами.

3. Отыскание минимального покрытия функции. Для нахождения минимального покрытия необходимо удалить из таблицы покрытия некоторые лишние простые импликанты. С этой целью в методе Квайна и Мак-Класки реализуется следующий алгоритм решения задачи покрытия.

3.1. Выделение ядра Квайна. Если в каком-либо столбце таблицы покрытий имеется только одна единица, то импликанта, стоящая в строке, является существенной (обязательной) и должна входить в ядро Квайна, а следовательно, и в минимальное покрытие, поскольку не используя ее, невозможно покрыть все конституенты.

3.2. Вычеркивание строк и столбцов, покрываемых импликантами, входящими в ядро Квайна. Если в полученной таблице покрытия имеются столбцы, содержащие по одной единице, то операцию описываемую в п.3.1, следует повторить.

3.2. Поглощение лишних столбцов (сжатие по столбцам). Из матрицы вычеркивается тот столбец, в который полностью входит любой другой столбец. Иначе если в таблице покрытий имеется такая пара столбцов d_i и d_j , что $d_i \leq d_j$, то столбец d_j вычеркивается, поскольку покрытие вычеркнутого столбца может производиться путем покрытия оставшегося столбца.

3.3. Поглощение лишних строк (сжатие по строкам). Если в таблице покрытий имеется такая пара строк q_i и q_j , что $q_i \geq q_j$, то строка q_j вычеркивается (строка q_i поглощает строку q_j).

4. Последовательное применение двух преобразований (сжатие по столбцам и строкам). В результате исходная таблица покрытий приводится к так называемой «циклической», импликанты которой должны входить в минимальное покрытие функции.

5. Получение минимальной формы логической функции. Построение импликанты, соответствующее строкам, которые входят в минимальное покрытие, соединяются знаками дизъюнкции и образуют МДНФ булевой функции.

Пример. Минимизировать логическую функцию четырех переменных $Y = \bigvee_1 (0,1,2,4,5,7,8,10,12,14,15)$ методом Квайна и Мак-Класки.

Решение

Представим заданную логическую функцию в виде двоичных эквивалентов входных десятичных наборов, получим

$0_{10}=0000_2$	$8_{10}=1000_2$
$1_{10}=0001_2$	$10_{10}=1010_2$
$2_{10}=0010_2$	$12_{10}=1100_2$
$4_{10}=0100_2$	$14_{10}=1110_2$
$5_{10}=0101_2$	$15_{10}=1111_2$
$7_{10}=0111_2$	

1. Находим простые импликанты в соответствии с первым этапом алгоритма минимизации функции (таблица 1).

Таблица 1 – Нахождение простых импликант функции

	0 группа	1 группа	2 группа	3 группа	4 группа
0 – кубы	0000(0)*	0001(1)* 0010(2)* 0100(4)* 1000(8)*	0101(5)* 1010(10)* 1100(12)*	0111(7)* 1110(14)*	1111(15)*
1 – кубы	000–(0,1)* 00–0(0,2)* 0–00(0,4)* –000(0,8)*	0–01(1,5)* –010(2,10)* 010–(4,5)* –100(4,12)* 10–0(8,10)* 1–00(8,12)*	01–1 (5,7) 1–10 (10,14)* 11–0 (12,14)*	–111(7,15) 111–(14,15)	
2 – кубы	0–0– (0,1,4,5) 0–0– (0,4,1,5) –0–0 (0,2,8,10) –0–0 (0,8,2,10) – – 00 (0,4,8,12) – – 00 (0,8,4,12)	1– –0 (8,10,12,14) 1– –0 (8,12,10,14)			

Все не отмеченные кубы комплекса $K(Y)$ являются импликантами и образуют покрытие:

$$\Pi(Y) = \left\{ \begin{array}{l} 01-1 \\ -111 \\ 111- \\ 0-0- \\ -0-0 \\ --00 \\ 1--0 \end{array} \right\}$$

2. Строим таблицу покрытий матрицы Квайна функции Y (таблица 2)

Таблица 2 – таблица покрытий матрицы Квайна

Имплика- нта	Конституента единицы										
	0000 (0)	0001 (1)	0010 (2)	0100 (4)	0101 (5)	0111 (7)	1000 (8)	1010 (10)	1100 (12)	1110 (14)	1111 (15)
01-1 (5,7)					1	1					
-111 (7,15)						1					1
111- (14,15)										1	1
0-0- (0,1,4,5)	1	1		1	1						
-0-0 (0,2,8,10)	1		1				1	1			
--00 (0,4,8,12)	1			1			1		1		
1--0 (8,10,12,14)							1	1	1	1	

3. Определяем оптимальное покрытие. С помощью таб.2 устанавливаем, что импликанта 0-0- является существенной конституентой 0000, 0001, 0100, 0101, и импликанта -0-0 является существенной для столбцов 0000, 0010, 1000, 1010. Исходя из этого, на данном этапе в минимальное покрытие функции должны быть включены импликанты 0-0- и -0-0. В результате таблица 2 упрощается (таблица 4).

Таблица 4 – Решение задачи покрытия методом Квайна и Мак-Класки

№ строки	Импли- канта	Конституента единицы			
		0111 (7)	1100 (12)	1110 (14)	1111 (15)
1	01-1 (5,7)	1			
2	-111 (7,15)	1			1
3	111- (14,15)			1	1
4	--00 (0,4,8,12)		1		

5	1--0 (8,10,12,14)		1	1	
---	----------------------	--	---	---	--

Затем проводим сжатие таблицы 4, сжатие по столбцам невозможно. Строка 2 поглощает строку 1, а строка 5 – строку 4. Строки 1 и 4 удаляем из таблицы в результате получаем таблицу 5.

Таблица 5 – Решение задачи покрытия методом Квайна и Мак-Класки

№ строки	Импликанта	Конституента единицы			
		0111 (7)	1100 (12)	1110 (14)	1111 (15)
2	-111 (7,15)	1			1
3	111- (14,15)			1	1
5	1--0 (8,10,12,14)		1	1	

После сжатия по столбцам получаем таблицу 6, в которой импликанта 111- является лишней, так как не содержит единиц ни в одном столбце.

Таблица 6 – Решение задачи покрытия методом Квайна и Мак-Класки

№ строки	Импликанта	Конституента единицы	
		0111 (7)	1100 (12)
2	-111 (7,15)	1	
3	111- (14,15)		
5	1--0 (8,10,12,14)		1

После исключения ее из таблицы 6 импликанты -111 и 1--0 которые должны входить в минимальное покрытие функции таблица 7.

Таблица 7 – Решение задачи покрытия методом Квайна и Мак-Класки

№ строки	Импликанта	Конституента единицы	
		0111 (7)	1100 (12)
2	-111 (7,15)	1	
5	1--0 (8,10,12,14)		1

Таким образом, минимальное покрытие имеет вид

$$\Pi(Y) = \left\{ \begin{array}{l} 0-0- \\ -0-0 \\ -111 \\ 1--0 \end{array} \right\}$$

5. Записываем МДНФ $Y = \overline{X_1}\overline{X_3} \cup \overline{X_2}\overline{X_4} \cup X_2X_3X_4 \cup X_1\overline{X_4}$

РЕАЛИЗАЦИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ ПРИ ПОМОЩИ SCADA–СИСТЕМЫ TRACE MODE

Рассмотрим реализацию следующей логической функции $Y = \overline{X_1} \overline{X_3} \cup \overline{X_2} X_4 \cup X_2 X_3 X_4 \cup X_1 \overline{X_4}$ при помощи SCADA–системы TRACE MODE на языке Техно FBD.

Разработка любого проекта автоматизации всегда начинается с запуска Интегрированной среды разработки (ИСР). Для ее запуска необходимо выполнить команду TRACE MODE IDE 6 (base) из группы установки инструментальной системы в меню Программы WINDOWS или двойным

щелчком ЛК мыши по иконке  рабочего стола Windows рисунок 1;

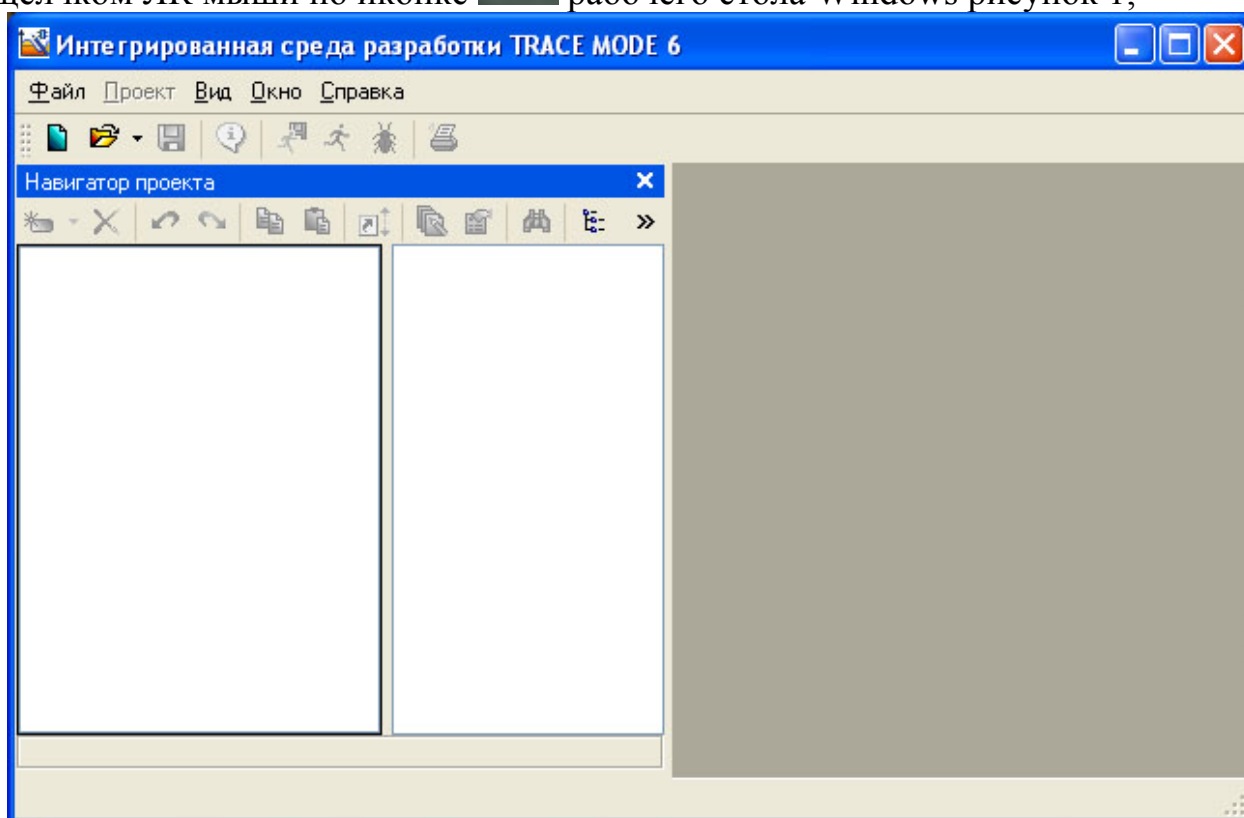


Рисунок 1– Интегрированная среда разработки

После запуска ИСР в меню **Файл** выбрать команду **Настройки ИС...** В появившемся окне выбрать **Уровень сложности** и настроить как показано на рисунке 2, а затем выбрать **Отладка** и настроить как на рисунке 3.

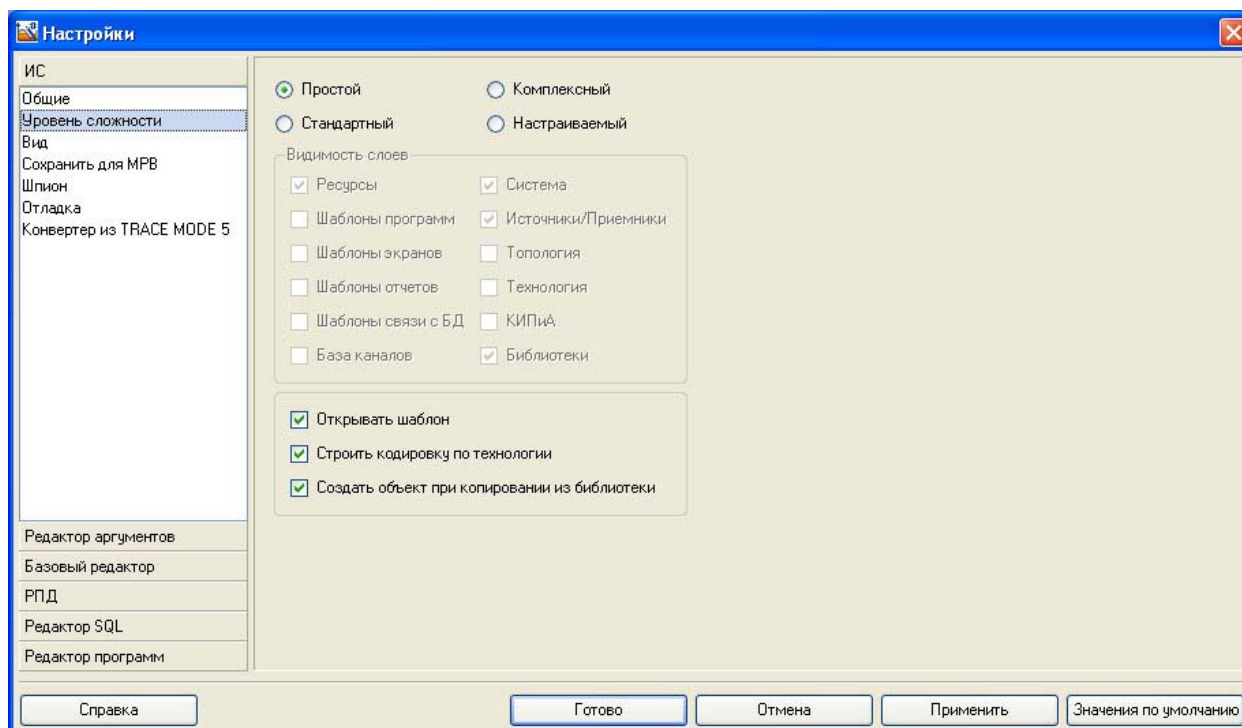


Рисунок 2 – Настройки ИСП

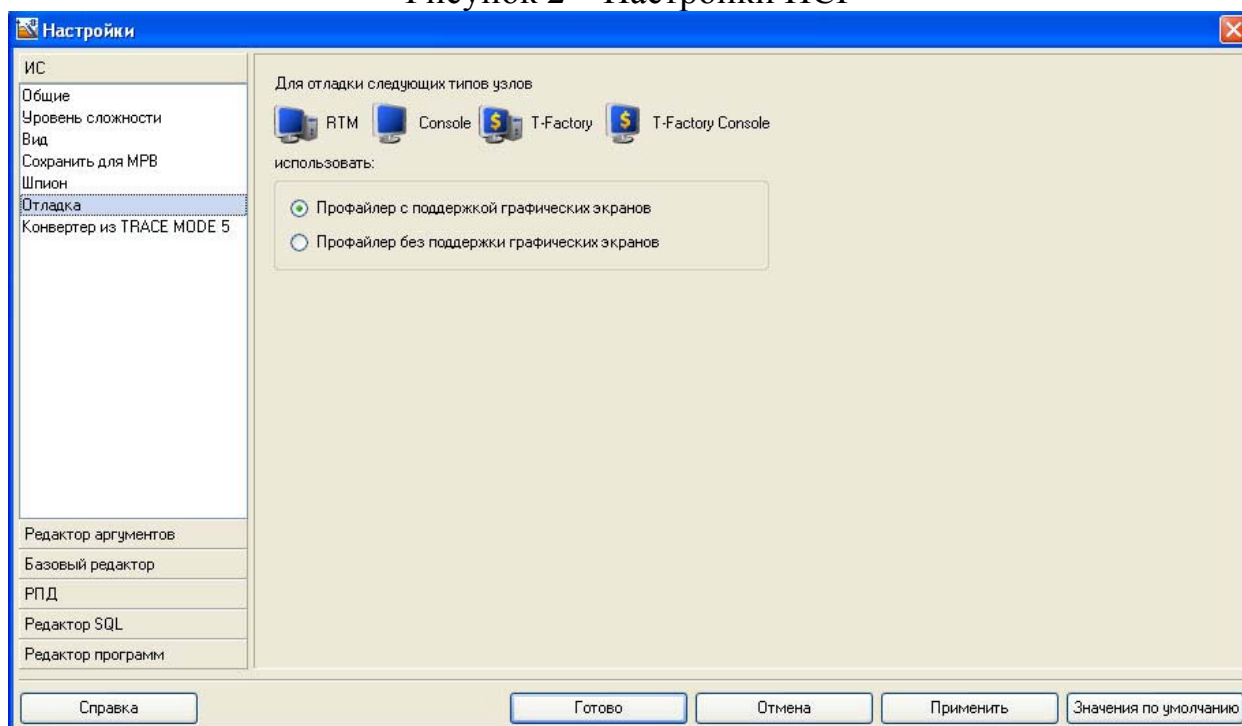



Рисунок 3 – Настройки ИСП

После проведенных настроек ИСП нажать кнопку Готово.

С помощью иконки  инструментальной панели создадим новый проект при этом в открывшемся на экране диалоге рисунок 4.

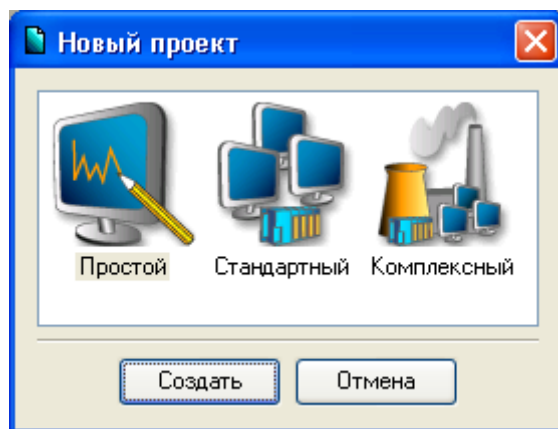


Рисунок 4 – Выбор типа проекта

Выберем **Простой** стиль разработки. После нажатия левой клавиши мыши (ЛК) на экранной кнопке **Создать**, в левом окне Навигатора проекта появится дерево проекта с созданным узлом **АРМ RTM_1**. Откроем узел **RTM_1** двойным щелчком ЛК, в правом окне **Навигатора проекта** отобразится содержимое узла – пустая группа **Каналы** и один канал класса «Вызов» **Экран#1**, предназначенный для отображения на узле **АРМ** графического экрана рисунок 5.

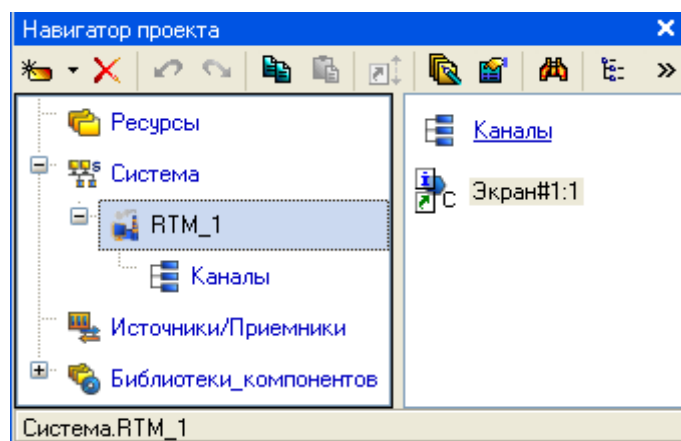


Рисунок 5 – Навигатор проекта

Создание графического экрана

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

Подготовим на экране вывод динамического текста для отображения численного значения входных переменных X_1 , X_2 , X_3 , X_4 и выходного значения Y путем указания динамизации атрибута графического элемента (ГЭ). Определим назначение аргумента шаблона экрана.

Создадим и разместим четыре ГЭ  как показано на рисунке 6.

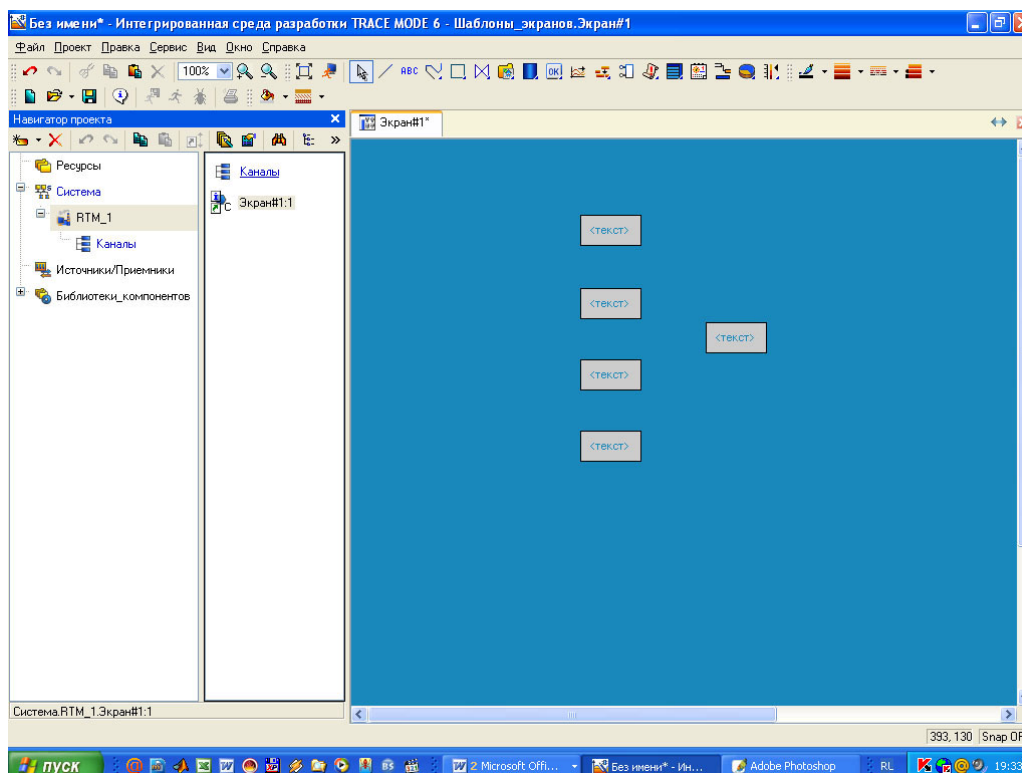


Рисунок 6 – Создание графических элементов

Двойным щелчком ЛК на строке Текст вызвать меню **Вид индикации** рисунок 7;

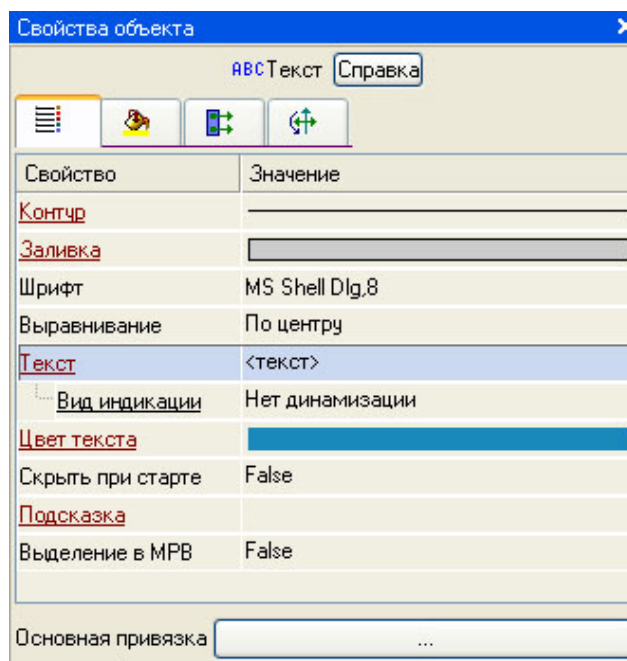


Рисунок 7 – Настройка индикации

В правом поле строки нажать ЛК и вызвать список доступных типов. Выбрать тип **Значение** рисунок 8.

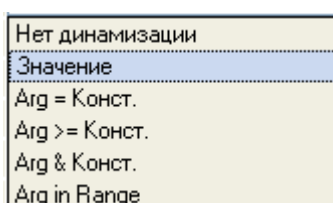


Рисунок 8 – Выбор типа динамизации

В открывшемся меню настройки параметров динамизации рисунок 9.

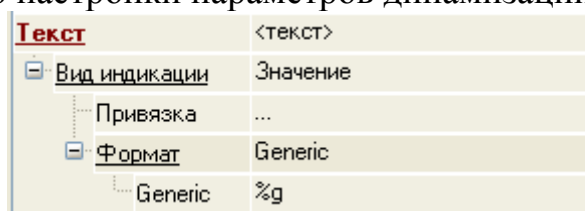



Рисунок 9 – Окно привязки

Выбрать свойство **Привязка**.

В открывшемся окне **Свойство привязки**, нажав кнопку  на его панели инструментов, создать пять аргументов экрана 4 входных и один выходной рисунок 10.

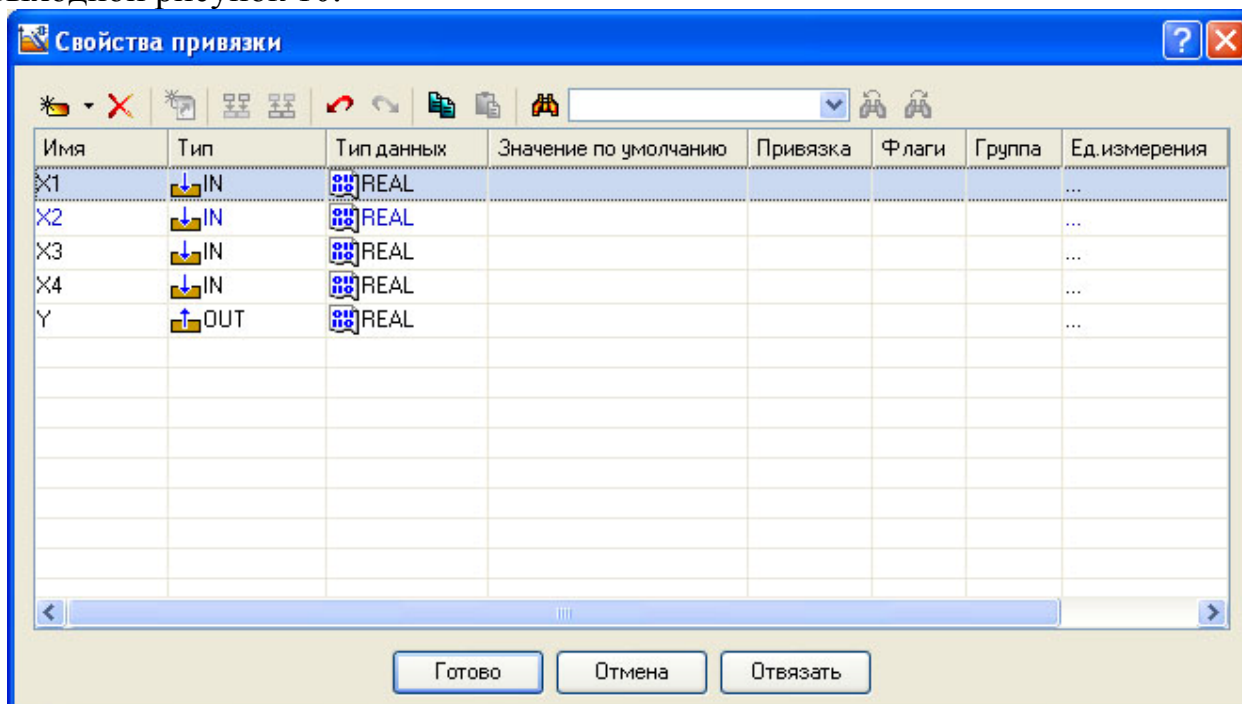



Рисунок 10 – Создание аргументов экрана

Двойным щелчком ЛК выделить имя аргумента и изменить его, введя с клавиатуры «X1» (завершить ввод нажатием клавиши Enter). Подтвердить связь с этим аргументом нажатием кнопки Готово; Аналогичным образом настроить все входные и выходные аргументы.

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

Выбрать на инструментальной панели графического редактора иконку ГЭ Кнопка - . С помощью мыши разместить его в поле экрана как показано на рисунке 11.

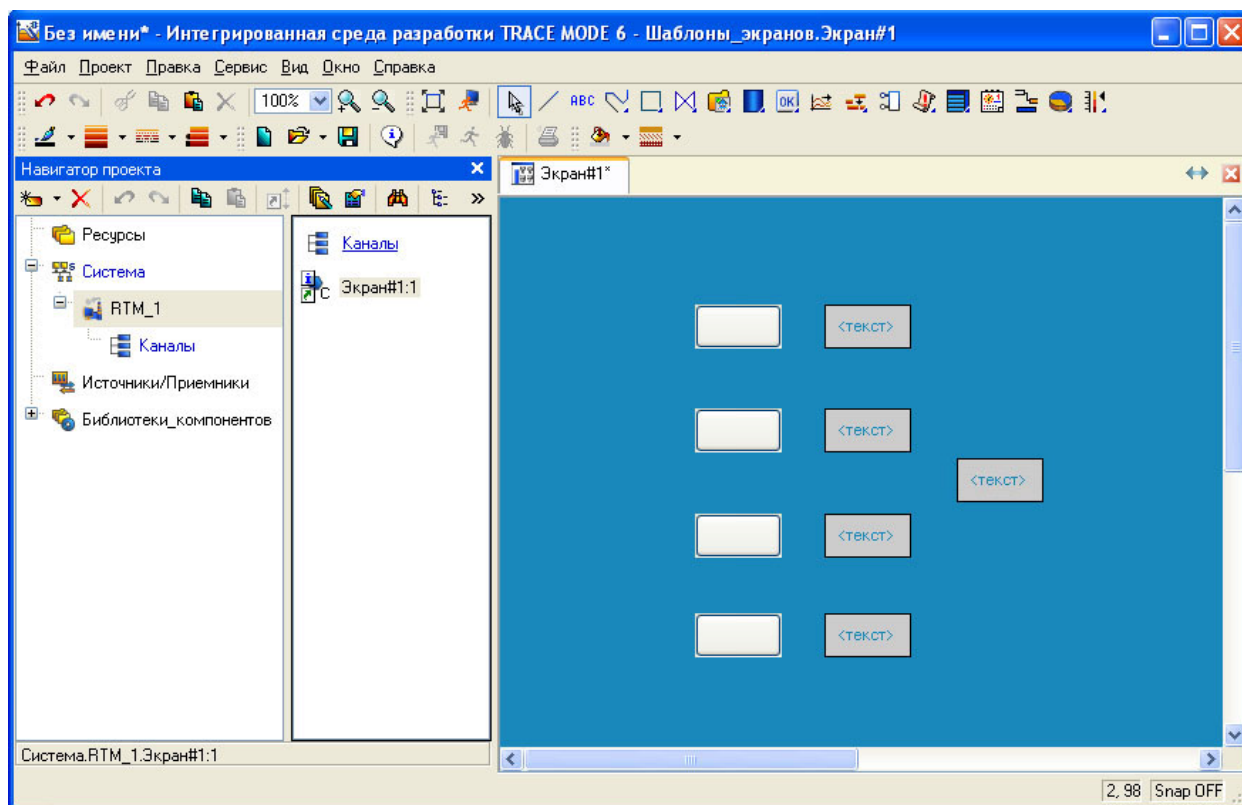




Рисунок 11 – Графический экран

Перейти в режим редактирования  первой кнопки, вызвать окно свойств ГЭ  рисунок 12.

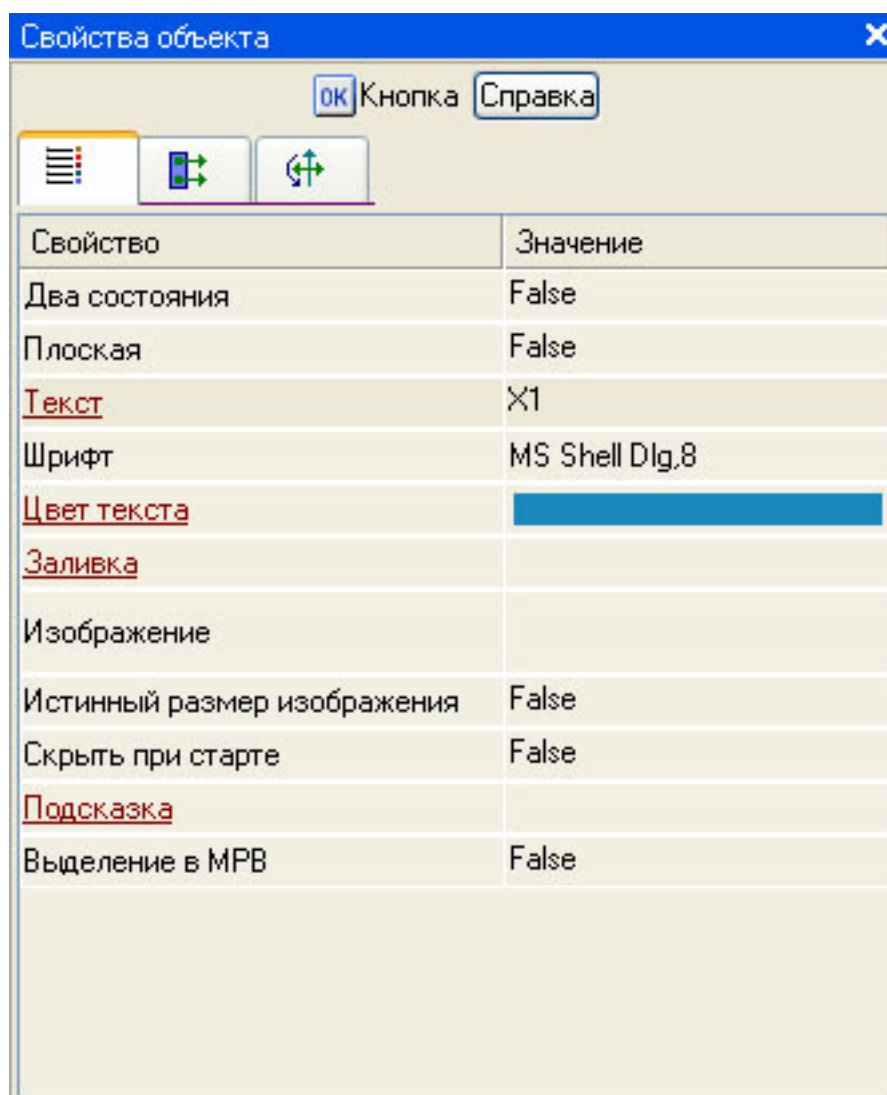



Рисунок 12 – Свойства графического элемента

В поле **Текст** ввести «X1». Открыть бланк **События**  и ПК раскрыть меню **По нажатию (pressed)**. Выбрать из списка команду **Добавить Передать значение**, раскрыть меню настроек выбранной команды рисунок 13.

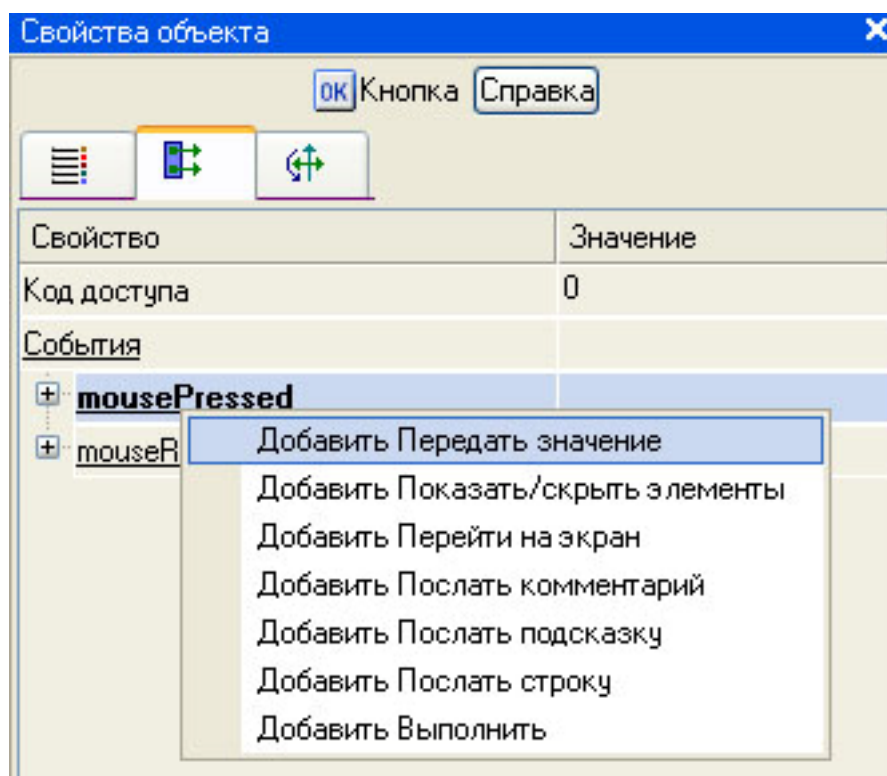


Рисунок 13 – Настройка типа передачи

В поле **Тип передачи (Send Type)** выбрать из списка **Ввести и передать (Enter & Send)** рисунок 14.

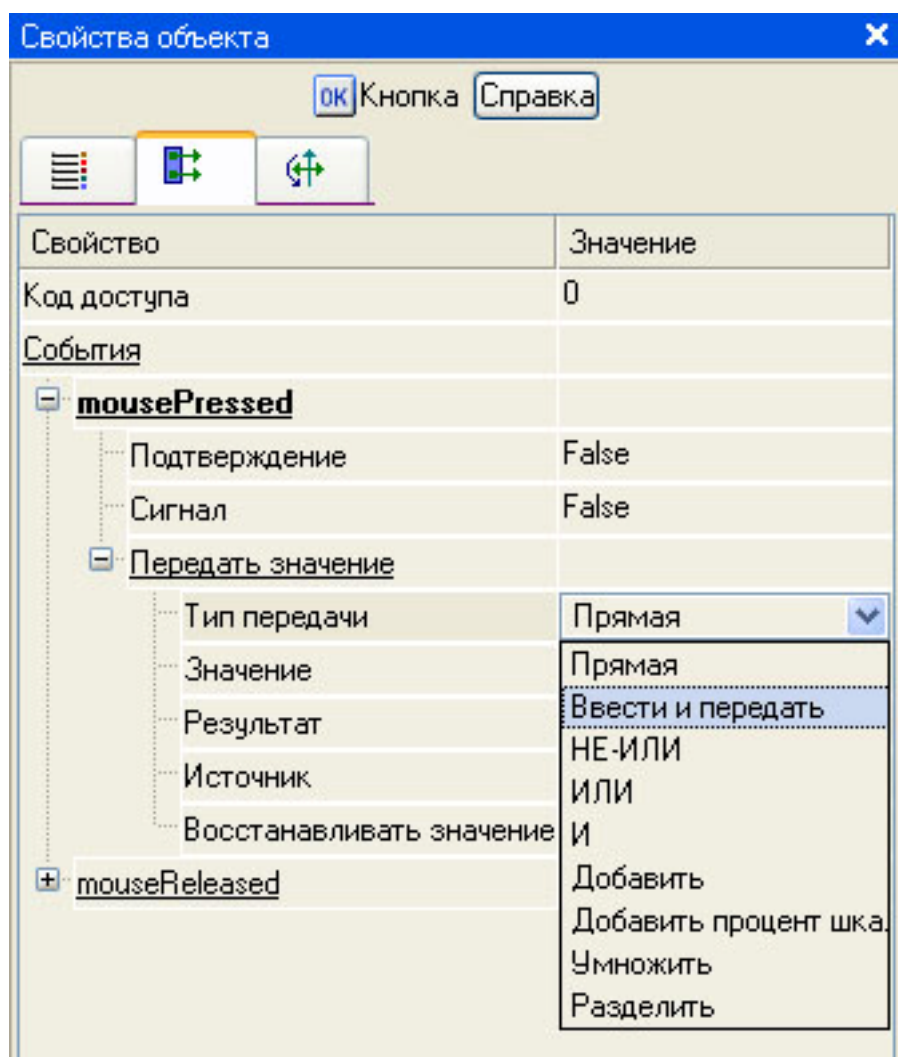


Рисунок 14 – Настройка типа передачи

ЛК в поле **Результат** вызвать табличный редактор аргументов и произвести привязку к аргументу X1. После привязки окно Свойства объекта выглядит следующим образом рисунок 15.

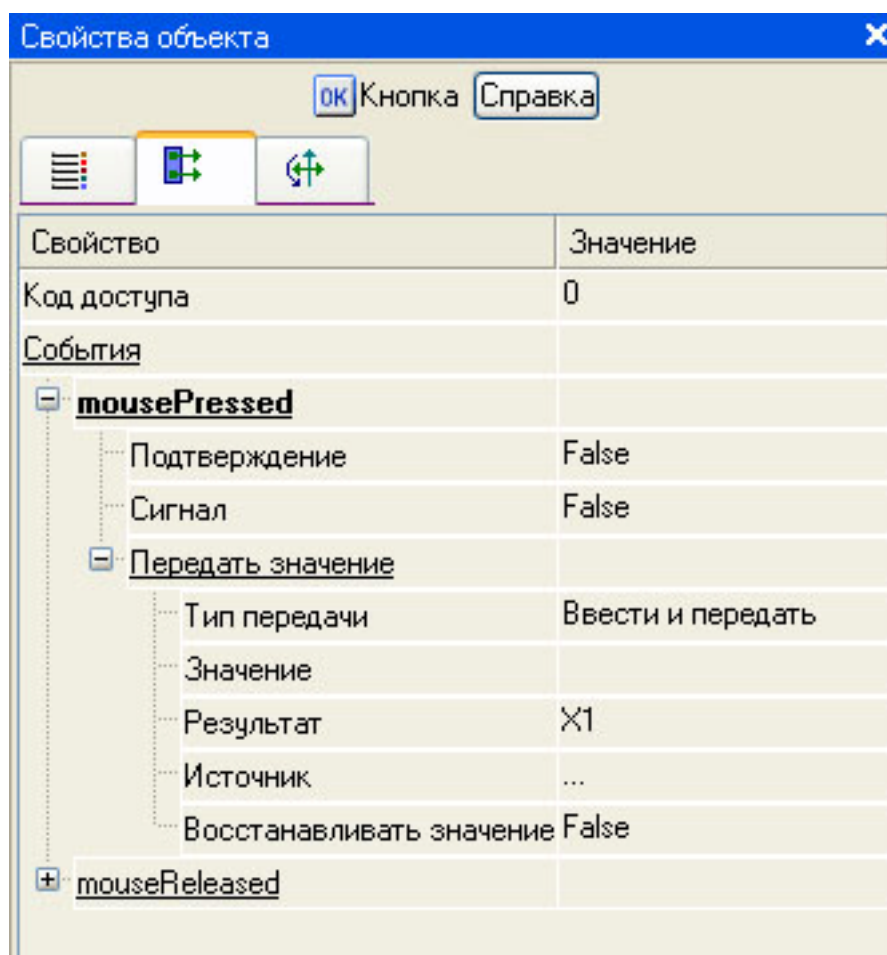


Рисунок 15 – Окончательный вид окна свойств графического элемента
Аналогичным образом настроить остальные кнопки x2, x3, x4.

Привязка аргумента экрана к каналу

Создадим по аргументам X1, X2, X3, X4 и выходу Y шаблона экрана новые каналы и отредактируем их привязку. В слое **Система** открыть узел **RTM_1**. С помощью ПК вызвать через контекстное меню свойства компонента **Экран#1** рисунок 16.

Свойства аргументов						
Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги	
X1	↓ IN	REAL		• X1:Реальное значение (Система.RTM_1)		
X2	↓ IN	REAL		• X2:Реальное значение (Система.RTM_1)		
X3	↓ IN	REAL		• X3:Реальное значение (Система.RTM_1)		
X4	↓ IN	REAL		• X4:Реальное значение (Система.RTM_1)		
Y	↑ OUT	REAL		• Y:Входное значение (Система.RTM_1)		

Рисунок 16 – Окно свойств экрана


Выбрать вкладку Аргументы, ЛК выделить аргументы X1, X2, X3, X4 и выход Y и с помощью иконки  создать новый канал. В результате, в узле **RTM_1** ,будут автопостроены следующие каналы рисунок 17.



Рисунок 17 – Автопостроены каналы

Создание программы на языке Техно FBD

Создадим программу, которая будет реализовывать заданную логическую функцию. Двойным щелчком ЛК открыть узел **RTM_1** и создать в нем компонент **Программа** рисунок 18.

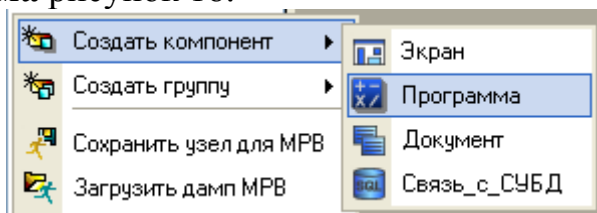


Рисунок 18 – Создание компонента Программа

Выделить компонент Программа#1 и ПК вызвать контекстное меню, выбрав в котором ЛК пункт **Редактировать шаблон**, перейти в режим редактирования программы рисунок 19.

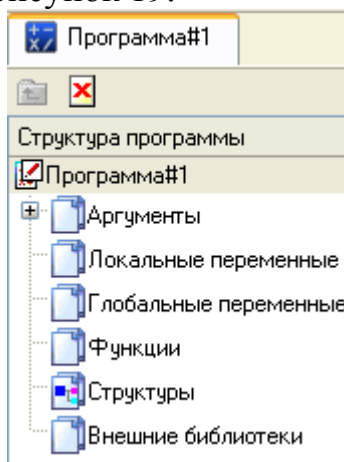



Рисунок 19 – Редактирование аргументов Программы

Выделением ЛК в дереве шаблона **Программа#1** строки Аргументы вызвать табличный редактор аргументов иконкой  создать в редакторе

аргументов четыре аргумента X1, X2, X3, X4 и выход Y. При этом первые 4 аргумента должны быть типа **IN**, а последний – **OUT** рисунок 20.

Имя	Тип	Тип данных	Значение по умолчанию
X1	↓ IN	REAL	
X2	↓ IN	REAL	
X3	↓ IN	REAL	
X4	↓ IN	REAL	
Y	↑ OUT	REAL	

Рисунок 20 – Аргументы программы

Выделить в дереве шаблона строку **Программа#1** и в открывшемся диалоге Выбор языка выбрать язык **FBD** рисунок 21.

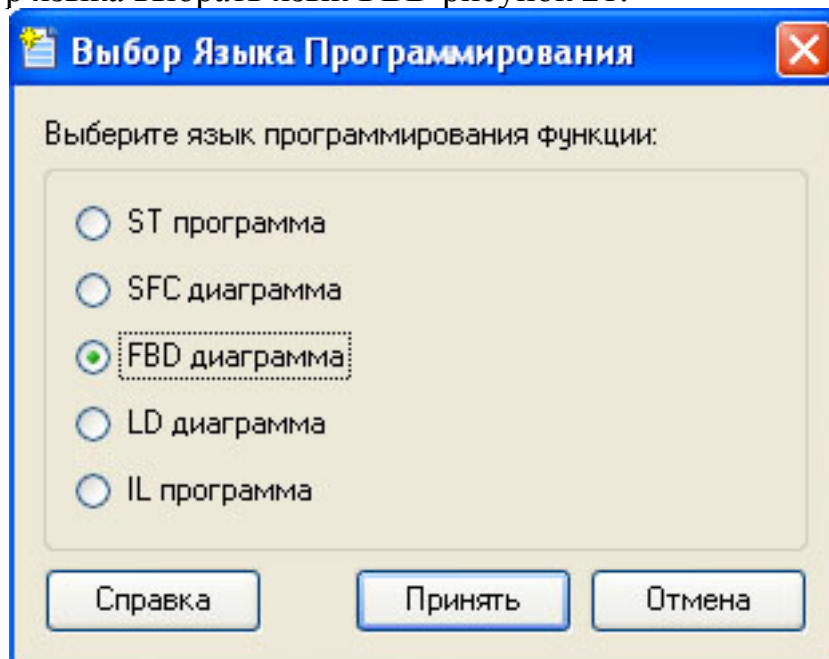



Рисунок 21 – Выбор языка программирования

По нажатию экранной кнопки **Принять** в открывшемся окне редактора программ с объявленными переменными создать программу в соответствии с заданием. Для выбора палитры FBD блоков необходимо ЛК мыши нажать на кнопку  после чего появится окно выбора FBD блоков рисунок 22. При разработки программы верхние входы FBD блоков не используются т.к. они предназначены для изменения порядка пересчета блоков, а информационными входами, являются входы начиная со второго.

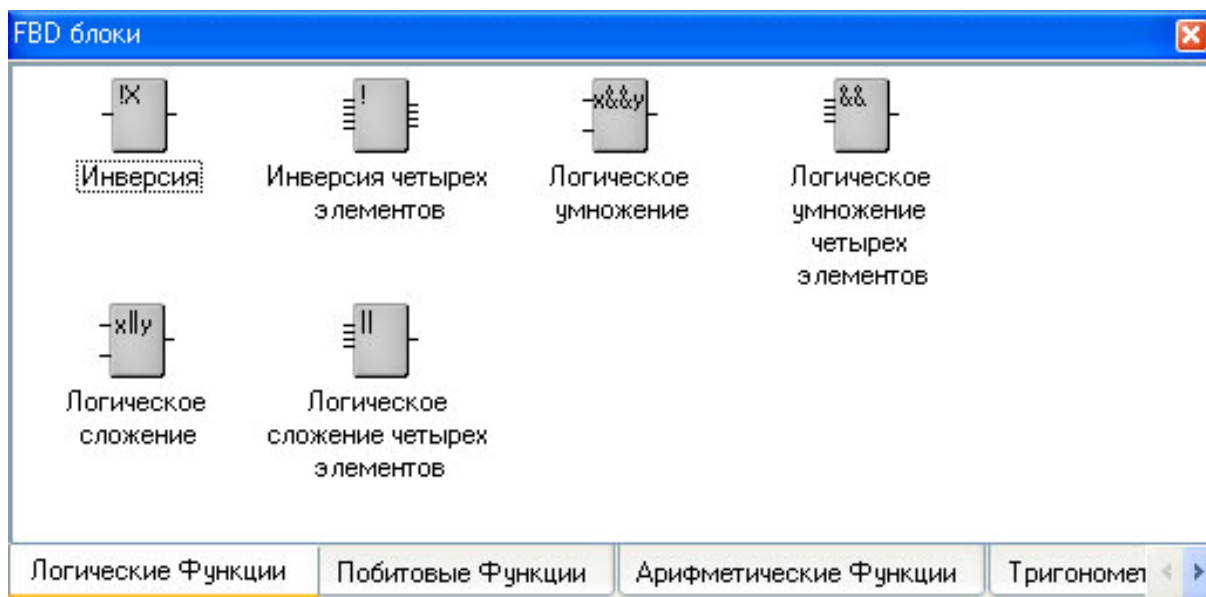


Рисунок 22 – Палитра FBD блоков

Для реализации логической функции выберем следующие блоки: из раздела *Логические Функции* **FBD блоки** инверсия (!X), логическое умножение (X&&Y и &&), логическое сложение (||). После размещения всех блоков для рассматриваемого примера программа будет выглядеть следующим образом рисунок 23.

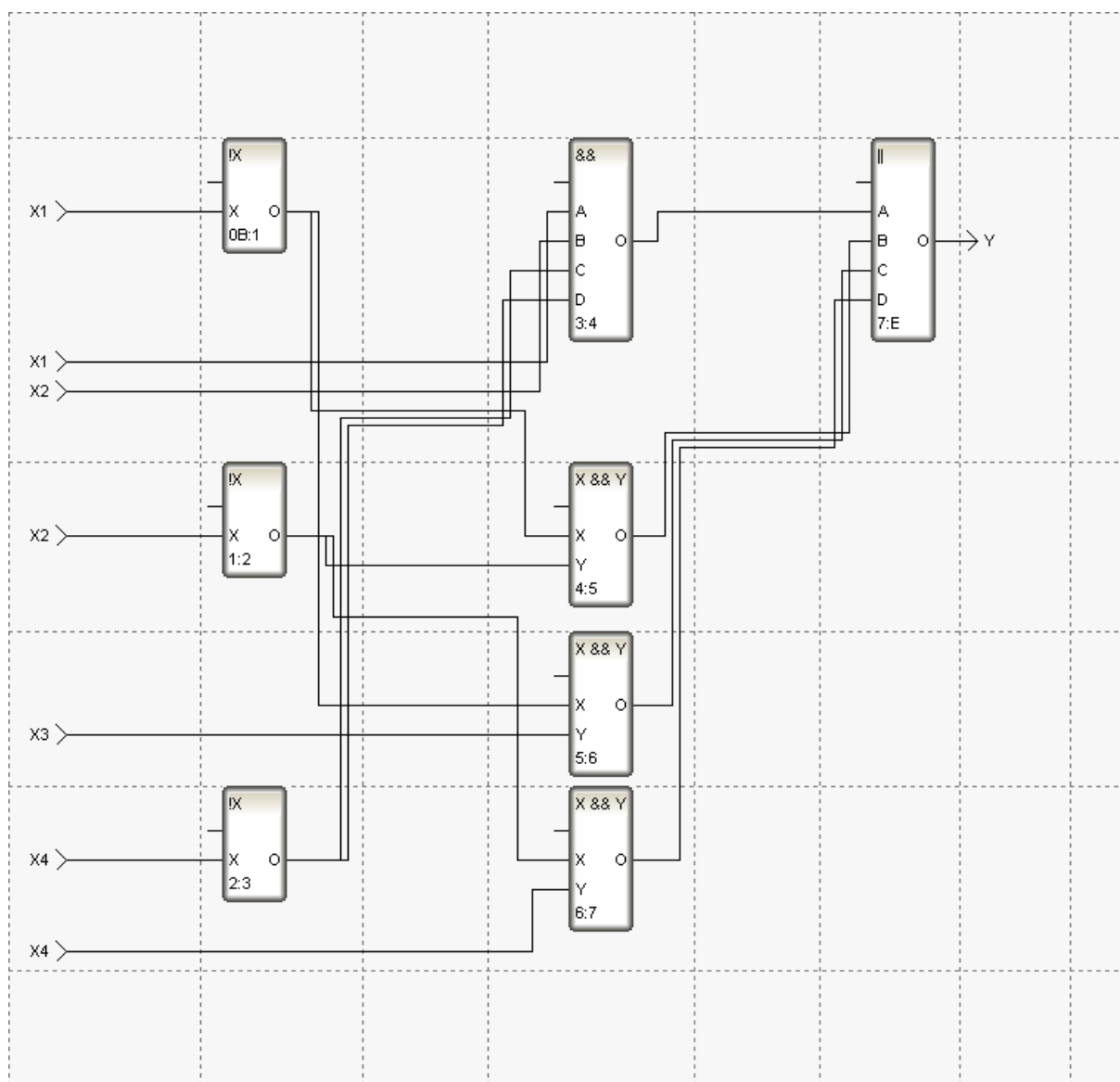




Рисунок 23 – Программа на языке Техно FBD

С помощью иконки  на инструментальной панели редактора или «горячей клавишей» F7 скомпилировать программу и убедиться в успешной компиляции в окне Выход (Output), вызываемого из инструментальной панели с помощью иконки  рисунок 24.

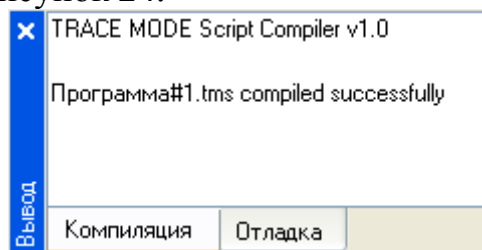


Рисунок 24 – Результат успешной компиляции программы

Привязка аргументов программы

Выполним привязку аргументов программы к атрибутам каналов. Вызвать свойства компонента **Программа#1** через контекстное меню. Выбрать вкладку **Аргументы**. Двойным нажатием в поле **Привязка** привязать аргументы

программы к атрибутам каналов – аргументы X1, X2, X3, X4 к реальному значению каналов X1, X2, X3, X4 рисунок 25.

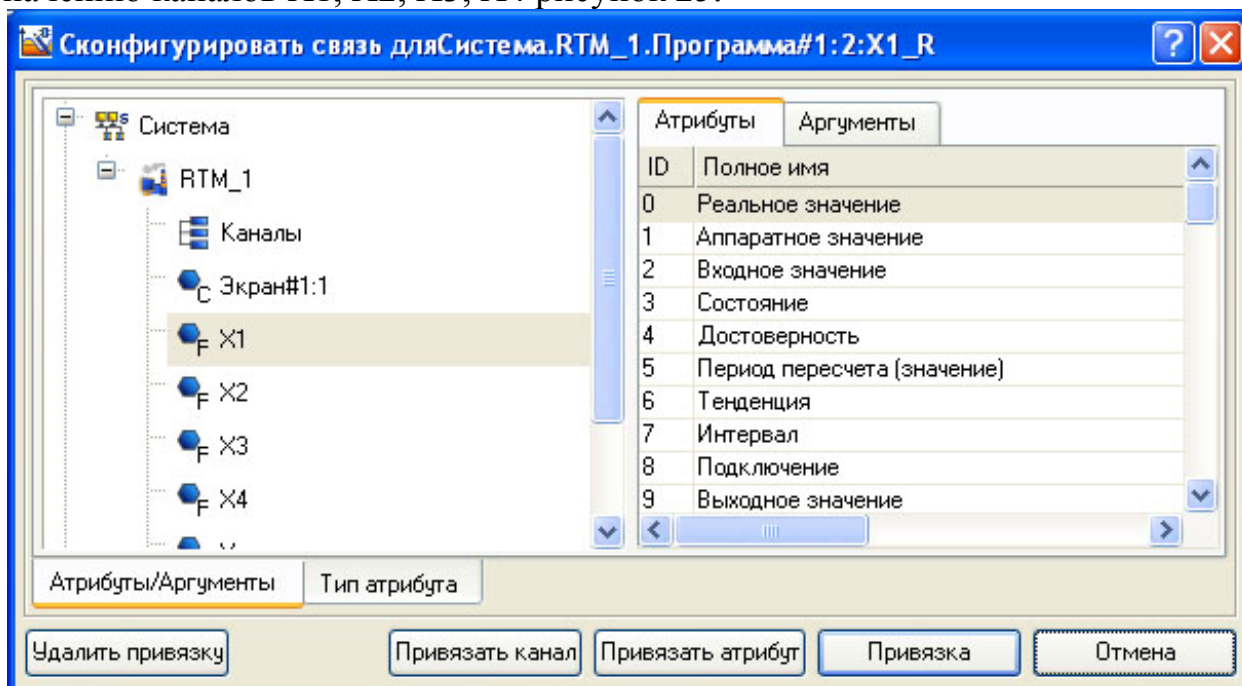


Рисунок 25 – Связь аргументов программы с аргументами экрана

Двойным щелчком в поле **Привязка** аргумента программы Y вызвать окно настройки связи, выбрать в левом окне канал класса «Вызов» **Экран#1**, а в правом окне выбрать вкладку **Аргументы** и указать в ней аргумент Y и кнопкой **Привязка** подтвердить связь рисунок 26.

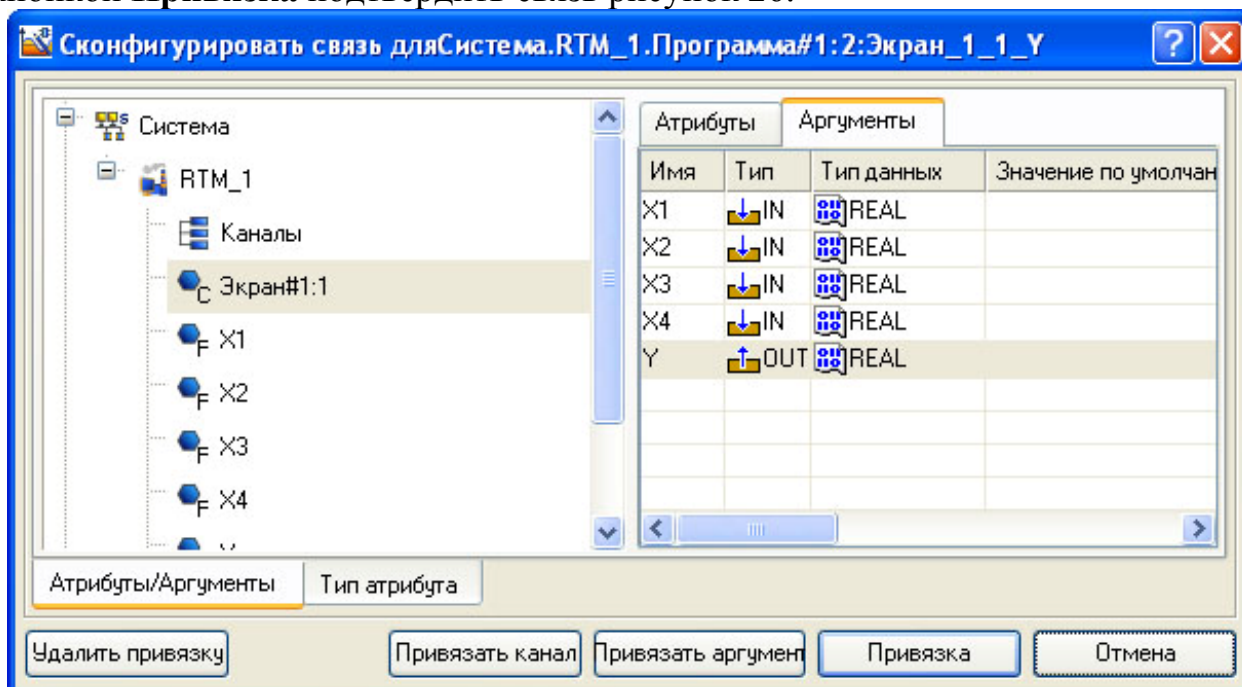


Рисунок 26 – Связь выходных аргументов

В результате, будем иметь рисунок 27.

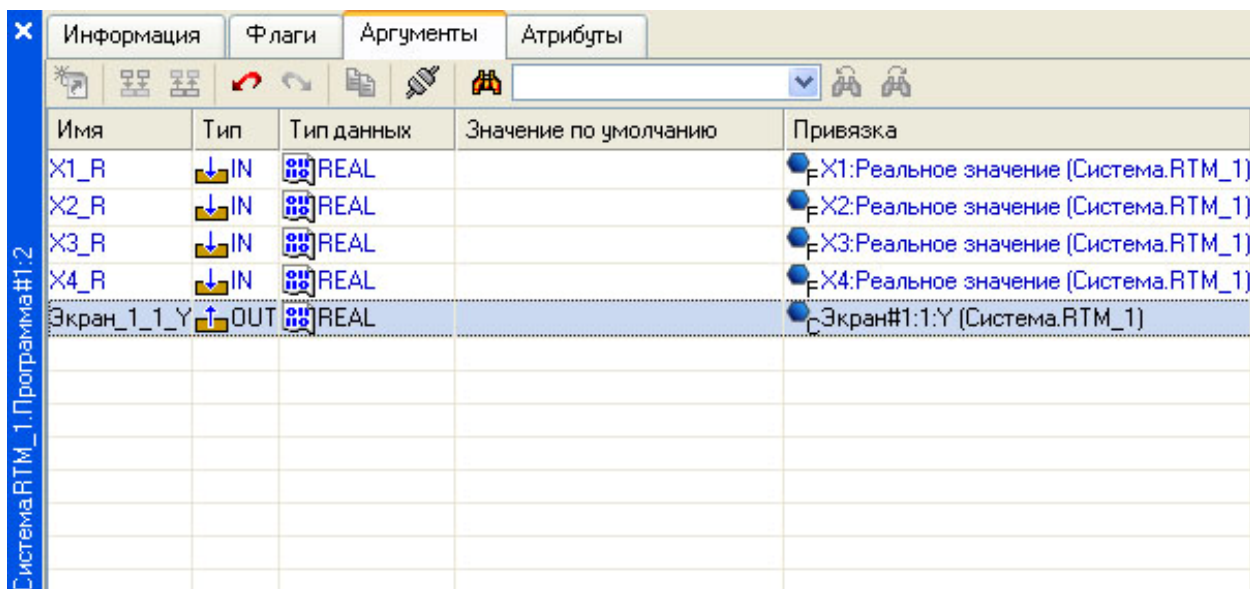


Рисунок 27 – Окончательная настройка связи

После закрыть окно свойств компонента **Программа#1**.

Запуск проекта

Сохранить проект с помощью иконки . На инструментальной панели выбрать команду и скомпилировать тем самым проект для запуска в реальном времени. В навигаторе проекта выделить узел RTM_1 рисунок 28.

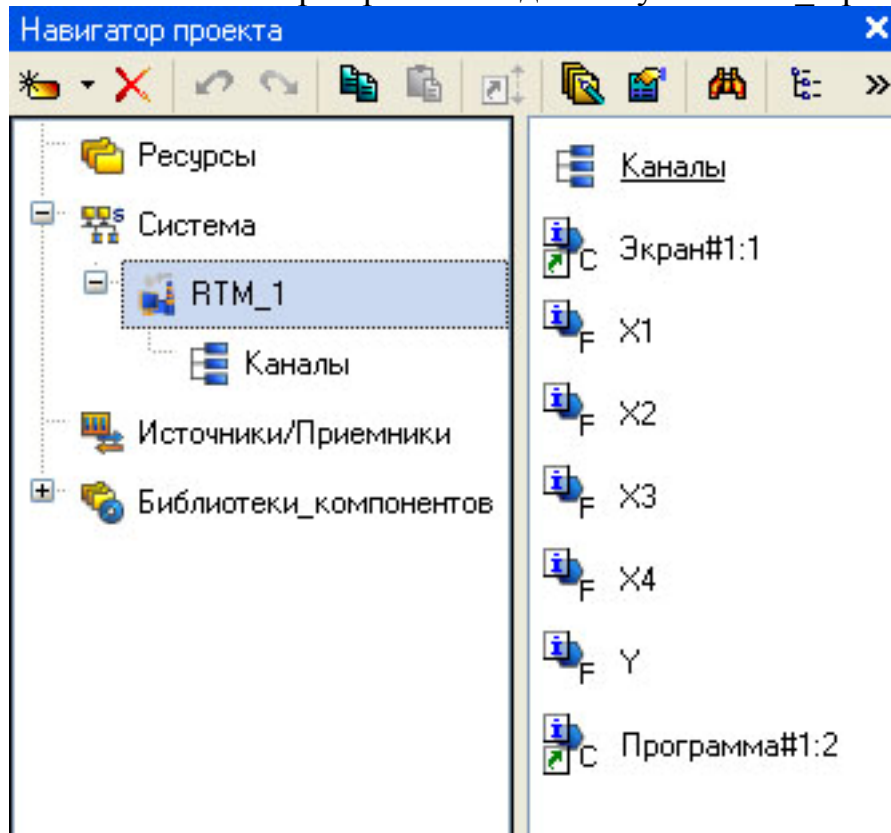


Рисунок 28 – Навигатор проекта

Выбрать иконку на инструментальной панели и запустить режим исполнения.

После реализации полученной МДНФ в TRACE MODE, необходимо выбрать технические средства автоматизации для реализации логической системы управления.

ВЫБОР ТЕХНИЧЕСКИХ СРЕДСТВ АВТОМАТИЗАЦИИ

Назначение и выбор ПЛК

Программируемые логические контроллеры (ПЛК) предназначены для сбора и анализа информации с первичных датчиков, измерения и сравнения параметров, логической обработки сигналов по заданным алгоритмам и выдачи управляющих воздействий (команд) на исполнительные механизмы.

При программировании промышленных ПЛК используются 5 языков программирования стандарта МЭК 6-1131/3.

В настоящее время используются системы автоматизации на базе ПЛК, связанных с персональным компьютером. Они получают все большее распространение благодаря удобству, доступности, дружественному интерфейсу и низкой стоимости.

Открытые протоколы, стандартизация отдельных компонентов и свершившийся всеобщий переход на контрактное производство стирает различия между категориями контроллерных устройств и даже между изделиями разных марок. Это позволяет собирать управляющие комплексы на базе микропроцессоров нового поколения из модулей разных производителей. Поэтому определить класс и тип контроллеров, наилучшим образом подходящий для решения конкретных производственных задач, целесообразнее всего исходя из соотношения цена/качество, сроков поставки и условий сервисного обслуживания, а не престижа торговой марки.

Алгоритм выбора ПЛК

При выборе ПЛК необходимо учитывать следующие основные факторы:

1. Характер применения (автономно, в качестве станции в распределенной сети, в качестве удаленной станции)
2. Функциональное назначение (ПИД-регулирование, управление системами тепло- и водоснабжения, измерение и счет данных, терморегулирование, аварийная защита и блокировка и т.д.)
3. Количество входов/выходов (цифровых и аналоговых)
4. Требуемая скорость передачи данных
5. Наличие автономного счетчика времени
6. Условия регистрации и хранения данных
7. Возможность самодиагностики
8. Требования к панели оператора
9. Язык программирования
10. Интерфейс
11. Каналы связи (проводной, беспроводной)
12. Режим и условия эксплуатации

ЗАДАНИЯ ДЛЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Номер варианта выбирается в соответствии с порядковым номером студента в экзаменационной ведомости. Студенты группы А–ХХ–2 в исходный вариант добавляют любую цифру от 0 до 31 отсутствующую в задании.

№ варианта	Исходная логическая функция
1	$f = \vee_1(0,4,6,9,10,14,15,16,19,21,22,24,26,27,28)$
2	$f = \vee_1(2,3,4,6,11,13,14,17,18,20,21,25,26,30,31)$
3	$f = \vee_1(1,5,7,8,9,14,16,23,24,25,26,27,28,29,31)$
4	$f = \vee_1(0,1,2,3,6,7,8,9,10,11,13,14,21,25,31)$
5	$f = \vee_1(0,1,2,4,6,11,13,15,18,20,21,22,23,24,31)$
6	$f = \vee_1(2,3,4,6,7,13,15,20,22,25,26,27,28,29,31)$
7	$f = \vee_1(1,2,5,9,10,11,12,15,19,21,25,26,28,29,31)$
8	$f = \vee_1(0,2,5,7,8,10,13,16,17,18,19,21,22,27,31)$
9	$f = \vee_1(0,5,6,7,8,9,10,13,14,17,22,23,28,30,31)$
10	$f = \vee_1(0,1,2,6,8,10,11,12,18,19,21,24,26,27,31)$
11	$f = \vee_1(2,4,5,7,8,9,11,12,13,17,20,22,25,26,30)$
12	$f = \vee_1(3,9,11,12,14,15,17,19,22,23,25,28,29,30)$
13	$f = \vee_1(0,2,3,7,11,14,16,18,19,20,23,25,26,27,29)$
14	$f = \vee_1(2,3,5,9,10,13,15,19,20,21,22,25,28,29,30)$
15	$f = \vee_1(0,1,8,10,11,12,14,15,17,19,20,21,25,26,30)$
16	$f = \vee_1(0,1,3,6,8,9,10,11,13,17,18,21,25,26,27)$
17	$f = \vee_1(1,5,6,8,10,11,12,13,15,16,24,26,28,30,31)$
18	$f = \vee_1(0,4,5,6,8,9,13,14,20,22,24,25,27,28,31)$
19	$f = \vee_1(2,6,7,8,9,11,12,15,17,18,19,20,23,29,30)$
20	$f = \vee_1(0,1,2,3,4,9,11,12,14,16,19,22,23,25,29)$
21	$f = \vee_1(4,5,7,10,11,12,14,16,22,23,24,25,27,28,30)$

22	$f = \vee_1(0,2,6,7,10,13,15,16,17,19,25,26,29,30,31)$
23	$f = \vee_1(0,3,5,6,9,10,11,12,13,15,20,25,26,27,31)$
24	$f = \vee_1(0,2,4,7,8,9,10,15,16,17,19,20,22,25,27)$
25	$f = \vee_1(0,2,4,8,11,13,14,18,19,20,21,23,25,30,31)$
26	$f = \vee_1(1,2,3,4,6,7,10,17,19,21,24,25,26,29,31)$
27	$f = \vee_1(0,3,8,9,10,14,15,17,20,22,23,24,27,30,31)$
28	$f = \vee_1(0,1,4,7,11,14,18,19,20,24,25,27,28,29,30)$
29	$f = \vee_1(0,3,4,6,9,13,14,16,17,18,19,20,28,29,30,31)$
30	$f = \vee_1(2,3,5,7,8,9,10,11,12,13,15,16,17,18,26)$