

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего профессионального
образования
**«МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)»**
ВОЛЖСКИЙ ФИЛИАЛ

Кафедра математики и информатики



Лабораторный практикум по дисциплине

«Аппаратно-программные комплексы»

для направления

09.03.01 «Информатика и вычислительная техника»

профиля

**«Автоматизированные системы обработки информации и
управления»**

ЧАСТЬ 1

**TRACE MODE В РАЗРАБОТКЕ КОМБИНАЦИОННЫХ
МОДЕЛЕЙ**

Для студентов очной формы обучения

Чебоксары 2015

ББК 22.11
К862

Рецензенты:

М.В. Максимова, кандидат физико-математических наук доцент кафедры математики и информатики Волжского филиала Московского автомобильно-дорожного государственного технического университета (МАДИ)

О.Н. Зайцев, кандидат технических наук профессор кафедры «Управление в технических системах и программирование» Чебоксарского политехнического института (филиала) ФГБОУ ВПО «Московский государственный машиностроительный университет (МАМИ)»

К 862 Изосимова Т.А., Ксенофонтова И.В. Лабораторный практикум по дисциплине «Аппаратно-программные комплексы». Часть 1. – Чебоксары: Волжский филиал Московского автомобильно-дорожного государственного технического университета (МАДИ), 2015. – 37 с.

Методическое пособие составлено в соответствии с действующей программой курса «Аппаратно-программные комплексы» для студентов направления 09.03.01 «Информатика и вычислительная техника». Предназначено для студентов очной формы обучения. В нем кратко изложены необходимые теоретические сведения и методические указания для выполнения лабораторных работ по дисциплине «Аппаратно-программные комплексы».

*Печатается по решению Учебно-методического совета
Волжского филиала МАДИ*

© Изосимова Т.А., Ксенофонтова И.В., 2015
© Волжский филиал МАДИ, 2015

Оглавление

Лабораторная работа №1. ОСНОВЫ АЛГЕБРЫ ЛОГИКИ	4
Лабораторная работа №2. ЛОГИЧЕСКИЙ СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ	8
Лабораторная работа №3. ПОСТРОЕНИЕ КОМБИНАЦИОННЫХ МОДЕЛЕЙ СРЕДСТВАМИ ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЫ TRACE MODE.....	11
Лабораторная работа №4. РАЗРАБОТКА ГРАФИЧЕСКОГО ИНТЕРФЕЙСА	17
Лабораторная работа № 5. РАЗРАБОТКА ПРОЕКТА АВТОМАТИЗАЦИИ СРЕДСТВАМИ TRACE MODE 5.....	22
6. ПРИЕМЫ РАБОТЫ С TRACE MODE 5	33
СПИСОК ЛИТЕРАТУРЫ	36

Лабораторная работа №1. ОСНОВЫ АЛГЕБРЫ ЛОГИКИ

Для анализа и синтеза схем в ЭВМ при алгоритмизации и программировании решения задач широко используется математический аппарат алгебры логики.

Алгебра логики – это раздел математической логики, значение всех элементов (функций и аргументов) которой определены в двухэлементном множестве: 0 и 1. Алгебра логики оперирует с логическими высказываниями.

Высказывание – это любое предложение, в отношении которого имеет смысл утверждение об его истинности или ложности. При этом считается, что высказывание удовлетворяет закону исключенного третьего, т.е. каждое высказывание или истинно, или ложно и не может быть одновременно и истинным, и ложным. Например, высказывание «частное от деления 10 на 2 равно 3» - ложное утверждение.

В алгебре логики все высказывания обозначаются буквами a , b , c и т.д. Простейшими операциями в алгебре логики являются операции логического сложения (рис. 1), умножения (рис.2) и отрицания (рис. 3).

♦ *Логическое сложение* – операция *ИЛИ* – *дизъюнкция*, для обозначения операции используют символы $+$ или \vee , в соответствии с международным стандартом логический блок изображается следующим образом (рис.1.1):

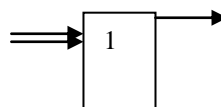


Рис. 1.1. Блок логического сложения.

♦ *Логическое умножение* – операция *И* – *конъюнкция*, для обозначения операции используют символы $*$ или \wedge , в соответствии с международным стандартом логический блок изображается следующим образом (рис. 1.2):

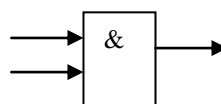


Рис. 1.2. Блок логического умножения.

♦ *Операция отрицания* – операция *НЕ* – *инверсия*, обозначаемая чертой над элементом, в соответствии с международным стандартом логический блок изображается следующим образом (рис. 1.3):

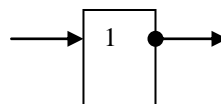


Рис. 1.3. Блок логической операции отрицания.

Наименьшим элементом алгебры логики является 0, наибольшим элементом 1.

В перечень машинных команд, которые используются при программировании входят и некоторые логические операции (OR, AND, XOR, NOT):

- ♦ логическое сложение: $0+0=0$
 $0+1=1$
 $1+1=1$
- ♦ логическое умножение: $0*0=0$
 $0*1=0$
 $1*1=1$

♦ исключающее ИЛИ: операция отрицания разнзначности, устанавливает 1 в тех битах результата, в которых исходные числа отличались друг от друга:

$$\begin{aligned} 0+0 &= 0 \\ 0+1 &= 1 \\ 1+1 &= 0 \end{aligned}$$

- ♦ операция отрицания: $\bar{1} = 0, \bar{0} = 1$

Законы алгебры логики:

1) комбинационные:

- тавтологии: $a+a=a$
 $a*a=a$
- сочетательный: $(a+b)+c=a+(b+c)$
 $(a*b)*c=a*(b*c)$
- переместительный: $a+b=b+a$
 $a*b=b*a$
- распределительный: $a*(b+c)=a*b+a*c$
 $a+(b*c)=(a+b)*(a+c)$
- поглощения: $a+a*b=a$
 $a*(a+b)=a$
- склеивания: $a*b+a*\bar{b}=a$
 $(a+b)(a+\bar{b})=a$

2) отрицания:

- двойного отрицания: $\bar{\bar{a}} = a$
- дополненности: $\bar{a}+a=1, a*\bar{a}=0$
- двойственности: $\overline{a+b} = \bar{a}*\bar{b}, \overline{a*b} = \bar{a}+\bar{b}$

Функция в алгебре логики – это алгебраическое выражение, содержащее элементы алгебры логики $a, b, c \dots$, связанные между собой операциями, определенными в этой алгебре:

$$f(a,b,c) = \bar{a} + \bar{a}*\bar{b}*c + \overline{a*c}$$

Согласно теоремам разложения функций на конституэнты (составляющие) любая функция может быть разложена на конституэнты «1»:

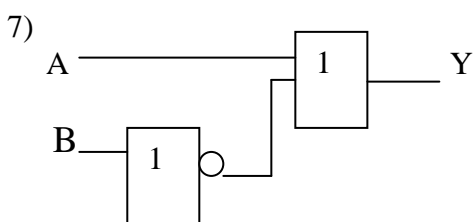
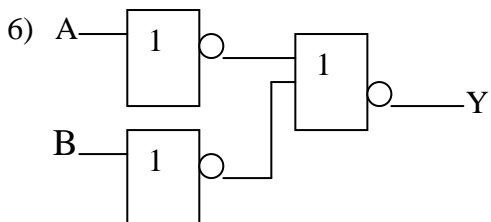
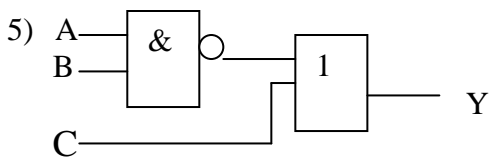
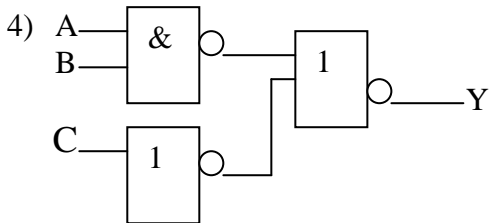
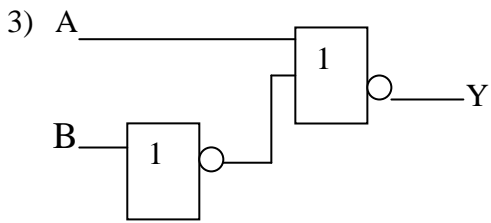
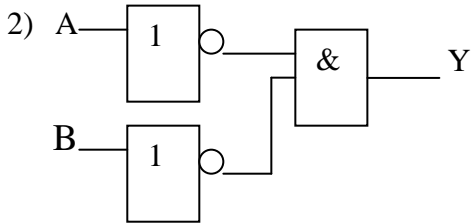
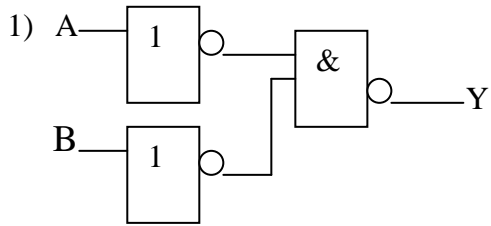
$$f(a) = f(1)*a + f(0)*\bar{a};$$

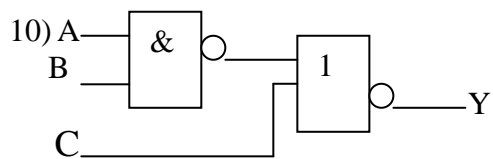
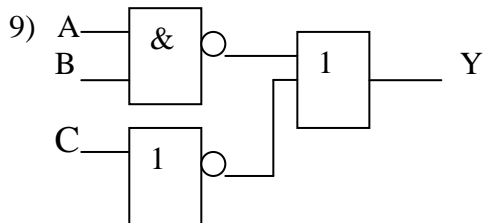
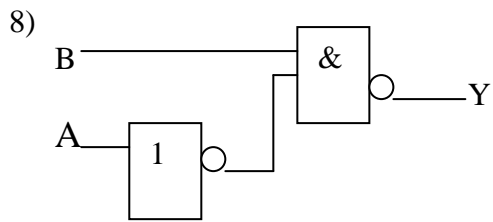
$$f(a,b) = f(1,b)*a + f(0,b)*\bar{a} = f(1,1)*a*b + f(1,0)*a*\bar{b} + f(0,1)*\bar{a}*b + f(0,0)*\bar{a}*\bar{b}$$

Эти соотношения используются для синтеза логических функций и вычислительных схем.

КОНТРОЛЬНЫЕ ЗАДАНИЯ

Задание 1.1. По заданной логической схеме составить логическое выражение и заполнить для него таблицу истинности:





Задание 1.2. По заданному логическому выражению составить логическую схему и построить таблицу истинности:

- 1) $A \text{ AND } B \text{ OR NOT } C$;
- 2) $A \text{ AND NOT } B \text{ OR } C$;
- 3) $\text{NOT } (A \text{ AND NOT } B) \text{ OR } C$;
- 4) $A \text{ OR NOT } (\text{NOT } B \text{ AND } C)$;
- 5) $\text{NOT } (A \text{ AND } B) \text{ OR NOT } C$;
- 6) $\text{NOT } (\text{NOT } A \text{ OR } B \text{ OR } C)$;
- 7) $A \text{ OR NOT } B \text{ AND } C$;
- 8) $\text{NOT } (A \text{ OR } B) \text{ AND NOT } C$;
- 9) $\text{NOT } A \text{ OR } B \text{ AND } C$;
- 10) $\text{NOT } (\text{NOT } A \text{ OR } B \text{ AND NOT } C)$.

Лабораторная работа №2. ЛОГИЧЕСКИЙ СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ

Рассмотрим *логический синтез* (создание) комбинационной схемы на примере одноразрядного двоичного сумматора, имеющего два входа («a» и «b») и два выхода («s» и «p») и выполняющего операцию арифметического сложения в соответствии с заданной таблицей 2.1:

Таблица 2.1

Таблица истинности

a	b	$f_1(a,b)=s$	$f_2(a,b)=p$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

где $f_1(a,b)=s$ – значение цифры суммы в данном разряде;

$f_2(a,b)=p$ – цифра переноса в следующий (старший) разряд.

Согласно соотношению, рассмотренному выше, можно записать:

$$s = f_1(a,b) = 0 * a * b + 1 * a * \bar{b} + 1 * \bar{a} * b + 0 * \bar{a} * \bar{b} = \bar{a} * b + a * \bar{b};$$

$$p = f_2(a,b) = 1 * a * b + 0 * a * \bar{b} + 0 * \bar{a} * b + 0 * \bar{a} * \bar{b} = a * b$$

Логическая схема сумматора, реализующего полученную функцию (рис.2.1):

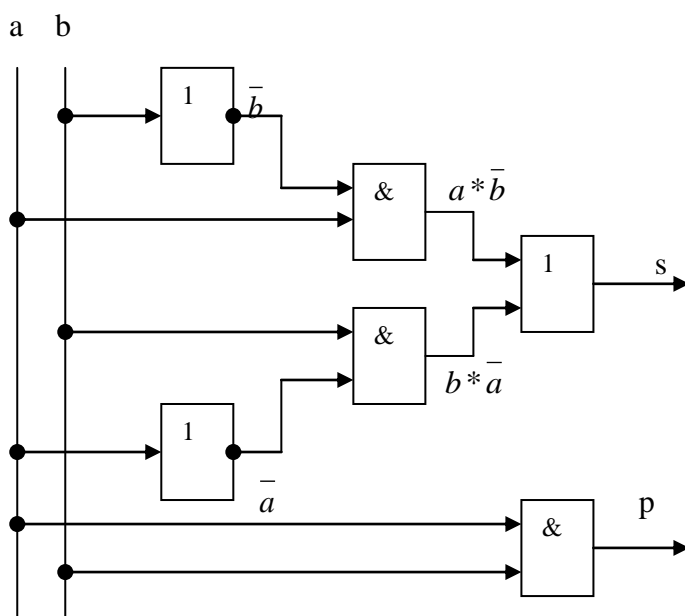


Рис. 2.1. Логическая схема сумматора.

Таким образом, комбинационные логические устройства строятся на основе логических элементов.

Рассмотрим построение еще нескольких комбинационных схем.

Пример 2.1. Требуется создать логическое устройство для подключения напряжения к агрегату. Агрегат может быть включен непосредственно ($A=1$)

или по команде с диспетчерского пункта ($B=1$). Агрегат работает только тогда, когда напряжение питания $U > U_{\min}$ (логическая функция $C=1$).

В соответствии с условиями задачи заполняем таблицу истинности, в которой записываем значение выходной функции F в зависимости от входных функций A, B, C для всех возможных вариантов их сочетаний. При трех входных функциях число сочетаний $N=2^3=8$ (таблица 2.2).

Таблица 2.2

Таблица истинности

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Запишем логическую функцию:

$$F = \bar{A} * B * C + A * \bar{B} * C + A * B * C$$

Составим логическую схему (рис. 2.2):

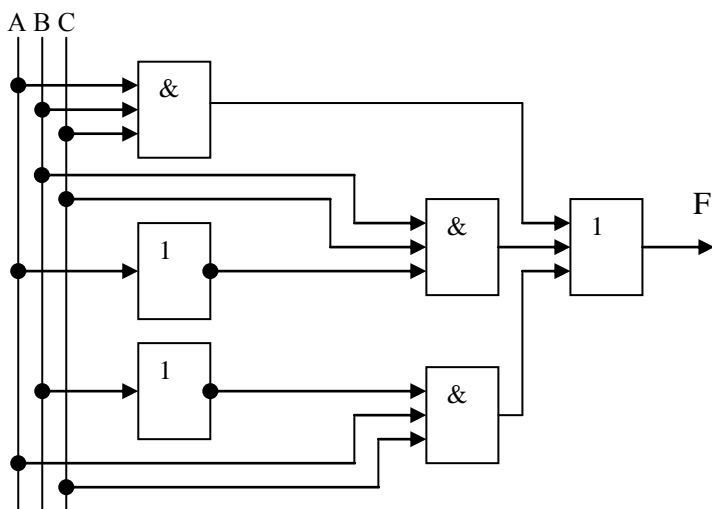


Рис. 2.2.

Пример 2.2. Произвести сортировку деталей на три группы по размеру, равному a , $2a$ и больше $3a$, открыв заслонки бункеров накопителей, предназначенных для отбора деталей каждого типа. Контроль размеров деталей можно осуществить тремя датчиками $d_1 - d_3$, установленными поперек роликового транспортера. Обозначив бункеры b_1, b_2 и b_3 , выделим следующие ситуации:

- 1) идет деталь размера a – перекрыт один из датчиков (d_1 , или d_2 , или d_3), открыт бункер b_1 (№ 4, 2, 1);
- 2) идет деталь $2a$ – перекрыты два датчика (d_1, d_2 , или d_2, d_3), открыт бункер b_2 (№ 6,3);

3) идет деталь размером $3a$ – перекрыты все три датчика, открыт бункер b_3 (№ 7).

Таблица истинности (табл. 2.3) составляется для всех возможных комбинаций командных датчиков. Число датчиков $n = 3$, число комбинаций $N = 8$. Незаполненные клетки состояний входов (№ 5) соответствуют нереальной ситуации, когда деталь перекроет датчики d_1 и d_3 , и не перекроет d_2 ; она может возникнуть лишь в результате неисправности датчика d_2 . Эти клетки заполняются нулями для предотвращения аварийной ситуации (все бункеры закрыты, звучит сигнал)».

Таблица 2.3

Таблица истинности

Номера комбинаций	Состояние					
	входов			выходов		
	d_1	d_2	d_3	b_1	b_2	b_3
0	0	0	0	0	0	0
1	0	0	1	1	0	0
2	0	1	0	1	0	0
3	0	1	1	0	1	0
4	1	0	0	1	0	0
5	1	0	1	0	0	0
6	1	1	0	0	1	0
7	1	1	1	0	0	1

Для демонстрации принципов действия разработанных комбинационных схем воспользуемся программным продуктом TRACE MODE, бесплатная базовая версия которого имеется в распоряжении любого технического вуза.

Лабораторная работа №3.

ПОСТРОЕНИЕ КОМБИНАЦИОННЫХ МОДЕЛЕЙ СРЕДСТВАМИ ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЫ TRACE MODE

TRACE MODE – это программный комплекс, предназначенный для разработки, настройки и запуска в реальном времени систем управления технологическими процессами. Все программы, входящие в TRACE MODE, делятся на две групп:

- 1) инструментальная система разработки АСУ;
- 2) исполнительные модули.

Инструментальная система включает в себя три редактора:

- Редактор базы каналов;
- Редактор представления данных;
- Редактор шаблонов.

В них разрабатываются: база данных реального времени, программа обработки данных и управления, графические экраны для визуализации состояния технологического процесса и управления им, а так же шаблоны для генерации отчетов о работе производства.

Исполнительные модули – это программы, под управлением которых запускается АСУ, созданная в инструментальной системе.

Рассмотрим решение примера из задания 2.2 (см. выше): A AND B OR NOT C, используя инструментальную систему TRACE MODE 5. Для этого необходимо выполнить следующую последовательность шагов:

- создать проект в редакторе базы каналов;
- создать узел;
- создать объект база;
- создать канал в объекте база;
- создать FBD-программу.

Разработка любого проекта всегда начинается в редакторе базы каналов. Чтобы загрузить этот редактор, надо выполнить команду **Редактор базы каналов** из группы установки инструментальной системы в меню **Программы WINDOWS**.

При этом осуществляется запуск редактора базы каналов и на экране появляется его окно, показанное на рисунке 3.1:

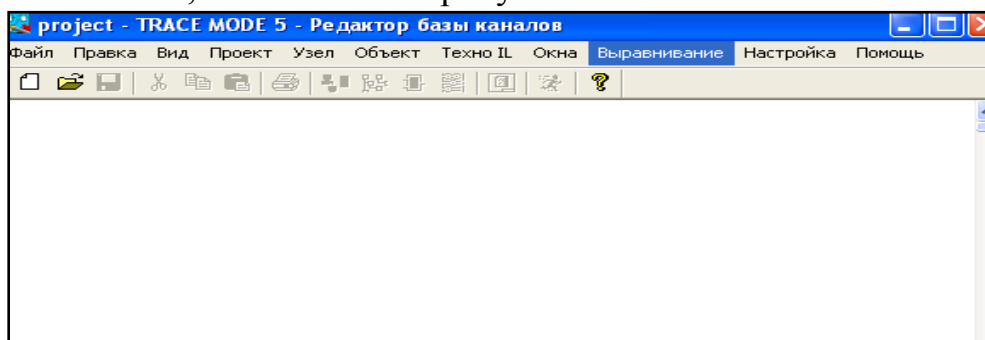


Рис. 3.1.

Для создания нового проекта следует нажать ЛК на иконке инструментальной панели, обозначенной на предыдущем рисунке. При этом на экран выводится следующий диалог (рис. 3.2):



Рис. 3.2.

Зададим произвольное имя проекту. Подтвердим завершение настройки параметров проекта нажатием ЛК на кнопке ОК. При этом диалог **Свойства проекта** исчезнет с экрана, а в заголовке окна редактора базы каналов и его строке статуса появится название нового проекта.

Теперь перейдем к созданию структуры проекта. Она включает в себя перечень **узлов** – операторских станций и контроллеров, которые работают под управлением TRACE MODE.

В нашем случае потребуется создать один узел - операторскую станцию.

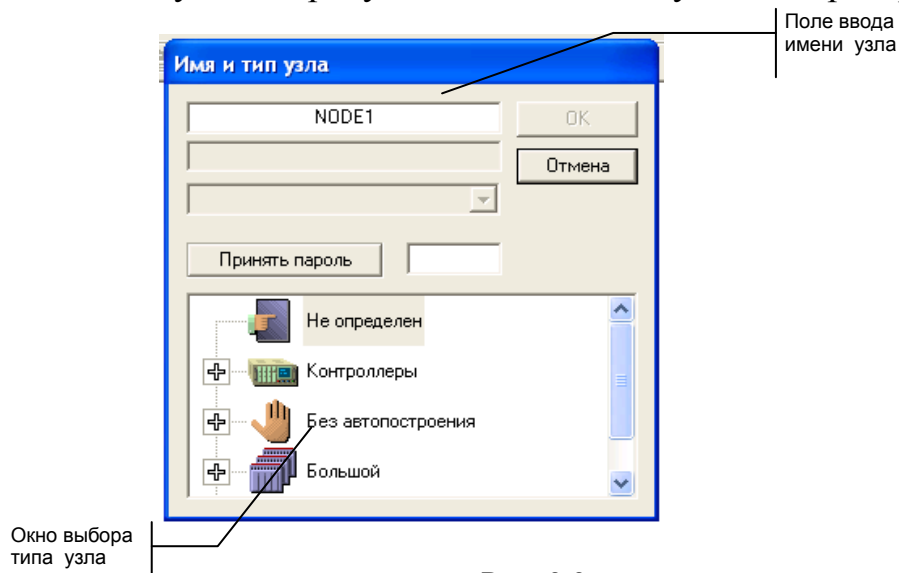


Рис. 3.3.

Для создания узла надо выполнить команду **Создать** из меню **Узел** или нажать ПК в рабочей области редактора базы каналов. При этом на экране появится диалог **Имя и тип узла**, показанный на рисунке 3.3. В окне выбора типа узла раскроем список узлов под названием **Большой**. Для этого нажмем ЛК в области [+], расположенной левее названия. Для примера выберем тип операторской станции МРВ. Подтвердим создание узла нажатием ЛК на кнопку ОК. При этом диалог **Имя и тип узла** исчезает с экрана, а в рабочем поле редактора базы каналов появляется обозначение созданного узла (рис. 3.4):

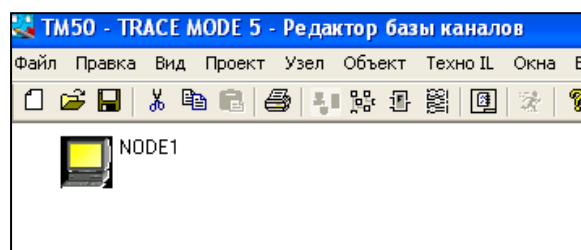


Рис. 3.4.

Перейдем теперь к созданию базы каналов нашего узла. Для этого дважды нажмем ЛК на его изображении в рабочем поле окна редактора базы каналов. При этом открывается окно редактирования базы каналов, в котором выводятся изображения объектов базы каналов. Для редактирования каналов объекта дважды нажмем ЛК на его изображении. При этом на экране появится диалог **Каналы объекта**, показанный на рисунке 3.5:

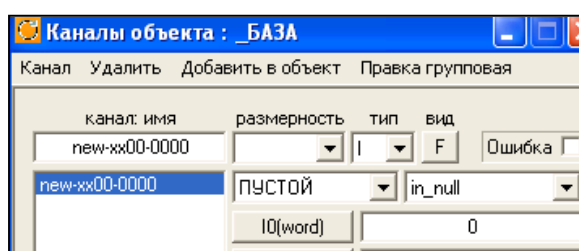


Рис. 3.5.

Следующим шагом нашей работы будет разработка FBD-программы, реализующей принцип действия рассматриваемой нами логической схемы.

Чтобы создать FBD-программу, следует сначала указать, на каком узле она будет использоваться. Для этого нужно либо войти в режим редактирования базы каналов этого узла, либо просто выделить его в окне структуры проекта. В нашем случае выделим узел **NODE 1**. Далее для перехода в окно редактирования FBD-программ надо выполнить команду **FBD-программы** из меню **Окна**. При этом на экране появляется диалог **FBD-программа**. В нем можно выбрать FBD-программу для редактирования или создать новую.

Для создания новой программы надо выполнить команду **Создать** из меню **Программа** диалога **FBD-программа**. По умолчанию создаваемой программе присваивается имя **FormN**, где **N** – ее номер по порядку в данном узле. Для изменения имени программы в данном диалоге предусмотрено специальное поле.

Создадим новую программу, присвоим ей имя **Задание2** и введем для нее комментарий, как показано на рисунке 3.6:

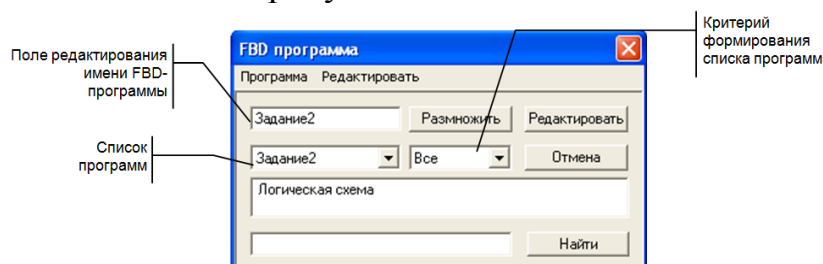


Рис. 3.6.

Для перехода к редактированию созданной FBD-программы следует нажать ЛК на кнопке **Редактировать**. При этом диалог **FBD-программа** исчезнет с экрана, а в рабочее поле редактора базы каналов будет выведена выбранная FBD-программа. Кроме того, на экране появится диалог управления редактированием **Меню FBD**.

В нашем случае, поскольку программа только что создана, рабочее поле редактора будет пустым. В нем будет только присутствовать диалог управления редактированием **Меню FBD**, показанный на рисунке 3.7:

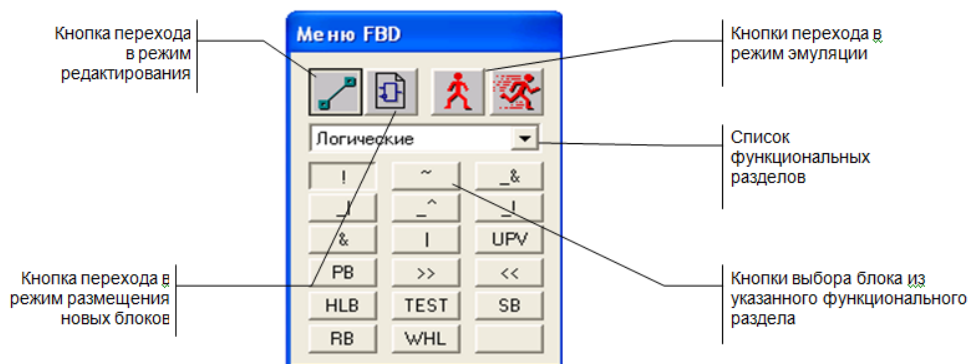


Рис. 3.7.

Перейдем в режим размещения новых блоков. Это делается нажатием ЛК на соответствующей кнопке диалога **Меню FBD** и выполним следующую последовательность действий:

- выбрать функциональный раздел **Логические**;
- выбрать блок побитового логического умножения (**_&**) и разместить его нажатием ЛК в рабочем поле;
- выбрать блок инверсии (**!**) и разместить его в рабочем поле;
- выбрать блок побитового логического сложения (**_|**) и разместить его в рабочем поле.

Далее следует перейти в режим редактирования связей. Теперь следует связать входы и выходы размещенных блоков, как показано на рисунке 3.8:

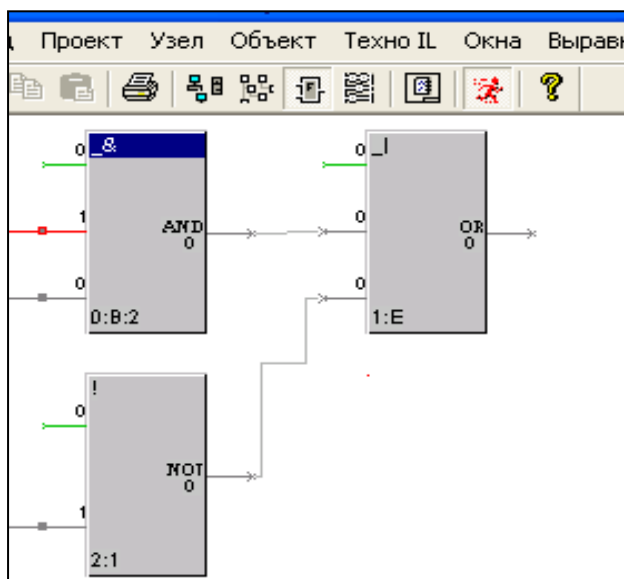


Рис. 3.8.

Для соединения входа одного блока с выходом другого надо сначала выделить с помощью ЛК этот вход, затем снова нажать ЛК (при этом будет слышен звуковой сигнал) и, удерживая ее, переместить курсор в область второго конца связи. Для уничтожения связи следует выделить ее со стороны входа блока и нажать клавишу **DEL**.

Далее следует описать переменные и константы данной программы. Для описания любой переменной программы следует дважды нажать ЛК на соответствующем входе или выходе функционального блока. При этом на экране появится следующий диалог (рис.3.9):

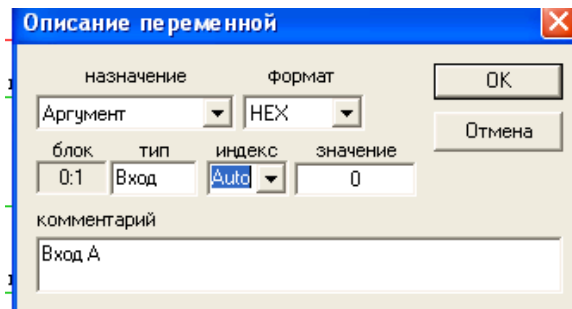


Рис. 3.9.

Настроим входы и выходы блоков следующим образом (табл.3.1):

Таблица 3.1

Блок	Вход, выход	Назначение	Комментарий
(&_)	IW1	аргумент	Вход А
	IW2	аргумент	Вход В
(!)	INB	аргумент	Вход С
(_)	OR	аргумент	Выход F

На этом разработка программы завершена. Затем запускается режим эмуляции, в котором задаются сигналы на входах блоков и при этом можно наблюдать, что происходит на выходе логической схемы.

КОНТРОЛЬНЫЕ ЗАДАНИЯ

Задание 1. Средствами TRACE MODE 5 разработать FBD-программу, реализующую функцию комбинационного устройства, в котором X – обозначение его входов, Y – обозначение его выходов:

1)

X ₁	X ₂	Y ₁	Y ₂	Y ₃	Y ₄
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

2)

X ₁	X ₂	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅
0	0	1	1	1	1	0
0	1	0	0	1	0	0
1	0	0	1	0	1	1

1	1	0	1	1	1	0
---	---	---	---	---	---	---

3)

X ₁	X ₂	X ₃	X ₄	Y ₁	Y ₂
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

4)

X ₁	X ₂	Y ₁	Y ₂	Y ₃	Y ₄
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

5)

X ₁	X ₂	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅
0	0	1	1	1	1	0
0	1	0	1	0	0	0
1	0	1	0	1	0	1
1	1	1	1	1	0	0

6)

X ₁	X ₂	X ₃	X ₄	Y ₁	Y ₂
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

7)

X ₁	X ₂	Y ₁	Y ₂	Y ₃	Y ₄
0	0	1	0	0	0
0	1	0	0	1	0
1	0	0	1	0	0
1	1	0	0	0	1

8)

X ₁	X ₂	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅
0	0	0	1	1	1	1
0	1	0	0	1	0	0
1	0	1	1	0	1	0
1	1	0	1	1	1	0

9)

X ₁	X ₂	X ₃	X ₄	Y ₁	Y ₂
1	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	0	0	1	1	1

10)

X ₁	X ₂	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅
0	0	1	1	1	1	0
0	1	0	0	0	1	0
1	0	1	0	1	0	1
1	1	1	0	1	1	0

Лабораторная работа №4. РАЗРАБОТКА ГРАФИЧЕСКОГО ИНТЕРФЕЙСА

Как уже было сказано выше, при управлении технологическим циклом необходимо формировать дискретную последовательность команд исполнительным элементам технологического объекта управления. Формирование команд осуществляется на основе логического анализа ситуации, о которой сообщают различные датчики.

Для разработки формальной модели управления технологическим циклом можно воспользоваться следующим редактором инструментальной системы TRACE MODE 5 – Редактором представления данных.

Рассмотрим решение подобной задачи на основе примера 2.2 (см. выше) «Контроль размеров детали». Составляется FBD программа (назовем ее **Control**) (рис. 4.1) и далее разрабатывается графическая база узла. Эта база включает в себя все графические фрагменты, которые выводятся на монитор операторской станции, осуществляющей контроль размеров деталей (рис 4.2).

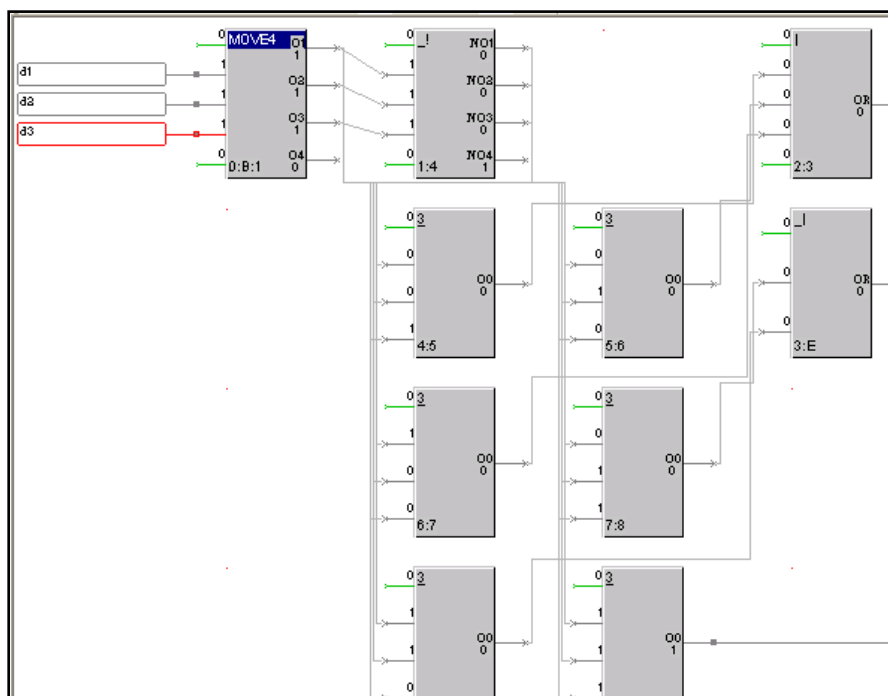


Рис. 4.1.

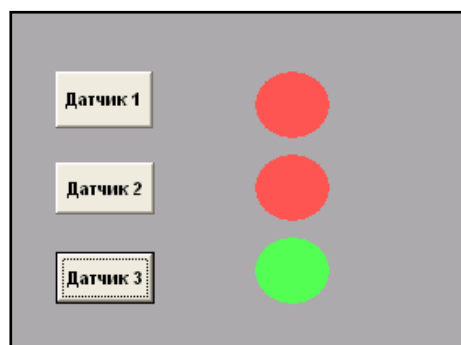


Рис. 4.2.

Прежде чем начать разработку графической базы узла, необходимо осуществить привязку разработанной FBD-программы к каналам данного узла.

Для создания каналов войдем в диалог **Каналы объекта** для объекта NODE1. Нам потребуется три входных (Д1, Д2, Д3) и три выходных канала (Б1, Б2, Б3) и еще один дополнительный входной канал, его имя и подтип **Управление**.

Далее свяжем разработанную FBD-программу с каналом **Управление**. Для этого откроем диалог **Реквизиты** канала **Управление**, дважды нажав ЛК на имени канала в списке. Войдем в бланк **Управление** диалога **Реквизиты** и в поле выбора FBD-программ укажем **Control**.

В этом бланке присутствует список для выбора программы, поле комментария к программе, список ее аргументов и констант, а также комментариев к настраиваемому элементу этого списка.

Для настройки любого аргумента или константы надо дважды нажать ЛК на соответствующей строке списка. Если выбрана константа, то на экране появится диалог **Значение переменной**. В нем можно ввести значение константы для данного вызова программы. При настройке аргумента на экран выводится диалог, в котором выбирается канал и его атрибут, со значением которого связывается настраиваемый аргумент. Свяжем вход d1 с реальным значением канала Д1 объекта NODE1, вход d2 – с реальным значением канала Д2 объекта NODE1, вход d3 - реальным значением канала Д3 объекта NODE1, выход b1 - с входом канала Б1 объекта NODE1, выход b2 - с входом канала Б2 объекта NODE1, выход b3 - с входом канала Б3 объекта NODE1. Завершим настройку нажатием ЛК на кнопке ОК диалога **Реквизиты**.

Разработка графического интерфейса для операторских станций осуществляется в редакторе представления данных. В него загружается структура проекта, созданная в редакторе базы каналов. Выбрав требуемый узел проекта, можно редактировать его графическую базу.

Для запуска редактора представления данных следует войти в папку **Программы** главного меню WINDOWS, затем - в папку **Trace Mode 5** и выполнить команду **Редактор представления данных**. При этом на экране появится окно редактора.

Редактор представления данных имеет главное меню, рабочую область, строку статуса навигатор проекта и четыре инструментальные панели. В навигаторе проекта выводятся список узлов проекта, состав их графических баз и списки загруженных графических библиотек. Инструментальные панели используются для выбора графических элементов и управления редактированием графической базы.

Рабочая область редактора при загрузке не содержит никаких изображений, а инструментальные панели недоступны. Для создания и редактирования графических экранов необходимо загрузить проект и загрузить или создать графическую базу любого узла проекта.

Загрузим разработанный выше проект в редактор представления данных, выполнив команду **Открыть** из меню **Проект**. При этом на экран выводится диалог выбора файла структуры проекта. После загрузки проекта в бланке **Экраны** навигатора проекта появляется список присутствующих в проекте узлов: **NODE1**.

Выберем нажатием ЛК на бланке **Экраны** навигатора проекта узел **NODE1**. Затем нажатием ПК войдем в меню узлов этого бланка, показанное на рисунке 4.3

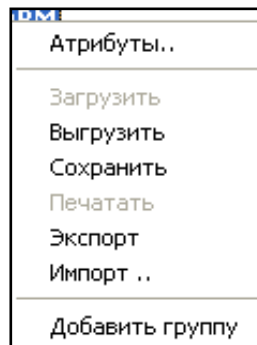


Рис.4.3.

Выполним команду **Добавить группу**. При этом для узла **NODE1** создастся группа экранов. Ей автоматически присваивается имя **Новая группа**. Далее выделим эту группу и нажмем ПК. При этом на экран выводится меню групп. Выполним в этом меню команду **Добавить экран**.

Разработка графического интерфейса заключается в размещении на экранах графической базы статических элементов рисования и динамических форм отображения. Графические элементы выбираются с помощью соответствующих инструментальных панелей. При этом на экран выводится диалог настройки их атрибутов (цвет фона, привязка к каналам и пр.).

В соответствии с рисунком 4.2 нам потребуются три **Кнопки** и три **Цветовых индикатора**. С помощью форм отображения **Кнопки** мы будем посылать значения на вход «черного ящика» (данные с датчиков), **Цветовые индикаторы** будут сообщать нам состояние на выходах «черного ящика» (открытие заслонок бункеров накопителей).

Чтобы создать и настроить форму отображения **Кнопка**, нажмем ЛК на соответствующей иконке инструментальной панели форм отображения, показанной на рисунке 4.4:

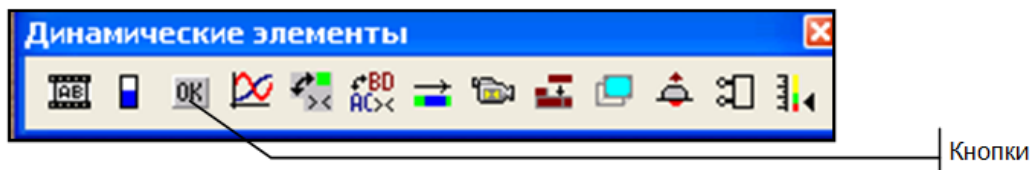


Рис.4.4.

Настроим атрибуты 1-й **Кнопки** как показано на рисунке 4.5:

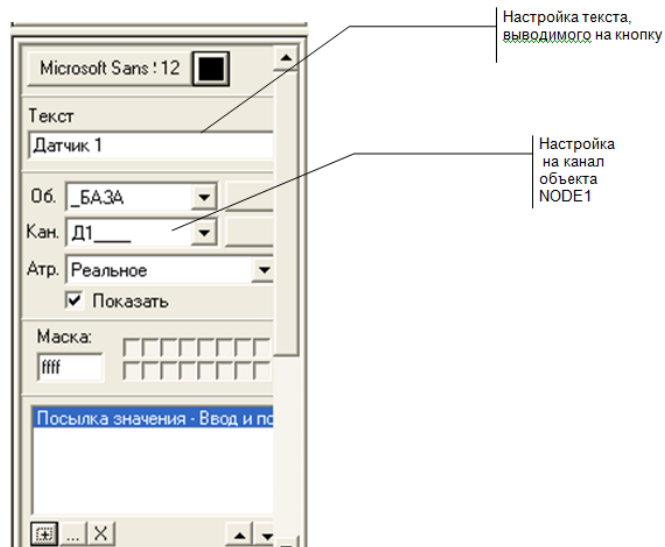


Рис.4.5.

Аналогично настроим атрибуты 2-й и 3-й кнопок, с привязкой к каналам Д2 и Д3 соответственно.

Для размещения цветowego индикатора, необходимо выбрать подложку, т.е. определить какой формы он будет. Выберем форму отображения **Эллипс** (рис.4.6) и поверх этой формы разместим **Цветовой индикатор на равенство** (рис. 4.7).



Рис. 4.6.



Рис. 4.7.

Настроим атрибуты формы отображения **Цветовой индикатор на равенство** как показано на рисунке 4.8:

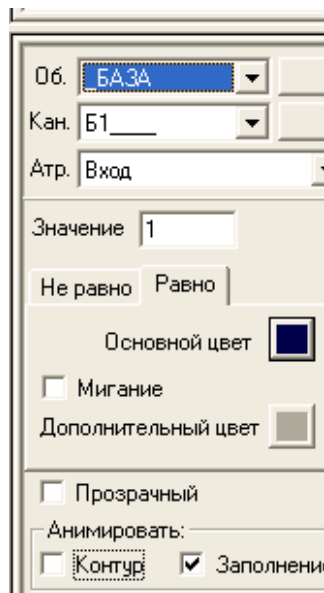


Рис. 4.8.

КОНТРОЛЬНЫЕ ЗАДАНИЯ

Задание 4.1. Для вхождения в автоматизированную систему пользователю нужно ввести имя и пароль. В зависимости от введенного имени и пароля пользователь получает доступ к определенным задачам. Например, имеются три пользователя и пять задач, первый пользователь имеет доступ к 1 и 3-й задаче (загорается соответствующий индикатор), второй - к 2, 4, 5-й задаче, третий – ко всем задачам».

<input type="text" value="имя"/>	<input type="radio"/> задача 1
<input type="text" value="пароль"/>	<input type="radio"/> задача 2
	<input type="radio"/> задача 3
	<input type="radio"/> задача 4
	<input type="radio"/> задача 5

Средствами TRACE MODE 5 разработать данный проект автоматизации, включающий в себя информационную структуру, математическую базу, графическую часть. При выполнении задания, необходимо разработать комбинационное устройство, которое имеет число входов равное числу пользователей автоматизированной системы и число выходов, равное числу решаемых в системе задач.

Задание 4.2. Необходимо обеспечить в двух соседних помещениях пожарную сигнализацию с возможностью определения возгорания в каждом помещении отдельно. В каждое помещение устанавливается по два извещателя: пожарный и дымовой. Средствами TRACE MODE 5 разработать проект системы пожарной сигнализации, включающий в себя информационную структуру, математическую базу, графическую часть.

Лабораторная работа № 5. РАЗРАБОТКА ПРОЕКТА АВТОМАТИЗАЦИИ СРЕДСТВАМИ TRACE MODE 5

Разработка проекта автоматизации средствами инструментальной системы TRACE MODE 5 включает в себя разработку информационной структуры проекта, математической базы и графического интерфейса.

1.1. Разработка структуры проекта и его математической базы

Откроем редактор базы каналов и создадим три узла: два контроллера и операторскую станцию. Назовем контроллеры PLC1 и PLC2, операторскую станцию - АРМ. Войдем в базу каналов для операторской станции, дважды щелкнув по узлу АРМ. Откроем диалог **Каналы объекта БАЗА** и создадим четыре канала (рис 5.1).

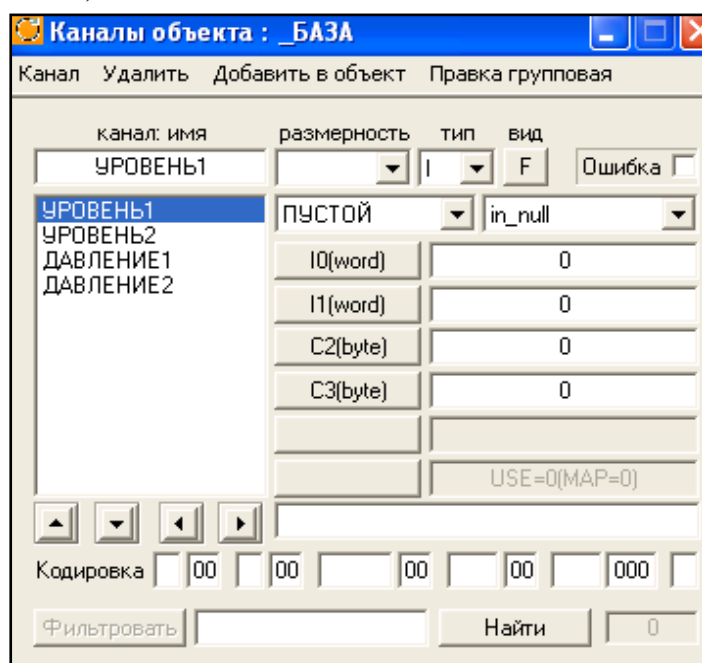


Рис. 5.1.

Создадим FBD-программу для узла АРМ. В ней будут случайным образом задаваться величины **Давления** и **Уровня**. Перейдем в режим размещения новых блоков и выполним следующую последовательность действий:

- выбрать функциональный раздел **Генераторы**;
- выбрать блок RND1 (выход блока является случайной величиной с законом распределения в диапазоне от 0 до 1);
- выбрать функциональный раздел **Арифметические**;
- выбрать блок (*);
- выбрать блок (+2).

Далее следует перейти в режим редактирования связей. Теперь следует связать входы и выходы размещенных блоков, как показано на рисунке 5.2.

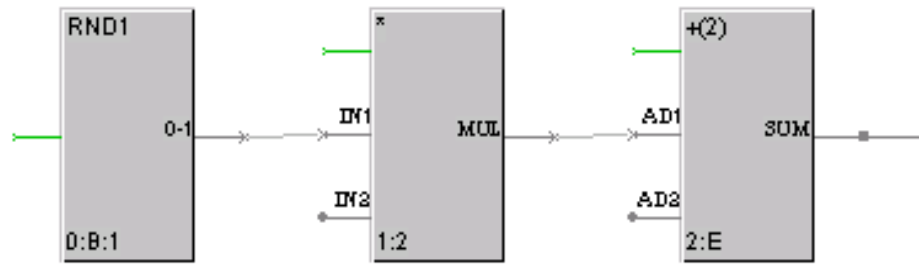


Рис. 5.2

Настроим входы и выходы блоков следующим образом (табл. 1.1):

Таблица 5.1

Блок	Вход, выход	Тип	Комментарий
(*)	IN2	константа	*rnd
(+2)	AD2	константа	+
	SUM	аргумент	out

Подключим FBD программу к каналам. Дважды нажав ЛК на имени первого канала в списке, войдем в бланк **Трансляция** диалога **Реквизиты** и в поле выбора FBD-программ укажем название нашей разработанной программы. Настроим аргументы и константы для каждого из каналов, как показано на рисунках 5.3 (а, б, в, г).

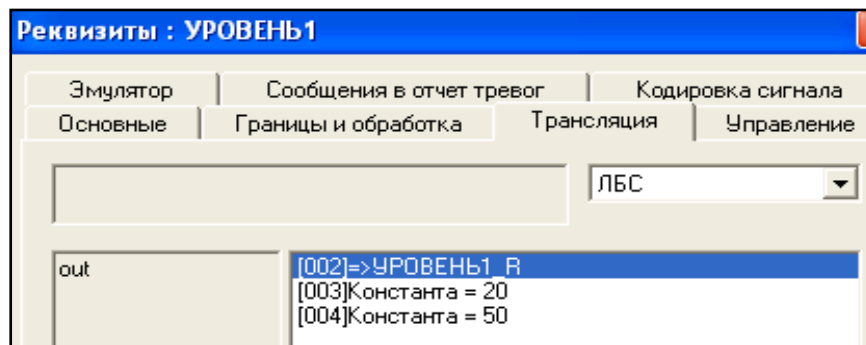


Рис. 5.3 (а).

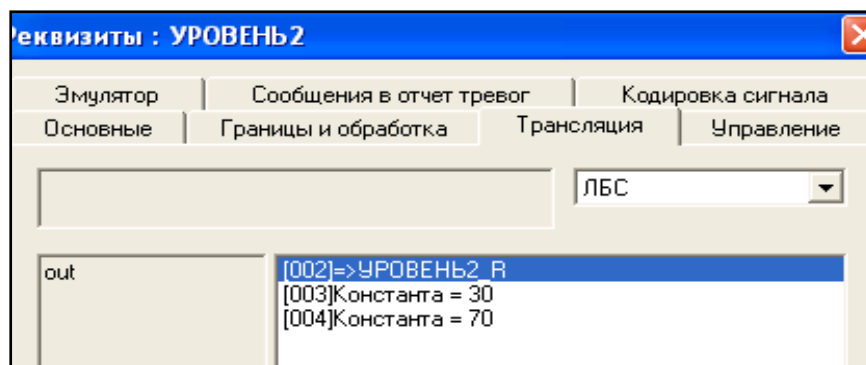


Рис. 5.3 (б).

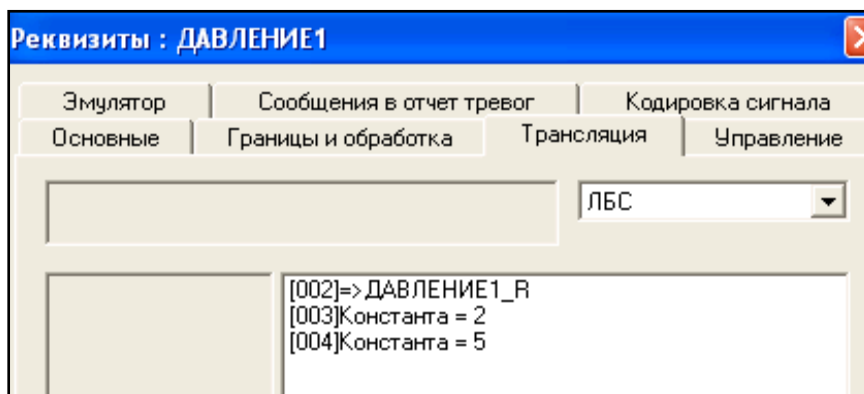


Рис. 5.3 (в).

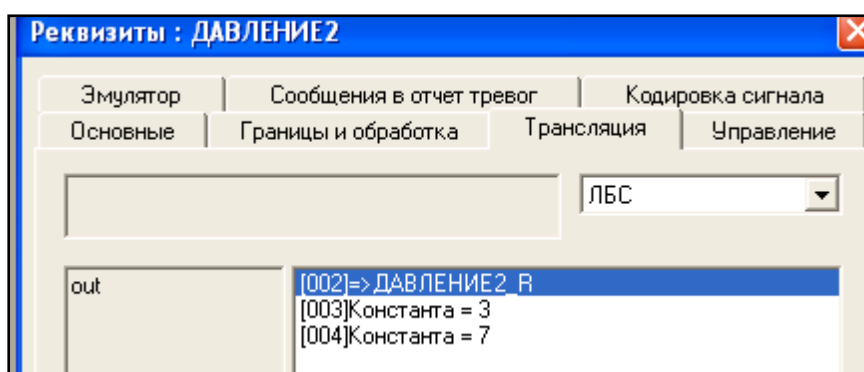


Рис. 5.3 (г).

1.2. Разработка графического интерфейса

Загрузим разработанный выше проект в редактор представления данных, выполнив команду **Открыть** из меню **Проект**. После загрузки проекта в бланке **Экраны** навигатора проекта появляется список присутствующих в проекте узлов: **ARM**, **PLC1** и **PLC2**.

Выберем нажатием ЛК на бланке **Экраны** навигатора проекта операторскую станцию **ARM**. Затем нажатием ПК войдем в меню узлов этого бланка, показанное на рисунке 5.4.

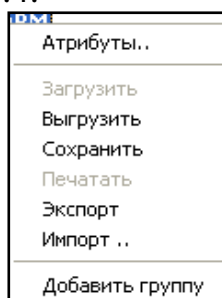


Рис. 5.4.

Выполним команду **Добавить группу**. При этом для узла **ARM** создастся группа экранов. Ей автоматически присваивается имя **Новая группа**. Далее выделим эту группу и нажмем ПК. При этом на экран выводится меню групп.

Выполним в этом меню команду **Добавить экран**. Снова войдем в это же меню и выполним ту же команду. Изменим имена экранов и группы. Первому экрану дадим имя **Участок 1**, второму - **Участок 2**, а группе - **Мнемосхемы**. Бланк **Экраны** примет следующий вид (рис. 5.5.):

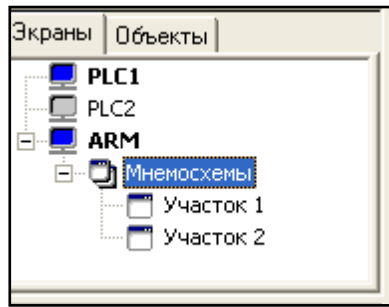


Рис. 5.5.

Разработка графического интерфейса заключается в размещении на экранах графической базы статических элементов рисования и динамических форм отображения. Графические элементы выбираются с помощью соответствующих инструментальных панелей. При этом на экран выводится диалог настройки их атрибутов (цвет фона, привязка к каналам и пр.).

Двойным нажатием ЛК на имени экрана **Участок 1** в бланке **Экраны** выведем его в рабочее поле редактора. Ниже показан статичный рисунок, который надо создать для этого экрана (рис. 5.6.).

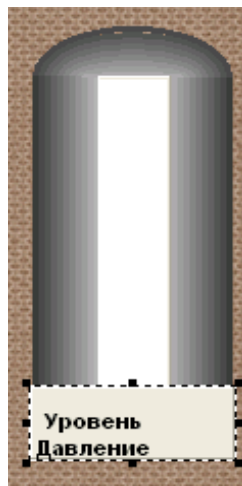


Рис. 5.6.

Рассмотрим последовательность его создания. Сначала разместим на экране объемный элемент – вертикальный цилиндр. Для этого выберем в панели элементов рисования соответствующую группу, как показано на рисунке 5.7.

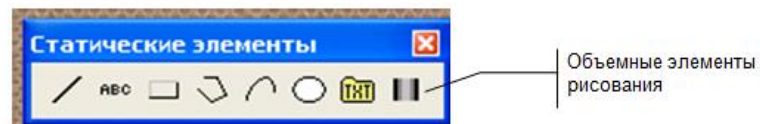


Рис. 5.7

При этом в левой части экрана под навигатором проекта выводится диалог настройки атрибутов для выбранного элемента рисования. Настроим атрибуты для цилиндра, как показано на рисунке 5.8.

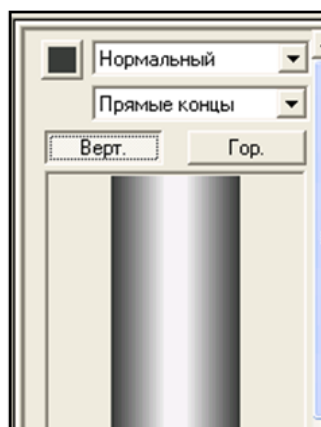


Рис. 5.8.

Далее разместим цилиндр на экране **Участок 1**. Теперь пририсовем к верхнему торцу цилиндра эллиптическую крышку. Для этого нажмем ЛК еще раз на иконке объемных элементов и в появившемся меню выберем элемент **Шар** (рис. 5.9).

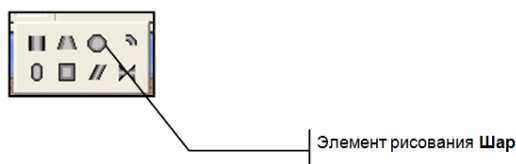


Рис.5.9.

Настроим атрибуты элемента рисования **Шар**, как показано на рисунке 5.10.

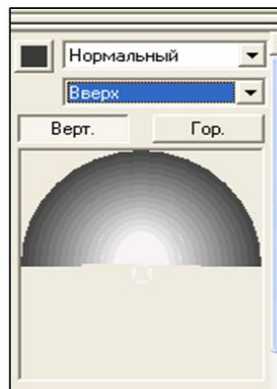


Рис. 5.10.

Нарисуем теперь рамку по центру емкости. В ней затем будет размещена гистограмма, отображающая уровень. Вторую рамку разместим под емкостью для вывода в ней числовых значений давления и уровня.

Нажмем дважды ЛК на иконке группы прямоугольников инструментальной панели элементов рисования, показанной на рисунке 5.11.

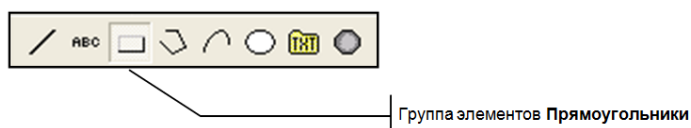


Рис. 5.11.

При этом на экране появится меню выбора элемента группы. Нажмем ЛК на элементе Рамка, показанном на рисунке 5.12.



Рис. 5.12.

После этого пиктограмма выбранного элемента появится в соответствующей ячейке инструментальной панели, а на экран будет выведен диалог настройки атрибутов элемента рисования **Рамка**. Вид этого диалога показан на рисунке 5.13.

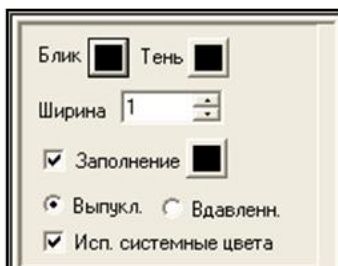


Рис. 5.13

Разместим две рамки, как показано на рисунке 5.14.

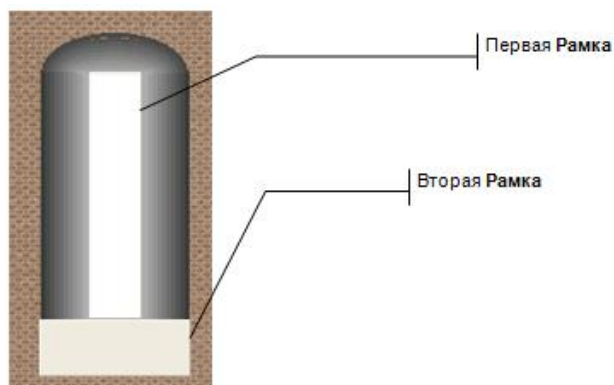


Рис. 5.14.

Теперь в рамке, размещенной под емкостью, сделаем две надписи: **Уровень** и **Давление**. Для этого нажмем ЛК на иконке статического текста в инструментальной панели элементов рисования (рис. 5.15).

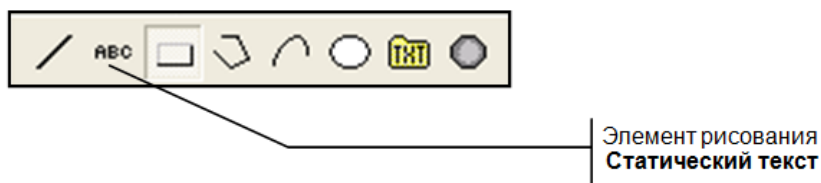


Рис. 1.15.

При этом на экран выводится диалог настройки атрибутов (рис.5.16).

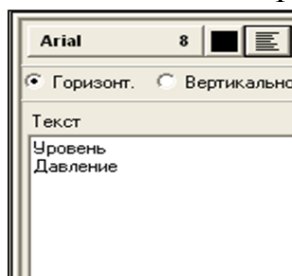


Рис. 5.16.

Перейдем теперь к размещению динамических элементов. Окончательный вид экрана **Участок 1** показан на рисунке 5.17:

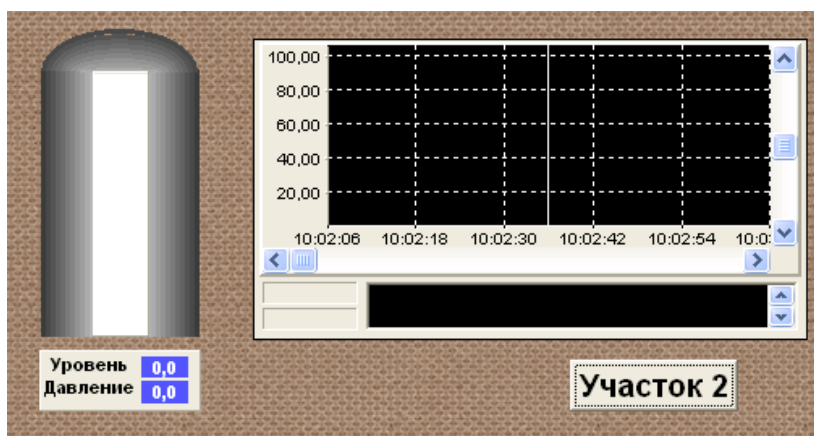


Рис. 5.17.

Чтобы показать уровень заполнения емкости, используем форму отображения **Гистограмма**. Выберем ее нажатием ЛК на соответствующей иконке инструментальной панели (рис. 5.18).

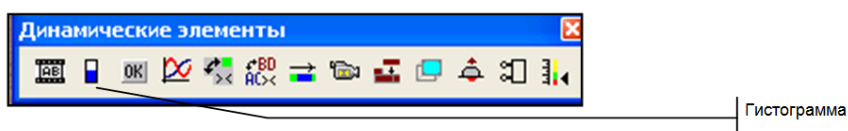


Рис. 5.18.

Настроим атрибуты гистограммы, как показано на рисунке 5.19.

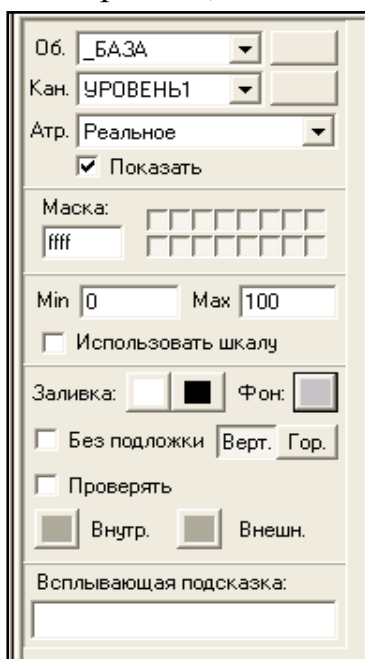


Рис. 5.19.

После этого разместим гистограмму внутри рамки, расположенной по центру емкости, размещая ее снизу вверх.

Для вывода значений параметров в виде чисел предназначена текстовая форма отображения. Для ее размещения нажмем ЛК на соответствующей

иконке инструментальной панели форм отображения, показанной на рисунке 5.20.

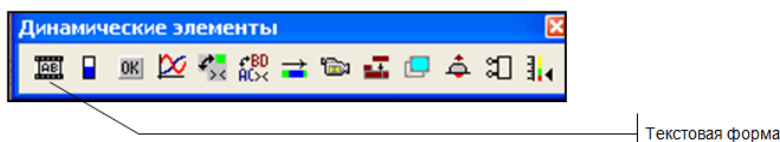


Рис. 5.20.

Настроим атрибуты текстовой формы, как показано на рисунке 5.21.

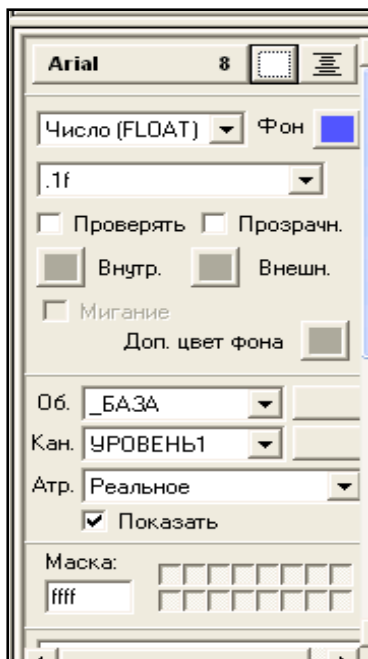


Рис. 5.21.

Свяжем данную форму отображения с реальным значением канала **УРОВЕНЬ1** из объекта **_БАЗА** и разместим ее на экране под изображением емкости напротив слова **Уровень**. Затем свяжем форму с каналом **ДАВЛЕНИЕ1** и разместим новую форму отображения напротив слова **Давление**. Рамка, расположенная под изображением емкости, будет выглядеть следующим образом (рис.5.22):

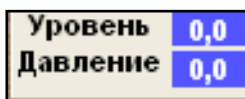


Рис.5.22.

Вывод трендов технологических параметров осуществляется с помощью специальной формы отображения (рис. 5.23).



Рис. 5.23

Настроим атрибуты этой формы отображения как показано на рис. 5.24 и перейдем к настройке линий графика.

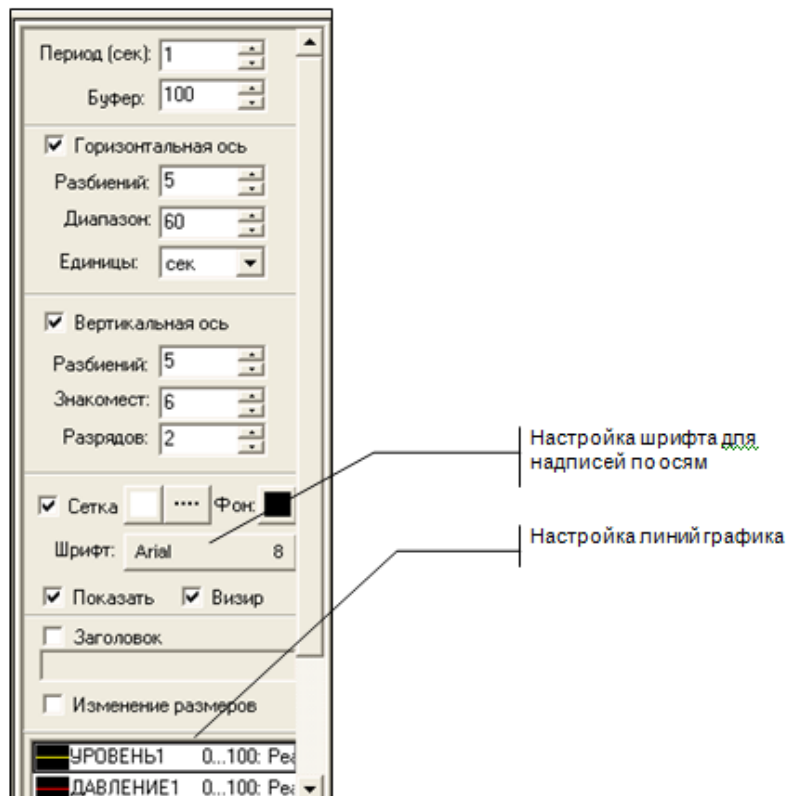


Рис. 5.24.

Для этого нажмем ЛК на кнопке **Добав**. При этом на экран выводится диалог **Атрибуты кривой**, показанный на рисунке 5.25:

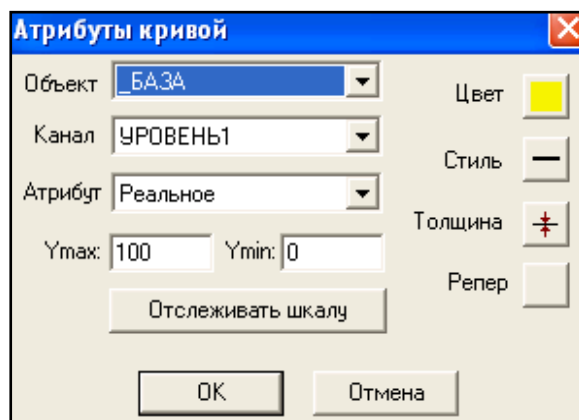


Рис.5.25.

Свяжем эту кривую с реальным значением канала **УРОВЕНЬ1**. Далее настроим диапазон вывода, цвет, стиль и толщину линии, как показано на рисунке. Подтвердим завершение настройки кривой нажатием ЛК на кнопке **ОК**. Повторим операции по настройке кривой. На этот раз свяжем ее с каналом **ДАВЛЕНИЕ1**.

Последней формой отображения, которую мы разместим на экране, будет кнопка, реализующая переход на экран **Участок 2**. Чтобы создать и настроить эту форму отображения, нажмем ЛК на соответствующей иконке инструментальной панели форм отображения, показанной на рисунке 5.26.

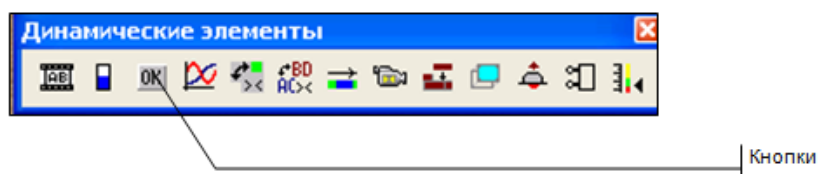


Рис. 5.26.

Окно **Атрибуты** для этой формы отображения показано на рисунке 5.27.

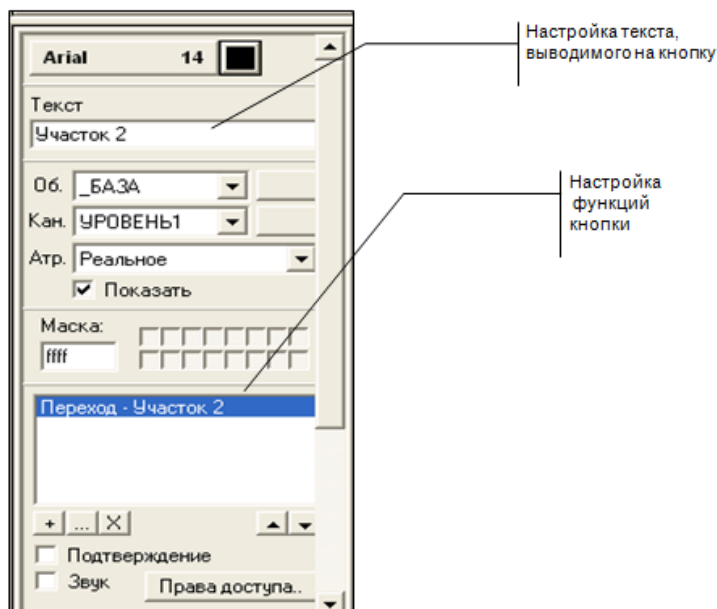


Рис. 1.27.

На этом разработка экрана **Участок 1** закончена.

Второй технологический участок идентичен первому. Информация на экране **Участок 2** будет представляться в том же виде, что и на экране **Участок 1**. Поэтому скопируем рисунок с первого экрана на второй и затем свяжем формы отображения с другими каналами.

На этом разработка графического интерфейса для данной задачи закончена. Для проверки функционирования размещенных на экранах форм отображения следует перейти в режим эмуляции. Переход в режим эмуляции реализуется нажатием ЛК на иконке системной инструментальной панели, показанной на рисунке 5.28.



Рис. 5.28.

Повторное нажатие этой кнопки позволяет остановить пересчет базы каналов. В окне тренда будут выведены кривые графиков давления и уровня (рис. 5.29).

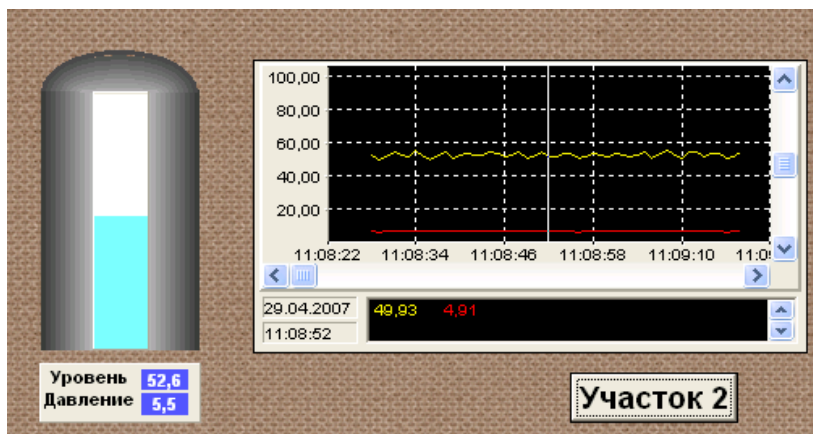


Рис. 5.29.

Контрольные задания

Задание 5.1. Разработать функциональный блок, выполняющий следующую функцию: если значение входит в заданный диапазон, то на выходе блока формируется 1, если не входит, то на выходе формируется 0.

Задание 5.2. При вводе номера, загорается соответствующий индикатор (рис. 5.30). Разработать проект автоматизации, включающий в себя информационную структуру, математическую базу и графический интерфейс.

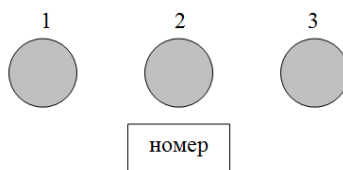


Рис. 5.30.

Задание 5.3. Вводится имя и пароль пользователя (рис. 5.31). При правильном вводе загорается зеленый цвет индикатора, при неправильном – красный. Разработать проект автоматизации, включающий в себя информационную структуру, математическую базу и графический интерфейс.



Рис. 5.31.

6. ПРИЕМЫ РАБОТЫ С TRACE MODE 5

Инструментальную систему TRACE MODE можно использовать для разработки виртуальных лабораторий.

Рассмотрим процесс разработки модели химического опыта «Нахождение зависимости оптической плотности раствора от концентрации и определение концентрации раствора по его оптической плотности с помощью уравнения калибровочного графика» приняв некоторые допущения в параметрах. Данный химический опыт можно представить в виде объекта, имеющего определенный набор входных и выходных величин (каналов объекта) (рис. 6.1). На основе этих данных в редакторе базы каналов создается информационная структура опыта. Далее разрабатывается математическая модель опыта, в виде FBD-программы (рис. 6.2).

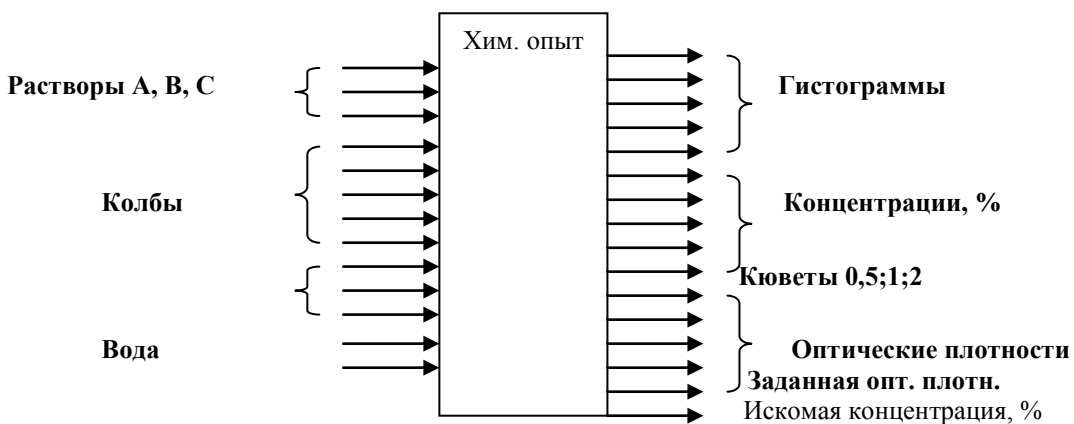


Рис. 6.1. Информационная структура опыта.

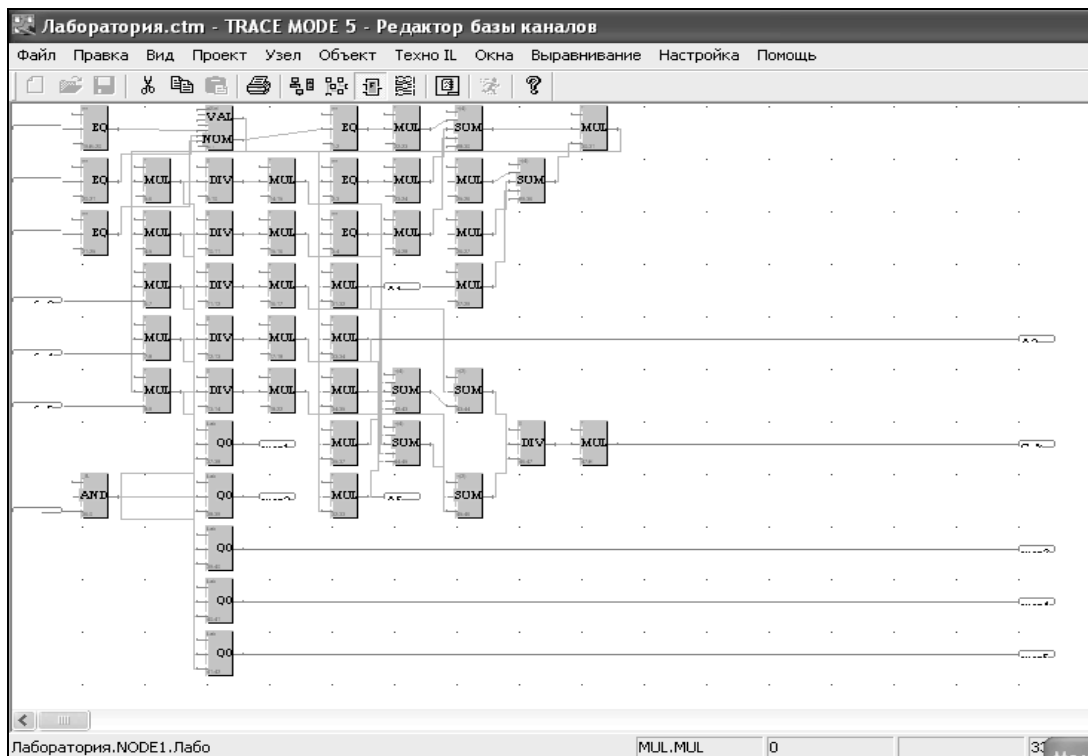


Рис.6.2. Математическая база опыта.

Следующий этап - разработка графического интерфейса лабораторной работы. Опыт разбивается на последовательность шагов, каждому из которых соответствует определенный экран. Таким образом, загружая последовательно экран за экраном и выполняя требуемые на них действия, осуществляется выполнение виртуального химического опыта (рис. 6.3 а, б, в, г, д . е).

Лабораторная работа №1.
Нахождение зависимости оптической плотности раствора от концентрации и определение концентрации раствора по его оптической плотности с помощью уравнения калибровочного графика.

1. Выберите один из трех исходных растворов. Из него приготовьте пять растворов с концентрацией в 50; 20; 10; 6,25 и 5 раз меньше исходного раствора. Для этого в пять мерных колб на 50 мл налейте исходного раствора: в первую - 1 мл, во вторую - 2,5; в третью - 5; в четвертую - 8; в пятую - 10 мл. Долейте колбы водой до метки.

2. На фотоколориметре измерьте оптическую плотность приготовленных растворов, начиная с раствора меньшей концентрации.

3. Зафиксируйте зависимость оптической плотности раствора от его концентрации.

4. Возьмите раствор неизвестной концентрации. На фотоколориметре измерьте оптическую плотность и с помощью уравнения калибровочного графика определите его концентрацию.

Структурная схема колориметра

1 — 2 — 3 — 4 — 5

Свет от лампы 1, проходит через систему линз 2, кювету 3 с исследуемым раствором, далее световой поток попадает на фотозлемент 4. Регистрирующим устройством служит микроамперметр 5, оцифрованный в делениях пропускания и оптической плотности.

Далее

Рис. 6.3. а.

Выбираем исходный раствор соли:

Раствор А

Раствор В

Раствор С

Далее

Рис. 2.3. б.

Готовим пять растворов с разной концентрацией:

Колба 1 Колба 2 Колба 3 Колба 4 Колба 5

вода

Далее

Рис. 2.3. в.

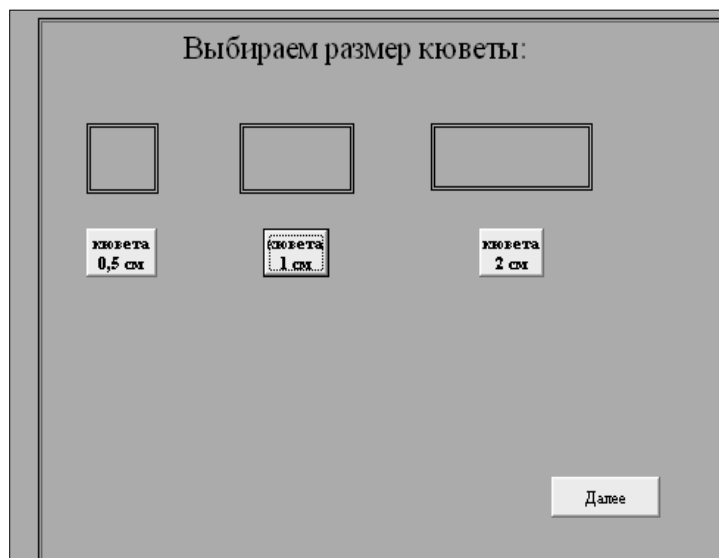


Рис. 2.3. г.

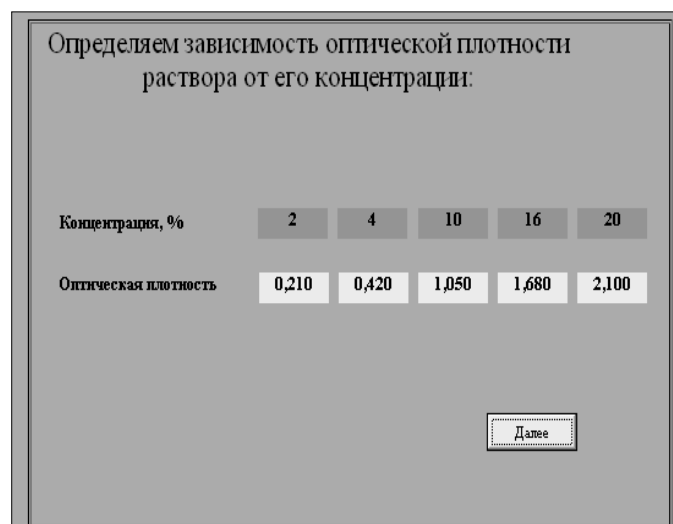


Рис. 2.3. д.

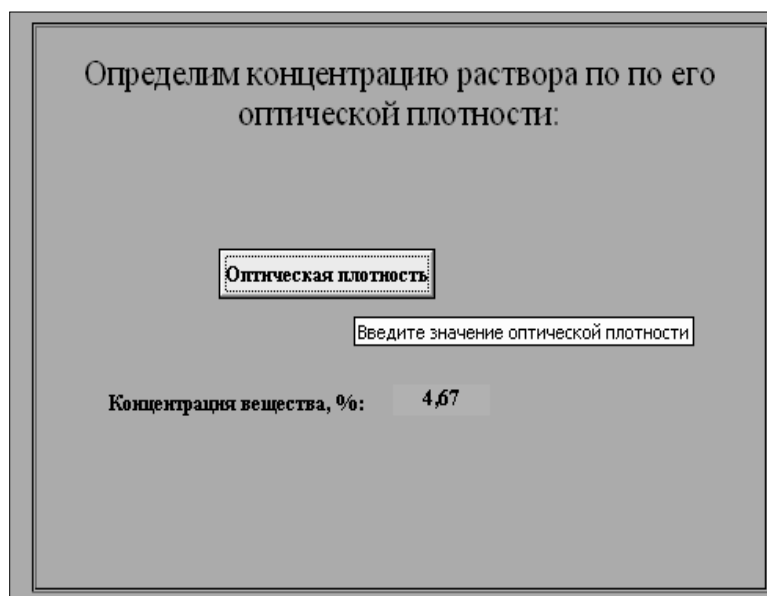


Рис. 2.3. е.

СПИСОК ЛИТЕРАТУРЫ

1. Гагарина Л.Г., Киселев Д.В., Федотова Е.Л. Разработка и эксплуатация автоматизированных информационных систем: учеб. пособие – М.: ИД «Форум»: ИНФРА-М, 2007
2. Емельянова Н.З., Партыка Т.Л., Попов И.И. Основы построения автоматизированных информационных систем: Учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2007
3. Гвоздева В. А., Лаврентьева И. Ю. Основы построения автоматизированных информационных систем: учебник.- М.: ИД «ФОРУМ»: ИНФРА-М, 2007

**Татьяна Анатольевна Изосимова
Инна Витальевна Ксенофонтова**

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

по дисциплине

«Аппаратно-программные комплексы»

для направления 09.03.01 «Информатика и вычислительная техника»
профиля

«Автоматизированные системы обработки информации и управления»

Часть 1

**TRACE MODE В РАЗРАБОТКЕ КОМБИНАЦИОННЫХ
МОДЕЛЕЙ**

Компьютерная верстка – Ксенофонтова И.В.

Подписано в печать . Формат 60x84/16.
Бумага писчая. Печать оперативная. Усл. печ.л.4 .
Тираж экз. Заказ № . Цена свободная

Федеральное государственное бюджетное образовательное учреждение
Высшего профессионального образования
«Московский автомобильно-дорожный
государственный технический университет (МАДИ)»
Волжский филиал