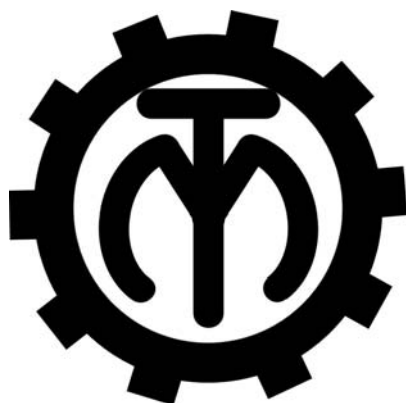


МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Технология машиностроения»

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ОБЛАСТИ АВТОМАТИЗАЦИИ И УПРАВЛЕНИЯ

*Методические рекомендации к практическим занятиям
для магистрантов направления подготовки
15.04.06 «Мехатроника и робототехника»
очной и заочной форм обучения*



Могилев 2023

УДК 658.012.011.56
ББК 30.2-5-05
К19

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Технология машиностроения» «22» ноября 2022 г.,
протокол № 6

Составитель канд. техн. наук, доц. М. Н. Миронова

Рецензент канд. техн. наук, доц. В. В. Кутузов

Методические рекомендации предназначены к практическим занятиям для магистрантов направления подготовки 15.04.06 «Мехатроника и робототехника» очной и заочной форм обучения по дисциплине «Компьютерные технологии в области автоматизации и управления». Изложены методики выполнения практических работ.

Учебно-методическое издание

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ОБЛАСТИ АВТОМАТИЗАЦИИ И УПРАВЛЕНИЯ

Ответственный за выпуск

В. М. Шеменков

Корректор

А. А. Подошевка

Компьютерная верстка

Е. В. Ковалевская

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 26 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2023

Содержание

Инструкция по охране труда при проведении практических занятий.....	4
1 Практическая работа № 1. Моделирование оптимального варианта раскроя материала	5
2 Практическая работа № 2. Оптимизация управления технологическим процессом	11
3 Практическая работа № 3. Изучение архитектурного построения SCADA-систем.....	15
4 Практическая работа № 4. Управление работой роботизированного производства с использованием SCADA-системы.....	29
Список литературы.....	37

Инструкция по охране труда при проведении практических занятий

Общие требования безопасности

1 Для работы на ПЭВМ в компьютерном классе допускаются студенты, прошедшие обучение и проверку знаний по мерам безопасности.

2 Студенты должны соблюдать правила внутреннего распорядка. Не допускается находиться в классах в верхней одежде, в состоянии алкогольного, токсического или наркотического опьянения.

3 При проведении практических работ необходимо соблюдать правила пожарной безопасности, знать места расположения первичных средств пожаротушения.

Требования безопасности перед началом работы

1 Внимательно изучить содержание и порядок проведения практической работы, а также безопасные приемы его выполнения.

2 В случае неисправности оборудования немедленно сообщить об этом преподавателю и до ее устранения к работе не приступать (работать на неисправном оборудовании запрещается).

Требования безопасности во время работы

Студенту при работе на ПЭВМ запрещается:

– прикасаться к задней стенке системного блока (процессора) при включенном питании;

– загромождать верхние панели устройств ненужными бумагами и посторонними предметами;

– допускать попадания влаги на поверхность системного блока, монитора, рабочую поверхность клавиатуры и другие устройства.

Требования безопасности по окончании работы

1 Произвести закрытие всех активных задач.

2 Отключить питание системного блока (процессора).

3 Осмотреть и привести в порядок рабочее место.

4 Предупредить преподавателя обо всех, даже малейших и незначительных, неисправностях оборудования.

1 Практическая работа № 1. Моделирование оптимального варианта раскроя материала

Цель работы: приобретение практических навыков линейного программирования симплексным методом.

Теоретические сведения

Симплекс-метод – алгоритм решения оптимизационной задачи линейного программирования путём перебора вершин выпуклого многогранника в многомерном пространстве.

Сущность метода: построение базисных решений, на которых монотонно убывает линейный функционал, до ситуации, когда выполняются необходимые условия локальной оптимальности.

Задача линейного программирования состоит в том, что необходимо максимизировать или минимизировать некоторый линейный функционал на многомерном пространстве при заданных линейных ограничениях.

Последовательность вычислений симплекс-методом можно разделить на две основные фазы:

- 1) нахождение исходной вершины множества допустимых решений;
- 2) последовательный переход от одной вершины к другой, ведущий к оптимизации значения целевой функции.

Симплексный метод – это метод последовательного улучшения плана. Этим методом можно решать задачи линейного программирования с любым количеством переменных и ограничений.

Симплексный метод включает три основных этапа:

- 1) построение начального опорного плана;
- 2) правило перехода к лучшему решению;
- 3) критерий проверки найденного решения на оптимальность.

При симплексном методе выполняются вычислительные процедуры (итерации) одного и того же типа в определенной последовательности до тех пор, пока не будет получен оптимальный план задачи или станет ясно, что его не существует.

1 *Построение начального опорного плана.* Задачу линейного программирования необходимо сначала привести к каноническому виду. При этом правые части ограничений должны быть неотрицательными.

Признаком возможности построения начального опорного плана служит наличие в каждом ограничении-равенстве с неотрицательной правой частью базисной переменной.

Базисной называют плановую переменную, которая входит только в одно уравнение (а в другие не входит), и при этом имеет коэффициент, равный единице.

2 Составление симплексных таблиц. Критерий оптимальности. Симплексный метод удобно применять, используя построение симплексных таблиц (таблица 1.1).

Таблица 1.1 – Первая симплексная таблица, соответствующая начальному плану

Базис	$c_{\bar{b}i}$	B	c_1	c_2	...	c_n	θ_i
			x_1	x_2	...	x_n	
$x_{\bar{b}1}$	$c_{\bar{b}1}$	b_1	a_{11}	a_{12}	...	a_{1n}	
$x_{\bar{b}2}$	$c_{\bar{b}2}$	b_2	a_{21}	a_{22}	...	a_{2n}	
...
$x_{\bar{b}m}$	$c_{\bar{b}m}$	b_m	a_{m1}	a_{m2}	...	a_{mn}	
$\Delta_j = Z_j - c_j$		Δ_0	Δ_1	Δ_2	...	Δ_n	

Здесь приняты следующие обозначения:

$x_{\bar{b}i}$ – базисные переменные;

$c_{\bar{b}i}$ – коэффициенты при базисных переменных в целевой функции;

b_i – правые части ограничений;

a_{ij} – коэффициенты при переменных в ограничениях;

c_j – коэффициенты при переменных в целевой функции.

Последняя строка в таблице ($\Delta_j = Z_j - c_j$) – это проверочная или оценочная строка.

Δ_0 – значение целевой функции, соответствующее построенному начальному плану. Его находят так: каждый элемент столбца $c_{\bar{b}i}$ умножают на соответствующий элемент столбца B и произведения складывают, т. е.

$$\Delta_0 = c_{\bar{b}1} b_1 + c_{\bar{b}2} b_2 + \dots + c_{\bar{b}m} b_m.$$

Δ_j – называют оценками или симплексными разностями и находят так: элементы столбца $c_{\bar{b}i}$ умножают на соответствующие элементы столбца x_j , складывают результаты и вычитают c_j .

Например,

$$\Delta_2 = c_{\bar{b}1} a_{12} + c_{\bar{b}2} a_{22} + \dots + c_{\bar{b}m} a_{m2} - c_2.$$

Оценки Δ_j базисных переменных всегда равны нулю.

Признак оптимальности опорного плана состоит в следующем: опорный план будет оптимальным тогда и только тогда, когда все оценки $\Delta_j \geq 0$ для задачи на max и $\Delta_j \leq 0$ для задачи на min.

Если критерии оптимальности не выполняются, то нужно перейти к нехудшему опорному плану, т. е. исключить из базиса некоторую переменную и ввести вместо нее новую из числа свободных переменных.

Переменная, которую нужно ввести в базис, соответствует той оценке Δ_j , которая не удовлетворяет условию оптимальности. Если таких оценок

несколько, то среди них выбирают наибольшую по абсолютной величине и соответствующую ей переменную вводят в базис. Столбец с этой переменной называют разрешающим.

Для определения переменной, которую нужно вывести из базиса, поступают так: элементы столбца B делят только на положительные элементы разрешающего столбца и находят симплексные отношения $\theta_i = \frac{b_i}{a_{ik}}$. Выбирают

из θ_i наименьшее. Оно называет ту переменную, которую нужно ввести в базис. Соответствующая строка таблицы называется разрешающей.

На пересечении разрешающего столбца и разрешающей строки находится разрешающий элемент.

При переходе от таблицы к таблице для контроля сравнивают Δ_0 , которое должно быть не меньше предыдущего для задачи на максимум и не больше предыдущего – для задачи на минимум.

При использовании симплексного метода возможны следующие случаи:

1) если в оценочной строке симплекс-таблицы оценка $\Delta_j = 0$ соответствует свободной переменной, то это означает, что задача линейного программирования имеет не единственный оптимальный план;

2) если при переходе от одного опорного плана к другому в разрешающем столбце нет положительных элементов, то это означает, что целевая функция задачи линейного программирования является неограниченной и оптимальных планов не существует.

Пример – Решить симплексным методом задачу линейного программирования.

$$F = 2x_1 + 3x_2 \rightarrow \max .$$

$$\begin{cases} x_1 + 3x_2 + x_3 = 18; \\ 2x_1 + x_2 + x_4 = 16; \\ x_2 + x_5 = 5; \\ 3x_1 + x_6 = 21. \end{cases}$$

$$x_j \geq 0; j = 1...6.$$

1 Определяем базисные переменные – это x_3, x_4, x_5, x_6 .

2 Заполняем первую таблицу (таблица 1.2).

Таблица 1.2 – Первая таблица решения задачи

Базис	$c_{\delta i}$	B	2	3	0	0	0	0	θ_i
			x_1	x_2	x_3	x_4	x_5	x_6	
x_3	0	18	1	3	1	0	0	0	$18 / 3 = 6$
x_4	0	16	2	1	0	1	0	0	$16 / 1 = 16$
x_5	0	5	0	1	0	0	1	0	$5 / 1 = 5$
x_6	0	21	3	0	0	0	0	1	–
$\Delta_j = Z_j - c_j$	$\Delta_0 = 0$	$\Delta_1 = -2$	$\Delta_2 = -3$	$\Delta_3 = 0$	$\Delta_4 = 0$	$\Delta_5 = 0$	$\Delta_6 = 0$		

Здесь Δ_1 и Δ_2 не удовлетворяют условию оптимальности, т. к. они меньше нуля. Выбираем среди них большее по модулю. Это Δ_2 . Следовательно, столбец переменной x_2 – разрешающий. Значит, в новый базис нужно ввести переменную x_2 .

Находим θ_i :

$$\theta_1 = \frac{b_1}{a_{12}} = \frac{18}{3} = 6; \theta_2 = \frac{b_2}{a_{22}} = \frac{16}{1} = 16; \theta_3 = \frac{b_3}{a_{32}} = \frac{5}{1} = 5.$$

Наименьшее из этих чисел – это число 5, что соответствует строке базисной переменной x_5 . Значит, строка базисной переменной x_5 – разрешающая, следовательно, из базиса нужно вывести переменную x_5 . Элемент $a_{32} = 1$ – разрешающий. Новый базис: x_3, x_4, x_2, x_6 .

Заполнение следующей таблицы начинаем со столбцов «Базис» и « $c_{\delta i}$ ». Потом заполняем разрешающую строку, разделив каждый ее элемент на разрешающий, т. е. на 1. Все элементы разрешающего столбца будут нулями, кроме разрешающего, который всегда равен 1. Столбцы под x_3, x_4, x_6 переписываем без изменения, т. к. эти переменные остались в базисе. Остальные элементы новой таблицы находим по правилу прямоугольника. Например,

$$b_2'' = \frac{16 \cdot 1 - 5 \cdot 1}{1} = 11;$$

$$a_{15} = \frac{1 \cdot 0 - 3 \cdot 1}{1} = -3.$$

Оценки Δ_j для новой таблицы можно находить по этому же правилу.

В целом, решение данной задачи симплексным методом можно представить в виде таблиц (таблица 1.3).

Таблица 1.3 – Таблицы решений задачи симплексным методом

Базис	c_{bi}	B	2	3	0	0	0	0	θ_i
			x_1	x_2	x_3	x_4	x_5	x_6	
x_3	0	18	1	3	1	0	0	0	6
x_4	0	16	2	1	0	1	0	0	16
x_5	0	5	0	1	0	0	1	0	5
x_6	0	21	3	0	0	0	0	1	-
$\Delta_j = Z_j - c_j$	0	0	-2	-3	0	0	0	0	
x_3	0	3	1	0	1	0	-3	0	3
x_4	0	11	2	0	0	1	-1	0	5,5
x_2	3	5	0	1	0	0	1	0	-
x_6	0	21	3	0	0	0	0	1	7
$\Delta_j = Z_j - c_j$	15	15	-2	0	0	0	3	0	
x_1	2	3	1	0	1	0	-3	0	-
x_4	0	5	0	0	-2	1	5	0	1
x_2	3	5	0	1	0	0	1	0	5
x_6	0	12	0	0	-3	0	9	1	12 / 9
$\Delta_j = Z_j - c_j$	21	21	0	0	2	0	-3	0	
x_1	2	6	1	0	-0,2	0,6	0	0	
x_5	0	1	0	0	-0,4	0,2	1	0	
x_2	3	4	0	1	0,4	-0,2	0	0	
x_6	0	3	0	0	0,6	-1,8	0	1	
$\Delta_j = Z_j - c_j$	24	24	0	0	0,8	0,6	0	0	

Оценочная строка таблицы показывает, что получен оптимальный план, т. к. все $\Delta_j \geq 0$.

X_{opt} – это значения b_i из столбца B , т. е.

$$x_1^* = 6, x_5^* = 1, x_2^* = 4, x_6^* = 3.$$

Свободные (небазисные) переменные $x_3^* = x_4^* = 0$.

Тогда

$$X_{opt} = (6; 4; 0; 0; 1; 3), Z_{max} = \Delta_0 = 24.$$

Задание

По заранее определенной программе выпуска требуется изготовить заготовки двух типоразмеров. Для раскроя стандартных заготовок материала на заготовки этих деталей предложено четыре варианта. Определить оптимальный вариант раскроя материала, при котором будет обеспечен выпуск требуемого количества заготовок, а расход стандартных заготовок материала на их выпуск будет минимальным.

Варианты заданий приведены в таблицах 1.4 и 1.5 и выбираются по номеру студента в списке учебной группы.

Таблица 1.4 – Варианты заданий для определения оптимального раскроя материала

Первая цифра списка	Номер детали	Программа выпуска деталей
0	1	360
	2	720
1	1	480
	2	780
2	1	320
	2	640

Таблица 1.5 – Варианты заданий для определения оптимального раскроя материала

Вторая цифра списка	Номер детали	Вариант раскроя материала			
		В-1	В-2	В-3	В-4
0	1	10	8	4	2
	2	0	4	12	18
1	1	6	8	4	5
	2	3	5	10	12
2	1	8	6	5	10
	2	2	4	10	8
3	1	4	2	6	5
	2	0	6	8	12
4	1	0	5	8	6
	2	10	4	2	3
5	1	0	6	8	10
	2	4	8	5	4
6	1	4	0	6	9
	2	6	5	7	12
7	1	3	5	0	2
	2	10	8	6	5
8	1	8	6	10	4
	2	6	12	2	5
9	1	4	0	9	5
	2	3	5	8	4

Порядок выполнения работы

1 Сформировать целевую функцию для решения задачи. Определить систему ограничений. Построить модель линейного программирования для решения задачи.

2 Найти решение задачи симплекс-методом.

3 Определить какое количество стандартных заготовок материала должно раскраиваться по каждому из вариантов, какие варианты не используются.

4 Показать, что данное решение является оптимальным относительно сформированной целевой функции.

5 Определить суммарный расход стандартных заготовок материала для полученного решения.

Контрольные вопросы

- 1 Целевая функция для решения задачи оптимального варианта раскрытия материала.
- 2 Назовите систему ограничений.
- 3 В чем заключается симплекс-метод?

2 Практическая работа № 2. Оптимизация управления технологическим процессом

Цель работы: приобретение практических навыков параметрической оптимизации переходов технологических операций механической обработки деталей машин.

Задание

Выполнить параметрическую оптимизацию технологических переходов продольного точения.

Порядок выполнения работы

1 Скопировать из папки «ПО_Пр2» в свою рабочую папку файл «Параметрическая оптимизация технологических операций.xls» и открыть его в своей рабочей папке.

2 Заполнить область исходных данных таблицы на основе данных, представленных на фрагменте операционной карты (рисунок 2.1). Пример форм для ввода исходных данных и для задания ограничений на область поиска экстремума целевой функции представлены на рисунках 2.2 и 2.3 соответственно.

3 Заполнить область нормативных режимов резания на основе данных, представленных в фрагменте операционной карты (см. рисунок 2.1). Пример формы для ввода данных, которые были рассчитаны по нормативам и служат для оценки эффективности оптимизации показан на рисунке 2.4.

№	Имя	Фамилия	Год	№	Лист	ИФ	Подпись	Дата		
				Белорусско-Российский университет		БРУ 60 14200003				
				Вал				020		
Наименование операции		Материал		Твердость	ВВ	МД	Профиль и размеры			
Токарная с ЧПУ		Сталь 18ХГТ ГОСТ 4543-71		170...217 НВ	К2	11,2	№95х266 14,6 1			
Обозначение устройства ЧПУ		Обозначение программы		Т _а	Т _б	Т _с	С _ж			
16ГС25Ф3				6,2	1,36	28	Эмульсия			
В		ПМ	В шц.В	L	f	i	S	n	V	
0 01	Установить А. Установить и закрепить заготовку									
T 02	Патрон 7100-0009 ГОСТ 2675-80; Центр А-1-5-У ГОСТ 4742-75									
0 03	1 Точить поверхности 1, 2, 3, 4, 5 предварительна									
T 04	Резец 2103-0711 ГОСТ 20872-80, 25×25, Т15К6; Штангенциркуль ШЦЦ-И-250-0,01 ГОСТ 166-89									
P 05				906	233	2,2	1	0,6	353	1105
P 06				75	206	2,605	3	0,6	415	97,7
P 07				65	27	2,5	2	0,6	482	98,4
P 08				66	178	2,25	2	0,6	482	99,9
P 09				60	149	3	1	0,6	508	95,7
0 10	2 Точить поверхности 1, 4 окончательно и фаски									
T 11	Резец 2103-0711 ГОСТ 20872-80, 25×25, Т15К6; Штангенциркуль ШЦЦ-И-250-0,01 ГОСТ 166-89; Микронметр 100-1 ГОСТ 6507-75									
T 12				896	31	0,5	1	0,2	616	173,6
P 13				65	33	0,5	1	0,2	850	173,6
OK										

Рисунок 2.1 – Фрагмент операционной карты, содержащий сведения о переходах, которые подлежат оптимизации


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Параметрическая оптимизация токарной операции														
2	Регулирование частоты вращения шпинделя бесступенчатое														
3	Регулирование подачи суппорта бесступенчатое														
4															
5	Иванов И.И.			гр.ТМ-091			Дата: 05.05.2014								
6															
7	Исходные данные												 Пуск		
8															
9	Номер операции		015 Токарная с ЧПУ Установы А и Б												
10	Характер обработки		Черновая , продольное точение												
11	Обрабатываемый материал		Сталь 40Х												
12	Вид заготовки		Прокат	Диаметр	100	Длина	600	Закрепление В патроне							
13	Количество переходов		12												
14	Номер перехода		2	4	6	8	9	11	14	17	20	23	26	29	
15	Диаметр поверхности, мм		41	51	61	71	81	91	41	50	60	70	80	90	
16	Длина поверхности, мм		100	100	100	100	100	100	100	100	100	100	100	100	
17	Глубина резания, мм		3	3	3	3	3	3	0,5	0,5	0,5	0,5	0,5	0,5	
18	Подача (нормативная), мм/об		0,5	0,6	0,7	0,8	0,9	0,9	0,2	0,15	0,15	0,15	0,15	0,15	
19	Врезание и перебег, мм.		2	2	2	2	2	2	0,5	0,5	0,5	0,5	0,5	0,5	
20	Шероховатость Ra, мкм		12	12	12	12	12	12	2,5	2,5	2,5	2,5	2,5	2,5	

Рисунок 2.2 – Пример формы для ввода исходных данных

22	Ограничения на область поиска экстремума целевой функции	
23		
24	1 Ограничения по техническим характеристикам станка	
25	Допустимое усилие в направлении подачи инструмента	8000 Н
26	Ограничение по мощности привода шпинделя станка	22 кВт
27	2 Ограничение по техническим характеристикам станочного приспособления	
28	Предельно допустимая частота вращения токарного патрона	3500 об/мин
29	Предельный допустимый передаваемый крутящий момент	1000 Н*м
30	3 Ограничение по прочности режущей пластины	
31	Максимальная допустимая глубина резания	4 мм
32	Максимальная допустимая подача	2,5 мм/об
33	4 Ограничения по условиям обеспечения устойчивого стружкодробления	
34	Верхняя граница по глубине резания	4 мм
35	Нижняя граница по глубине резания	0,1 мм
36	Нижняя граница по подаче	0,1 мм/об
37	Верхняя граница по подаче	1,5 мм/об
38	5 Ограничения по величине допустимых упругих деформаций заготовки	
39	Допустимая деформация в направлении перпендикулярном подаче	0,010 мм
40	6 Ограничения по допустимому диапазону скоростей резания	
41	Нижняя граница	50 м/мин
42	Верхняя граница	350 м/мин

Рисунок 2.3 – Пример формы для задания ограничений на область поиска экстремума целевой функции

Нормативные режимы резания												
Скорость резания, м/мин	161	156	153	150	147	147	252	252	252	252	252	252
Частота вращения, об/мин	1247	976	797	671	578	514	1955	1603	1336	1145	1002	890
Подача, мм/об	0,5	0,6	0,7	0,8	0,9	0,9	0,15	0,15	0,15	0,15	0,15	0,15
Подача, мм/мин	624	585	558	537	520	463	293	240	200	172	150	134
Период стойкости, мин	30	30	30	30	30	30	30	30	30	30	30	30
Шероховатость Ra, мкм	5,35	6,31	7,25	8,18	9,10	9,10	1,64	1,64	1,64	1,64	1,64	1,64
Сила резания Pz, Н.	2933	3376	3803	4216	4618	4618	185	185	185	185	185	185
Мощность, кВт	7,7	8,6	9,5	10,3	11,1	11,1	0,8	0,8	0,8	0,8	0,8	0,8
Основное время, мин	0,16	0,17	0,18	0,19	0,19	0,22	0,34	0,42	0,50	0,58	0,67	0,75
Себестоимость перехода, р.	258	277	292	306	317	356	522	637	764	892	1019	1146

Рисунок 2.4 – Пример формы для ввода данных, которые были рассчитаны по нормативам и служат для оценки эффективности оптимизации

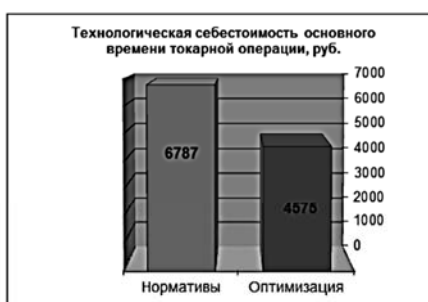
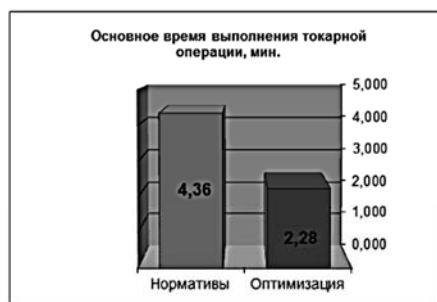
4 На листах «БД» и «Цены» ввести сведения о станке, режущем инструменте, коэффициентах и показателях степеней, которые используются в уравнениях математической модели процедуры параметрической оптимизации так, как это показано на рисунке 2.5.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R		
2	База данных нормативов для расчета режима продольного точения																			
4	Справочник нормативных данных для расчета скорости резания V																			
5																				
6	Номер записи	Обрабатываемый материал	Предел прочности стали, МПа или HB для чугунов и цветных сплавов	Инструментальный материал	Рекомендуемый нормативный период стойкости инструмента T, мин	Коэффициент K _T	Показатель степени n _T	Коэффициент K _{Tn}	Коэффициент K _{KT}	Коэффициент S _{Таб}	Показатель степени n _у	Показатель степени m _у	Коэффициент K _{Фy}	Коэффициент K _{Фm}	Коэффициент K _{Фс}	Коэффициент, учитывающий использование СОЖ K _с	Коэффициент, учитывающий наличие изношенных покрывной режущей пластины K _{из}			
7	1	Сталь 40Х	980	T15K6	30	0,95	1,00	0,9	1,00	350	0,15	0,15	0,35	0,7	1	1	1,4	2,5		
8	Коэффициент	K _{MV} = 0,727				Общий поправочный коэффициент на скорость резания										K _с = 1,60				
9						Новое значение скоростного коэффициента C _V = C _{V,таб} * K _V										C _V = 561,09				

Рисунок 2.5 – Образец таблицы базы данных

5 Для запуска программы оптимизации выполнить двойной щелчок по кнопке «Пуск», расположенной справа от области ввода исходных данных (см. рисунок 2.2). На основе полученных результатов оптимизации строятся гистограммы, которые наглядно демонстрируют эффективность параметрической оптимизации технологической операции (рисунок 2.6).

S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG
Результаты параметрической оптимизации токарной операции														
Оптимальные режимы резания														
Скорость резания, м/мин	171	171	172	172	170	171	312	311	311	312	311	311	311	311
Частота вращения, об/мин	1330	1070	900	770	670	600	2420	1980	1650	1420	1240	1100		
Подача, мм/об	0,89	0,89	0,89	0,89	0,89	0,89	0,27	0,27	0,27	0,27	0,27	0,27	0,27	0,27
Подача, мм/мин	1184	952	801	685	596	534	653	535	446	383	335	297		
Период стойкости, мин	19,5	19,5	19,1	19,4	19,8	19,4	12,7	12,7	12,7	12,6	12,7	12,7		
Шероховатость, Ra мкм	8,54	8,53	8,52	8,53	8,55	8,53	2,50	2,50	2,50	2,50	2,50	2,50		
Мощность, кВт	12,5	12,5	12,6	12,5	12,5	12,5	1,4	1,4	1,4	1,4	1,4	1,4		
Сила резания P _z , Н	4418	4417	4413	4416	4421	4417	275	275	275	275	275	275		
Основное время, мин	0,08	0,11	0,12	0,15	0,17	0,19	0,15	0,19	0,22	0,26	0,30	0,34		
Себестоимость перехода, р.	159	198	236	275	314	353	319	389	466	545	622	700		

**Показатели эффективности параметрической оптимизации токарной операции**

Общее сокращение основного времени токарной операции	1000 шт	91,45 %
Сокращение основного времени при годовом объеме выпуска		34,69 час
Снижение технологической себестоимости основного времени		48,35 %
Снижение себестоимости основного времени на годовом объеме выпуска		2212191 руб

Рисунок 2.6 – Образец формы для оценки эффективности параметрической оптимизации технологической операции

6 После завершения программы параметрической оптимизации произвести анализ полученных результатов и составить отчет по работе.

Содержание отчета

- 1 Цель работы.
- 2 Программа параметрической оптимизации.
- 3 Результаты оптимизации переходов.
- 4 Выводы.

Контрольные вопросы

- 1 Что представляет собой функциональная математическая модель процесса обработки, используемая в задачах параметрической оптимизации?
- 2 Какие параметры технологического перехода могут рассматриваться в качестве входных и выходных параметров модели?
- 3 В чем заключается параметрическая оптимизация технологических процессов?
- 4 Какие параметры технологического процесса могут быть выбраны в качестве критерия оптимизации?

3 Практическая работа № 3. Изучение архитектурного построения SCADA-систем

Цель работы: изучение архитектурного построения SCADA-систем.

Теоретические сведения

На начальном этапе развития (1980-е гг.) каждый производитель микропроцессорных систем управления разрабатывал свою собственную SCADA-программу. Такие программы могли взаимодействовать только с узким кругом контроллеров, и по всем параметрам были закрытыми (отсутствие набора драйверов для работы с устройствами различных производителей и средств их создания, отсутствие стандартных механизмов взаимодействия с другими программными продуктами и т. д.).

С появлением концепции открытых систем (начало 1990-х гг.) программные средства для операторских станций становятся самостоятельным продуктом.

Одной из первых задач, поставленных перед разработчиками SCADA, стала задача организации многопользовательских систем управления, т. е. систем, способных поддерживать достаточно большое количество АРМ пользователей (клиентов). В результате появилась клиент-серверная технология или архитектура.

Клиент-серверная архитектура характеризуется наличием двух взаимодействующих самостоятельных процессов – клиента и сервера, которые, в общем случае, могут выполняться на разных компьютерах, обмениваясь данными по сети. По такой схеме могут быть построены системы управления технологическими процессами, системы обработки данных на основе СУБД и т. п.

Клиент-серверная архитектура (рисунок 3.1) предполагает, что вся информация о технологическом процессе от контроллеров собирается и обрабатывается на сервере ввода/вывода (сервер базы данных), к которому по сети подключаются АРМ клиентов.

Под станцией-сервером в этой архитектуре следует понимать компьютер со специальным программным обеспечением для сбора и хранения данных и последующей их передачи по каналам связи оперативному персоналу для контроля и управления технологическим процессом, а также всем заинтересованным специалистам и руководителям.

По определению сервер является поставщиком информации, а клиент – ее потребителем. Таким образом, рабочие станции операторов/диспетчеров, специалистов, руководителей являются станциями-клиентами. Обычно клиентом служит настольный ПК, выполняющий программное обеспечение конечного пользователя.



Рисунок 3.1 – Клиент-серверная архитектура

Программное обеспечение клиента – это любая прикладная программа или пакет, способные направлять запросы по сети серверу и обрабатывать получаемую в ответ информацию. Естественно, функции клиентских станций, а следовательно, и программное обеспечение, различны и определяются функциями рабочего места, которое они обеспечивают.

Количество операторских станций, серверов ввода/вывода (серверов БД) определяется на стадии проектирования и зависит, прежде всего, от объема перерабатываемой в системе информации. Для небольших систем управления функции сервера ввода/вывода и станции оператора (HMI) могут быть совмещены на одном компьютере.

В сетевых распределенных системах средствами SCADA/HMI стало возможным создавать станции (узлы) различного функционального назначения: станции операторов/диспетчеров, серверы с функциями HMI, «слепые» серверы (без функций HMI), станции мониторинга (только просмотр без прав на управление) для специалистов и руководителей и другие.

SCADA-программы имеют в своем составе два взаимозависимых модуля: *Development* (среда разработки проекта) и *Runtime* (среда исполнения). В целях снижения стоимости проекта эти модули могут устанавливаться на разные компьютеры. Например, станции оператора, как правило, являются узлами *Runtime* (или *View*) с полным набором функций человеко-машинного интерфейса. При этом хотя бы один компьютер в сети должен быть типа *Development*. На таких узлах проект разрабатывается, корректируется, а также может и исполняться. Некоторые SCADA-системы допускают внесение изменений в проект без остановки работы всей системы. Программное обеспечение SCADA-серверов позволяет создавать полный проект системы управления, включая базу данных и HMI.

Важным аспектом в структурном построении сетевых систем управления является *структура базы данных реального времени* (централизованная или распределенная). Каждая из структур в SCADA/HMI-системах реализуется разными разработчиками по-разному. От реализации существенно зависят эффективность обеспечения единства и целостности базы данных, ее надежность, возможности модификации и т. д.

В одних случаях для доступа к данным на компьютере-клиенте создается своя база данных, копируемая с удаленных серверов. Дублирование данных может привести к определенным проблемам с точки зрения целостности базы данных и производительности системы управления. При модификации базы данных с такой организацией, например, при введении дополнительной переменной потребуются изменения в каждой сетевой копии, использующей эту переменную.

В других случаях компьютерам-клиентам не требуются копии баз данных. Они получают необходимую им информацию по сети от сервера, в задачу

которого входит поддержание базы данных. Серверов может быть несколько, и любая часть данных хранится только в одном месте, на одном сервере. Поэтому и модификация базы данных производится только на одном компьютере – сервере базы данных, что обеспечивает ее единство и целостность. Такой подход к структурному построению системы снижает нагрузку на сеть и дает еще целый ряд преимуществ.

С точки зрения структурного построения SCADA-пакетов различают:

- системы, обеспечивающие полный набор базовых функций HMI;
- системы, состоящие из модулей, реализующих отдельные функции HMI.

Системы, обеспечивающие полный набор базовых функций, могут комплектоваться дополнительными опциями, реализующими необязательные в применении функции контроля и управления.

Во втором случае система создается полностью модульной (сервер ввода/вывода, сервер алармов, сервер трендов и т. д.).

Для небольших проектов все модули могут исполняться на одном компьютере. В проектах с большим количеством переменных модули можно распределить на несколько компьютеров в разных сочетаниях. Вариант клиент-серверной архитектуры такой системы представлен на рисунке 3.2.

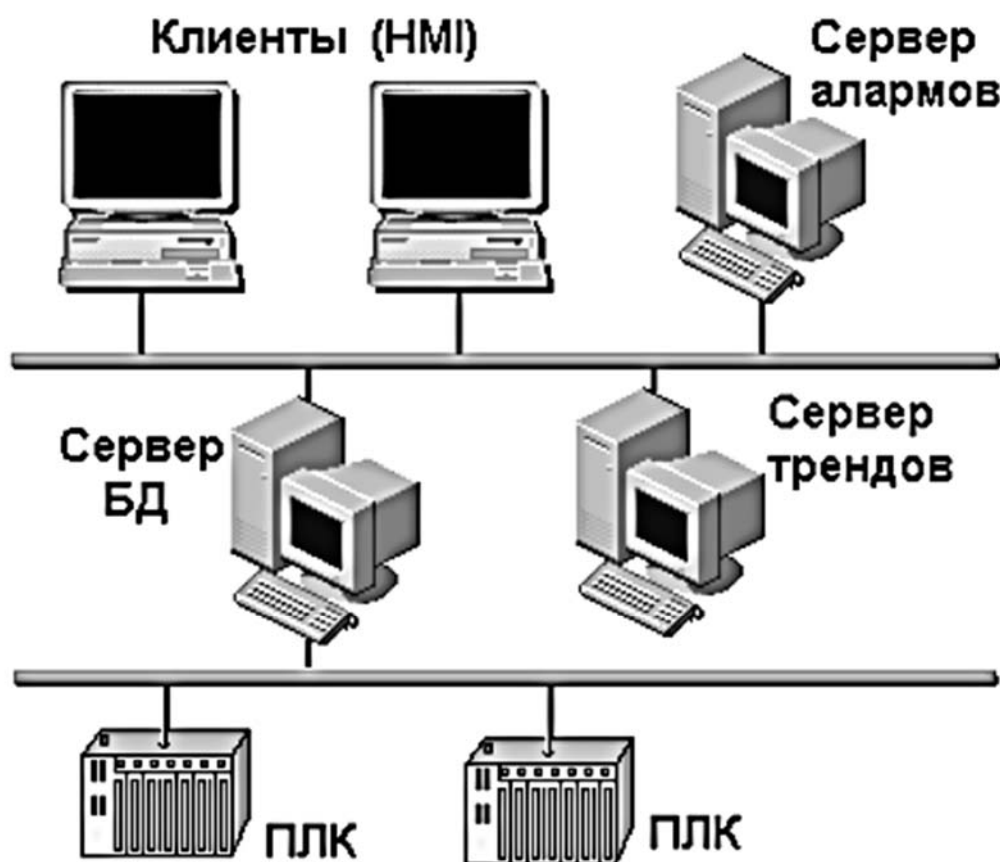


Рисунок 3.2 – Архитектура модульной SCADA

В клиент-серверной архитектуре системы управления, представленной на рисунке 3.2, функции сбора и хранения данных, управления алармами и трендами распределены между тремя серверами. Функция HMI реализуется на станциях-клиентах.

Например, SCADA Citect имеет в своем составе пять функциональных модулей (серверов или клиентов).

I/O – сервер ввода/вывода. Обеспечивает передачу данных между физическими устройствами ввода/вывода и другими модулями Citect.

Display – клиент визуализации. Обеспечивает операторский интерфейс: отображение данных, поступающих от других модулей Citect, и управление выполнением команд оператора.

Alarms – сервер алармов. Отслеживает данные, сравнивает их с допустимыми пределами, проверяет выполнение заданных условий и отображает алармы на соответствующем узле визуализации.

Trends – сервер трендов. Собирает и регистрирует трендовую информацию, позволяя отображать развитие процесса в реальном масштабе времени или в ретроспективе.

Reports – сервер отчетов. Генерирует отчеты по истечению определенного времени, при возникновении определенного события или по запросу оператора.

В одной сети можно использовать только один сервер алармов, сервер трендов и сервер отчетов. В то же время допускается использование нескольких серверов ввода/вывода (I/O Server). Количество компьютеров с установленным модулем *Display* (обеспечивающим операторский интерфейс) в сети практически не ограничено.

SCADA как открытая система. Распространение архитектуры «клиент-сервер» стало возможным благодаря развитию и широкому внедрению в практику концепции открытых систем. Главной причиной появления и развития концепции открытых систем явились проблемы взаимодействия программно-аппаратных средств в локальных компьютерных сетях. Решить эти проблемы можно было только путем международной стандартизации программных и аппаратных интерфейсов.

Концепция открытых систем предполагает свободное взаимодействие программных средств SCADA с программно-техническими средствами разных производителей. Это актуально, т. к. для современных систем автоматизации характерна высокая степень интеграции большого количества компонент. В системе автоматизации, кроме объекта управления, задействован целый комплекс программно-аппаратных средств: датчики и исполнительные устройства, контроллеры, серверы баз данных, рабочие места операторов, АРМы специалистов и руководителей и т. д. (рисунок 3.3). При этом в одной системе могут быть применены технические средства разных производителей.

Очевидно, что для эффективного функционирования в этой разнородной среде SCADA-система должна обеспечивать высокий уровень сетевого взаимодействия.

Реализация этой задачи требует от SCADA-системы наличия типовых протоколов обмена с наиболее популярными промышленными сетями, такими как Profibus, ControlNet, Modbus и другими.

С другой стороны, SCADA-системы должны поддерживать интерфейс и со стандартными информационными сетями (Ethernet и др.) с использованием стандартных протоколов (TCP/IP и др.) для обмена данными с компонентами распределенной системы управления.

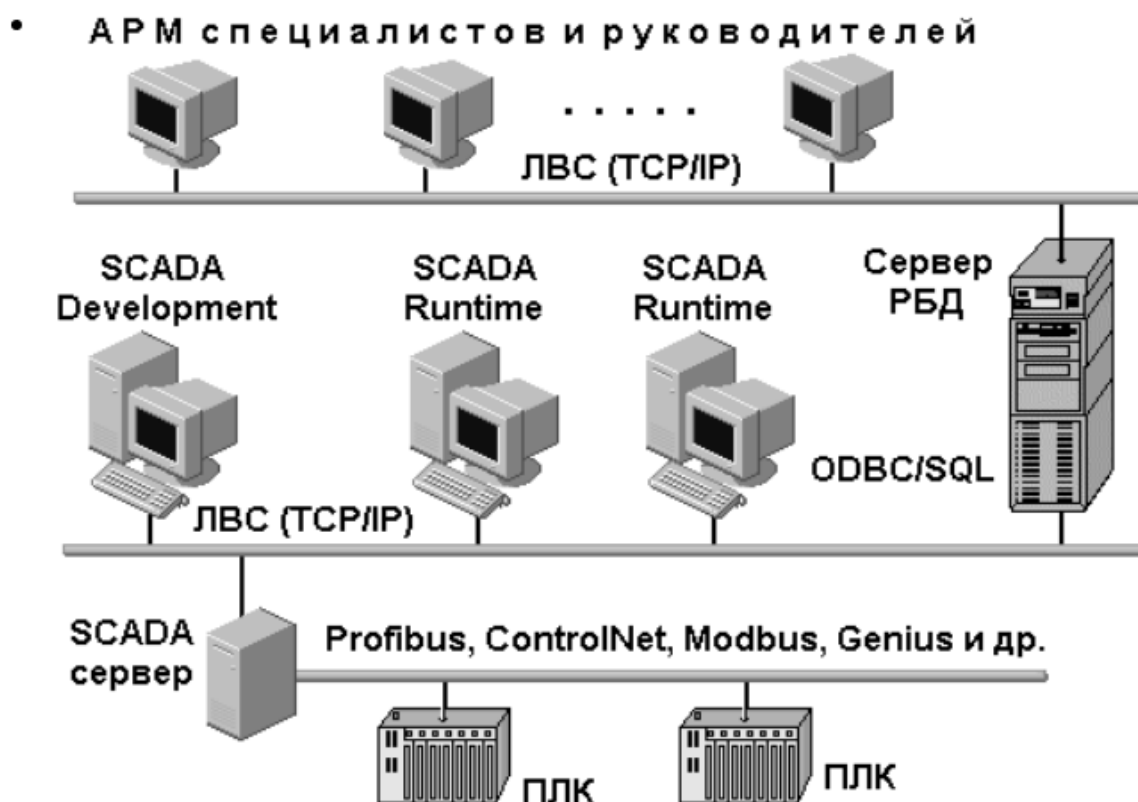


Рисунок 3.3 – Интеграция SCADA в систему управления

Практически любая SCADA-система имеет в своем составе базу данных реального времени и подсистему архивирования данных. Но подсистема архивирования не предназначена для длительного хранения больших массивов информации (месяцы и годы). Информация в ней периодически обновляется, иначе для нее просто не хватит места. Рассматриваемый здесь класс программного обеспечения (SCADA-системы) предназначен для обеспечения текущей и архивной информацией оперативного персонала, ответственного за непосредственное управление технологическим процессом.

Информация, отражающая хозяйственную деятельность предприятия (данные для составления материальных балансов установок, производств,

предприятия в целом и т. п.), хранится в реляционных базах данных (РБД) типа Oracle, Sybase и т. д. В эти базы данных информация поставляется либо с помощью ручного ввода, либо автоматизированным способом (посредством SCADA-систем). Таким образом, выдвигается еще одно требование к программному обеспечению SCADA – наличие в их составе протоколов обмена с типовыми базами данных.

Наиболее широко применимы два механизма обмена:

1) ODBC (Open Data Base Connectivity – взаимодействие с открытыми базами данных) – международный стандарт, предполагающий обмен информацией с РБД посредством ODBC-драйверов. Как стандартный протокол компании Microsoft, ODBC поддерживается и наиболее распространенными приложениями Windows;

2) SQL (Structured Query Language) – язык структурированных запросов.

Программное обеспечение SCADA должно взаимодействовать с контроллерами для обеспечения человеко-машинного интерфейса с системой управления (см. рисунок 3.3). К контроллерам через модули ввода/вывода подключены датчики технологических параметров и исполнительные устройства (на рисунке 3.3 не показаны).

Информация с датчика записывается в регистр контроллера. Для ее передачи в базу данных SCADA-сервера необходима специальная программа, называемая драйвером. Драйвер, установленный на сервере, обеспечивает обмен данными с контроллером по некоторому физическому каналу. Но для реализации обмена необходим и логический протокол.

После приема SCADA-сервером сигнал попадает в базу данных, где производится его обработка и хранение. Для отображения значения сигнала на мониторе рабочей станции оператора информация с сервера должна быть передана по сети клиентскому компьютеру. И только после этого оператор получит информацию, отображенную изменением значения, цвета, размера, положения и т. п. соответствующего объекта операторского интерфейса.

Большое количество контроллеров с разными программно- аппаратными платформами и постоянное увеличение их числа заставляло разработчиков включать в состав SCADA-системы большое количество готовых драйверов (до нескольких сотен) и инструментарий для разработки собственных драйверов к новым или нестандартным устройствам нижнего уровня.

Для взаимодействия драйверов ввода/вывода и SCADA до недавнего времени использовались два механизма:

1) DDE (Dynamic Data Exchange – динамический обмен данными) – обмен по собственным (известным только фирме-разработчику) протоколам (рисунок 3.4);

2) технология OPC (OLE for Process Control).



Рисунок 3.4 – Обмен информацией с помощью DDE-протокола

Технология OPC (OLE for Process Control) основана на технологии OLE (Object Linking and Embedding – встраивание и связывание объектов). Под объектами здесь подразумеваются так называемые компоненты, которые представляют собой готовые к использованию мини-приложения. Встраивая и связывая эти компоненты, можно разрабатывать приложения компонентной архитектуры.

Этот новый подход к разработке приложений получил название технологии COM (Component Object Model – модель компонентных объектов). Теперь приложение-клиент может удаленно вызывать те или иные функции этих объектов так, как будто объекты находятся «рядом». Объект может находиться и в самом деле рядом (в адресном пространстве приложения) – тогда это просто COM.

Если же объект находится в другой программе на том же компьютере или на другом узле сети, то это DCOM-Distributed (распределенная) COM.

OPC представляет собой коммуникационный стандарт, поддерживающий взаимодействие между полевыми устройствами, контроллерами и приложениями разных производителей. Стандарт OPC описывает компонентные объекты, методы и свойства (базирующиеся на технологии OLE/COM) для серверов данных реального времени, таких как PLC, DCS, систем архивирования данных и других, и обеспечивает передачу информации, содержащейся на этих серверах, стандартным OLE-клиентам.

OPC-взаимодействие основано на клиент-серверной архитектуре. OPC-клиент (например, SCADA), вызывая определенные функции объекта OPC-сервера, подписывается на получение определенных данных с определенной частотой. В свою очередь, OPC-сервер, опросив физическое устройство, вызывает известные функции клиента, уведомляя его о получении данных и передавая сами данные. Таким образом, при OPC-взаимодействии используются как прямые COM-вызовы (от клиента к серверу), так и обратные (от сервера к клиенту).

Изложить идею технологии OPC можно на примере стандартов на шины для персонального компьютера (ПК). К шине ПК можно подключать широкий класс устройств, производимых целым рядом компаний, и все они будут иметь возможность взаимодействовать друг с другом, поскольку используют одну и ту же стандартную шину. Также и унифицированный интерфейс OPC позволяет различным программным модулям, производимым самими различными компаниями, взаимодействовать друг с другом.

OPC-сервер отвечает за получение данных от соответствующего устройства управления процессом. На каждом сервере имеется некоторое количество OPC-групп, которые представляют собой логические коллекции данных, запрос на получение которых поступает от клиента. Группы на сервере могут быть доступны нескольким клиентам одновременно или лишь одному клиенту.

Каждая OPC-группа содержит набор OPC-элементов, в которых хранятся данные, поступившие от соответствующего устройства управления процессами. Запрос клиента серверу на получение данных реализуется посредством указания идентификатора элемента. По уникальному идентификатору сервер умеет находить нужное значение в соответствующем устройстве (например, контроллере). Для программируемого логического контроллера (ПЛК) идентификатор элемента обычно соответствует номеру регистра. Дополнительно сервер может снабжать полученные данные меткой времени.

Таким образом, любое устройство, для которого есть OPC-сервер, может использоваться вместе с любой современной SCADA-системой.

Стандарт OPC разрабатывался специально для использования в промышленной автоматизации и поэтому имеет вполне содержательную концептуальную сторону.

Стандарт состоит из трех основных спецификаций:

- 1) доступ к данным реального времени (Data Access);
- 2) обработка тревог и событий (Alarms & Events);
- 3) доступ к историческим данным (Historical Data Access).

Соответственно, OPC-серверов тоже может быть три вида, хотя допускается совмещать все эти функции в одном сервере. OPC-серверы физических устройств обычно являются только серверами данных.

Раньше разработчикам клиентских приложений приходилось писать множество драйверов (см. рисунок 3.5) для взаимодействия с каждым из используемых управляющих устройств (контроллеров).



Рисунок 3.5 – Схема взаимодействия управляющих устройств

Стандарт OPC позволяет написать лишь один-единственный драйвер (рисунок 3.6) для доступа к данным, поступающим в едином формате от самых различных источников.



Рисунок 3.6 – Схема взаимодействия управляющих устройств при использовании стандарта OPC

OPC-интерфейс допускает различные варианты обмена: с физическими устройствами, с распределенными сетевыми системами управления и с любыми приложениями (рисунок 3.7).

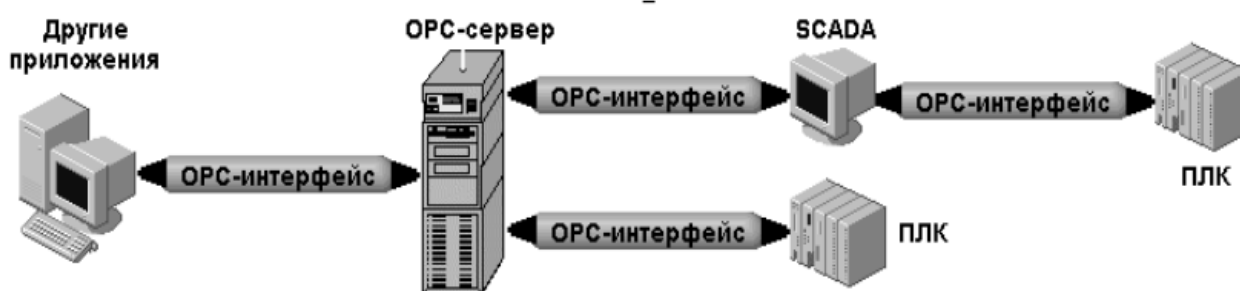


Рисунок 3.7 – Обмен данными по OPC-интерфейсу

Использование технологии OPC позволяет конечным пользователям выбирать программно-аппаратные средства, наиболее отвечающие их потребностям, независимо от того, кто их производит.

Другим преимуществом описываемой технологии является то, что при ее использовании снижаются риски и стоимость интеграционных работ. Все используемые в системе компоненты работают на одной и той же технологии.

При разработке систем автоматизации может потребоваться создание собственных программных модулей (не предусмотренных в SCADA-системе) и их включение в систему автоматизации. Поэтому свойство открытости SCADA-систем является очень важной характеристикой программных продуктов этого класса.

Открытость SCADA-системы означает возможность доступа к спецификациям системных вызовов, реализующих тот или иной системный сервис. Это может быть и доступ к графическим функциям, функциям работы с базами данных и т. д.

С другой стороны, сегодня в мире существует множество компаний, занимающихся разработкой различных программных компонентов для SCADA-систем, например, ActiveX-объектов. Их использование при разработке систем автоматизации упрощает и ускоряет процесс проектирования. Этот процесс все больше начинает напоминать процесс «сборки» прикладного программного обеспечения из готовых компонентов. Снижаются требования к квалификации программистов – количество задач, решаемых системой с помощью программ собственной разработки на высокоуровневых языках типа С или Visual Basic уменьшается. Все это способствует расширению области применения SCADA-систем.

Открытость программного обеспечения SCADA обеспечивается целым рядом факторов, а именно:

- возможностью создания собственных программных модулей и использования программных модулей разработки других компаний;
- наличием специальных драйверов для связи SCADA с наиболее популярными контроллерами разных фирм;
- наличием специальных инструментальных средств для создания новых драйверов;
- возможностью их работы в типовых операционных системах;
- наличием типовых программных интерфейсов (DDE, OLE, OPC, ActiveX, ODBC, SQL и др.), связывающих программное обеспечение SCADA с другими программно-аппаратными средствами системы управления, включая и СУБД.

Сейчас уже можно сказать, что современные системы SCADA хорошо структурированы и представляют собой готовые к применению и согласованные по функциям и по всем интерфейсам наборы программных продуктов и вспомогательных компонентов.

Организация доступа к SCADA-приложениям. SCADA-приложения, по определению, являются потребителями технологических данных, но, с другой стороны, они должны быть и их источником. Информация со SCADA-приложений потребляется многочисленными клиентами (прежде всего, специалистами и руководителями среднего звена).

Для автоматизированного доступа к информации реального времени с любого рабочего места необходимо установить компьютер, подключенный к локальной сети. Организованное таким образом автоматизированное рабочее место (АРМ) предназначено для реализации вполне определенных функций. Поэтому программное обеспечение компьютера (системное и прикладное) должно обеспечить соответствующий данному АРМ набор пользовательских услуг. К их числу можно отнести:

- объем предоставляемой информации;
- форма представления информации;
- реализуемые функции (только информационные или с возможностью выдачи управляющих воздействий);
- протяженность и надежность канала связи «источник-потребитель»;
- простота освоения пользователем и т. д.

За системным и прикладным программным обеспечением, которое необходимо компьютеру АРМ для получения удаленного доступа к производственной информации, закрепился термин «клиентское приложение». Клиентские приложения различного типа могут предоставлять информацию в любом объеме и приемлемом для пользователя виде.

Клиент-серверная организация SCADA-систем предполагает применение клиентских приложений двух типов: с возможностью передачи управляющих воздействий с клиентского приложения и чисто мониторинговые приложения. Пользователю необходимо лишь определить достаточный набор услуг.

Самыми простыми и распространенными клиентскими приложениями в настоящее время являются клиенты в локальной сети (рисунок 3.8). Такие клиентские приложения в SCADA-системах традиционно объединяются с серверными приложениями протоколами локальных сетей. Часто таким протоколом является TCP/IP.



Рисунок 3.8 – Организация доступа к информации через локальную сеть

Для организации АРМ потребуется компьютер достаточно хорошей конфигурации и лицензионное программное обеспечение SCADA. Когда речь идет об организации большого количества автоматизированных рабочих мест на базе программного обеспечения SCADA, то такое решение может оказаться дорогостоящим. К тому же, большинство пользователей SCADA-приложений, в отличие от операторов/диспетчеров, относится к категории нерегулярных, т. е. подключается к системе периодически по мере необходимости.

Технология сервер/терминал. Постоянное появление новых версий программного обеспечения, предъявляющих все более высокие требования к производительности клиентских ПК, привело к тому, что некоторые компании-разработчики программного обеспечения решили разработать технологию, которая бы обеспечила выполнение всех высокопроизводительных вычислений на сервере, оставляя клиентским компьютерам роль терминалов.

Технология сервер/терминал поддерживает режим клиентских сессий, когда один сервер обслуживает несколько клиентов, функционирующих независимо друг от друга. При этом каждый терминал получает свой ресурс: память, время центрального процессора, доступ к дискам сервера и приложениям. Когда клиент запускается, терминальный сервер регистрирует его, предоставляя доступ к ресурсам сервера. Windows Terminal Server создает виртуальный дисплей, изображение которого отображается на локальном мониторе. Операции ввода, активизируемые клиентом с клавиатуры и мыши, обслуживаются сервером. Добавление нового клиента заключается лишь в подключении нового терминала к сети.

Терминальные пользователи имеют доступ к данным, мнемосхемам, трендам, алармам с возможностью обмена информацией в реальном времени без необходимости установки SCADA-системы на локальном компьютере (терминале).

Терминал может играть роль как станции оператора/диспетчера, так и АРМ нерегулярных пользователей (технологов, специалистов службы КИП и т. п.), которые могут иметь доступ к необходимой оперативной информации о технологическом процессе и оборудовании (рисунок 3.9).

Для организации взаимодействия между сервером и терминалом/клиентом используются стандартные протоколы:

- для ОС Windows – Microsoft RDP (Remote Desktop Protocol);
- для ОС Linux/CE – Citrix ICA (Independent Computing Architecture).

Применение терминал/серверной технологии позволяет создавать более экономичные решения за счет того, что:

- приложения устанавливаются и поддерживаются администратором только на сервере;
- обновление программного обеспечения выполняется только один раз и только на сервере;
- терминальные клиенты могут быть реализованы на различных и, что особенно важно, недорогих платформах.

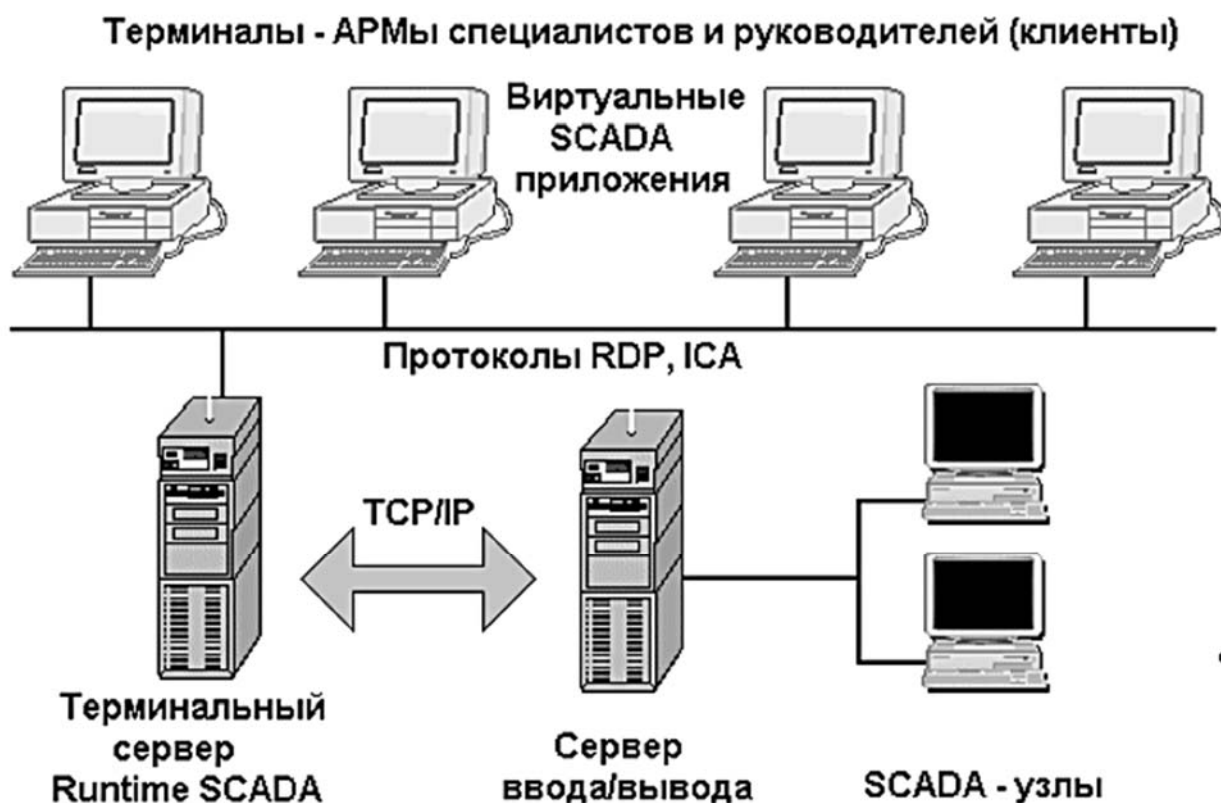


Рисунок 3.9 – Архитектура терминал-сервер

При работе в терминальном режиме вся обработка информации производится на сервере. Его конфигурация зависит от установленных на сервере приложений и от количества обслуживаемых им терминалов. При обработке высокоскоростных приложений для большого количества терминалов (десятки) может потребоваться достаточно дорогостоящий сервер (большая оперативная память).

Используя новые архитектурные возможности, компании-разработчики SCADA-систем стали предлагать терминальные сервисы, поддерживающие выполнение SCADA-приложений в режиме сессии.

Задание

Изучить архитектурное построение SCADA-систем и ответить на контрольные вопросы.

Контрольные вопросы

1 Дайте определения следующим терминам и понятиям: SCADA, интегрированная среда разработки проекта, исполнительные модуль, узел проекта, канал.

2 Дайте характеристику клиент- серверной архитектуре.

3 Что понимают под станцией-сервером?

4 Какие модули включают SCADA-программы?

5 Дайте классификацию SCADA-пакетов с точки зрения их структурного построения.

6 Охарактеризуйте открытые SCADA-системы.

7 Способна ли SCADA-система производить логические и математические вычисления?

4 Практическая работа № 4. Управление работой роботизированного производства с использованием SCADA-системы

Цель работы: приобретение практических навыков работы со SCADA-системами.

Теоретические сведения

Программный пакет MasterSCADA (рисунок 4.1) предназначен для проектирования систем диспетчерского управления и сбора данных.

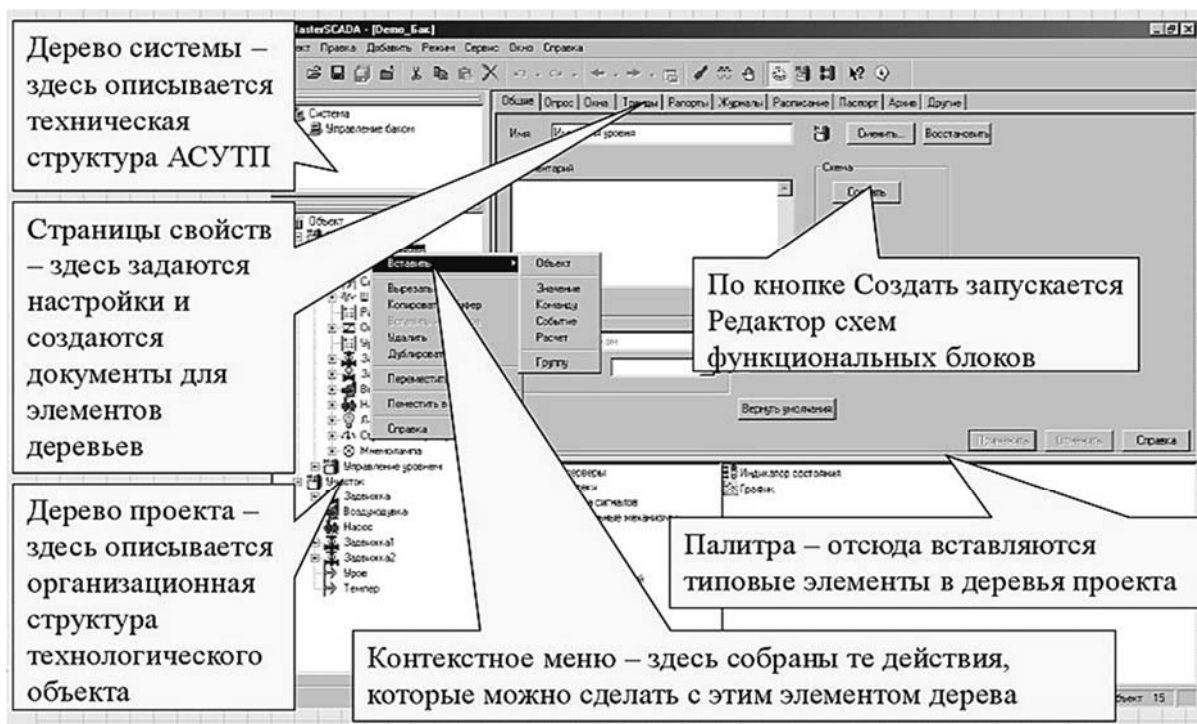


Рисунок 4.1 – Основное окно MasterSCADA

Объект в MasterSCADA – это основная единица разрабатываемой системы, соответствующая реальному технологическому объекту (цеху, участку, аппарату, насосу, задвижке, датчику и т. п.), управляемому разрабатываемой с помощью MasterSCADA системой. Каждый объект имеет набор свойств и документов.

Свойства объекта – период и способ опроса, восстановление после перезапуска, ограничение доступа в режиме исполнения.

Документы объекта – архивы, мнемосхемы, изображения, тренды, журналы, отчеты.

Обработка измерений. Датчики с аналоговым выходом подключаются к системе контроля и управления через аналоговые входы контроллеров, многоканальных измерительных приборов, модулей удаленного ввода сигналов, реже – через цифровые интерфейсы одноканальных датчиков (рисунок 4.2). Эти интеллектуальные приборы обычно проводят ту или иную первичную обработку параметров: фильтрацию, линеаризацию, масштабное преобразование из кодов АЦП в нормализованную шкалу (например, процентную) или физические единицы.

Шкала параметра. Для унификации задач первичной обработки в MasterSCADA предусмотрено понятие шкалы параметра. Шкала определяет диапазон изменения (поступающее в MasterSCADA с аналогового входа контроллера значение сигнала будет автоматически масштабироваться к этому диапазону), верхние (аварийная и предупредительная) и нижние границы контроля значения, а также граница контроля скорости изменения параметра в единицу времени.

Интервал	Начало	Конец	Формат	Ед. измерения
Проценты	0.00	100.00	f2	Процент
Градусы	0.0	100.0	f1	Градус
Температура 180	0.0	180.0	f1	Градус С
Расход 16	0.00	16.00	f2	Куб. метр в час
Давление 10	0.00	10.00	f2	Килограмм на кв. см
Уровень 100	0.0	100.0	f1	Процент

Рисунок 4.2 – Окно единиц измерений

Для шкалы задается (выбирается из имеющегося списка или создается новая) единица измерения и определяется точность отображения числового значения параметра в документах проекта.

В составе понятия «шкала параметра» нет алгоритмов первичной обработки (фильтрации, линеаризации и др.). Это связано с тем, что большая часть современного оборудования производит такую обработку уже на уровне модулей аналого-цифрового преобразования, снабженных микропроцессором и встроенным программным обеспечением.

MasterSCADA позволяет выбрать для шкалы методику поверки (набор контрольных точек и число измерений в каждой из них) и автоматизировать процедуру фиксации измерений, их статистической обработки и формирования стандартной отчетности (рисунок 4.3).

Обработка значения. Для обработки значения в MasterSCADA имеются обширные библиотеки алгоритмов (функциональных блоков). Эти блоки могут использоваться как в рамках проектной иерархии объекта, так и в традиционных графических схемах функциональных блоков.

Основные блоки, относящиеся к обработке параметров, сосредоточены в библиотеках «Обработка сигналов», «Вычисления» и некоторых других в палитре MasterSCADA.

Среди них есть динамические преобразования (фильтрация, дифференциал, ограничитель скорости, запаздывание и т. п.); интервальные обработки (на скользящем или дискретном интервале): максимум, минимум, интеграл, среднее; вычисление расходов жидкостей и газов (воды, мазута, пара, природного газа), а также алгоритмы не только для преобразования единичного параметра, но и для многопараметрической обработки данных.

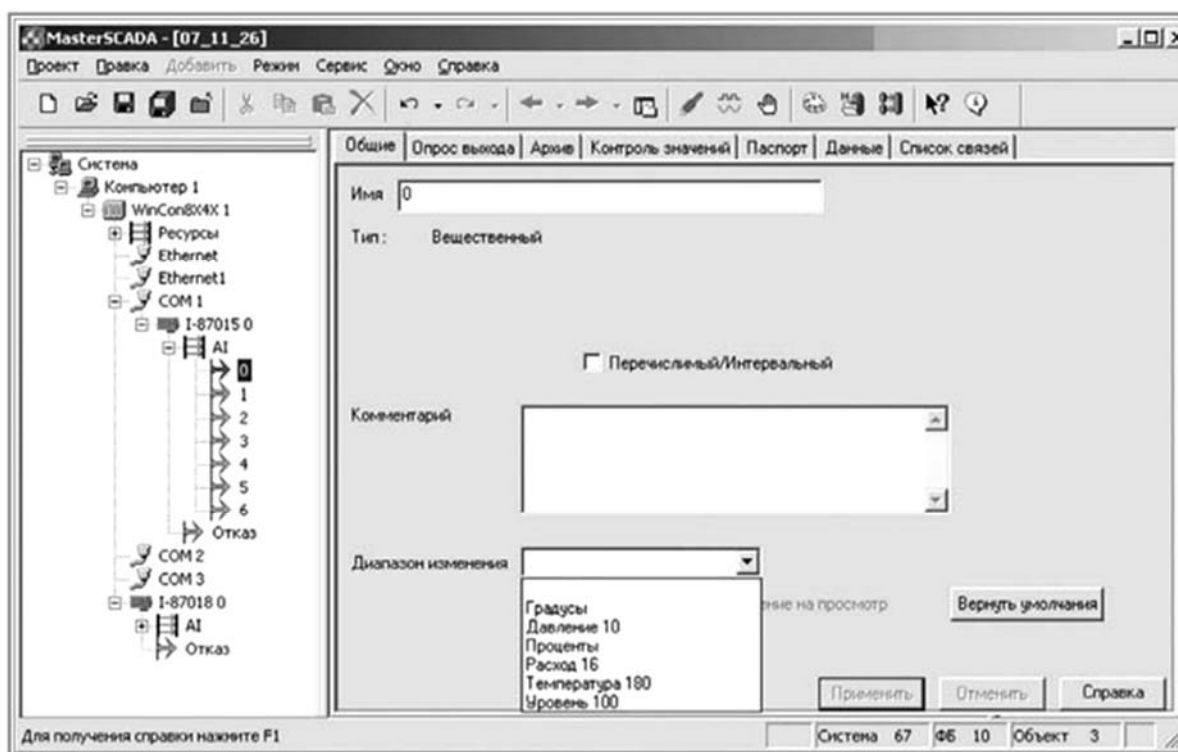


Рисунок 4.3 – Окно статической обработки результатов измерений

Использование параметра в расчетах. Помимо функциональных блоков параметр может быть задействован в формульных вычислениях. Для этого необходимо использовать стандартную переменную MasterSCADA «Расчет» (или «Событие» – расчет с логическим результатом).

Используемый в формуле параметр должен быть перетащен на страницу формулы в список используемых переменных. В самой формуле используются формальные параметры с условными именами. Конкретный фактический параметр, являющийся источником данных для формального параметра, в любой момент может быть заменен на другой, перетаскиванием нового параметра в строку списка без необходимости редактирования формулы.

В формуле можно использовать арифметические, логические, побитовые операции и операции сравнения, а также ряд встроенных функций и констант – всего около сотни операций, включая работу с такими свойствами параметра, как метка времени, признак нарушения границы, архивное значение в определенный момент времени или за период (интеграл, среднее и др.).

Наряду со стандартными операциями в формуле используется заимствованная из языков процедурного программирования конструкция условного вычисления, подобная ЕСЛИ-ТОГДА-ИНАЧЕ. (IF-THEN-ELSE).

Отображение параметра на мнемосхемах. Текущее значение той или иной переменной оператор может увидеть на мнемосхеме. Для этого достаточно перетащить параметр из проекта на поле мнемосхемы. Перетаскивание с помощью левой клавиши мыши дает отображение параметра в виде строки, в котором по умолчанию указано проектное имя параметра и его значение с заданным в шкале количеством знаков после запятой и необходимой единицей измерения.

При выходе значения за контрольную границу значение будет соответственно менять цвет. Если перетащить параметр правой кнопкой мыши, то откроется полный список возможных графических представлений параметра, соответствующих типовым щитовым приборам: столбчатому индикатору (вертикальному или горизонтальному), стрелочному прибору, самописцу. Их шкалы также отражают заданные контрольные границы.

Анимация мнемосхем с помощью параметра. Для визуального отображения изменения значения параметра в интуитивно понятной форме используют изменения визуальных свойств частей изображения мнемосхемы, соответствующих графическому представлению об технологическом оборудовании, для контроля за которым измеряется данный параметр. Например, уровень жидкости в емкости удобно показывать не только цифрой, но и высотой заливки изображения емкости другим цветом, символизирующим заполняющую емкость жидкость. Аналогично, температура этой жидкости может отображаться цветом этой заливки, а ее плотность или вязкость – степенью растривания.

Такая анимация (динамизация) свойств легко делается в MasterSCADA и может быть применена не только для «родных» библиотечных элементов, но и для свойств любого позаимствованного извне элемента управления ActiveX. Для этого необходимо выделить элемент на мнемосхеме, открыть закладку «Входы» панели свойств элемента. В поле любого свойства перетащить параметр,

по изменению значения которого должно меняться и данное свойство. После того как связь между свойством элемента мнемосхемы и параметром дерева объектов будет обозначена, потребуется настроить интервалы динамизации (рисунок 4.4).

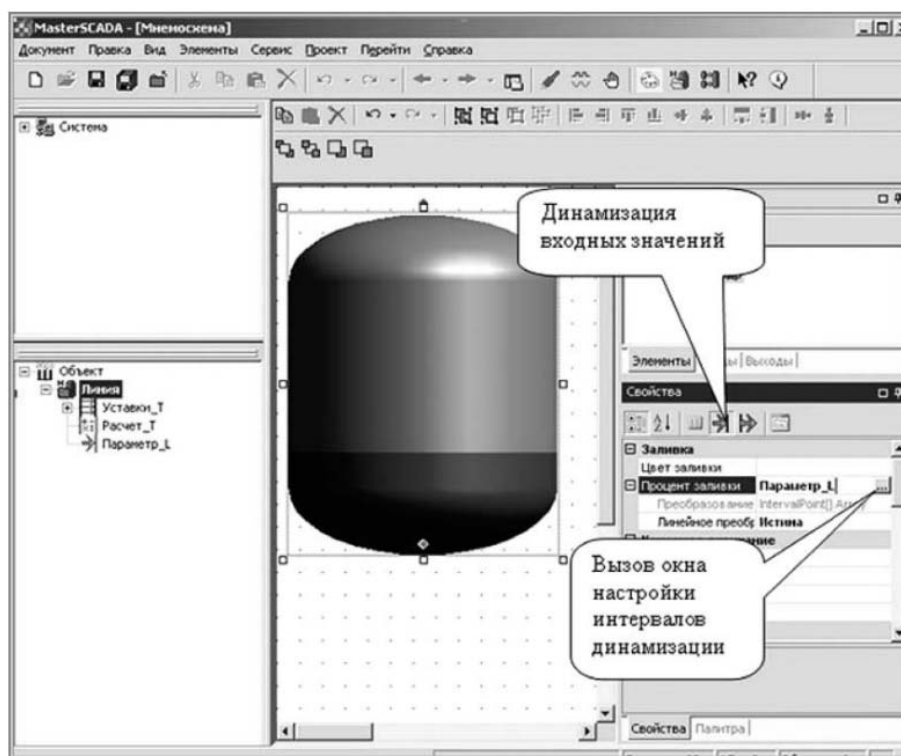


Рисунок 4.4 – Окно динамизации

MasterSCADA предусматривает два вида изменения свойств – линейное (пропорционально изменению параметра, например, плавное изменение уровня заливки или ее цвета) и ступенчатое, после пересечения значением границы интервала (например, для отображения каких-либо состояний оборудования).

Использование параметра на трендах. Изменения параметра во времени можно увидеть не только при помощи элемента мнемосхемы «График», но и в специальном приложении «Тренд».

Тренды MasterSCADA отображают любой параметр на всю глубину его сбора, причем на едином графике, без разделения на текущие и архивные данные. Тренды позволяют анализировать точки пересечения графиком контрольных или иных заданных оператором границ, сравнивать два параметра, строить графики их взаимной зависимости и многое другое (рисунок 4.5).

Чтобы поместить параметр на тренд, достаточно перетащить его в любое место окна графика, при этом образуется перо, которое по умолчанию будет иметь имя и шкалу, унаследованные от исходного параметра, в дальнейшем параметры пера можно переопределить.

Выбор параметров для просмотра на тренде возможен не только в инструментальной системе, но и непосредственно оператором в процессе функционирования исполнительной системы.

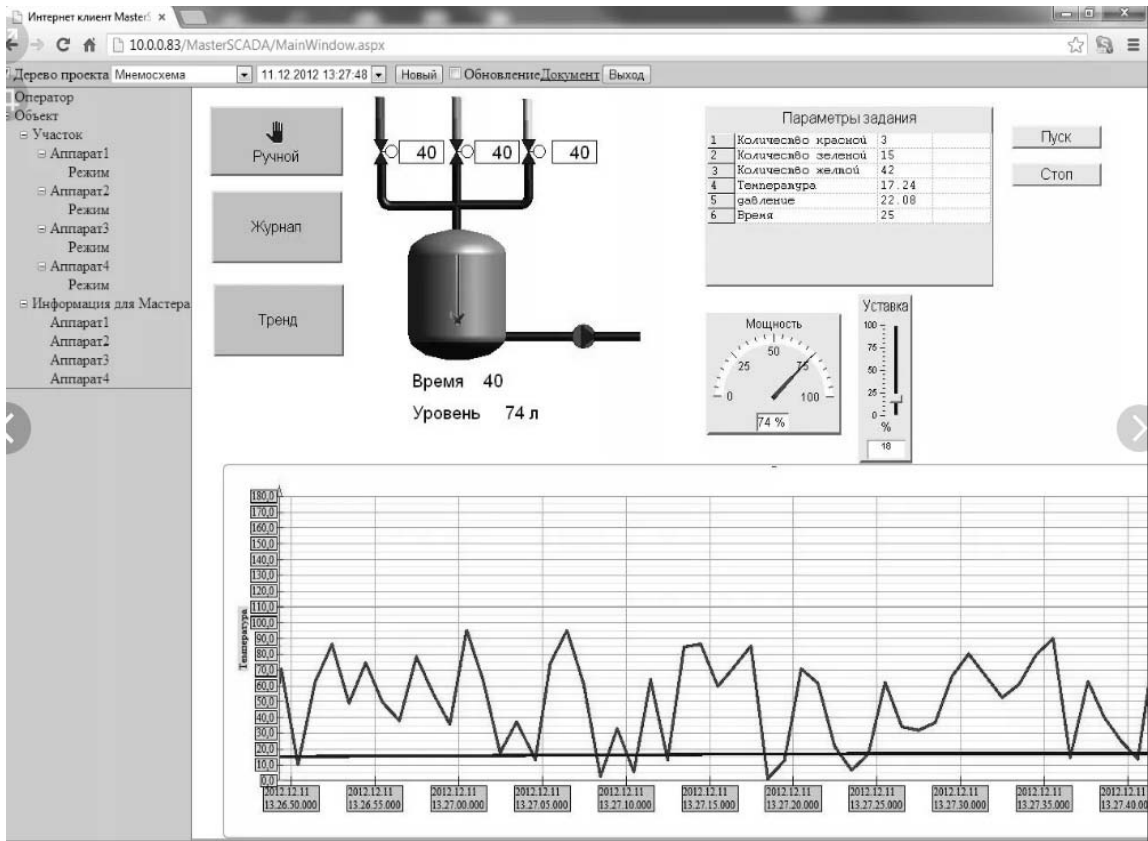


Рисунок 4.5 – Окно приложения «Тренд»

Тренд в MasterSCADA может просматриваться не только в графическом, но и в табличном виде, при необходимости он может быть сохранен в различных форматах (.jpg, .txt, .xls и др.).

Не всегда мнемосхемы и графики дают полную информацию о ситуации на объекте или состоянии оборудования. Иногда оператору надо описать ее детальнее.

Не менее важно эту информацию сохранить для последующего просмотра с целью анализа текущей обстановки. Для этих целей служат текстовые сообщения, формируемые автоматически (по нарушению заданных границ) (рисунок 4.6).

Автоматические сообщения (в MasterSCADA они называются системные) могут быть настроены или отключены на соответствующих закладках дерева системы для системы в целом или для отдельных компьютеров в частности.

Вычисляемые (пользовательские) сообщения формируются на закладке переменных типа «Событие» (рисунок 4.7).

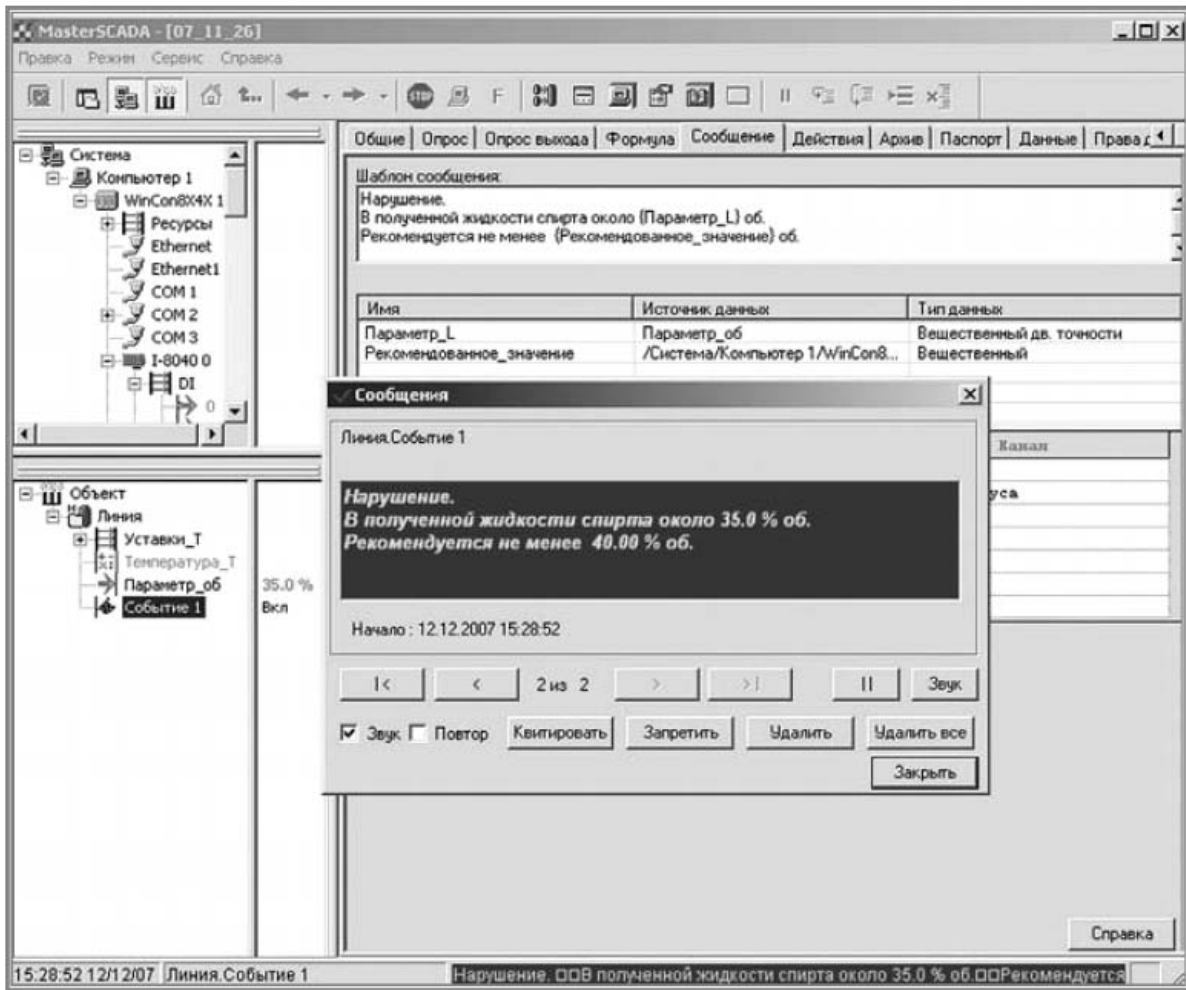


Рисунок 4.6 – Окно автоматического сообщения о нарушении заданных границ параметров процесса

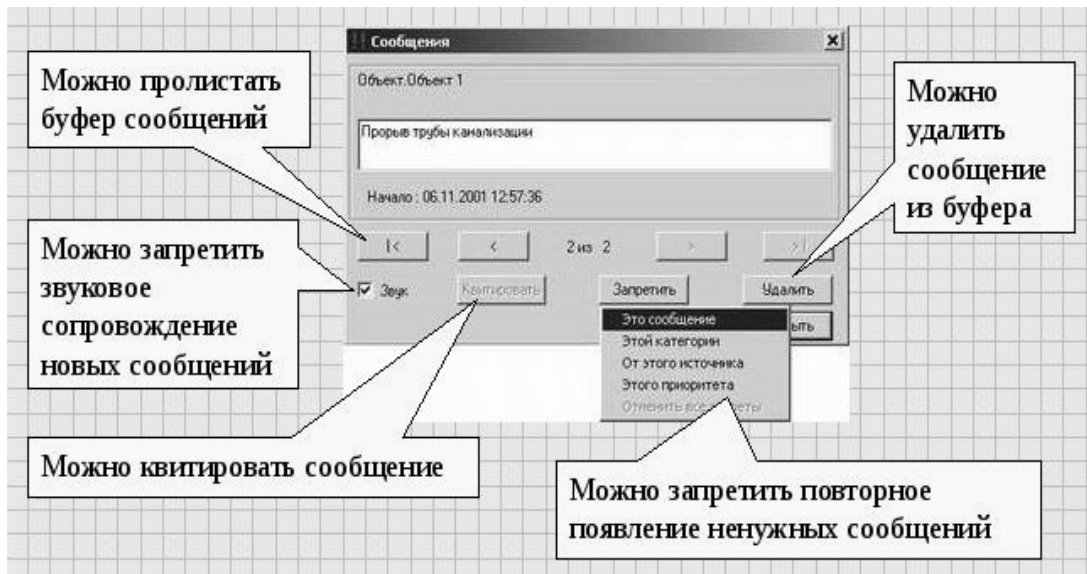


Рисунок 4.7 – Окно настройки пользовательских сообщений

Порядок выполнения работы

1 Получить у преподавателя задание на создание простого проекта управления РТК на основе автоматического регулирования параметров процесса.

2 Составить мнемосхему для управления работой РТК в среде MasterSCADA.

3 В режиме имитации работы РТК провести мониторинг контролируемых параметров.

Содержание отчёта

1 Название и цель работы.

2 Общий вид мнемосхемы управления работой РТК.

3 Контролируемые параметры мнемосхемы.

4 Расчетные граничные значения контролируемых параметров.

5 Вывод по работе.

Контрольные вопросы

1 Архитектура MasterSCADA.

2 Дайте определение объекту в MasterSCADA.

3 OPC-сервер в MasterSCADA.

4 Опишите модуль расчета системы MasterSCADA.

5 Опишите модуль отображения управляемых переменных системы MasterSCADA.

6 Что представляет собой мнемосхема?

Список литературы

1 **Шишов, О. В.** Современные технологии и технические средства информатизации : учебник / О. В. Шишов. – Москва : Инфра-М, 2021. – 462 с.

2 **Трофимов, В. Б.** Интеллектуальные автоматизированные системы управления технологическими объектами: учебное пособие / В. Б. Трофимов, С. М. Кулаков. – 2-е изд., испр. – Москва ; Вологда : Инфра-Инженерия, 2020. – 256 с.

3 **Беккер, В. Ф.** Технические средства автоматизации. Интерфейсные устройства и микропроцессорные средства: учебное пособие / В. Ф. Беккер. – 2-е изд. – Москва : РИОР ; Инфра-М, 2020. – 152 с.

4 **Юсупов, Р. Х.** Основы автоматизированных систем управления технологическими процессами: учебное пособие / Р. Х. Юсупов. – Москва : Инфра-Инженерия, 2018. – 132 с.

5 **Андрейчиков, А. В.** Интеллектуальные информационные системы и методы искусственного интеллекта : учебник / А. В. Андрейчиков, О. Н. Андрейчикова. – Москва : Инфра-М, 2021. – 530 с.

6 **Пятаева, А. В.** Интеллектуальные системы и технологии : учебное пособие / А. В. Пятаева, К. В. Раевич. – Красноярск : Сиб. федер. ун-т, 2018. – 144 с.

7 **Юревич, Е. И.** Теория автоматического управления : учебник / Е. И. Юревич. – 3-е изд. – Санкт-Петербург : БХВ-Петербург, 2007. – 560 с.

8 **Перфильев, Д. А.** Интеллектуальные системы поддержки принятия решений : учебное пособие / Д. А. Перфильев, К. В. Раевич, А. В. Пятаева. – Красноярск : Сиб. федер. ун-т, 2018. – 136 с.

9 **Родина, А. А.** Технологическое оснащение автоматизированных производств на базе РТК : учебное пособие / А. А. Родина, А. Н. Колодин, Б. Н. Хватов. – Тамбов : Тамбов. гос. техн. ун-т, 2014. – 82 с.