

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

**Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования**

**ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
(ДГТУ)**

Факультет Энергетики и нефтегазопромышленности

Кафедра Автоматизация и математическое моделирование в нефтегазовом
комплексе

Конспект лекций

По дисциплине Технологические процессы нефтегазового комплекса как
объекты автоматизации

По направлению 150404 Автоматизация технологических процессов и
производств

Форма обучения заочная.

Ростов-на-Дону
2023

АППАРАТНЫЕ СРЕДСТВА АВТОМАТИЗАЦИИ ПРОЦЕССОВ НЕФТЕГАЗОВОГО КОМПЛЕКСА

1. Общие характеристики систем автоматики

Техническими элементами систем автоматического и автоматизированного управления технологическими процессами являются средства получения, преобразования и регистрации первичной информации, аналоговые и цифровые вычислительные средства, исполнительные устройства.



Рис. 1 - Элемент системы автоматики.

Средства получения, преобразования и регистрации первичной информации — это датчики и вторичные приборы. К аналоговым средствам относятся усилители, исполнительные устройства и регуляторы. Исполнительными устройствами являются технические элементы, оказывающие непосредственное физическое воздействие на технологический процесс.

При рассмотрении общих характеристик технических элементов будем считать, что любой из них можно представить как преобразователь входного сигнала x в выходной y , реализующий функциональную зависимость $y = F(x)$ (рис. 1). Рассмотрим три основные характеристики технических элементов: коэффициент преобразования, погрешность и порог чувствительности.

Коэффициент преобразования — это отношение выходной величины к входной:

$$k = y/x \quad (1)$$

или отношение их приращений:

$$k' = \Delta y / \Delta x. \quad (2)$$

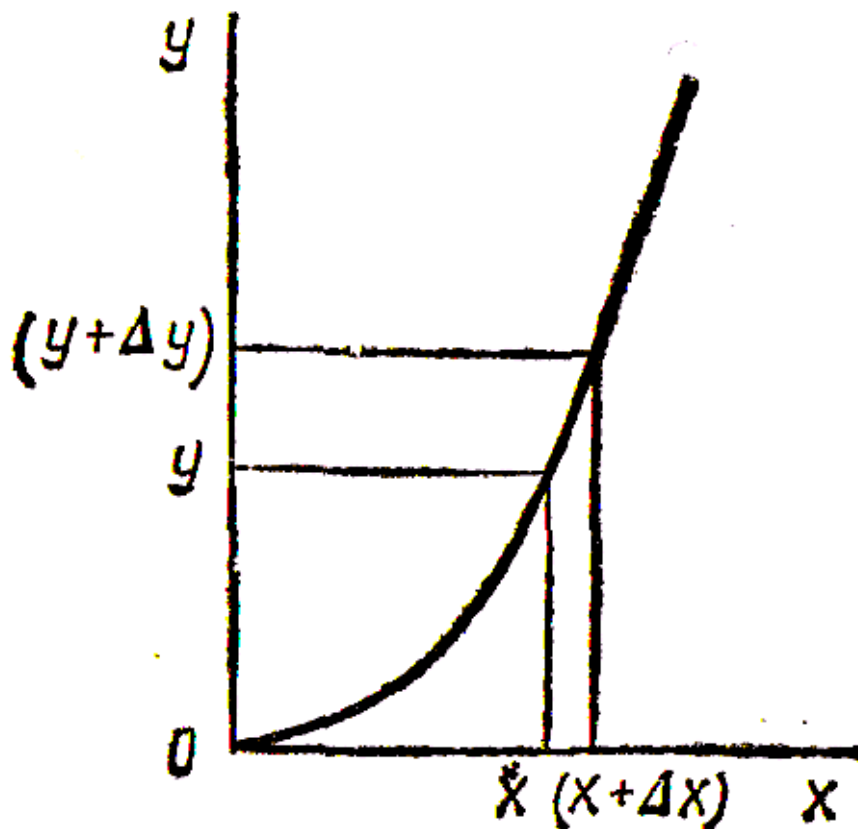


Рис. 2 - Схема определения коэффициента преобразования.

Единица измерения коэффициента образования обусловлена единицами измерения величин y и x . Иногда используют безразмерный относительный коэффициент преобразования η , которым называют отношение относительных приращений выходной и входной величин (рис. 2):

$$\eta = (\Delta y / y) / (\Delta x / x) = (\Delta y / \Delta x) / (y / x), \quad (3)$$

или при $\Delta x \rightarrow 0$

$$\eta' = (dy / dx) / (y / x). \quad (4)$$

Если функция $y = F(x)$ является пропорциональной зависимостью, то $k = k' = \text{const}$ и $\eta = \eta' = 1$. Применительно к отдельным функциональным элементам коэффициент преобразования называется по-разному. Для датчиков — это чувствительность, для усилителей — коэффициент усиления (по току, напряжению или мощности) и т. п. Требования к значению коэффициента преобразования обусловлены назначением элемента. Так, например, применительно к датчикам требуется максимальная чувствительность, а к стабилизаторам — минимальная; к усилителям предъявляется требование максимального коэффициента усиления.

Нужно отметить, что в усилителях широко применяется обратная связь для увеличения коэффициента усиления или повышения его стабильности. В первом случае вводят положительную обратную связь, а во втором — отрицательную.

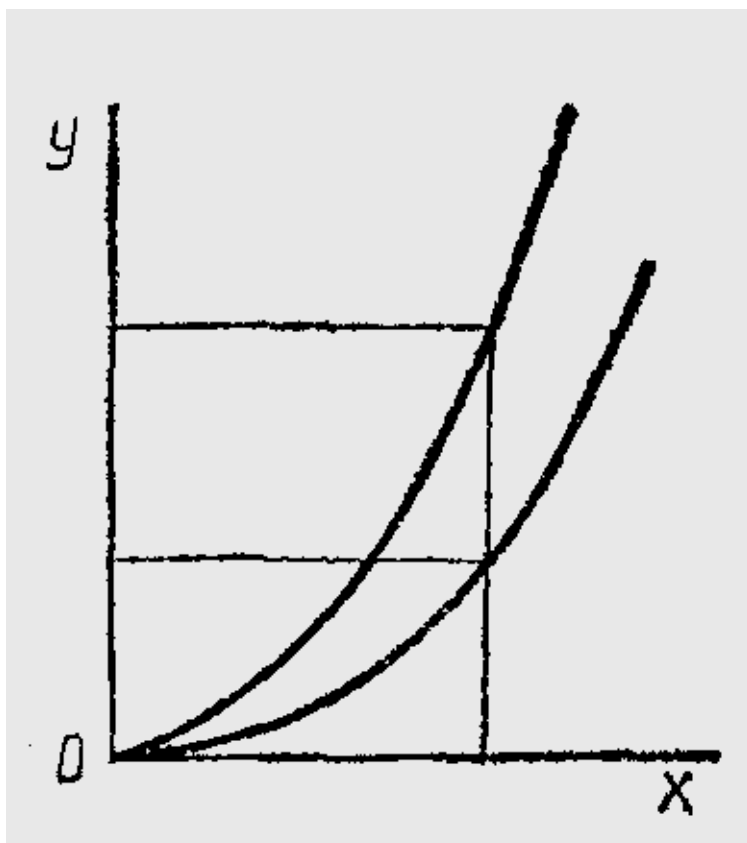


Рис. 3- Схема определения погрешности.

Погрешность — отклонение выходной величины y от истинного значения вследствие изменения внутренних свойств элемента или деления погрешности внешних условий работы. При наличии погрешности характеристика $y' = F(x)$ отлична от градуировочной $y = F(x)$ (рис. 3). Погрешность может иметь различные названия в зависимости от причин, вызывающих ее,— температурная, частотная, колебания напряжения питания, нестабильность (из-за изменения параметров с течением времени) и т. п. При характеристике свойств элементов используют понятия абсолютной, относительной и приведенной относительной погрешности.

Абсолютная погрешность

$$\Delta y = y' - y, \quad (5)$$

где y — значение выходной величины, а y — ее градуировочное значение.

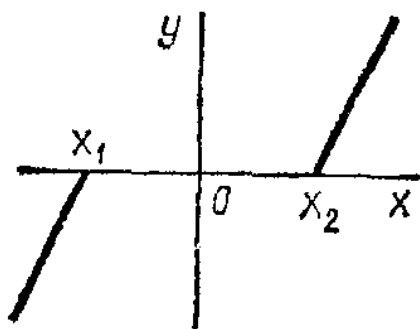


Рис. 4 - Порог чувствительности при наличии трения

Относительная погрешность (%)

$$\delta = (\Delta y / y) \cdot 100; \quad (6)$$

приведенная относительная погрешность (%)

$$\delta_{\pi} = (\Delta y / y_{\max}) \cdot 100, \quad (7)$$

где y_{\max} — максимальное значение выходной величины, определяющее диапазон ее изменения.

Обычно для оценки точности элементов используется приведенная относительная погрешность.

Порог чувствительности — минимальное по абсолютной величине приращение входной величины ($\Delta x_1 \Delta x_2$), вызывающее изменение выходной величины y (рис. 4). Интервал $[x_1 x_2]$ называют зоной нечувствительности.

Причиной существования порога чувствительности является наличие люфта, трения или гистерезиса у различных исполнительных устройств (двигателей, реле), а также наличие дрейфа и шумов на выходе усилителей.

2. Преобразователи неэлектрических параметров технологических процессов

Чувствительные элементы, преобразующие контролируемую или регулируемую величину в выходной сигнал, удобный для дистанционной передачи или дальнейшей обработки, называют *датчиками*.

Обычно в системах автоматического управления для передачи информации используются электрические сигналы. Поэтому широко применяются датчики, преобразующие неэлектрический сигнал в электрический. Датчики могут классифицироваться по назначению: датчики температуры, давления, уровня, линейных и угловых перемещений, состава веществ, оптических величин и т. п. Возможна классификация и по параметру датчика, изменяющемуся в результате преобразования измеряемой величины, датчики активного сопротивления, емкостные, индуктивные и т. п.

Датчики активного сопротивления (резисторные датчики). Принцип действия резисторных датчиков основан на изменении их электрического

сопротивления R при изменении длины l , площади сечения S или удельного электрического сопротивления ρ . Сопротивление датчика

$$R = \rho l / S, \quad (8)$$

где R — сопротивление, Ом; ρ — удельное электрическое сопротивление материала проволоки, Ом*м; l — длина проводника датчика, м; S — площадь сечения, м².

Основной характеристикой такого датчика, измеряющего перемещение, является зависимость сопротивления от перемещения, т. е. $R = f(l)$. При этом чувствительность (η) датчика

$$\eta = dR/dl = \rho/S. \quad (9)$$

Такие датчики могут работать в цепях постоянного и переменного тока. Разновидностью резисторных датчиков являются потенциометрические датчики и тензорезисторы (или тензодатчики).

Потенциометрические датчики используют в системах автоматического управления (САУ) для измерения угловых и линейных перемещений. На рис. 4 показана схема датчика линейного перемещения. Измеряемая величина x преобразуется в перемещение движка потенциометра, которому соответствует выходное напряжение ($U_{\text{вых}}$). Потенциометрические датчики используют в схемах на постоянном и переменном токе. Конструктивное исполнение датчиков различно.

Переменные резисторы выполняются из обмоточного провода, металлических пленок, полупроводниковых **материалов**. Широко применяются потенциометры непрерывной намотки. Их каркас выполняется плоским или цилиндрическим в случае преобразования поступательных перемещений и кольцевым для преобразования угловых перемещений. В качестве обмоточного провода используются материалы с высоким удельным электрическим сопротивлением (манганин, константан, нихром). Для повышения износостойчивости провода применяются сплавы из благородных металлов: платина с иридием, платина с палладием. Диаметр намоточного провода выбирается по требуемой точности и сопротивлению. Так, для высокого класса точности диаметр провода 0,03—0,1, а для низкого класса точности 0,3—0,4 мм. Обмоточный провод датчика покрывают слоем эмали или окислов. Подвижный токосъемный контакт, выполненный в виде щетки или движка, изготовляют из сплавов платины с иридием, платины с бериллием, серебра или фосфористой бронзы. Он скользит по виткам намотки потенциометра, защищенным в месте соприкосновения от изоляции.

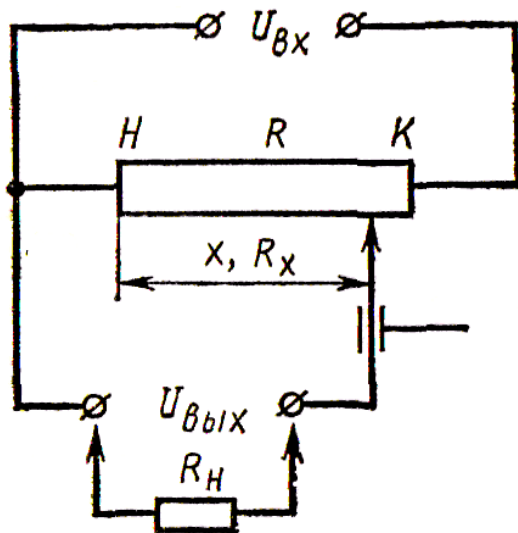


Рис. 5 - Схема датчика линейного перемещения

Для обеспечения надежной работы датчика усилие на скользящий контакт составляет $(0,5 \text{ :- } 15) \cdot 10^{-2} \text{ Н}$.

Основной характеристикой потенциометрического датчика является зависимость выходного напряжения $U_{\text{вых}}$ от перемещения x т. е. $U_{\text{вых}} = f(x)$, которая может быть линейной и нелинейной. Для потенциометра, изображенного на рис. .8, зависимость $U_{\text{вых}}(x)$ имеет вид

$$U_{\text{вых}} = U_{\text{вх}} \alpha / [1 + (\alpha/\beta)(1 - \alpha)], \quad (10)$$

где $\alpha = x/L$ — отношение перемещения движка датчика к длине всей намотки; $\beta = R_H/R$ — отношение сопротивления нагрузки к полному сопротивлению потенциометра.

Если сопротивление нагрузки значительно превышает сопротивление потенциометра, т. е. $R_H > R$, то

$$U_{\text{вых}} \cong U_{\text{вх}} \alpha = U_{\text{вх}} x/l. \quad (11)$$

На рис. 5 показана зависимость относительного изменения выходного напряжения $U_{\text{вых}}/U_{\text{вх}}$ от α при различных значениях коэффициента β .

Чувствительность датчика при малых перемещениях

$$\eta = [dU_{\text{вых}}/dx]_{x=0} = U_{\text{вх}}/l. \quad (12)$$

Проволочные датчики имеют достаточно высокую точность, но их недостатком является ступенчатость характеристики (28) и (29) из-за дискретного изменения сопротивления датчика при перемещении щетки (рис. 6). Ширина ступеньки $l_{\text{ш}}$ характеристики (рис. 7) равна шагу намотки провода, а высота R_l — сопротивлению одного витка. Идеализированной характеристикой $Y(x)$ является прямая, проведенная через середины ступенек, при этом абсолютная погрешность или зона нечувствительности

$$\Delta R_H = \pm [R/(2w)], \quad (13)$$

где R — сопротивление датчика; w — число витков обмотки датчика.
Относительная погрешность

$$\delta = \pm (\Delta R_H/R) = \pm [1/(2w)]. \quad (14)$$

Датчик имеет также технологические погрешности.

У датчиков высокого класса точности эта погрешность составляет не более 0,01 %, а для среднего класса точности — от 0,25 до 0,5 %.

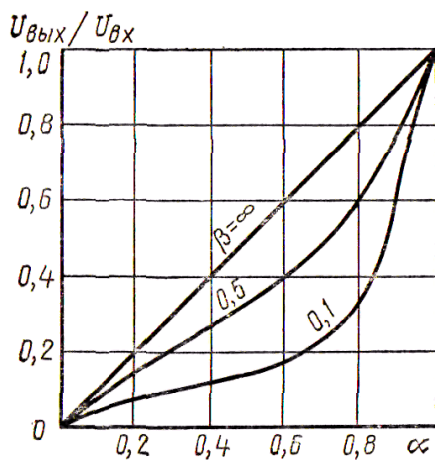


Рис. 6 Зависимость выходного напряжения датчика от перемещения

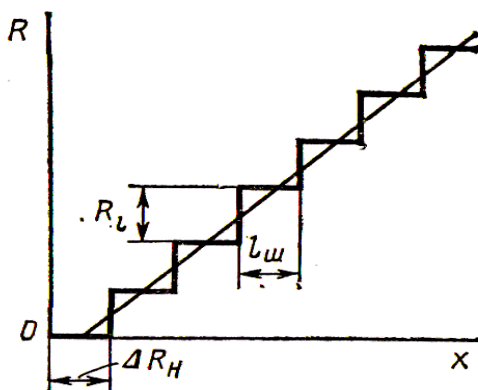


Рис. 7 Характеристика датчика линейного перемещения

Динамические свойства потенциметрических датчиков зависят от характера нагрузки. Если нагрузка активная, то датчик эквивалентен безынерционному звену с передаточной функцией

$$W(s) = k = R_x R_H / (r R_x + R R_H), \quad (15)$$

где R — полное сопротивление потенциометра, Ом; R_x — сопротивление потенциометра между точкой H и движком (см. рис. .8), Ом; r — сопротивление потенциометра между движком и точкой K (см. рис. .8), Ом.

Если нагрузка реактивная, то датчик эквивалентен апериодическому или форсирующему звену. Например, для индуктивной нагрузки передаточная функция датчика

$$W(s) = k(1 + s\tau)/(1 + sT), \quad (16)$$

где $k = R_x R_H / (r R_x + r R_H + R_x R_H)$, $\tau = L_H / R_H$,

$$T = L_n R / (r R_a + r R_n + R_x R_n).$$

Здесь L_n — индуктивность нагрузки; R_n — активная составляющая сопротивления нагрузки.

Тензорезисторы используют в системах автоматики для измерения сил и деформаций твердых тел. Существуют проволочные, фольговые, пленочные и полупроводниковые тензорезисторы. Действие проволочных, пленочных или фольговых тензорезисторов основано на изменении сопротивления проволоки, пленки или фольги при их сжатии или растяжении. Значительный прогресс достигнут в изготовлении наклеиваемых фольговых тензорезисторов. Они имеют ряд преимуществ перед проволочными, обеспечивают лучшую теплоотдачу и высокую эффективность использования при наклеивке на упругий элемент. Сопротивление фольговых тензорезисторов достигает несколько сотен ом, длина может составлять 3 мм, а толщина — от 0,025 до 0,012 мм. Проволочный тензорезистор представляет собой свернутую в спираль проволоку диаметром не выше 0,05 мм, наклеенную на полоску бумаги длиной до 40 мм и имеющую медные выводы (рис. 8).

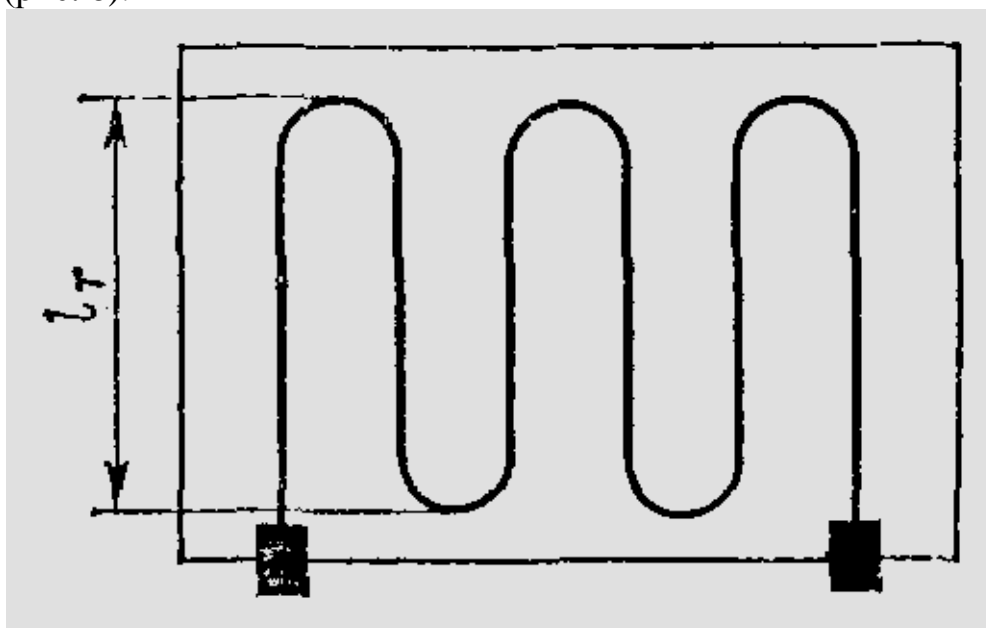


Рис 8. - Проволочный тензодатчик

В последние годы широко используются пленочные тензорезисторы. Технология их изготовления заключается в вакуумной возгонке тензочувствительного материала и последующей его конденсации на основание. В качестве тензочувствительного материала используются как металлические сплавы, так и полупроводники. Важным достоинством пленочных и фольговых тензорезисторов является то, что при их изготовлении можно получить решетку любого рисунка.

Относительная тензочувствительность тензорезистора

$$\eta = (\Delta R/R)/(\Delta l/l_T), \quad (17)$$

где R — сопротивление тензодатчика, Ом; ΔR — изменение сопротивления, Ом; l_T — длина тензочувствительного элемента, м; Δl — изменение длины датчика, вызванное измеряемой деформацией, м.

Тензочувствительные элементы изготавливают из различных сплавов.

В табл. 1 приведены величины относительной тензочувствительности для различных материалов.

Т а б л и ц а 1.

Материал проволоки	η	Материал проволоки	η
Манганин	0,5	Копель	2,4
Константен	19-2,1	Элинвар	3,2—3,5
Эдванс	2,1	Платиноиридий	5,8
Нихром	2,1—2,3	Никель	12,6

Сопротивление тензорезистора составляет обычно 100—200 Ом, измеряемая деформация не превышает 1 %. Проволочные тензорезисторы обладают высокой температурной стабильностью.

В качестве полупроводниковых тензорезисторов используют пластины, сделанные из кристаллов германия или кремния, наклеенные на бумажное основание, и с металлическими выводами (рис. .9). Длина тензорезистора составляет 5—15 мм. Полупроводниковые тензорезисторы по сравнению с проволочными обладают высокой относительной чувствительностью, которая находится в пределах от —100 до 200. Сопротивление тензорезистора 50 Ом — 5 кОм. Высокое значение относительной чувствительности позволяет снизить требования к усилителям и использовать их для измерения в расширенном диапазоне деформаций при наличии больших электромагнитных помех.

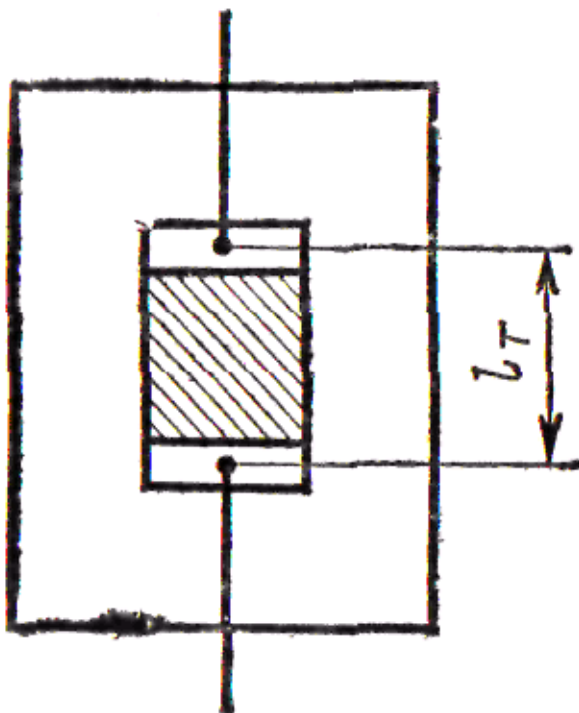


Рис. 9 - Проводниковый тензодатчик.

Нужно отметить, что разработаны тензодатчики давления модульного типа в виде гибридной интегральной схемы, в которой на общем керамическом основании размещены датчик и операционный усилитель. Тензорезисторы соединены по мостовой схеме, элементы которой нанесены на поверхность кремниевой пластинки диффузионным способом. Коэффициент чувствительности такого тензодатчика составляет 90, погрешность- 2 %. Отметим, что металлические тензорезисторы обладают высокой точностью и стабильностью; они рассчитаны на широкую область применения. Полупроводниковые тензорезисторы обычно применяются для динамических измерений в широком диапазоне частот. Как правило, тензорезисторы включаются в цепь или по схеме делителя напряжения или по мостовой схеме. Схема включения тензорезистора в цепь делителя напряжения показана на рис. 10.

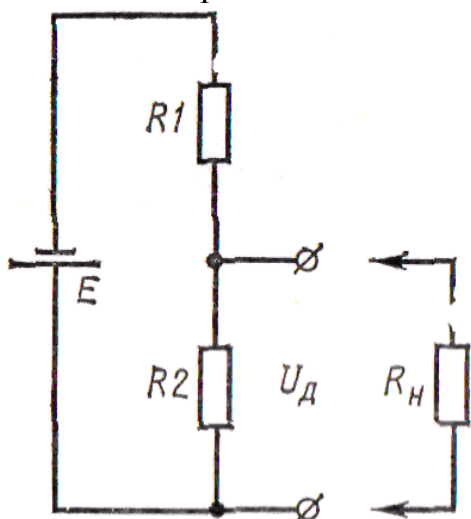


Рис 10 - Схема включения тензодатчика

Цепь делителя R_1 , R_2 питается от источника напряжения E . Выходное напряжение U_d — напряжение на тензорезисторе — подается на нагрузочное сопротивление R_n . Тензорезистор может быть включен и как сопротивление R_1 и как сопротивление R_2 . В схему моста тензорезистор включается так, как показано на рис. 10. В динамическом отношении тензодатчики эквивалентны безынерционному звену.

Емкостные датчики. Принцип действия емкостных датчиков основан на преобразовании линейных или угловых перемещений в изменение емкости конденсатора. Емкость C любого плоского конденсатора

$$C = \epsilon_0 \epsilon S / \delta, \quad (18)$$

где ϵ_0 — диэлектрическая проницаемость вакуума, равная $8,85 \cdot 10^{-12}$ Ф/м; ϵ — диэлектрическая проницаемость среды между пластинами; S — площадь перекрытия пластин, м²; δ — зазор между пластинами, м.

Относительная чувствительность плоского емкостного датчика с переменным зазором

$$\eta = \epsilon_0 \epsilon S / \delta^2. \quad (19)$$

Емкость датчика углового перемещения (рис. 11)

$$C = \epsilon_0 \epsilon S (1 + 2\alpha/\pi) / \delta, \quad (20)$$

где S — площадь перекрытия пластин при $\alpha = 0$; α — угол поворота пластины.

Чувствительность такого датчика

$$\eta = dC/d\alpha = 2\epsilon_0 \epsilon S / (\pi \delta). \quad (21)$$

Емкость датчика можно изменять, изменяя зазор между пластинами, площадь перекрытия пластин или диэлектрическую проницаемость диэлектрика.

В табл.2 указаны способы построения емкостных датчиков и области их применения.

Так как сила электростатического взаимодействия между пластинами очень мала, то рассматриваемые датчики находят

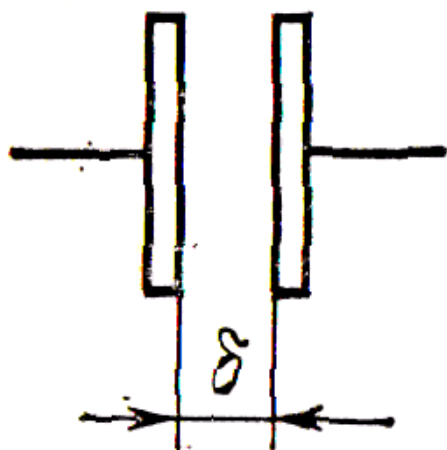


Рис. 11 - Емкостный датчик углового перемещения

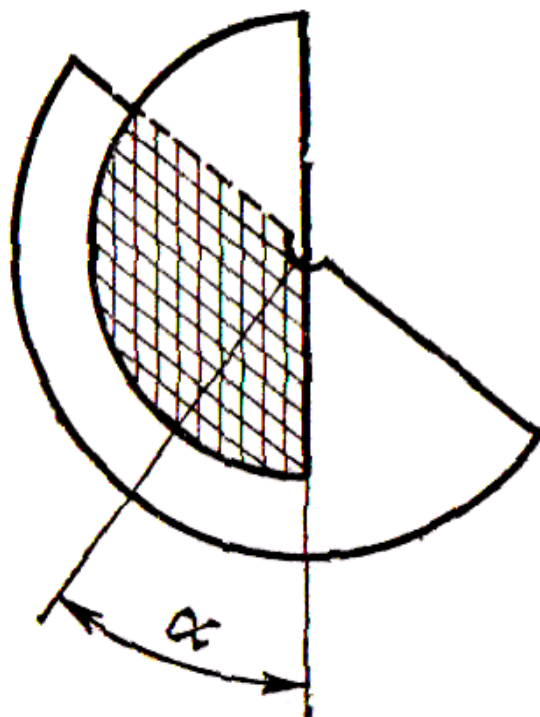


Рис. 12 - Емкостный датчик

широкое применение в тех случаях, когда недопустимы большие измерительные усилия. Они удобны для измерения вибраций с высокой частотой. Все емкостные датчики работают на переменном токе повышенной частоты (до 1000 Гц), что является их недостатком, так как усложняется схема из-за необходимости устранения паразитных емкостей. В настоящее время разработаны датчики с переменной площадью, работающие на частоте 500 Гц.

Наиболее распространенными схемами включения емкостных датчиков являются контурные и мостовые схемы. Последние в основном применяются с дифференциальными емкостными преобразователями (рис.13). Это дает возможность получить большую чувствительность при компенсации некоторых вредных внешних факторов (например, температуры). Емкости дифференциального датчика $C_1 = C_0 + \Delta C$ и $C_2 = C_0 - \Delta C$ включаются в соседние плечи моста; в два других плеча включаются или взаимосвязанные индуктивности, или полуобмотки трансформатора источника питания $U_{пит}$, или низкоомные сопротивления. Для исключения влияния емкостей кабеля в схеме предусматривают специальные меры. Сигнал на выходе $U_{вых}$ такого моста практически линейно зависит от изменения емкости датчика.

Таблица 2.

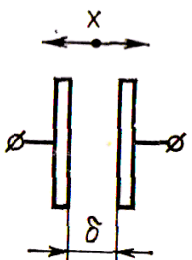
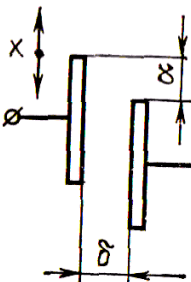
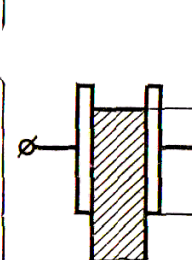
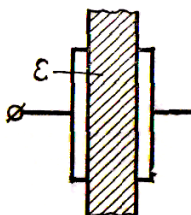
Основные сведения	Вид датчика			
	с переменным зазором δ	с переменной площадью S	с переменным уровнем среды с диэлектрической проницаемостью ϵ	с переменным значением диэлектрической проницаемости ϵ
Конструктивная схема датчика				
Область применения	Измерение малых перемещений; толщины покрытий 0—1 мм; давления	Измерение линейных или угловых перемещений более 1 мм; счет количества изделий	Измерение уровня вещества; защитная сигнализация; счет количества изделий	Измерение температуры t° , давления p , напряженности электрического поля; определение диэлектрических свойств веществ; измерение величины радиации

Схема включения емкостного датчика в контурную цепь показана на рис. 14, где LC —контур, емкость C_d которого образована датчиком и питается от генератора G стабильной частоты. Вследствие изменения емкости датчика изменяется напряжение U_K на настроенном контуре.

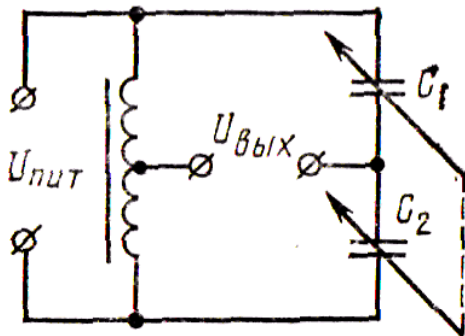


Рис. 13- Дифференциальная схема включения емкостного датчика

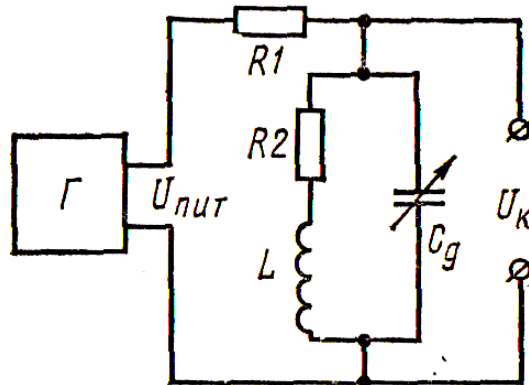


Рис. 14 - Схема включения емкостного датчика в резонансный контур

3. Вторичные приборы

Для наблюдения за ходом технологического процесса необходимо осуществлять контроль и регистрацию (запись) основных его параметров. Для этого в системы автоматики включаются вторичные приборы, работающие в комплекте с датчиками, выходной сигнал которого они измеряют. К вторичным приборам относятся милливольтметры и пирометрические вольтметры, миллиамперметры, автоматические потенциометры, автоматические уравновешенные мосты, автоматические электронные интеграторы и т.п. Все многообразие этих приборов можно разделить на две большие группы. Первая из них — небалансные (некомпенсационные, неуравновешенные) системы, вторая — балансные (компенсационные, уравновешенные). *Небалансная система* непосредственно измеряет

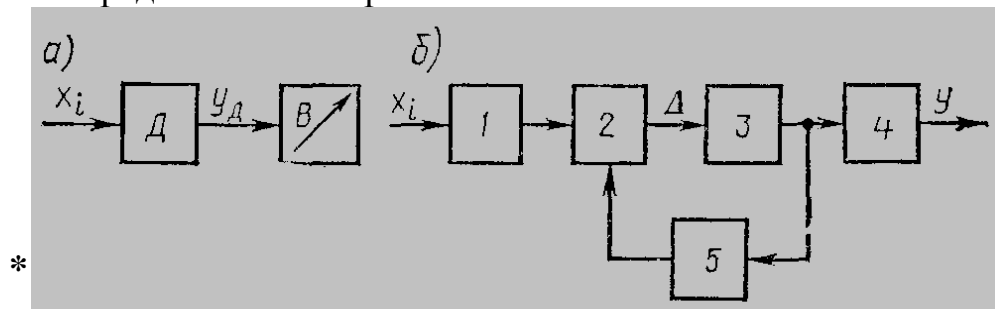


Рис. 15. Схемы включения вторичных приборов

или записывает выходной сигнал датчика (рис. 15, а), где D — датчик, а B — воспроизводящий орган. Таким образом используются пирометрические милливольтметры, которые работают совместно с термосопротивлениями или другими датчиками, выходной величиной которых является напряжение. В качестве воспроизводящего органа в таких приборах иногда используется логометр, что уменьшает ошибку от колебаний напряжения питания.

Наиболее точными системами измерения являются *балансные системы* (или компенсаторы). Они построены на принципе балансирования (уравновешивания) выходной величины датчика равной ей величиной (рис. 15, б). Сигнал с датчика / поступает в орган 2, в котором он сравнивается с сигналом уравновешивающего органа 5. Если рассогласование D равно нулю, то система находится в покое. Если изменяется контролируемый параметр X_i то появляется рассогласование ($\Delta \neq 0$), которое действует через усилитель 3 на балансирующий орган 5 до тех пор, пока сигнал рассогласования не станет равным нулю. При этом воспроизводящее устройство 4 регистрирует измеряемую величину. Такой прибор-компенсатор является самостоятельной системой автоматического управления, а поэтому эта измерительная система обладает рядом преимуществ перед небалансной. В частности, в ней отсутствуют погрешности, вызываемые колебаниями напряжения питания, нестабильностью коэффициента усиления усилителя и др. Кроме того, эти системы обеспечивают большую выходную мощность.

Вторичные приборы классифицируются также и по виду балансируемой величины. Если балансная система уравновешивает деяние ЭДС от внешнего магнитного поля. Выходное напряжение усилителя 3 поступает на фазочувствительный усилитель мощности, построенный на двойном триоде.

На выход усилителя включена управляющая обмотка w_y балансирующего двигателя. **Аноды** двойного триода питаются противоположным по фазе напряжением от сети переменного тока. Каждая половина лампы открыта лишь при положительном напряжении на аноде. Если $AU = 0$, то на сетки триода не подается напряжение и тогда в обмотке w_y отсутствует переменный ток, а есть лишь постоянная составляющая и четные гармоники, которые не создают вращающего момента. Когда появляется напряжение небаланса D U т[^]О, то в обмотке w_y появляется ток основной частоты 50 Гц. В зависимости от знака AU магнитный поток в обмотке w_y опережает или отстает по фазе на 90° от потока в обмотке возбуждения w_B двигателя 4, питающейся от сети. Это обеспечивается за счет включения емкости C в цепь обмотки тсⁱ_В. В результате взаимодействия этих потоков создается вращающий момент и двигатель 4 вращается в таком направлении, при котором осуществляется компенсация. Вал двигателя через механическую передачу связан с движком реохорда. Движок реохорда, в свою очередь, перемещает стрелку или перо записывающего устройства. В момент компенсации, т. е. при $AU = 0$, двигатель останавливается. Каждому положению движка реохорда соответствует положение указателя на шкале или пера на диаграмме, которые градуированы в единицах измеряемой величины.

В измерительной схеме потенциометра предусмотрены переключатель I , механическая связь M между двигателем 4 и движком реостата RJ и нормальный элемент $HЭ$ для корректировки тока в реохорде, а также никелевое сопротивление RNI для компенсации изменения температуры холодных спаев термопары. Для устранения влияния внешнего магнитного поля на входе потенциометра включают фильтр, состоящий из емкости C_{ϕ} и сопротивления R_{ϕ} .

Автоматические электронные потенциометры обладают высоким быстродействием и точностью, которая обусловлена тем, что прибор, по существу, является астатической системой автоматического управления.

Промышленность выпускает различные модификации потенциометров. К показывающим приборам относятся потенциометры марки КВП1, к записывающим — КСП1, КСП2. Погрешность измерения составляет 0,25—1 % от предела измерения; погрешность записи не превышает 1 % от предела измерения. Многоточечные автоматические потенциометры позволяют регистрировать измеряемые параметры процесса в 12 точках. Время прохождения указателем шкалы для различных типов приборов составляет от 1 до 16 с.

Электронные автоматические мосты. Автоматические уравновешенные мосты выполняются не только в виде астатических (или значительно реже статических) систем непрерывного балансирования, но и в виде систем с развращающим преобразованием, т. е. систем периодического балансирования. На рис. 16 показана функциональная схема автоматического моста непрерывного балансирования.

Принцип действия такого моста имеет много общего с работой автоматического потенциометра. Основными узлами моста являются измерительная схема 1, усилитель переменного тока 2 и балансирующий двигатель 3. Мост питается от сети переменного тока. Измерительная схема 1 выполняется в виде мостовой схемы, в одно из плеч которой включен датчик сопротивления D (термосопротивление, резисторный датчик или тензодатчик). При изменении контролируемого параметра в диагонали моста между точкой a и движком реохорда P появляется переменное напряжение небаланса ΔU , которое усиливается усилителем 2, состоящим из нескольких каскадов усиления напряжения и фазочувствительного усилителя мощности, на входе которого включена обмотка управления w_y балансирующего двигателя 3. Вал двигателя через механическую передачу связан с движком реохорда и указателем или самописцем.

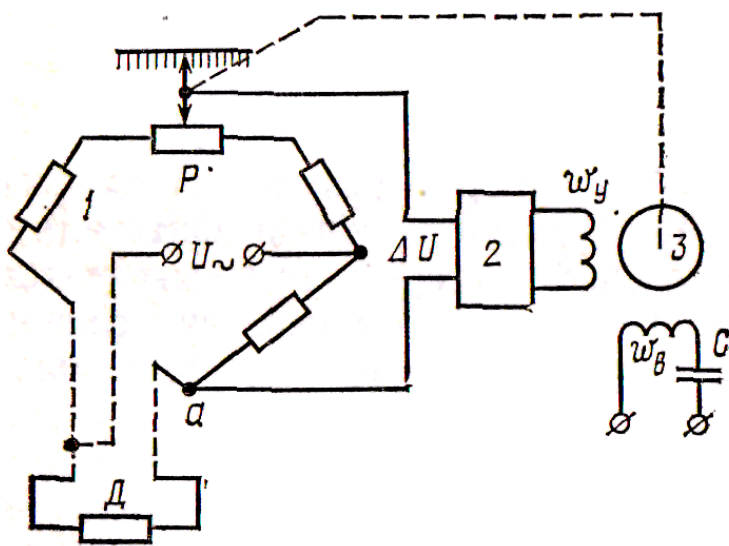


Рис. 16 - Функциональная схема автоматического моста непрерывного балансирования.

Балансирующий двигатель перемещает движок реохорда до тех пор пока не уравновесит мост.

В зависимости от того, увеличивается или уменьшается контролируемая величина, изменяется фаза напряжения небаланса на 180° и, следовательно, направление вращения двигателя.

Описанный мост, подобно автоматическому потенциометру, является астатической системой управления. На точность измерения в такой системе изменения напряжения питания и коэффициента усиления усилителя не влияют. Практически на погрешность влияет наличие зоны нечувствительности. Такой мост можно выполнить одноточечным и многоточечным с регистрацией показаний на одной диаграмме, при этом в систему вводится специальный многоточечный переключатель, поочередно подключающий датчики. Возможно использование в таких приборах мостовых схем с емкостными или индуктивными сопротивлениями. При этом двигатель 3 будет перемещать сердечник индуктивного или пластину емкостного датчика.

Схема автоматического моста с развертывающим преобразованием показана на рис. 40. Измерительным устройством служит мостовая схема, в которую включены круговой реохорд P и датчик D (термосопротивление или резисторный), управляемый первичным измерителем $ПИ$. К одной из диагоналей моста подводится постоянное напряжение U . Движок K реохорда вращается двигателем $Дв$ через редуктор $Ред1$, при этом напряжение компенсации U_K изменяется по пилообразной кривой (рис. 17). Напряжение рассогласования $\Delta U = U_K - U_d$ поступает на вход усилителя 2 на выход которого включены обмотки электромагнитов $\mathcal{E}M_1$ и $\mathcal{E}M_2$ записывающего устройства. В момент компенсации $\Delta U = U_K - U_d = 0$ обмотки электромагнитов обесточиваются и печатающая планка $Пл$ прижимает

до нулевой линии диаграммы соответствует измеряемой величине и пропорционально углу поворота движка реохорда. Диаграммная лента перемещается двигателем $Dв$ через редуктор $Ред2$.

The graph shows two voltage signals over time t . The vertical axis is labeled U and the horizontal axis is labeled t . A sawtooth wave, labeled U_K , has a constant slope and resets to zero at regular intervals. A smooth curve, labeled U_d , starts at the origin and increases monotonically, approaching the peak voltage of the sawtooth wave.

от предела измерения. Для разных типов приборов время оборота оси реохорда составляет от 1 до 16 с, а потребляемая мощность — от 15 до 60 В·А; ширина диаграммной ленты 100—500 мм. Вторичные приборы питаются от сети частотой 50 Гц, напряжением 220 В.

4. Усилительные устройства

Усилительные устройства в системах автоматики служат для усиления сигнала рассогласования по мощности или напряжению до величины, необходимой для управления исполнительными устройствами.

В системах автоматики применяются все типы усилителей — неэлектрические и электрические усилители. К неэлектрическим относятся механические, гидравлические и пневматические усилители, а к электрическим — электронные, полупроводниковые, ионные, магнитные, электромашинные усилители. При выборе усилителя учитывается мощность входного и выходного сигналов, т. е. оценивается требуемый коэффициент усиления по мощности. Кроме того, оцениваются параметры источника сигнала — измерительной схемы (выходное сопротивление, значение остаточного напряжения и т. п.) и параметры нагрузки (характер нагрузки, входное сопротивление, потребляемая мощность и т. п.). К усилителям в системах автоматики предъявляются следующие требования: постоянная времени должна быть минимальной, характеристика должна быть близка к линейной, зона нечувствительности не должна превышать допустимую.

Таблица 3.

Класс усилителя	Вид усилителя	Коэффициент усиления по мощности	Постоянная времени, с
Электронные лампы	Постоянного тока	$10^4—10^6$	$10^{-8}—10^{-9}$
	Переменного »	$10^6—10^8$	$10^{-3}—10^{-5}$
Полупроводниковые	Постоянного »	$10^3—10^5$	$10^{-6}—10^{-7}$
	Переменного »	$10^4—10^6$	$10^{-3}—10^{-6}$
Магнитные	С выходом на постоянном токе	$10^3—10^4$	$10^{-1}—10^{-2}$
	С выходом на переменном токе	$10^4—10^5$	$10^{-2}—10^{-3}$
Электромашинные (ЭМУ)	Обычный генератор	$10^2—10^3$	$1,0—10^{-2}$
	Генератор с самовозбуждением	$10^2—10^4$	$5 \cdot 10^{-1}—10^{-1}$
	ЭМУ с поперечным полем	$10^4—5 \cdot 10^5$	$10^{-1}—10^{-3}$
Гидравлические	Дроссельные	$10^4—10^6$	$10^{-1}—10^{-2}$
	Струйные	$10^3—10^4$	$10^{-2}—10^{-3}$
Пневматические	Дроссельные	$10^5—10^7$	$10^{-2}—10^{-3}$
	Струйные	$10^3—10^5$	$10^{-3}—10^{-4}$

В табл. 3 приведены основные характеристики некоторых классов усилителей. Выбор того или иного усилителя для системы автоматического управления обусловлен типами измерительного и исполнительного устройств. Если для управления исполнительным устройством необходим сигнал переменного тока, а измерительное устройство работает на

постоянном токе, то в усилительное устройство включается модулятор. Примером простейшего модулятора является вибропреобразователь, работа которого рассмотрена выше при описании принципа действия автоматического потенциометра.

В системах автоматизированного привода конечным каскадом усилительного устройства является усилитель мощности. При управлении двигателем в качестве усилителя мощности часто используются электромашинные или магнитные усилители. При этом в схему усилителя включается демодулятор. Демодулятор преобразует сигнал переменного тока в сигнал постоянного тока, который питает обмотки управления двигателя. Поэтому усилительные устройства не только усиливают сигнал, поступающий с измерительного устройства, но и преобразуют его для управления исполнительным устройством. Рассмотрим особенности работы отдельных классов усилителей в системах автоматики.

Полупроводниковые усилители. Из электрических усилителей в маломощных системах наиболее перспективны полупроводниковые из-за их малых габаритов и массы, экономичности, большой

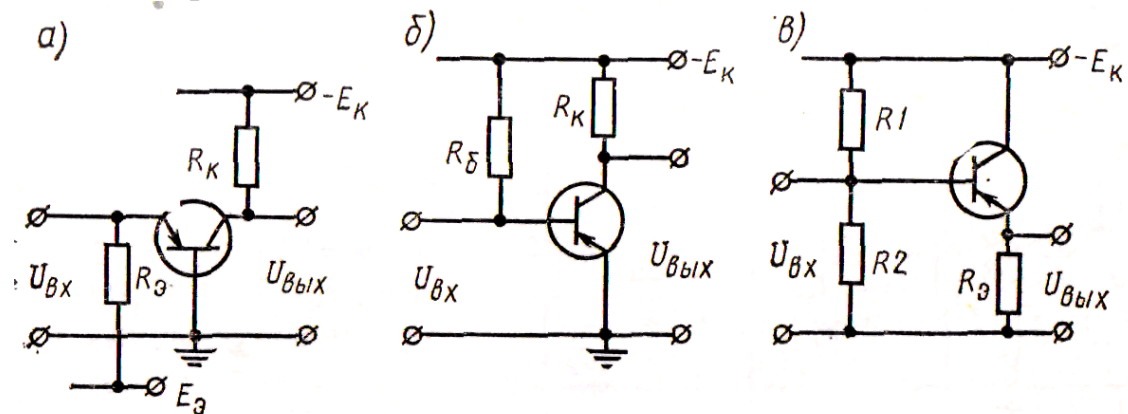


Рис. 19 - Схемы полупроводниковых усилителей

надежности и малой постоянной времени. Современная полупроводниковая техника позволяет выпускать микроэлементы в виде твердых схем, которые являются монолитным блоком и эквивалентны обычным схемам. Эти схемы обладают повышенной надежностью и комплектностью. Существенным недостатком полупроводниковых усилителей является нестабильность их параметров, в частности температурная.

Существует три схемы включения полупроводниковых усилителей: с общей базой (рис. 19, а), с общим эмиттером (рис. 19, б) и с общим коллектором (эмиттерный повторитель, (рис. 19, в). На схемах приняты следующие обозначения: R_3 , R_6 , R_K —соответственно сопротивления эмиттера, базы и коллектора; делитель напряжения R_1 , R_2 предназначен для задания потенциала базы; все схемы питаются от источника напряжения E_K .

Магнитные усилители. Магнитные усилители — это электромагнитные устройства, позволяющие получать значительные изменения мощности переменного тока посредством управления постоянным током малой мощности. Простейший дроссельный магнитный усилитель представлен на рис. 48. Усилитель состоит из сердечника, выполненного в виде О, С-, П-, Ш- или М-образной формы, на который нанесены две половины рабочей обмотки w_{\sim} и обмотка управления w_y постоянного тока; последовательно с рабочей обмоткой в цепь переменного тока включено нагрузочное сопротивление R_H . Изменение мощности в нагрузке значительно превышает мощность управления постоянным током P_y . Принцип действия магнитного усилителя основан на нелинейном характере кривой намагничивания ферромагнитных материалов.

Статическая характеристика — зависимость тока I_H в цепи переменного тока от тока управления I_y — показана на рис. 21. По виду статической характеристики магнитные усилители делятся на одноктактные (нереверсивные) и двухтактные (реверсивные). Усилитель с характеристикой, показанной на рис. 21, является одноктактным. У двухтактных усилителей при изменении полярности сигнала управления фаза выходного сигнала изменяется на 180° . Реверсивный усилитель строится на базе двух или нескольких одноктактных усилителей. Принцип их действия основан на суммировании выходных напряжений или токов одноктактных усилителей.

Двухтактный усилитель, выполненный по дифференциальной схеме, показан на рис. 22, а его статическая характеристика — на рис. 23

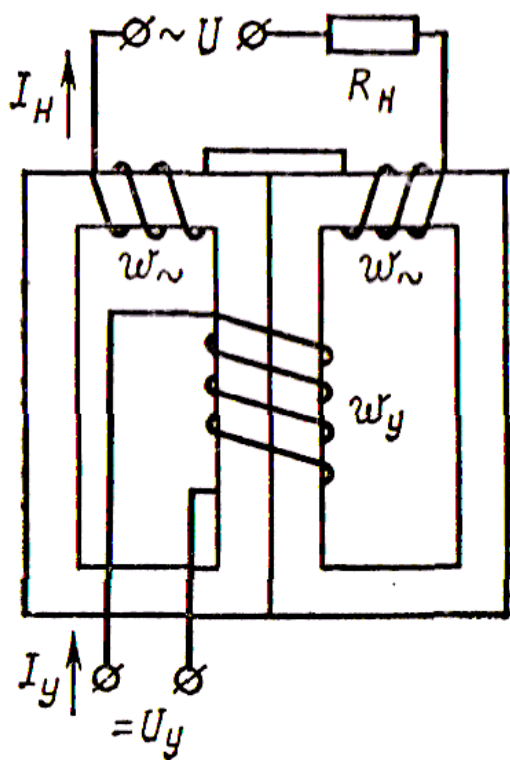


Рис. 20. Дроссельный магнитный усилитель

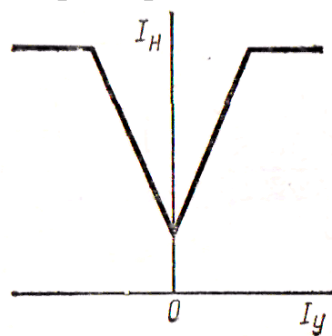


Рис. 21. Характеристика дроссельного усилителя

По способу включения обратной связи магнитные усилители классифицируют на усилители с внешней, внутренней и смешанной обратной связью; при этом используется гибкая и жесткая обратная связь. Если обратная связь внешняя, то сигнал обратной связи подается на отдельную обмотку. Для создания внутренней обратной связи последовательно с рабочими обмотками включают однополупериодные выпрямители.

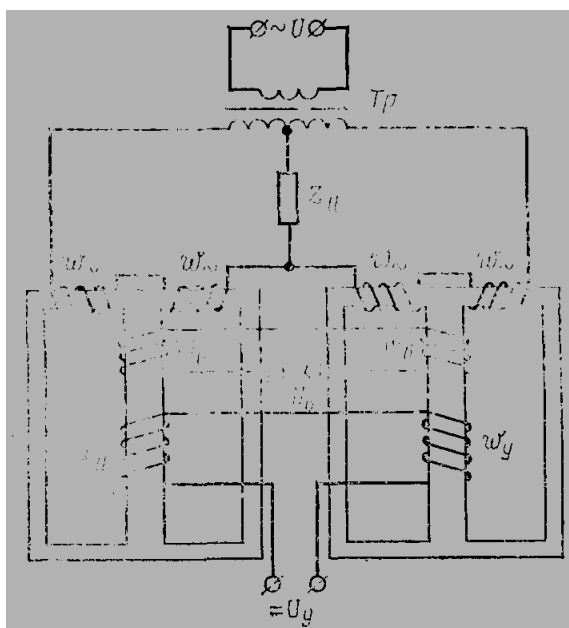


Рис. 22. Двухтактный магнитный усилитель

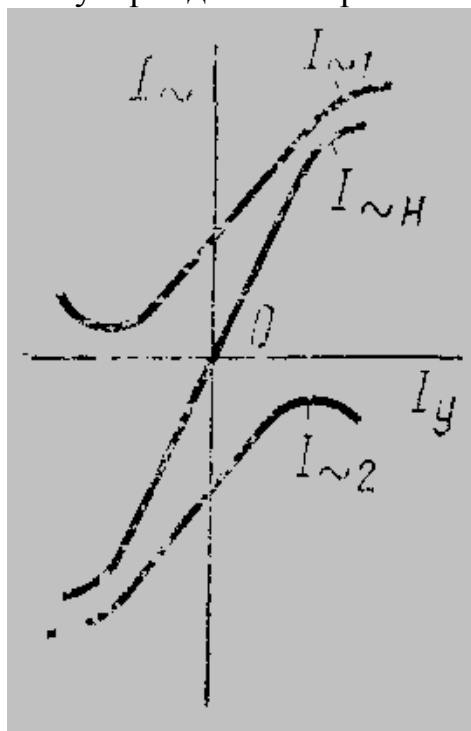


Рис. 23. - Характеристика двухтактного

Для повышения коэффициента усиления и быстродействия используется положительная обратная связь, а для улучшения стабильности статической характеристики - отрицательная обратная связь. Внешняя обратная связь возможна по току и по напряжению.

По способу включения нагрузки различают магнитные усилители с последовательным и параллельным соединением нагрузки и рабочих обмоток. В динамическом отношении магнитные усилители эквивалентны апериодическому звену с передаточной функцией

$$W(s) = k/(1 + sT).$$

Для магнитного усилителя с последовательным включением нагрузки, без обратной связи и с выходом на переменном токе (см. рис.24) параметры k и T вычисляются следующим образом.

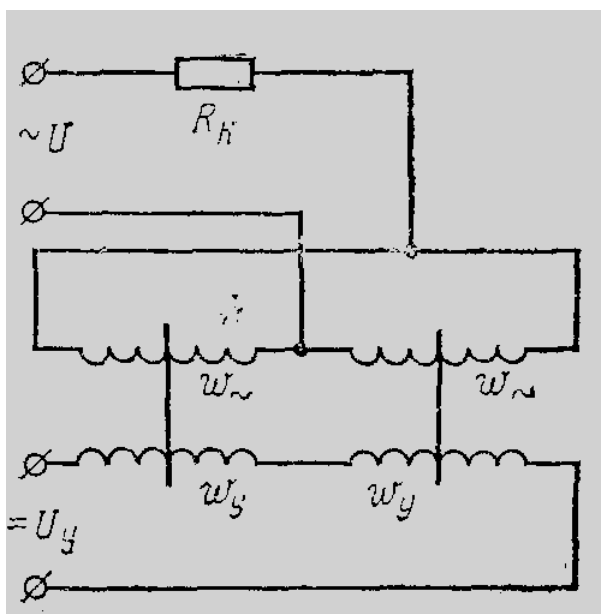


Рис. 24. Магнитный усилитель с параллельным включением нагрузки

Коэффициент k является или коэффициентом усиления по току

$$k = k_I = \Delta I_H / \Delta I_y = w_y / w_{\sim}, \quad (22)$$

или коэффициентом усиления по напряжению

$$\begin{aligned} k &= k_U = \Delta U_{\text{вых}} / \Delta U_{\text{вх}} = k_I R_H / R_y = \\ &= (w_y / w_{\sim}) (R_H / R_y), \end{aligned} \quad (23)$$

или коэффициентом усиления по мощности

$$\begin{aligned} k &= k_P = \Delta P_H / \Delta P_y = k_I k_U = \\ &= (w_y / w_{\sim})^2 (R_H / R_y), \end{aligned} \quad (24)$$

где ΔP_H — изменение мощности, выделяемой в нагрузке, Вт; ΔP_y — изменение мощности в обмотке управления, Вт. Постоянная времени

$$T = L_y / R_y = [1 / (4f)] (w_y / w_{\sim})^2 (R_H / R_y) = k_P / (4f), \quad (25)$$

где L_y — индуктивность обмотки управления, Гн; f — частота напряжения питания, Гц.

Если магнитный усилитель выполнен по схеме с параллельным включением нагрузки, без обратной связи и с выходом на переменном токе (рис. 24), то коэффициент усиления по напряжению

$$k_U = 2 (w_y / w_{\sim}) (R_H / R_y), \quad (26)$$

а постоянная времени

$$T = 2L_y / R_y. \quad (27)$$

При наличии в магнитных усилителях положительной обратной связи постоянная времени

$$T = [k_p / (4f)] (1 - k_{o.c.}), \quad (28)$$

где $k_{o.c.}$ — коэффициент положительной обратной связи.

Для усилителей средней мощности постоянная времени находится в пределах 0,05—0,1 с.

Электромашинные усилители. Электромашинные усилители (ЭМУ) используются для управления двигателями постоянного тока.

Электромашинный усилитель представляет собой генератор постоянного тока, якорь которого вращается приводным двигателем. Отличительным свойством ЭМУ является высокий коэффициент усиления по мощности. Зависимость выходного напряжения от тока управления линейна в широких пределах. К недостаткам ЭМУ можно отнести большую инерционность в сравнении с электронными усилителями, большую массу и габариты.

В зависимости от конструкции и способа возбуждения существуют следующие типы ЭМУ: с независимым возбуждением; со смешанным возбуждением; с поперечным полем. Наиболее широкую область применения имеют ЭМУ с поперечным полем. Этот тип усилителя имеет коэффициент усиления по мощности в пределах от 500 до 10^4 . В статоре располагаются обмотки возбуждения и компенсационная обмотка. Для уменьшения вихревых токов статор набирается из тонких листов. Ротор имеет две пары щеток: продольные и поперечные. Поперечные щетки замкнуты накоротко. К продольным щеткам подключается цепь нагрузки. Зазор между статором и ротором составляет примерно 0,2—0,3 мм.

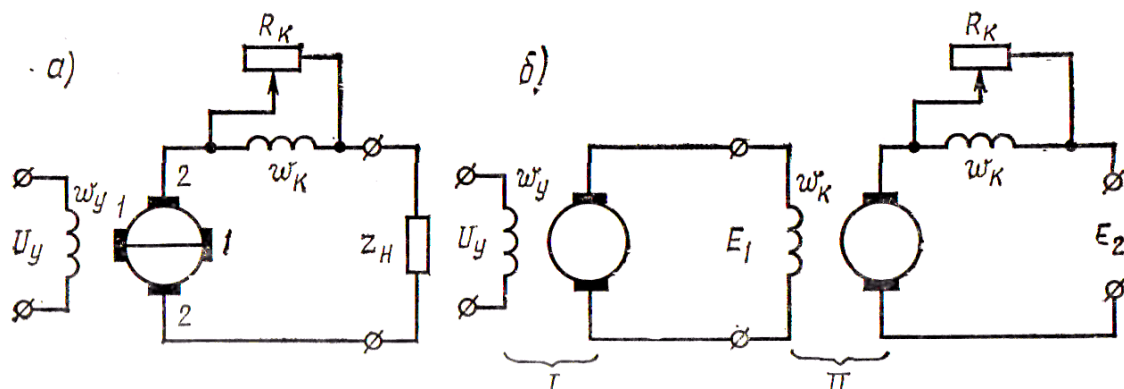


Рис. 25. Электромашинный усилитель

Принцип работы ЭМУ можно объяснить по схеме (рис. 25, а). Напряжение управления подается в обмотку возбуждения w_y , являющуюся управляемой обмоткой усилителя. Якорь ЭМУ вращается приводным двигателем с постоянной угловой скоростью ω , при этом в поперечной цепи 1—2 наводится небольшая ЭДС. Поскольку поперечные щетки замкнуты накоротко, в этой цепи проходит большой ток I_K , поток которого наводит в продольной цепи 2—2 ЭДС E_2 . Если на выходе ЭМУ включена нагрузка R_H , то в цепи проходит ток I_H , который создает поток реакции якоря Φ_H . Для компенсации этого потока в ЭМУ размещена компенсационная обмотка w_K .

создающая поток Φ_k , равный Φ_n и противоположный по направлению. Поэтому ЭМУ условно можно рассматривать как двухкаскадный усилитель: каскад I — от цепи управления к поперечной цепи, каскад II — от поперечной цепи к продольной (рис.25, б).

5. Исполнительные устройства

Исполнительные устройства служат для преобразования управляющего сигнала в перемещение органа регулирования. По характеру перемещения органа регулирования исполнительные устройства делятся на прямоходные и поворотные. По виду потребляемой энергии их классифицируют на гидравлические, пневматические, электрические и комбинированные (электрогидравлические и электропневматические).

Наиболее надежными являются гидравлические устройства. Конструктивно **гидравлические** и пневматические устройства выполняются в виде мембранных, поршневых, сильфонных и лопастных. Эти устройства позволяют получать наибольшее усиление для перемещения **органа** регулирования.

К электрическим исполнительным устройствам относятся электромагнитные и электромашинные устройства.

Электромагнитные исполнительные устройства. К устройствам этого типа можно отнести реле, соленоиды и электромагнитные **муфты**. Реле — элемент, в котором изменение входной электрической величины преобразуется в механическое перемещение, замыкающее или размыкающее контакты. Электромагнитные реле применяются для переключения мощных цепей тока, и их называют *контакторами*. Соленоид представляет собой катушку, внутри которой расположен якорь (рис.26). При подаче напряжения на катушку якорь приходит в движение и перемещает орган регулирования.

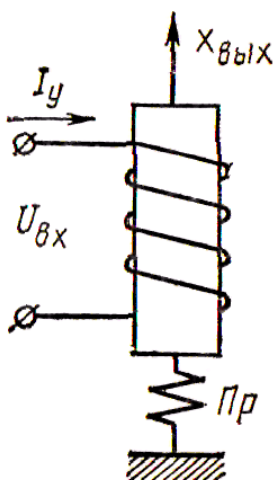


Рис. 26. - Схема соленоида

Передаточная функция соленоида

$$W(s) = X_{\text{вых}}(s)/U_{\text{вх}}(s) = \\ = k_c / [(1 + sT_\theta)(1 + T_2s + T_1^2s^2)], \quad (29)$$

где $X_{\text{вых}}$ — перемещение якоря; $T_\theta = L_0/R_0$ — постоянная времени электромагнита; R_0, L_0 — активное сопротивление и индуктивность катушки, соответствующие начальному положению якоря;

$T_1 = \sqrt{m/k}$ (m — масса подвижных частей; k_n — жесткость пружины); $T_2 = k/k_n$ (ξ_d — коэффициент демпфирования); $k_z = 2k_0I/(k_nR_0)$ — коэффициент передачи соленоида (k_0 — коэффициент пропорциональности между силой электромагнита и током I_v в катушке).

В системах автоматики применяются электромагнитные муфты следующих типов: фрикционные, порошковые и муфты скольжения. Фрикционная муфта состоит из двух полумуфт, насаженных на ведущий и ведомый валы. Когда на обмотку ведущей полумуфты подается напряжение, то полумуфты притягиваются и благодаря трению между ними ведомая полумуфта приводится во вращение, перемещая орган регулирования. В муфтах скольжения момент вращения на ведомый вал передается за счет магнитного поля, создаваемого обмоткой, расположенной на ведущей полумуфте, при вращении которой в ведомой полумуфте индуцируется ток; при взаимодействии двух магнитных полей возникает момент вращения, приводящий в движение ведомый вал.

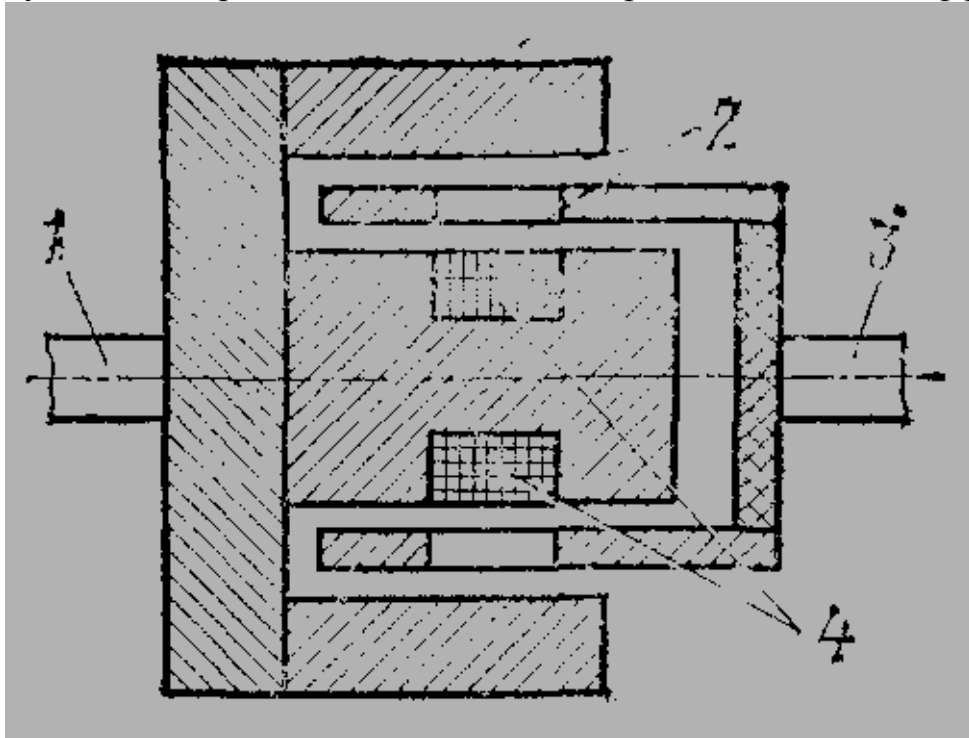
Принцип действия порошковой муфты (рис. 27) основан на изменении вязкости ферромагнитного порошка, заполняющего зазоры 2 муфты. Ведущий вал 1 муфты связан с приводным двигателем, а ведомый вал 3 — с органом регулирования. Когда на обмотку управления 4 подается напряжение U_y , то изменяется вязкость ферромагнитной массы и возникает сила сцепления между полумуфтами, а ведомый вал приходит во вращение, перемещая орган регулирования. При увеличении потока управления возрастает и момент, передаваемый на ведомый вал. При отсутствии тока в обмотке управления момент не равен нулю и составляет примерно сотые доли максимального момента, т. е. $M_0 = 0,01M_{\text{тах}}$. Это объясняется наличием трения в порошке. Для получения линейной характеристики между моментом и током управления $M = k_y I_y$

создают начальный ток подмагничивания.

Передаточная функция порошковой муфты

$$W(s) = \omega(s)/U_y(s) = \\ = k_m / [(T_y s + 1)(T_m s + 1)], \quad (30)$$

где ω — угловая скорость ведомого вала, рад/с; k_H — коэффициент



передачи

Рис. 27. - Схема порошковой муфты

муфты, определяемый по характеристике $M = f(I_y)$; $T_y = L_y/R_y$ — постоянная времени обмотки управления;

L_y и R_y — индуктивность и активное сопротивление обмотки управления; $T_m = J\omega_{ном}/M_{ном}$ — механическая постоянная времени (J — суммарный момент инерции, приведенный к ведомому валу).

Если за выходную величину принять угол поворота θ ведомого вала, то передаточная функция будет иметь вид

$$W(s) = k_m / [s(1 + T_y s)(1 + T_m s)], \quad (31)$$

так как

$$\theta = \int_0^t \omega dt.$$

Значение постоянной времени T_m находится в пределах 0,03—0,25 с.

Электромашинные исполнительные устройства. К электромашинным исполнительным устройствам относятся двигатели различных типов. Двигатели постоянного тока обычно используются в системах совместно с электромашинными усилителями. Для управления такими двигателями применяют тиристорные преобразователи. Двигатели переменного тока применяют в системах автоматического управления различных классов благодаря их простоте, надежности, малой инерционности и удобства управления.

Из двигателей постоянного тока используют двигатели с независимым возбуждением (рис. 28). Управление может осуществляться или со стороны

якоря (якорное управление), или со стороны обмотки возбуждения OB (полюсное управление). Соответствующие схемы приведены на рис. 29, а, б. Изменение направления и частоты вращения двигателя осуществляется при полюсном управлении путем изменения полярности и значения напряжения $U_{BX} = U_n$ (рис. 29, а), при этом на якорь через балластное сопротивление r_b подают постоянное напряжение U_b . При управлении со стороны якоря на обмотку возбуждения OB подают постоянное напряжение U_b а управление скоростью и направлением движения осуществляется путем изменения значения и полярности напряжения на якоре: $U_{ax} = U_a$ (рис. 29, б). Полюсное управление по сравнению с якорным позволяет уменьшить мощность усилительного устройства, но при этом ухудшается быстродействие. Поэтому управление со стороны якоря используется шире.

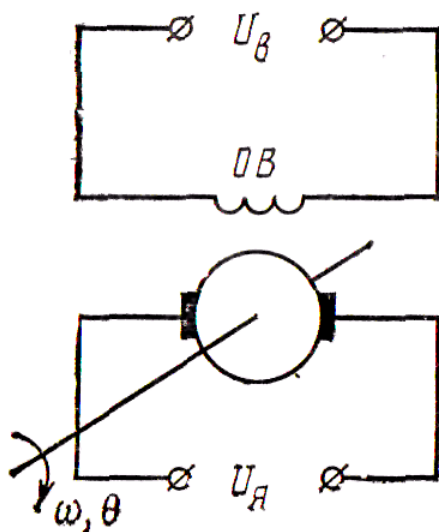


Рис. 28. Двигатель постоянного тока

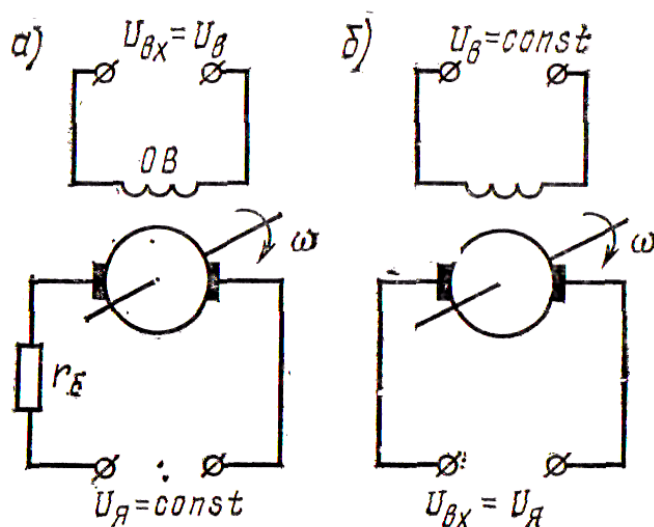


Рис. 29. Схемы управления двигателями постоянного тока

Вращающий момент $M_{вр}$ двигателя возникает вследствие взаимодействия полей возбуждения и якоря; он пропорционален потоку возбуждения Φ_b и току якоря I_a :

$$M_{вр} = c_1 I_a \Phi_a, \quad (32)$$

где c_2 — коэффициент пропорциональности.

При постоянном потоке возбуждения Φ_a вращающий момент

$$M_{вр} = c'_1 I_a. \quad (33)$$

Если для простоты момент нагрузки считать равным нулю, то момент (76) расходуется на преодоление инерции якоря и приводимого в движение органа регулирования, т. е.

$$M = J (d\omega/dt), \quad (34)$$

где J — приведенный момент инерции, $\text{Н} \cdot \text{м} \cdot \text{с}^2$; ω — угловая скорость вала двигателя, рад/с .

Уравнение цепи якоря можно представить в виде

$$U_{\text{я}} = I_{\text{я}} R_{\text{я}} + L_{\text{я}} (dI_{\text{я}}/dt) + e, \quad (35)$$

где $L_{\text{я}}$, $R_{\text{я}}$ — индуктивность и активное сопротивление якоря; $e = c_2 \omega$ — противо-ЭДС.

Можно составить дифференциальное уравнение двигателя, связывающее скорость вращения вала с напряжением управления $U_{\text{я}}$:

$$T_{\text{я}} T_{\text{м}} (d^2 \omega / dt^2) + T_{\text{м}} (d\omega / dt) + \omega = k_{\text{д}} U_{\text{я}}, \quad (36)$$

где $T_{\text{я}} = L_{\text{я}} / R_{\text{я}}$ — постоянная времени цепи якоря; $T_{\text{м}} = IR_{\text{я}} / (c_1 c_2) = I (\omega_{\text{хх}} / M_{\text{п}})$ — электромеханическая постоянная времени ($\omega_{\text{хх}}$ — угловая скорость холостого хода, рад/с; $M_{\text{п}}$ — пусковой момент двигателя); $k_{\text{д}} = 1/c_2 = \omega_{\text{хх}} / U_{\text{яном}}$ — коэффициент передачи.

На основании уравнения передаточная функция двигателя имеет вид

$$W(s) = \omega(s) / U_{\text{я}}(s) = k_{\text{д}} / (T_{\text{я}} T_{\text{м}} s^2 + T_{\text{м}} s + 1). \quad (37)$$

Если за выходную величину двигателя считать угол поворота вала θ , то

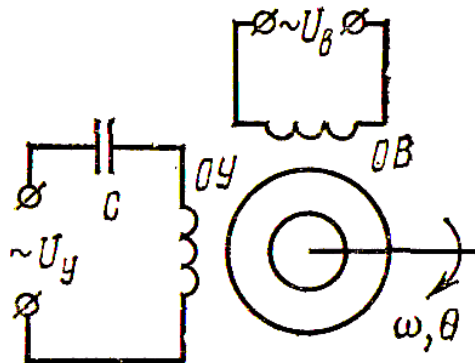
$$W(s) = \theta(s) / U_{\text{я}}(s) = k_{\text{д}} / [s (T_{\text{я}} T_{\text{м}} s^2 + T_{\text{м}} s + 1)]. \quad (38)$$

Постоянная цепи якоря, находится в пределах $(2 \div 5) \cdot 10^{-3}$ с, а электромеханическая постоянная — $(2 \div 15) \cdot 10^{-2}$ с. Так как

$T_{\text{я}} < T_{\text{м}}$ более чем на порядок, то передаточная функция принимает вид

$$W(s) = \omega(s) / U_{\text{я}}(s) = k_{\text{д}} / (T_{\text{м}} s + 1). \quad (39)$$

При этом двигатель эквивалентен апериодическому звену. Из двигателей переменного тока наиболее распространен индукционный двухфазный



двигатель (рис. 30).

Рис. 30. - Индукционный двухфазный двигатель

Эти двигатели имеют две обмотки статора: обмотку возбуждения OB и обмотку управления OU . Обмотки статора расположены в пространстве под углом

90° . Обмотка возбуждения питается от сети переменного тока, а обмотка управления — от усилителя системы, причем это напряжение сдвинуто по фазе относительно напряжения возбуждения на $\pm 90^\circ$. Амплитуда напряжения возбуждения постоянна, а амплитуда напряжения управления изменяется в зависимости от изменения управляемой величины. Ротор такого двигателя выполняется в виде полого тонкостенного алюминиевого или

стального цилиндра или в виде беличьей, клетки. При взаимодействии эллиптического поля, создаваемого обмотками статора, с полями вихревых токов, наводимых в роторе, возникает момент, приводящий в движение ротор, который перемещает регулирующий орган системы.

В динамическом отношении асинхронный двухфазный двигатель рассматривается как апериодическое звено, если за выходную величину принята угловая скорость вала двигателя, т. е. его передаточная функция описывается формулой. Если же выходной величиной является угол поворота θ вала двигателя, то

$$W(s) = k_d / [s(1 + sT_m)]. \quad (40)$$

В табл. 4 приведены постоянные времени двухфазных двигателей в зависимости от типов ротора.

Таблица 4.

Тип ротора	Частота тока в цепи возбуждения, Гц	Постоянная времени, с
Полый немагнитный	50 400—500	0,006—0,03 0,025—0,1
Полый стальной	50 400—500	1—2 1,5—3,0
Полый типа беличьей клетки	50 400—500	0,1—0,2 0,3—1,5

6. Агрегатные унифицированные системы

При решении задач автоматизации технологических процессов экономически более выгодно использовать универсальные регуляторы общепромышленного назначения, нежели в каждом отдельном случае создавать специализированные устройства. Универсальные регуляторы благодаря конструктивным особенностям могут применяться для управления различными процессами. Проектирование и производство таких систем рекомендовано Государственной системой промышленных приборов и средств автоматизации (ГСП). В состав ГСП входят устройства получения, преобразования и представления информации о ходе процесса, устройства передачи этой информации, а также исполнительные устройства, вырабатывающие необходимые воздействия на объект. При этом в ГСП предусмотрена унификация входных и выходных сигналов элементов и блоков системы, благодаря чему обеспечивается энергетическое и конструктивное сопряжение при создании агрегатных комплексов с широкими функциональными возможностями. По способу подключения датчика различают три принципа построения регулятора: приборный, аппаратный и агрегатный.

Регуляторы приборного типа получают входной сигнал от специальных вторичных датчиков-преобразователей, встроенных в соответствующий измерительный прибор.

Регуляторы аппаратного типа получают входные сигналы непосредственно от первичных датчиков. Входное устройство при этом выполняется сменным, что дает возможность включать различные типы датчиков.

Агрегатный принцип построения реализуется при стандартных сигналах на выходе датчиков; он дает возможность создавать различные системы из взаимозаменяемых блоков, предусмотренных номенклатурой ГСП.

В промышленных регуляторах используется три вида энергии: электрическая, гидравлическая и пневматическая. В соответствии с этим регуляторы классифицируют на электрические, гидравлические и пневматические. Каждый из них имеет определенные преимущества и недостатки.

Электрические регуляторы обеспечивают возможность передачи сигналов на большие расстояния, что очень важно, так как в промышленных условиях датчики и органы регулирования могут быть разнесены территориально.

Для гидравлических и пневматических регуляторов характерны значительные мощности, большие скорости и высокая надежность исполнительных механизмов, а также взрыво- и пожаробезопасность.

Иногда используются комбинированные регуляторы: электрогидравлические или электропневматические. В таких регуляторах датчики и измерительные устройства выполняют электрическими, а исполнительные элементы — гидравлическими или пневматическими.

Регуляторы различных типов воспроизводят одни и те же стандартные законы регулирования: пропорциональный (П), интегральный (И), пропорционально-интегральный (ПИ), пропорционально-дифференциальный (ПД), пропорционально-интегриродифференциальный (ПИД).

ПИД — закон регулирования, описывающийся следующим уравнением:

$$y = k_0 x + k_1 \int_0^t x dt + k_2 (dx/dt), \quad (41)$$

где y — выходная величина регулятора; x — входная величина; k_0, k_1, k_2 — постоянные коэффициенты.

При отсутствии каких-либо составляющих в законе регулирования формируются другие законы, упомянутые выше.

При $k_1 = k_2 = 0$ получим закон П регулирования:

$$y = k_0 x; \quad (42)$$

при $k_1 = k_2 = 0$ — закон И:

$$y = k_1 \int_0^t x dt; \quad (43)$$

при $k_1 = 0$ — закон ПД:

$$y = k_0 x + k_2 (dx/dt); \quad (44)$$

при $k_2 = 0$ — закон ПИ:

$$y = k_0 x + k_1 \int_0^t x dt. \quad (45)$$

При настройке регулятора коэффициенты k_0 , k_1 , k_2 могут изменяться. Практически регулятор воспроизводит законы. В частности, в передаточной функции регулятора при исследовании динамики нужно учитывать инерционность элементов схемы. В регуляторах помимо линейных законов используются и нелинейные. Применяется двух- или трехпозиционное регулирование.

В случае двухпозиционного регулятора закон регулирования имеет вид

$$y = \begin{cases} y_{\max} & \text{при } x > 0; \\ y_{\min} & \text{при } x < 0. \end{cases} \quad (46)$$

Для трехпозиционного регулятора закон регулирования имеет вид

$$y = \begin{cases} y_{\max} & \text{при } x > a; \\ 0 & \text{при } -a < x < a; \\ y_{\min} & \text{при } x < -a, \end{cases} \quad (47)$$

где a — зона нечувствительности регулятора.

ПРОГРАММНЫЕ СРЕДСТВА АВТОМАТИЗАЦИИ ПРОЦЕССОВ НЕФТЕГАЗОВОГО КОМПЛЕКСА

1. Потоки информации в автоматизированной системе управления предприятием. Общие сведения.

В настоящее время получили распространение различного рода автоматизированные системы. Каждая из таких систем реализует специфический набор задач и функций, которые как раз и определяют структуру системы, аппаратные средства и программные средства этой системы (рис.1.1.).

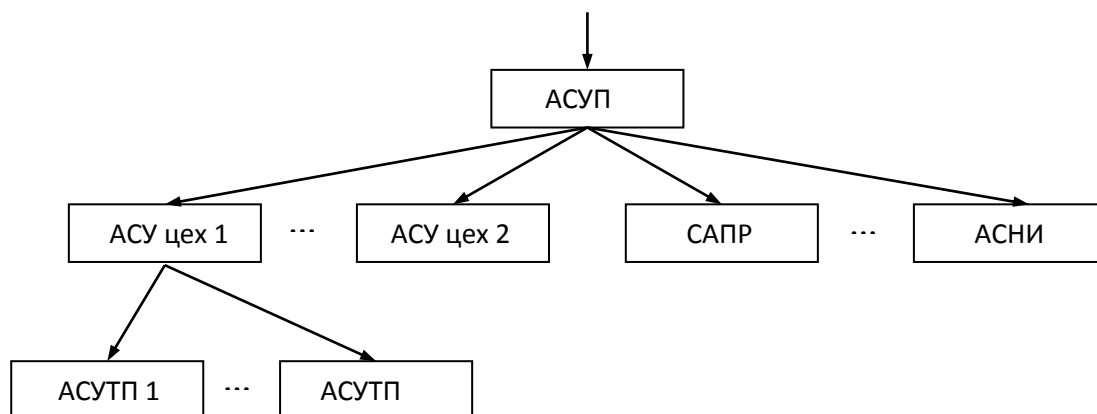


Рис.1.1 Архитектура автоматизированная системы управления предприятием

ОАСУ — отраслевая автоматизированная система управления;

АСУП — автоматизированная система управления предприятия;

АСНИ — автоматизированная система научных исследований;

АСУТП — автоматизированная система управления технического процесса;

СУГАЛ — система управления гибкой автоматизированной линии;

СУГАП — система управления гибкого автоматизированного производства.

В рамках каждого типа автоматизированных систем обычно можно выделить целый ряд так называемых функциональных подсистем, которые реализуют конкретные задачи по управлению объектом (рис.1.2.).

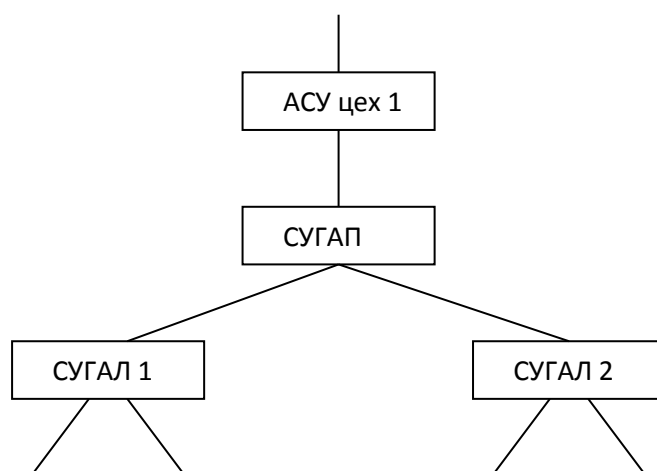


Рис.1.2. Функциональная подсистема АСУП

Например, в рамках АСУП можно выделить подсистемы — ОКП, БУ, УК, МТС, СиРП, УиОП.

ОКП — оперативно календарного планирования

БУ — бухгалтерского учета

УК — учета кадров

МТС — материально технического снабжения

СиРП — сбыта и реализации продукции

УиОП — управления и оперативного планирования

В рамках АСУТП выделяются подсистемы — СиПОИ, РиСП, ОР.

СиПОИ — сбора и первичной обработки информации

РиСП — регулирования и стабилизации параметров

ОР — организации режимов

Кроме функциональных подсистем имеют место обеспечивающие подсистемы. С их помощью обеспечивается нормальное функционирование системы.

Любая информационная система содержит информационное и программное обеспечение.

Каждая из функциональных подсистем использует ту или иную информацию об объекте (предметной области) в ранее создаваемых автоматизированных системах эта информация хранилась обычно в виде массивов данных, которые и использовались программами той или иной функциональной подсистемой.

Более перспективным в настоящее время считается подход когда в рамках автоматизированной системы выделяется специальная обеспечивающая подсистема (ИО), в которой концентрируется вся информация и из которой функциональные подсистемы могут получать данные необходимые для ее функционирования. Ядром такой обеспечивающей подсистемы (ИО) являются автоматизированные банки данных (АБД).

Рассмотри АСУП предприятия с выделением подсистемы. Для такой системы имеют место следующие особенности:

1. Преобладают задачи или функции связанные с большим количеством расчетов.
2. Структура хранимой информации не отличается большой сложностью, но объемы ее велики.
3. Имеет место жестко заданный перечень запросов, которые определяются функциями автоматизированной системы.
4. Информационная подсистема должна формировать документы и отчеты с заданными формами.

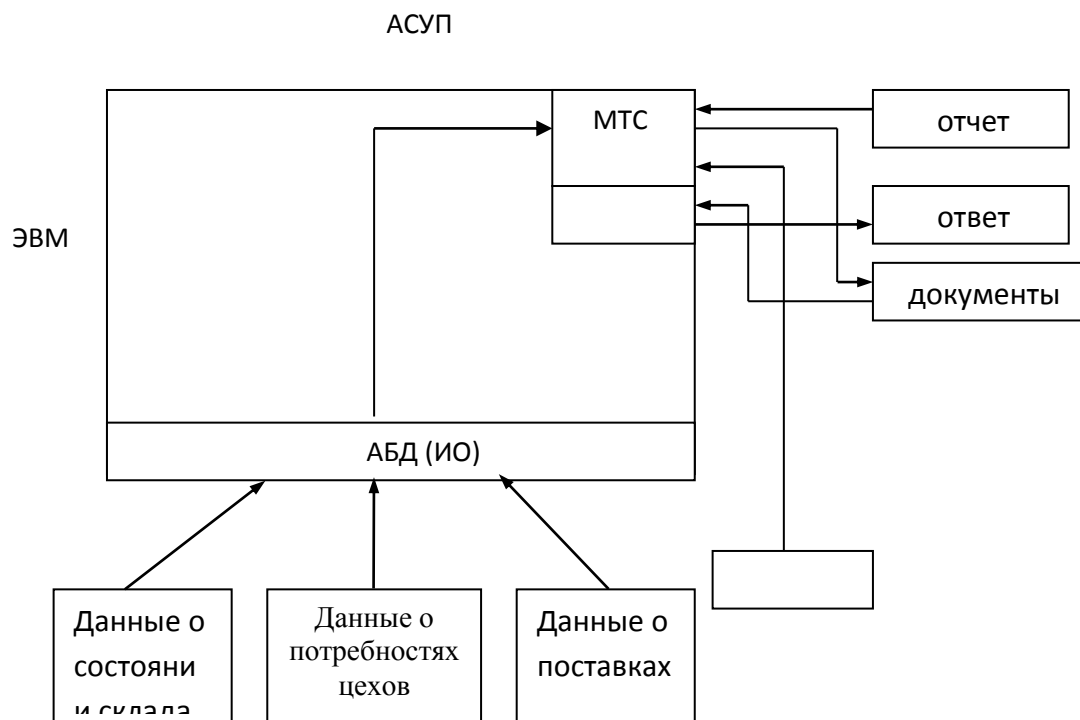


Рис.1.3. АСУП с выделением подсистемы



Рис.1.4 Потоки данных в АС

Спецификой такого рода автоматизированной системы является следующее:

1. Должен быть обеспечен высоко оперативный режим работы в реальном масштабе времени.
2. Наличие сложных расчетов для реализации моделирования поведения объекта и выработке правильных управляющих воздействий.
3. Большой объем вычислений.

В общем случае АБД – это подсистема АС включающая в свой состав комплекс специальных методов и средств (математических, технических, программных, лингвистических и организационных) для поддержки динамической информационной модели предметной области с целью решения задач управления или обслуживания потребителей.

Обычно к АБД предъявляются требования:

1. Система должна удовлетворять информационной потребности пользователя.

2. Система должна обеспечивать заданный уровень достоверности хранящейся информации.
3. Система должна обеспечивать требуемую производительность при обработке запроса.
4. Система должна обеспечивать доступ к данным пользователей в соответствии с их полномочиями.
5. Система должна обеспечивать удобство обращения пользователя за информацией.
6. Система должна обеспечивать возможность реорганизации и расширения.

2. Обобщенная архитектура системы управления объектами добычи, подготовки и транспорта нефти и газа

На рис.2.1 представлена архитектура многоуровневой системы управления, обобщающая многочисленные применения таких систем для управления технологическими процессами нефтяной и газовой промышленности (SCADA).



Рис.2.1 Обобщенная структурная схема системы управления.

Как правило, это двухуровневые системы, и именно на этих уровнях реализуется непосредственное управление технологическими процессами. Специфика каждой конкретной системы управления определяется используемой на каждом уровне программно - аппаратной платформой.

Нижний уровень - уровень объекта (контроллерный) - включает различные датчики (измерительные преобразователи) для сбора информации о ходе технологического процесса, электроприводы и исполнительные устройства для реализации регулирующих и управляющих воздействий. Датчики поставляют информацию локальным контроллерам (PLC), которые могут обеспечить реализацию следующих функций:

- сбор, первичная обработка и хранение информации о состоянии оборудования и параметрах технологического процесса;
- автоматическое логическое управление и регулирование;
- исполнение команд с пункта управления;
- самодиагностика работы программного обеспечения и состояния самого контроллера;
- обмен информацией с пунктами управления.

Так как информация в контроллерах предварительно обрабатывается и частично используется на месте, существенно снижаются требования к пропускной способности каналов связи.

Информация с локальных контроллеров может направляться в сеть диспетчерского пункта непосредственно, а также через контроллеры верхнего уровня (см. рис.2.1). В зависимости от поставленной задачи контроллеры верхнего уровня (концентраторы, коммуникационные контроллеры) реализуют различные функции. Некоторые из них перечислены ниже:

- сбор данных с локальных контроллеров;
- обработка данных, включая масштабирование;
- поддержание единого времени в системе;

- синхронизация работы подсистем;
- организация архивов по выбранным параметрам;
- обмен информацией между локальными контроллерами и верхним уровнем;
- работа в автономном режиме при нарушениях связи с верхним уровнем;
- резервирование каналов передачи данных и др.

Верхний уровень - диспетчерский пункт (ДП) - включает одну или несколько станций управления, представляющих собой автоматизированное рабочее место (АРМ) диспетчера/оператора. Здесь же может быть размещен сервер базы данных. На верхнем уровне могут быть организованы рабочие места (компьютеры) для специалистов, в том числе и для инженера по автоматизации (инжиниринговые станции). Часто в качестве рабочих станций используются ПЭВМ типа IBM PC различных конфигураций.

Станции управления предназначены для отображения хода технологического процесса и оперативного управления. Эти задачи и призвано решать программное обеспечение SCADA, ориентированное на разработку и поддержание интерфейса между диспетчером/оператором и системой управления, а также на обеспечение взаимодействия с внешним миром.

Все аппаратные средства системы управления объединены между собой каналами связи. На нижнем уровне контроллеры взаимодействуют с датчиками и исполнительными устройствами посредством физических линий, а с блоками удаленного и распределенного ввода/вывода - с помощью специализированных сетей.

Связь удаленных контроллеров с контроллерами верхнего уровня (концентраторами) часто реализуется по радио и телефонным каналам. В случае небольших расстояний локальные контроллеры объединяются между собой и с верхним уровнем управляющими сетями на базе витой пары, оптоволоконна.

Связь различных АРМ оперативного персонала и специалистов между собой, с контроллерами верхнего уровня, а также с вышестоящим уровнем осуществляется посредством информационных сетей (витая пара, оптоволокно).

Спектр реализаций RTU в таких системах управления достаточно широк. Конкретная реализация RTU зависит от области применения. Это могут быть промышленные компьютеры (PC-совместимые контроллеры) или программируемые логические контроллеры (PLC/ПЛК). На российском рынке представлена широкая гамма контроллеров самых различных конфигураций и назначений.

Что касается программного продукта типа SCADA, то сейчас на рынке присутствует несколько десятков открытых SCADA-пакетов, обладающих практически одинаковыми функциональными возможностями. Тем не менее, каждый SCADA-пакет является по-своему уникальным, и его выбор для конкретной системы автоматизации по-прежнему остается актуальным.

Выбор коммуникационного программного обеспечения (протоколов обмена информацией) для конкретной системы управления определяется многими факторами, в том числе и типом применяемых контроллеров, и выбранным SCADA-пакетом.

3. Программно-аппаратные средства автоматизации процессов нефтегазового комплекса

Для управления непрерывными технологическими процессами (заводскими) существуют специализированные программно-аппаратные средства, которые получили название Distributed Control Systems – DCS (распределенные системы управления - РСУ).

Если контроллеры в свое время пришли в автоматизацию для обработки дискретных сигналов, то DCS-системы изначально

предназначались для работы с аналоговыми сигналами. Именно аналоговые сигналы датчиков температуры, давления, уровня, расхода определяют состояние технологических процессов переработки нефти и газа. Одна из основных задач управления такими процессами – стабилизация большого количества технологических параметров, которые часто являются взаимозависимыми. Но это не означает, что современные DCS-системы не адаптированы для работы с дискретными сигналами.

Любая DCS-система – это система, включающая в себя все компоненты системы управления: контроллеры (управляющие процессоры), сети и интерфейсы связи, программное обеспечение станций операторов, инженеринговых станций. Все эти программно-аппаратные средства называются системой, более того, интегрированной системой, так как взаимодействие всех компонентов такой системы (и программных, и аппаратных) обеспечено фирмой - производителем. Понятно, что в этом случае можно говорить о высокой степени готовности этих средств и высокой надежности, так как это взаимодействие “оттачивается” многими годами в “лабораторных” условиях специалистами самой фирмы.

Этого не скажешь о SCADA-системах, когда созданием системы управления занимаются фирмы - системные интеграторы. Приходится “стыковать” программно-аппаратные средства разных производителей в рамках реального времени, отпущенного на разработку проекта. Для приобретения опыта (который, как известно, приходит не сразу) требуется время.

В рамках этой лекции трудно описать все тонкости и нюансы выбора того или иного подхода к созданию системы управления. Этот выбор определяется очень многими факторами и в каких-то случаях очевиден и однозначен. В других случаях возможны варианты выбора и того, и другого подхода. При этом должны быть выдвинуты дополнительные критерии, способные склонить чашу весов в ту или иную сторону.

Утверждать можно лишь следующее. Сегодня выбор DCS-системы обойдется потребителю дороже (в среднем) по сравнению с выбором PLC и SCADA. Но желание сэкономить, в свою очередь, может привести к отрицательному результату.

4. Информационное обеспечение автоматизированных систем управления нефтегазового комплекса

4.1 Информация и данные

В окружающем нас мире имеют место либо физические тела, либо физические поля. Последние находятся в состоянии непрерывного изменения, которые сопровождаются обменом энергией и выделением сигналов в окружающей среде. При взаимодействии этих сигналов с другими телами или объектами, в последних также возникают определенные изменения их свойств или состояний. Это и называется регистрацией сигнала. Эти изменения можно наблюдать или измерять. В результате этого процесса образуются данные. Данные – это зарегистрированные сигналы. Данные несут в себе информацию о событиях реального мира, однако они не тождественны информации.

Информация – это продукт взаимодействия данных и адекватным им методам.

Определение не является единственным. В настоящее время нет строгого общепризнанного определения информации, хотя имеются

различные понятия об информации. Причем оказывается, что в разных научных дисциплинах это понятие интерпретируется по-разному.

4.2 Свойства информации

Информация обладает следующими основными свойствами:

1) Динамика, т.е. информация не является статическим объектом. Она изменяется во времени и существует только в момент взаимодействия данных и метода.

2) Объективность и субъективность. Данные сами по себе являются объективными, т.к. они являются результатом регистрации объективно существующих сигналов. В тоже время методы являются субъективными. Но тот метод, который используется для обработки с данными, может быть субъективным, если другой работает с теми же данными. Более объективной принято считать информацию, в которую методы вносят меньше субъективных элементов. Так, например результат наблюдения фотоснимка какого либо объекта дает более объективную информацию, чем результат наблюдения рисунка этого же объекта.

3) Полнота информации. Полнота информации во многом характеризует качество информации и определяет достаточность данных для принятия решения.

4) Достоверность информации. Данные возникают в момент регистрации сигнала, но не все сигналы являются полезными, т.е. всегда имеет место какой-то уровень посторонних сигналов. В результате полезные данные сопровождаются определенным уровнем информационных шумов. При увеличении уровня шумов достоверность информации снижается. И в таком случае для получения того же количества информации и того же сигнала необходимо использовать либо больше данных, либо более совершенные методы.

5) Адекватность информации – это степень соответствия реальному процессу или объекту.

6) Доступность информации – это мера возможности получить ту или иную информацию. На степень доступности влияют одновременно и степень доступности к данным и доступность к адекватным методам для их интерпретации.

7) Актуальность информации – степень соответствия информации текущему моменту. Поскольку информационные процессы происходят во времени, то адекватная и достоверная, но устаревшая информация может приводить к ошибочным решениям.

4.3 Инфологический и даталогический аспекты проектирования автоматизированной информационной системы

То, что мы имеем в виде данных об объекте или о предметной области, в ЭВМ, может, при использовании определенных методов стать информацией, т.е. приобрести смысловое содержание (семантику). Если фиксация сигнала об объекте осуществляется с помощью естественного языка, то данные и их семантика фиксируются совместно, но если используется ЭВМ для отображения состояния такого объекта, то в нем обычно данные и их интерпретация разделены, например, на различные поля, это связано с двумя причинами:

1. ЭВМ не имели и не имеют достаточных возможностей для обработки текста на привычном языке.

2. Память ЭВМ конечна, и поэтому семантика возлагается на пользователя, а хранятся только данные, причем пользователь может не сам интерпретировать данные, а разработать для этого программу.

При создании АБД имеют место инфологический и даталогический аспекты проектирования.

Инфологический – употребляется при рассмотрении вопросов связанных со смысловым содержанием данных независимо от их способа представления в памяти систем; при создании АИС выделяется этап инфологического проектирования, в его рамках решаются следующие основные вопросы:

- о каких объектах или явлениях реального мира требуется накапливать и обрабатывать информацию в АИС;
- какие основные характеристики объектов и какие взаимосвязи должны быть учтены в информационной системе;
- уточняется вводимые в ИС понятия об объектах, их характеристик и взаимосвязи.

При даталогическом проектировании исходят из возможностей имеющихся средств восприятия и хранения, а также обработки данных для решения следующих вопросов:

- разработки соответствующих форм представления данных;
- при выборе моделей и метода представления и преобразования данных;
- при формулировании правил смысловой интерпретации полученных данных.

4.4 Понятия БД, системы управления БД (СУБД).

На первых периодах разработки и внедрения автоматизированных информационных систем, в их программном обеспечении использовался так называемый позадачный подход. При таком подходе, при создании сложных АС возникают проблемы:

- избыточность данных;
- сложность внесения изменений в исходные данные, так как при корректировке данных для одной программы, необходимо корректировать и для других программ;

– так как имеет место взаимосвязь между структурой данных и текстом программ, то изменение в организации или структуре данных приводит к необходимости изменения и самих прикладных программ;

– любая АС является развивающейся, а при этом внедряются новые программы, в которых если даже используется ранее применяющиеся данные, их все равно необходимо включить в новую версию программы.

Кроме них есть и другие недостатки, которые стимулировали появление новых подходов к организации данных:

1. Интеграция и централизация данных, когда все данные накапливаются, обрабатываются и хранятся централизованно, создавая динамически обновляющуюся информационную модель предметной области.
2. Обеспечение независимости прикладных программ и самих данных.

Реализация этих принципов в том или ином виде приводит к созданию единого для всех задач (прикладных) блока (базы) данных, а во-вторых – к созданию пакета управляющих программ для манипулирования этими данными. Таким образом из состава прикладных программ $ПП_1$ и $ПП_N$ АИС, используется позадачный подход в составе АИС, построенная на основе банка данных. Во-первых в БД включены сами данные $Д_1 \dots Д_N$. Фрагменты ПП, обеспечивающие манипулирование данными как бы включены в состав СУБД и, наконец, сами прикладные программы, основанные на новом принципе (рис.4.1.). Из них остаются только те, которые поддерживают алгоритм конкретной задачи.

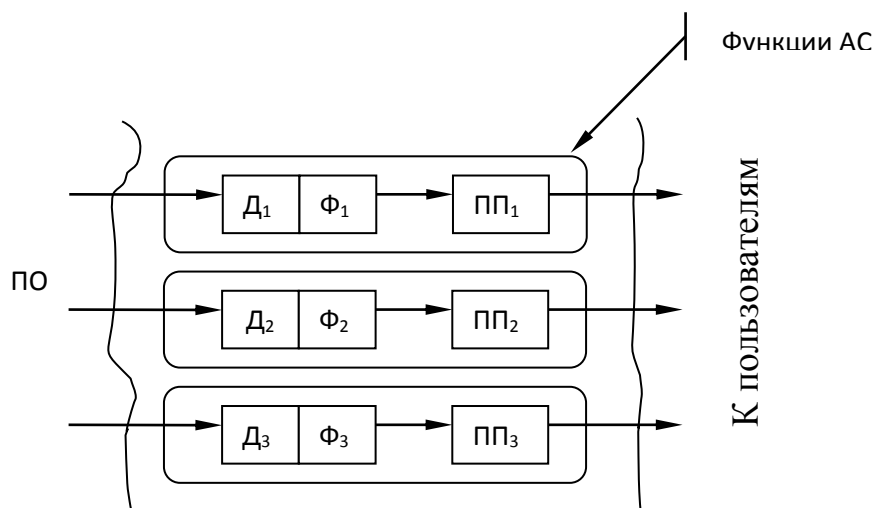


Рис.4.1.Функции информационного обеспечения АС

По сравнению с традиционными позадачными подходами, когда каждое приложение обеспечено монопольными файлами. Централизованное управление имеет следующие преимущества:

- сокращение избыточности хранимых данных, то есть в этом случае может быть обеспечена минимально необходимая избыточность хранения данных, если же выявляется факт использования несколькими ПП одних данных, то такие данные заносятся и хранятся в одном экземпляре (кроме резерва):
- устранение противоречивости хранимых данных, следствием устранения избыточности данных является устранение возможности устранения противоречивости одних и тех же данных в различных приложениях;
- многоаспектное использование данных, то есть централизованное управление в полной мере позволяет реализовать принцип однократного ввода данных и многократного использования данных;
- возможность соблюдения стандартов, в настоящее время при проектировании АИС учитываются корпоративные, ведомственные, промышленные и интернациональные стандарты;

- обеспечение возможности санкционированного доступа к данным, интеграция данных приводит к тому, что данные, использованные разными потребителями, хранятся на общих носителях и могут пересекаться самым различным образом, этим самым повышается возможность доступа к секретным данным и т.п., а также возможность непреднамеренного искажения данных;
- возможность обеспечения целостности данных, задача обеспечения данных состоит в обеспечении их правильности, точности, достоверности при хранении в БД.

4.5 Банк данных, как автоматизация системы. Компоненты БД

На рис.4.2. представлена структура АБД, такая структура действительно соответствует структуре автоматизированных систем. Она является человеко-машинной и предназначена для сбора и обработки данных, используемых в различных сферах человеческой деятельности.

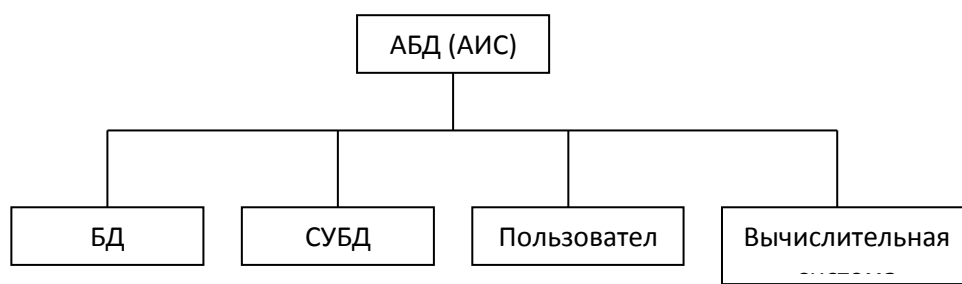


Рис.4.2. Структура АБД

БД предназначена для поддержания адекватной модели предметной области, с целью использования ее в различных приложениях. БД должна обладать следующими основными качествами:

- интегрированностью, направленную на решение комплекса задач;
- структурированностью (модульностью), позволяющей отражать какую-то предметную область в виде адекватной модели;
- независимостью описания данных прикладных программ;
- взаимосвязанностью данных, что позволяет используя одно данные получать новые данные на их основе.

В зависимости от того, кто использует или разрабатывает АИС (АБД), возникает как бы уровни абстрагирования или представления о БД.

Наиболее распространена трехуровневая архитектура БД (рис.4.3.).

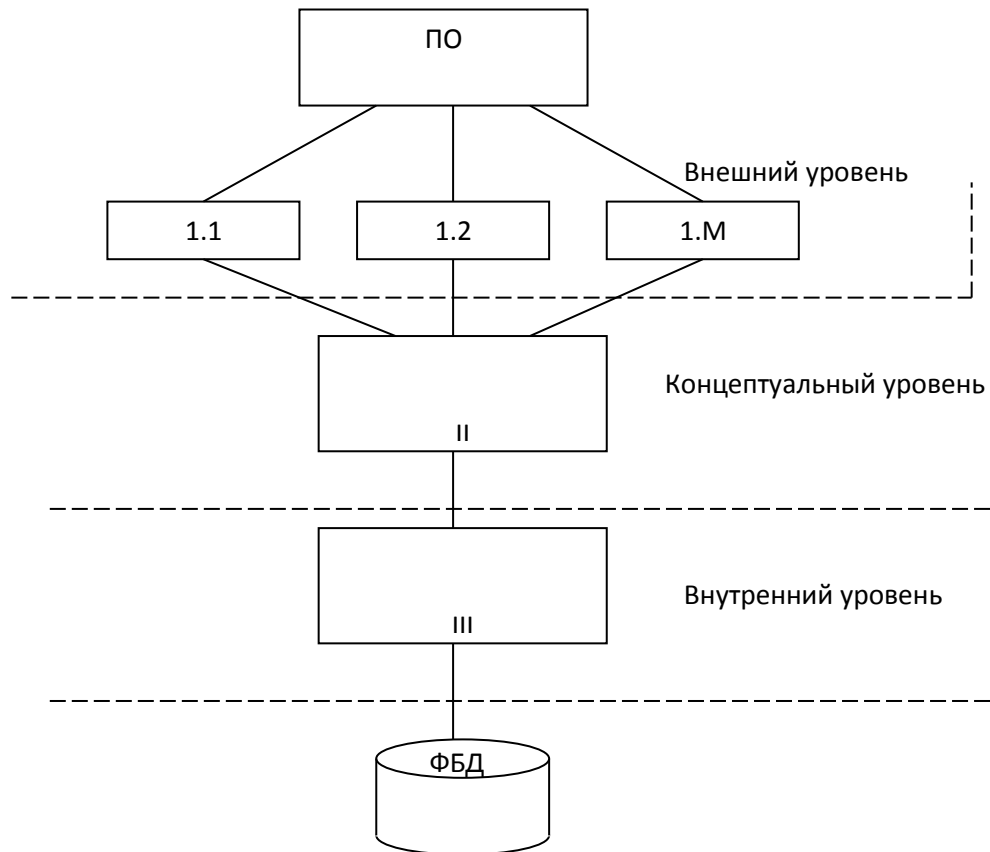


Рис.4.3. Трехуровневая архитектура БД

Внешнее представление – это содержимое БД, каким видит его определенный пользователь. Для таких пользователей БД – это хранилище сведений (на стипендии, оценках, ценах) о тех объектах, которые понятны специалистам в этой предметной области (ПО).

Второй уровень – концептуальный уровень представления соответствует представлению проектировщика АИС о логической организации данных или о структуре данных. Этот уровень во многом совпадает со внешним представлением, но его отличие состоит в привязке к средствам реализации АИС (СУБД). Описание БД на концептуальном уровне задается на языке описания ФБД, используемого в СУБД.

Концептуальное представление состоит из множества экземпляров каждого типа концептуальной записи. На каждом уровне идет обобщение индивидуальных представлений пользователя. Определения концептуальной схемы могут включать и другие дополнительные средства – средства обеспечения безопасности, целостности.

Третий уровень – внутренний уровень представления (абстрагирования). Это представление всей БД в виде многих экземпляров каждого типа хранимой записи. Но внутренние представления, такое, как и первые два уровня не связаны с физическим уровнем (физические записи).

Другими словами, внутренне представление предполагает, что память – бесконечное линейное адресное пространство.

Внутренние представления обычно описываются с помощью внутренней схемы, которая определяет различные типы хранимых записей, существующие индексы, способы представления хранимых полей, последовательность хранимых записей и т.д.

СУБД – специальный пакет программ, посредством которого реализуется централизованное управление БД и обеспечивается доступ к ним.

Упрощенно эти функции реализуются следующим образом:

1. Пользователь выдает запрос на доступ к данным (либо выбором какого-то пункта меню, либо сформировав запрос на языке SQL).
2. СУБД воспринимает этот запрос и анализирует.
3. СУБД просматривает внешнюю схему пользователя, концептуальную схему и внутреннюю схему.

4. После этого СУБД выполняет необходимые операции преобразования данных от одного уровня к другому и в результате инициирует через ОС передачу данных между физической БД и пользователем.

Основные функции СУБД:

1. Обеспечение возможности описания данных: СУБД должна обеспечить определение или описание данных (внешние схемы, концептуальные схемы в виде внутренних схем) и обеспечить преобразование этих описаний в форму соответствующих объектов или другими словами СУБД должна включать в себя языковой процессор для одного языка описания данных.

2. Обработка данных: СУБД должна уметь обрабатывать запросы пользователя на выборку, изменение, удаление существующих записей или на добавление новых данных в базу, или другими словами СУБД должна включать в себя процессор языка манипулирования данными.

3. Обеспечение безопасности и целостности данных: СУБД должна контролировать пользовательские запросы и пресекать попытки нарушения правил безопасности и целостности, определенных проектировщиком БД или администратором БД.

4. Обеспечение восстановления и дублирования данных: СУБД или другой связанный с ней компонент должны осуществлять необходимый контроль за своевременностью процедур дублирования данных и за реализацией процедур восстановления данных при аппаратных сбоях, программных сбоях и т.д.

5. Поддержание словаря данных. Обеспечение функции словаря данных: Этот словарь должен содержать данные о данных (например формы различных схем, перекрестные ссылки, указывающие, какие прикладные программы какую часть БД используют; какие терминалы подключены к системе в данный момент и т.д.).

6. Обеспечение требуемой производительности: СУБД должна выполнять все вышеуказанные функции (особенно манипулирование данными), за временные интервалы, не превышающие требуемых.

Пользователей АИС (или БД) можно разделить на три группы

1. Прикладные программисты – специалисты, которые создают прикладные программы, взаимодействующие с БД. Прикладные программы могут быть написаны на универсальных языках, либо с использованием языков конкретных СУБД, в любом случае они реализуют какие-то операции, связанные со вставкой, выборкой, удалением данных, либо их обновлением.

2. Конечные пользователи могут получать доступ к данным. Во-первых, используя так называемый интерфейс, основанный на меню и формах. Во-вторых, используя так называемый командный интерфейс, или процессор языка запроса. В этом случае пользователь должен указать команду или выражение высокого уровня (например Select/Insert) обычно с использованием высокого языка (например SQL) и в результате получить те или иные необходимые ему данные. Пользователь может получить доступ к БД используя приложения или программы, разработанные прикладными программистами.

3. Администратор БД – лицо (группа лиц), реализующее управление БД.

Если рассмотреть АСУ, то можно выделить объект управления, т.е. БД, и можно выделить управляющий орган, или систему управления, т.е. администратора БД (рис.4.4).

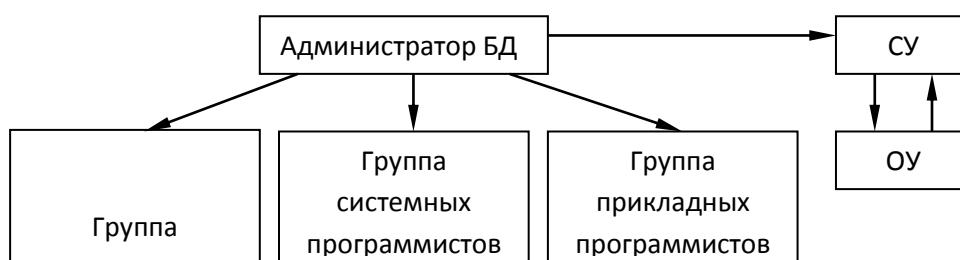


Рис.4.4. СУБД

Основные функции администратора БД:

1. Определение концептуальной схемы – первый этап создания АИС, состоит в том, что специалисты, хорошо знающие эту область, формируют внешние представления или внешние схемы данных, о предметной области. Этот этап обычно называется этапом инфологического проектирования, но и на этом этапе администратор должен уже принимать участие, чтобы иметь возможность создать соответствующую концептуальную схему.

2. Определить внутреннюю схему данных — этот этап, называется этапом физического проектирования.

3. Взаимодействие с пользователями — в функцию администратора БД входит обеспечение пользователей необходимыми данными, консультации по разработке приложений, обеспечение консультаций по техническим и другим вопросам, при возникновении проблем в процессе эксплуатации БД.

4. Определение и реализация мер по обеспечению безопасности и целостности. (например: одна мера по защите данных от некомпетентного использования, а вторая, по обеспечению секретности определенной части данных).

5. Определение процедур резервного копирования и восстановления.

6. Управление производительностью и реагирования на изменяющиеся требования к АИС (АБД).

Архитектура БД. Для обеспечения независимости ПП от данных, для обеспечения целостности, безопасности данных при построении БД используется т.н. модели данных.

Внешняя модель данных (ВшМД), концептуальная модель данных (КМД) и внутренняя модель данных (ВнМД). Эти модели соответствуют уровням представления или абстрагирования БД. Существует несколько терминов: модель данных (МД), схема МД (СхМД) и отображение МД одной на другую.

Схема данных – это инструментарий, позволяющий отображать предметную область на определенном уровне абстрагирования.

СУБД реализует обмен данными между рабочими областями ПП и ФБД (рис.4.5). Любой запрос ПП, сформулированный на ЯНД, поступает в СУБД. Имея соответствие схемы моделей и описание отображение между моделями, СУБД обращается к материалам доступа ОС для выполнения необходимых действий уже на физическом уровне. Такое взаимодействие между СУБД, ОС и МД может быть представлено в виде трехуровневой архитектуры данных.

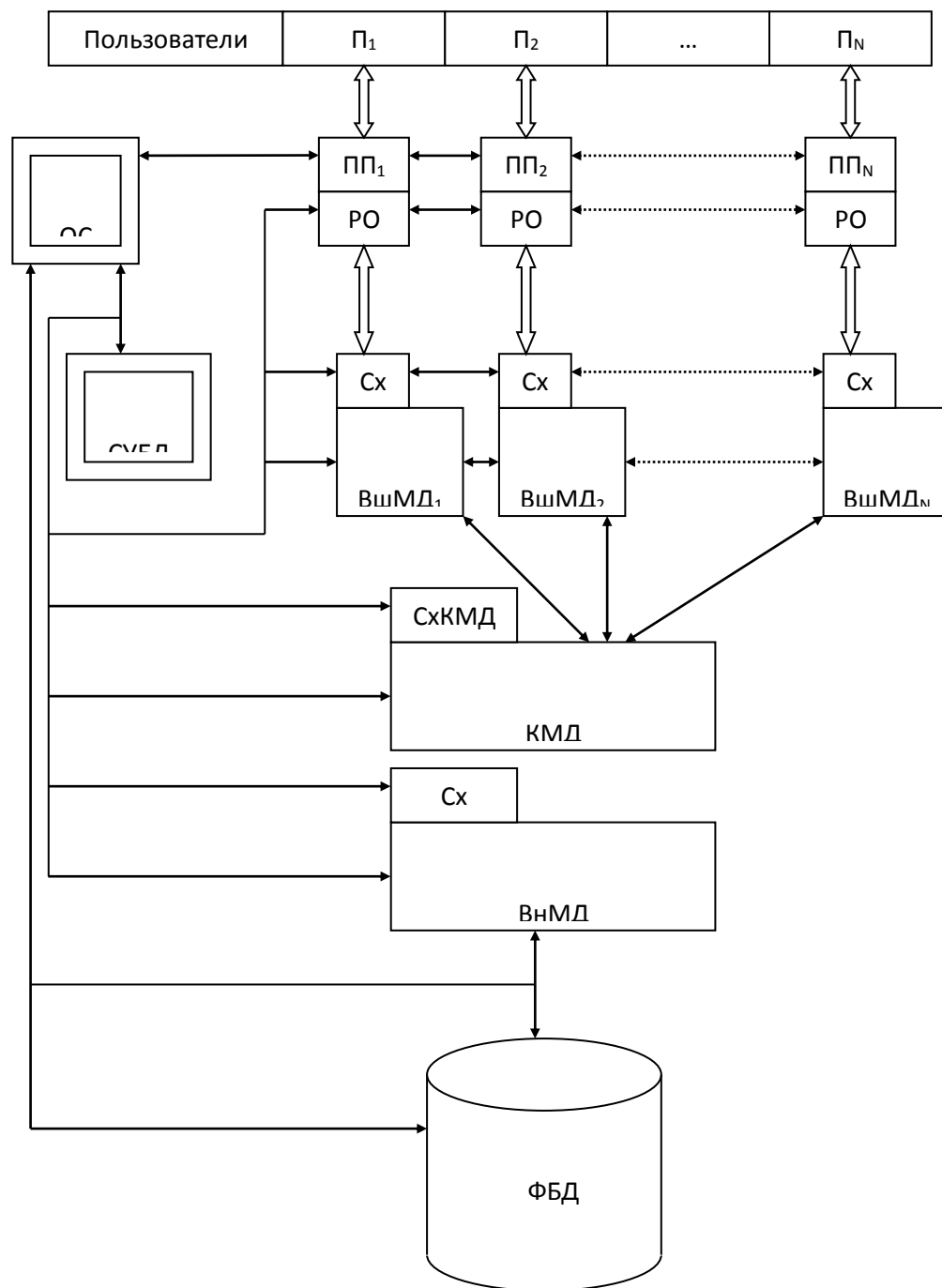


Рис.4.5. Схема потоков данных

Последовательность действий СУБД при формировании записи ВШМД для ПП по запросу следующая (рис.4.6.):

1. ПП обращается к СУБД для чтения или записи из ФБД в виде ВШМД;

2. СУБД используются схемы ВшМД и КМД и используя отображения ВшМД на КМД, определяет какие записи КМД необходимы для формирования требуемой записи ВшМД;

3. Используя схемы КМД и ВнМД и описания отображения КМД на ВнМД, СУБД определяет, какие хранимые записи необходимы для построения затребованных записей КМД и какая совокупность физических записей необходима для считывания с магнитного диска;

4. СУБД выдает запрос ОС на считывание, в свою буферную область памяти, необходимой записи из ФБД;

5. ОС, с помощью своих методов доступа, считывает из физической памяти затребованные СУБД физические записи, и помещает их в системный буфер СУБД, кроме того, сообщение ОС о выполнении этого запроса добавляется к сообщениям СУБД, и также помещается в буферную память;

6. На основании имеющихся схем моделей и описаний соответствующих отображений, СУБД формирует в буферной памяти запись ВшМД в таком виде, в каком требует ПП;

7. СУБД пересылает сформулированную запись ВшМД в рабочую область (РО) ввода/вывода ПП;

8. СУБД передает в ПП свои сообщения и сообщения ОС о результатах выполнения запроса;

9. ПП обрабатывает запись, поступившую в РО ввода/вывода, и выдает результат пользователю.

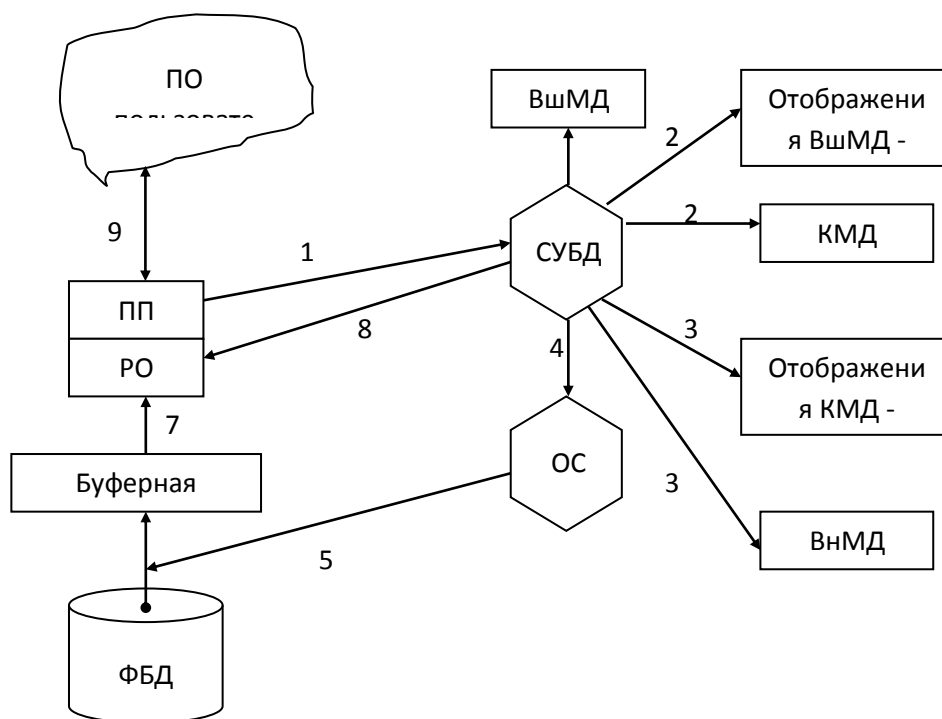


Рис.4.6. Последовательность действий СУБД при формировании записи ВшМД для ПП по запросу

Наличие в БД процессов обмена информацией между пользователем и системой, между администратором БД (и АдБД) и системой, а также между моделями данных различных уровней представления данных требует унификации этих процессов, т. е. разработки соответствующих интерфейсов. Эти интерфейсы разрабатываются с помощью ЯОД и ЯМД на соответствующих уровнях.

В типовом варианте БД обычно используются следующие интерфейсы:

1. Интерфейс пользователя. Используется язык внешнего уровня, при подготовке исходных текстов ПП или формулировке запросов;
2. Имеет место при трансляции ПП или запроса с языка внешнего уровня данные поступают на вход СУБД в объектных кодах;
3. Между КМД и АдБД;

4. Между ВшМД и АдБД;
5. Между ВнМД и АдБД.

Кроме того, для написания и коррекции схем моделей администратор БД может быть обеспечен соответствующими средствами или языками.

Кроме рассмотренных уже уровней абстрагирования в БД существует ещё один, им предшествующий. Модель этого уровня содержит информацию о ПО в виде, независимом от используемой СУБД (на естественном языке). Это естественный, информационный уровень абстрагирования, связанный с фиксацией и описанием выделенных сведений о ПО. Модель этого уровня называется инфологической моделью ПО.

4.6 Классификация ИС и БД

ИС (БД) могут классифицироваться по различным признакам:

- 1) В зависимости от масштаба (ИС рабочей группы, отдела предприятия)
- 2) По характеру обработки информации:
 - оперативная обработка транзакций;
 - системы оперативного анализа;
 - системы поддержки принятия решений;
 - хранилище данных.
- 3) В зависимости от архитектуры:
 - ИС с централизованной архитектурой;
 - ИС с архитектурой клиент-сервер;
 - Корпоративные ИС с использованием интернет-технологий;
 - Географически распределенной ИС.

4.6.1 ИС с централизованной архитектурой

Такая архитектура является традиционной и характеризуется тем, что имеется ЭВМ (центральный узел, HOST), на котором располагается и БД и СУБД и приложения.

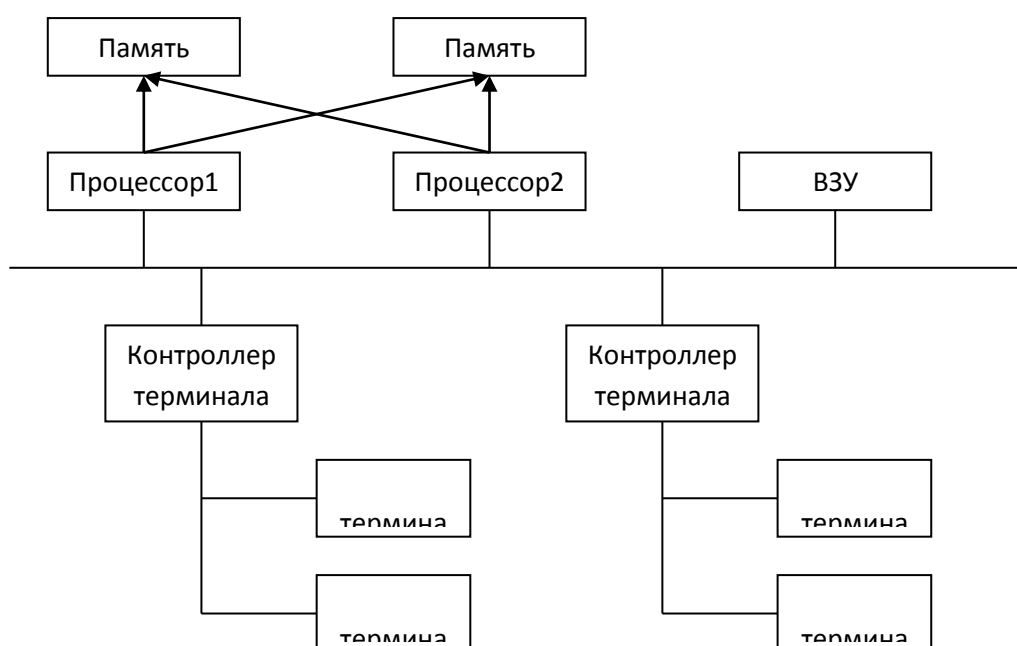


Рис.4.7. ИС с централизованной архитектурой

4.6.2 ИС с архитектурой клиент-сервер

При рассмотрении ИС можно, на более высоком уровне обобщения архитектуры, выделить такие компоненты – сервер (или машины БД, СУБД) и набора клиентов (различные приложения).

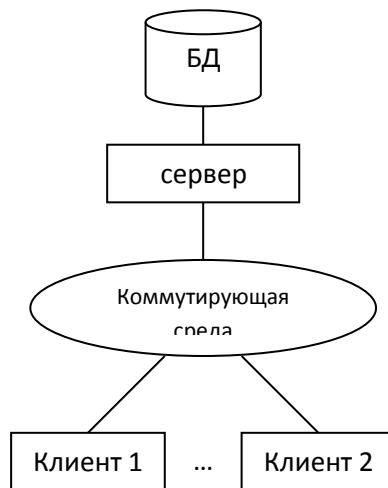


Рис.4.8 ИС с архитектурой клиент-сервер

В рамках такой архитектуры может быть выделена разновидность — файл-сервер. В этом случае компьютер-сервер и его база (ВЗУ) используется как хранилище данных, а обработка на файл-сервере не производится. В архитектуре клиент-сервер на сервере может осуществляться обработка данных.

В последнее время в некоторых случаях машины клиентов могут иметь свои собственные сохраняемые данные, и машина сервера тоже может иметь свои приложения. То есть, каждый компьютер может выступать и в роли сервера, и в роли клиента. В этом случае говорят, что такая структура поддерживает полную систему БД. В рамках этой архитектуры описывается два варианта доступа:

- клиент может получать доступ в каждый момент времени только к одному из серверов;
- клиент может получить доступ к любому числу серверов одновременно (распределенная система данных).

5. Классификация программных средств автоматизированных систем управления технологическими процессами нефтегазового комплекса

В типовой архитектуре АСУТП НГК явно просматриваются два уровня:

- уровень контроллеров, взаимодействующих с объектом управления посредством датчиков и исполнительных устройств;
- уровень оперативного управления технологическим процессом, основными компонентами которого являются серверы, рабочие станции операторов/диспетчеров, АРМ специалистов.

Каждый из этих уровней функционирует под управлением специализированного программного обеспечения (ПО). Разработка этого ПО или его выбор из предлагаемых в настоящее время на рынке программных средств зависит от многих факторов, прежде всего от решаемых на конкретном уровне задач. Различают базовое и прикладное программное обеспечение (рис.5.1.).

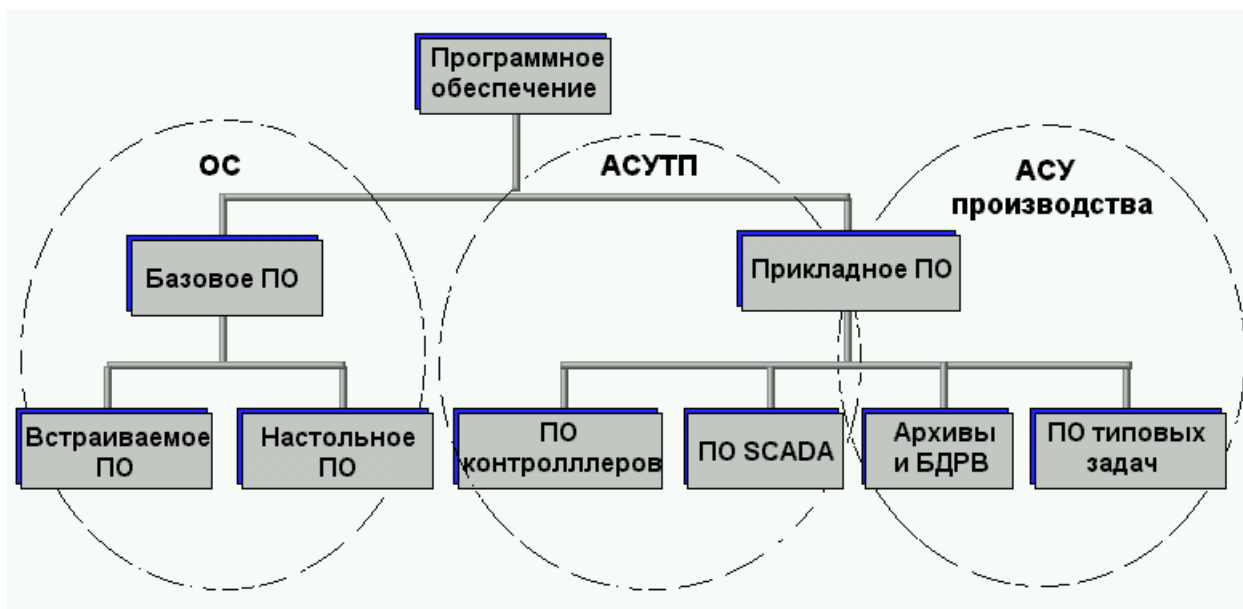


Рис.5.1. Классификация программных средств системы управления

Базовое ПО включает в себя различные компоненты, но основным из них является операционная система (ОС) программно-технических средств АСУТП. Каждый уровень АСУТП представлен «своими» программно-техническими средствами: на нижнем уровне речь идет о контроллерах, тогда как основным техническим средством верхнего уровня является компьютер. В соответствии с этим в кругу специалистов появилась и такая классификация: встраиваемое и настольное программное обеспечение.

Очевидно, требования, предъявляемые к встраиваемому и настольному ПО, различны. Контроллер в системе управления наряду с функциями сбора информации решает задачи автоматического непрерывного или логического управления. В связи с этим к нему предъявляются жесткие требования по времени реакции на состояние объекта и выдачи управляющих воздействий на исполнительные устройства. Контроллер должен гарантированно откликаться на изменения состояния объекта за заданное время.

Для решения подобных задач рекомендуется применение ОС реального времени (ОСРВ). Такие операционные системы иногда называют детерминированными, подразумевая под этим гарантированный отклик за заданный промежуток времени. Большинство микропроцессорных устройств (в том числе контроллеры и компьютеры) используют механизм прерываний работы процессора. В ОС реального времени, в отличие от ОС общего назначения (не гарантирующих времени исполнения), прерываниям присвоены приоритеты, а сами прерывания обрабатываются за гарантированное время.

Выбор ОС зависит от жесткости требований реального времени. Для задач, критичных к реакции системы управления, в настоящее время применяются такие операционные системы реального времени, как OS-9, QNX, VxWorks. В системах с менее жесткими требованиями к реальному времени возможно применение версий Windows NT/CE, точнее их расширений реального времени.

OS-9 относится к классу Unix-подобных операционных систем реального времени и предлагает многие привычные элементы среды Unix. Все функциональные компоненты OS-9, включая ядро, иерархические файловые менеджеры, систему ввода/вывода и средства разработки, реализованы в виде независимых модулей. Комбинируя эти модули, разработчик может создавать системы с самой разной конфигурацией - от миниатюрных автономных ядер, ориентированных на ПЗУ контроллеров, до полномасштабных многопользовательских систем разработки.

OS-9 обеспечивает выполнение всех основных функций операционных систем реального времени: управление прерываниями, межзадачный обмен информацией и синхронизация задач.

Операционная система QNX разработки канадской фирмы QNX Software Systems Ltd. является одной из наиболее широко используемых систем реального времени. QNX гарантирует время реакции в пределах от нескольких десятков микросекунд до нескольких миллисекунд (в зависимости от быстродействия ПЭВМ и версии QNX). Кроме того, высокая эффективность QNX в задачах управления в реальном времени обеспечивается такими свойствами, как многозадачность (до 250 задач на одном узле), встроенные в ядро системы сетевые возможности, гибкое управление прерываниями и приоритетами, возможность выполнения задач в защищенном и фоновом режимах.

Операционная система QNX нашла применение как на нижнем уровне АСУТП (ОС для контроллеров), так и на верхнем уровне (ОС для программного обеспечения SCADA).

Операционная система реального времени VxWorks предназначена для разработки ПО встроенных компьютеров, работающих в системах «жесткого» реального времени. К операционной системе VxWorks прилагается и инструментальная среда Tornado фирмы Wind River Systems со средствами разработки прикладного программного обеспечения. Его разработка ведется на инструментальном компьютере в среде Tornado для

последующего исполнения на целевом компьютере (контроллере) под управлением VxWorks.

ОС VxWorks поддерживает целый ряд компьютерных платформ, в том числе Intel 386/486/Pentium, PowerPC, DEC Alpha. К платформам, поддерживаемым инструментальной средой Tornado, относятся Sun (Solaris), HP 9000/400,700, DEC Alpha, PC (Windows 95 и NT) и другие.

Операционная система Windows знакома всем как настольная система. Но это, прежде всего, относится к платформам Windows 3.xx/95, в которых действительно отсутствует поддержка реального времени. Ситуация резко изменилась с появлением Windows NT. Сама по себе Windows NT не является операционной системой реального времени в силу ряда ее особенностей. Система поддерживает аппаратные (а не программные) прерывания, отсутствует приоритетная обработка отложенных процедур и др. Но в конце XX века ряд фирм предприняли серьезные попытки превратить Windows NT в ОС жесткого реального времени. И эти попытки увенчались успехом. Компания VenturCom разработала модуль Real Time Extension (RTX) - подсистему реального времени (РВ) для Windows NT. Эта подсистема имеет собственный планировщик со 128 приоритетами прерываний, который не зависит от NT. Максимальное время реакции на прерывание составляет 20-80 мкс вне зависимости от загрузки процессора. Теперь при каждом прерывании от таймера приоритет передается критичным по времени задачам. А в оставшееся от их работы время могут выполняться «медленные» процессы: ввод/вывод, работа с диском, сетью, графическим интерфейсом и т. п.

32-разрядная Windows CE была создана компанией Microsoft для малых компьютеров (калькуляторов), но в силу ряда достоинств стала претендовать на роль стандартной ОС реального времени. К числу этих достоинств относятся:

- открытость и простота стыковки с другими ОС семейства Windows;

- время реакции порядка 500 мкс;
- значительно меньшие по сравнению с другими ОС Windows требования к ресурсам памяти и возможность построения бездисковых систем.

А в 1999 году компанией Direct by Koyo ОС Windows CE была впервые установлена на платформу микроPLC.

Выбор операционной системы программно-технических средств верхнего уровня АСУТП определяется прикладной задачей (ОС общего пользования или ОСРВ). Но наибольшую популярность и распространение получили различные варианты ОС Windows (Windows NT/2000). Ими оснащены программно-технические средства верхнего уровня АСУТП, представленные персональными компьютерами (ПК) разной мощности и конфигурации - рабочие станции операторов/диспетчеров и специалистов, серверы баз данных (БД) и т. д.

Такая ситуация возникла в результате целого ряда причин и тенденций развития современных информационных и микропроцессорных технологий.

Вот несколько основных аргументов в пользу Windows:

- Windows имеет очень широкое распространение в мире, в том числе и в России, в связи с чем легко найти специалиста, который мог бы сопровождать системы на базе этой ОС;
- эта ОС имеет множество приложений, обеспечивающих решение различных задач обработки и представления информации;
- ОС Windows и Windows-приложения просты в освоении и обладают типовым интуитивно понятным интерфейсом;
- приложения, работающие под управлением Windows, поддерживают общедоступные стандарты обмена данными;
- системы на базе ОС Windows просты в эксплуатации и развитии, что делает их экономичными как с точки зрения поддержки, так и при поэтапном росте;

- Microsoft развивает информационные технологии (ИТ) для Windows высокими темпами, что позволяет компаниям, использующим эту платформу «идти в ногу со временем».

Также следует учитывать и то, что неотъемлемой частью верхнего уровня АСУ ТП является человек, время реакции которого на события недетерминировано и зачастую достаточно велико. Да и сама проблема реального времени на верхнем уровне не столь актуальна.

В 90-х годах широкое распространение получила ОС реального времени QNX. Имеется множество примеров использования QNX на всех уровнях иерархической структуры АСУТП (от контроллеров до серверов и рабочих станций). Но в последние годы активность компании на рынке SCADA-систем значительно снизилась, что привело и к снижению числа продаж этого программного продукта. Объясняется это тем, что еще в 1995 году компания QNX Software Systems Ltd. объявила об «уходе» во встроенные системы.

С точки зрения разработки системы управления предпочтительна такая программная архитектура, в которой ПО всех уровней управления реализовано в единой операционной системе. В этом случае «автоматически» снимаются все вопросы, связанные с вертикальным взаимодействием различных программных компонент системы управления. Но на практике это далеко не так. Достаточно часто в разрабатываемых системах контроля и управления нижний и верхний уровни реализуются в разных ОС. И наиболее характерна ситуация, когда на уровне контроллера используется ОС реального времени, а на уровне оператора/диспетчера SCADA-система функционирует под Windows NT. Без специализированных решений по организации взаимодействия между подсистемами здесь не обойтись.

Для функционирования системы управления необходим и еще один тип ПО - прикладное программное обеспечение (ППО).

Известны два пути разработки прикладного программного обеспечения систем управления:

- создание собственного прикладного ПО с использованием средств традиционного программирования (стандартные языки программирования, средства отладки и т.д.);
- использование для разработки прикладного ПО существующих (готовых) инструментальных средств.

Первый вариант является наиболее трудоемким. Применение высокоуровневых языков требует соответствующей квалификации разработчиков в теории и технологии программирования, знания особенностей конкретной операционной системы, тонкостей аппаратного обеспечения (контроллеров). С точки зрения основных критериев - стоимости и времени разработки - этот вариант неприемлем в большинстве случаев.

Второй вариант является более предпочтительным. Почему? А потому, что на сегодняшний день в мире уже создано несколько десятков инструментальных систем, хорошо поддерживаемых, развиваемых и нашедших применение при создании десятков и сотен тысяч проектов автоматизации. Эти проверенные временем программные средства упрощают (разработчики интерфейсов - не высокочеловеческие программисты, а специалисты по автоматизации), ускоряют и значительно удешевляют процесс разработки.

С точки зрения области применения готовые инструментальные средства можно разделить на два класса:

- средства, ориентированные на разработку программ управления внешними устройствами, контроллерами - CASE-системы (Computer Aided Software Engineering);
- средства, ориентированные на обеспечение интерфейса оператора/ диспетчера с системой управления – SCADA-системы

(Supervisory Control And Data Acquisition - диспетчерское управление и сбор данных).

Контроллеру требуется программа, в соответствии с которой он взаимодействует с объектом. В одних случаях речь идет только о сборе данных с объекта, в других - о логическом управлении (например, выполнении блокировок). Наконец, одно из основных применений контроллера - реализация функций непрерывного управления отдельными параметрами или технологическим аппаратом (процессом) в целом.

Фирмы, производящие оборудование для построения систем автоматизации, всегда стремились сопровождать свою продукцию набором программных инструментов, с помощью которых пользователь по определенным правилам и соглашениям мог бы описывать логику работы контроллера. На раннем этапе развития этих программных средств набор поддерживаемых ими функций обеспечивался нестандартными языками. Со временем правила и соглашения совершенствовались и на определенном этапе были оформлены в виде специальных языков программирования, образовав то, что сейчас называется CASE-инструментарием.

В 1992 году Международная Электротехническая Комиссия (МЭК, IEC - International Electrotechnical Commission,) взяла под контроль процессы, связанные с развитием этого типа прикладного ПО. Были выдвинуты требования открытости системы, выполнение которых позволило бы унифицировать программные средства и упростить разработку:

- возможность разработки драйверов для контроллеров самими пользователями, т.е. сопровождение программных продуктов по программированию контроллеров специальными инструментальными средствами;
- наличие коммуникационных средств (интерфейсов) для взаимодействия с другими компонентами системы управления;
- возможность портации ядра системы на ряд программно-аппаратных платформ.

На рынке появилось большое количество пакетов, удовлетворяющих вышеописанным требованиям. Практически во всех этих пакетах среда разработки реализована в Windows-интерфейсе, имеются средства загрузки разработанного приложения в исполнительную систему.

Названия некоторых из этих пакетов приведены ниже:

- RSLogix 500, RS Logix 5, RSLogix 5000 фирмы Rockwell Software для программирования контроллеров различных семейств Allen-Bradley;
- DirectSOFT для контроллеров семейства Direct Logic фирмы Kooyo;
- пакеты PL7 и Concept - ПО для программирования контроллеров различных семейств компании Schneider Electric;
- пакеты STEP 5, STEP 7 Micro, STEP 7 для программирования контроллеров семейств S5 и S7 фирмы Siemens;
- пакет Toolbox для конфигурирования контроллеров семейства Moscad;
- пакет TelePACE для программирования контроллеров серий

TeleSAFE Micro 16 и SCADAPack фирмы Control Microsystems.

Стандартом МЭК 1131-3 определены пять языков программирования контроллеров: три графических (LD, FBD, SFC) и два текстовых (ST, IL).

LD (Ladder Diagram) - графический язык диаграмм релейной логики. Язык LD применяется для описания логических выражений различного уровня сложности.

FBD (Function Block Diagram) - графический язык функциональных блоковых диаграмм. Язык FBD применяется для построения комплексных процедур, состоящих из различных функциональных библиотечных блоков - арифметических, тригонометрических, регуляторов и т.д.).

SFC (Sequential Function Chart) - графический язык последовательных функциональных схем. Язык SFC предназначен для использования на этапе

проектирования ПО и позволяет описать «скелет» программы - логику ее работы на уровне последовательных шагов и условных переходов.

ST (Structured Text) - язык структурированного текста. Это язык высокого уровня, по мнемонике похож на Pascal и применяется для разработки процедур обработки данных.

IL (Instruction List) - язык инструкций. Это язык низкого уровня класса ассемблера и применяется для программирования эффективных, оптимизированных процедур.

В конце 90-х годов появились открытые программные продукты ISaGRAF, InControl (Wonderware), Paradym (Intellution), предназначенные для разработки, отладки и исполнения программ управления как дискретными, так и непрерывными процессами.

Сейчас уже можно сказать, что подавляющее большинство контроллеров и систем управления обслуживается программными продуктами, реализующими стандарт МЭК 1131-3.

Широкое применение в России нашел пакет ISaGRAF французской компании CJ International.

Основные возможности пакета:

- Поддержка всех пяти языков стандарта МЭК 1131-3 плюс реализация языка Flow Chart как средства описания диаграмм состояний. При этом ISaGRAF позволяет смешивать программы и процедуры, написанные на разных языках, а также вставлять кодовые последовательности из одного языка в коды, написанные на другом языке.
- Наличие многофункционального отладчика, позволяющего во время работы прикладной задачи просматривать состояние программного кода, переменных, программ и многое другое.
- Поддержка различных протоколов промышленных сетей.
- Реализация опций, обеспечивающих открытость системы для доступа к внутренним структурам данных прикладной ISaGRAF-

задачи, а также возможность разработки драйверов для модулей ввода/вывода, разработанных самим пользователем, и возможность переноса ISaGRAF-ядра на любую аппаратно-программную платформу.

- Набор драйверов для работы с контроллерами различных фирм-производителей: PEP Modular Computers, Motorola Computer Group и др.

- Наличие дополнительных интерактивных редакторов для описания переменных, констант и конфигураций ввода/вывода.

- Встроенные средства контроля за внесением изменений в программный код ISaGRAF-приложения и печати отчетов по разработанному проекту с большой степенью детализации, включая печать таблиц перекрестных ссылок для программ и отдельных переменных.

- Полное документирование этапов разработки.

Программные средства верхнего уровня АСУТП (SCADA-пакеты) предназначены для создания прикладного программного обеспечения пультов контроля и управления, реализуемых на различных компьютерных платформах и специализированных рабочих станциях. SCADA - пакеты позволяют при минимальной доле программирования на простых языковых средствах разрабатывать многофункциональный интерфейс, обеспечивающий оператора/диспетчера не только полной информацией о технологическом процессе, но и возможностью им управлять.

В своем развитии SCADA - пакеты прошли тот же путь, что и программное обеспечение для программирования контроллеров. На начальном этапе (80-е годы) фирмы-разработчики аппаратных средств создавали собственные (закрытые) SCADA-системы, способные взаимодействовать только со «своей» аппаратурой. Начиная с 90-х годов, появились универсальные (открытые) SCADA - программы.

Понятие открытости является фундаментальным, когда речь идет о программно-аппаратных средствах для построения многоуровневых систем автоматизации. Более подробно об этом будет сказано ниже.

Сейчас на российском рынке присутствует несколько десятков открытых SCADA-пакетов, обладающих практически одинаковыми функциональными возможностями. Но это совсем не означает, что любой из них можно с одинаковыми усилиями (временными и финансовыми) успешно адаптировать к той или иной системе управления, особенно, если речь идет о ее модернизации. Каждый SCADA-пакет является по-своему уникальным, и его выбор для конкретной системы автоматизации, обсуждаемый на страницах специальной периодической прессы почти на протяжении последних десяти лет, по-прежнему остается актуальным.

Ниже приведен перечень наиболее популярных в России SCADA-пакетов.

- Trace Mode/Трейс Моуд (AdAstrA) - Россия;
- InTouch (Wonderware) - США;
- FIX (Intellution) - США;
- Genesis (Iconics Co) - США;
- Factory Link (United States Data Co) - США;
- RealFlex (BJ Software Systems) - США;
- Sitex (Jade Software) - Великобритания;
- Citect (CI Technology) - Австралия;
- WinCC (Siemens) - Германия;
- RTWin (SWD Real Time Systems) - Россия;
- САРГОН (НВТ - Автоматика) - Россия;
- MIK\$Sys (МИФИ) - Россия;
- Cimplicity (GE Fanuc) - США;
- RSView (Rockwell Automation) - США и многие другие.

Последовательность представления пакетов в приведенном выше перечне в достаточной степени случайна. Констатируется лишь сам факт

существования той или иной системы. Предлагается исходить из предпосылки, что SCADA-пакет существует, если с помощью него уже реализовано хотя бы несколько десятков проектов. Вторая предпосылка - нет абсолютно лучшей SCADA-системы для всех случаев применения. SCADA - это всего лишь удобный инструмент в руках разработчика, и ее адаптация к конкретной системе автоматизации - вопрос квалификации и опыта.

До недавнего времени задача регистрации информации в реальном времени могла быть решена либо на уровне программного обеспечения концентратора (контроллера верхнего уровня), либо на уровне SCADA-системы. При этом речь идет о больших потоках данных о процессе, поступающих от большого количества датчиков (нескольких сот или тысяч) в реальном масштабе времени и с высокой частотой (периоды опроса – порядка секунд и даже долей секунд). На уровне АСУТП эта информация нужна для оперативного управления технологическим процессом.

Данные технологических процессов специфичны. Они, как правило, могут быть представлены в виде временных рядов «значение – время». Для их сбора и хранения практически любой SCADA-пакет имеет в своем составе подсистему регистрации исторических данных (архив) с возможностью последующей выборки требуемых для анализа данных и их представления в виде трендов.

Но такие архивы не предназначены для длительного хранения больших объемов информации. К тому же, речь здесь идет о так называемых локальных архивах. Архив SCADA-пакета хранит информацию о переменных лишь одного конкретного технологического процесса. Но предприятие имеет в своем составе целый ряд технологических процессов, системы управления которыми выполнены, как правило, на различной программно-аппаратной платформе.

В получении оперативных и объективных технологических данных сегодня заинтересованы практически все службы предприятия. Однако характер необходимой информации различен для различных уровней

управления. На верхнем уровне (АСУП) нужна только интегрированная (предварительно подготовленная) информация о технологических процессах (данные типа «нарастающим итогом», средних значений за определенные промежутки времени, общее количество произведенных продуктов и т.д.).

Для хранения такой информации хорошо адаптированы базы данных реляционного типа (РБД). Данные в этих базах статичны, связаны многими отношениями, должны быть легко выбираемы по различным сложным критериям. Однако РБД не приспособлены для хранения огромного количества значений параметров, получаемых от SCADA-систем и накапливаемых за достаточно длительное время (до трех и более лет).

В результате, информация, имеющаяся и успешно используемая в АСУТП, недоступна для верхнего уровня.

Таким образом, назрела необходимость создания и внедрения в процесс управления так называемых исторических архивов производственных данных или баз данных реального времени (БДРВ) масштаба предприятия.

Во - первых, такие системы должны обеспечить сбор данных с различных источников производственной информации на предприятии (SCADA-систем, DCS-систем, лабораторных систем - LIMS, различных СУБД и т. п.) и их долговременное хранение в едином формате. Во-вторых - обеспечить доступ к информации специалистам и руководителям всех уровней и служб по стандартным протоколам с помощью специализированных клиентских приложений.

Такие системы от различных производителей (в том числе и от производителей SCADA-систем) уже появились в России и с каждым днем находят все более широкое применение. Среди них IndustrialSQL Server – компонент интегрированного пакета FactorySuite (Wonderware), iHistorian – компонент семейства Intellution Dynamics и другие.

Существует целый ряд задач управления, не перекрываемых ни классом АСУП, ни классом АСУТП. Частично эти задачи не перекрываются

из-за отсутствия возможностей программного обеспечения этих уровней системы управления. Среди них находятся и задачи, решение которых может оказать решающее влияние на эффективность предприятия в целом: диспетчеризация производства, оперативное планирование, управление качеством продукции и многие другие.

Наличие базы данных реального времени масштаба предприятия – это только лишь предпосылка для их решения (необходимое, но недостаточное условие). Ряд разработчиков инструментальных систем предлагают использовать с этой целью специальный тип программных продуктов. Это могут быть небольшие системы, предназначенные для решения отдельных типовых задач, например, системы расчета и согласования материальных балансов. Появился ряд интегрированных систем, поддерживающих, наряду с функциями хранения и представления информации, решение задач расчета тепловых и материальных балансов, планирования, оптимизации и т.п. К наиболее известным программным продуктам этого класса ПО относятся InfoPlus компании Aspen Tech, «Калькулятор качества» фирмы ПЕТРОКОМ, PI System (Plant Information System) компании OSIsoft.

Современное развитие информационных технологий (ИТ) создало предпосылки для успешной интеграции всех уровней управления многоуровневой системы и создания интегрированной информационной системы предприятия.

5. 1 Общая характеристика программного обеспечения SCADA

5.1.1 Основные функции SCADA-систем

Программное обеспечение типа SCADA предназначено для разработки и эксплуатации автоматизированных систем управления технологическими процессами. Резонно задать вопрос: а что же все-таки первично – разработка или эксплуатация? И ответ в данном случае

однозначен – первичным является эффективный человеко-машинный интерфейс (HMI), ориентированный на пользователя, т. е. на оперативный персонал, роль которого в управлении является определяющей. SCADA – это новый подход к проблемам человеческого фактора в системах управления (сверху вниз), ориентация в первую очередь на человека (оператора/диспетчера), его задачи и реализуемые им функции.

Такой подход позволил минимизировать участие операторов/диспетчеров в управлении процессом, но оставил за ними право принятия решения в особых ситуациях.

А что дала SCADA-система разработчикам? С появлением SCADA они получили в руки эффективный инструмент для проектирования систем управления, к преимуществам которого можно отнести:

- высокую степень автоматизации процесса разработки системы управления;
- участие в разработке специалистов в области автоматизируемых процессов (программирование без программирования);
- реальное сокращение временных, а, следовательно, и финансовых затрат на разработку систем управления.

Прежде, чем говорить о функциональных возможностях ПО SCADA, предлагается взглянуть на функциональные обязанности самих операторов/диспетчеров. Каковы же эти обязанности? Следует сразу отметить, что функциональные обязанности операторов/диспетчеров конкретных технологических процессов и производств могут быть существенно разными, да и сами понятия «оператор» и «диспетчер» далеко не равнозначны. Тем не менее, среди многообразия этих обязанностей оказалось возможным найти общие, присущие данной категории работников:

- регистрация значений основных технологических и
хозрасчетных
параметров;
- анализ полученных данных и их сопоставление со сменно-суточными
заданиями и календарными планами;
- учет и регистрация причин нарушений хода
технологического
процесса;
- ведение журналов, составление оперативных рапортов, отчетов и
других документов;
- предоставление данных о ходе технологического процесса
и
состоянии оборудования в вышестоящие службы и т. д.

Раньше в операторной (диспетчерской) находился щит управления (отсюда - щитовая). Для установок и технологических процессов с несколькими сотнями параметров контроля и регулирования длина щита могла достигать нескольких десятков метров, а количество приборов на них измерялось многими десятками, а иногда и сотнями. Среди этих приборов были и показывающие (шкала и указатель), и самопишущие (кроме шкалы и указателя еще и диаграммная бумага с пером), и сигнализирующие. В определенное время оператор, обходя щит, записывал показания приборов в журнал. Так решалась задача сбора и регистрации информации.

В приборах, обслуживающих регулируемые параметры, имелись устройства для настройки задания регулятору и для перехода с автоматического режима управления на ручное (дистанционное) управление. Здесь же, рядом с приборами, находились многочисленные кнопки, тумблеры и рубильники для включения и отключения различного технологического оборудования. Таким образом, решались задачи дистанционного управления технологическими параметрами и оборудованием.

Над щитом управления (как правило, на стене) находилась мнемосхема технологического процесса с изображенными на ней технологическими аппаратами, материальными потоками и многочисленными лампами сигнализации зеленого, желтого и красного (аварийного) цвета. Эти лампы начинали мигать при возникновении нештатной ситуации. В особо опасных ситуациях предусматривалась возможность подачи звукового сигнала (сирена) для быстрого предупреждения всего оперативного персонала. Так решались задачи, связанные с сигнализацией нарушений технологического регламента (отклонений текущих значений технологических параметров от заданных, отказа оборудования).

С появлением в операторной/диспетчерской компьютеров было естественным часть функций, связанных со сбором, регистрацией, обработкой и отображением информации, определением нештатных (аварийных) ситуаций, ведением документации, отчетов, переложить на компьютеры. Еще во времена первых управляющих вычислительных машин с монохромными алфавитно-цифровыми дисплеями на этих дисплеях усилиями энтузиастов-разработчиков уже создавались «псевдографические» изображения - прообраз современной графики. Уже тогда системы обеспечивали сбор, обработку, отображение информации, ввод команд и данных оператором, архивирование и протоколирование хода процесса.

Хотелось бы отметить, что с появлением современных программно-технических средств автоматизации, рабочих станций операторов/диспетчеров, функционирующих на базе программного обеспечения SCADA, щиты управления и настенные мнемосхемы не канули безвозвратно в лету. Там, где это продиктовано целесообразностью, щиты и пульты управления остаются, но становятся более компактными.

Появление УВМ, а затем и персональных компьютеров вовлекло в процесс создания операторского интерфейса программистов. Они хорошо владеют компьютером, языками программирования и способны писать

сложные программы. Для этого программисту нужен лишь алгоритм (формализованная схема решения задачи). Но беда в том, что программист, как правило, не владеет технологией, не «понимает» технологического процесса. Поэтому для разработки алгоритмов надо было привлекать специалистов-технологов, например, инженеров по автоматизации.

Выход из этой ситуации был найден в создании методов «программирования без реального программирования», доступных для понимания не только программисту, но и инженеру-технологу. В результате появились программные пакеты для создания интерфейса «человек-машина» (Man/Humain Machine Interface, MMI/HMI). За рубежом это программное обеспечение получило название SCADA (Supervisory Control And Data Acquisition – супервизорное/диспетчерское управление и сбор данных), так как предназначалось для разработки и функциональной поддержки АРМов операторов/диспетчеров в АСУТП. А в середине 90-х аббревиатура SCADA (СКАДА) уверенно появилась и в лексиконе российских специалистов по автоматизации.

Оказалось, что большинство задач, стоящих перед создателями программного обеспечения верхнего уровня АСУ ТП различных отраслей промышленности, достаточно легко поддается унификации, потому что функции оператора/диспетчера практически любого производства достаточно унифицированы и легко поддаются формализации.

Таким образом, базовый набор функций SCADA-систем предопределен ролью этого программного обеспечения в системах управления (HMI) и реализован практически во всех пакетах. Это:

- сбор информации с устройств нижнего уровня (датчиков, контроллеров);
- прием и передача команд оператора/диспетчера на контроллеры и исполнительные устройства (дистанционное управление объектами);

- сетевое взаимодействие с информационной системой предприятия (с вышестоящими службами);
- отображение параметров технологического процесса и состояния оборудования с помощью мнемосхем, таблиц, графиков и т.п. в удобной для восприятия форме;
- оповещение эксплуатационного персонала об аварийных ситуациях и событиях, связанных с контролируемым технологическим процессом и функционированием программно-аппаратных средств АСУ ТП с регистрацией действий персонала в аварийных ситуациях.
- хранение полученной информации в архивах;
- представление текущих и накопленных (архивных) данных в виде графиков (тренды);
- вторичная обработка информации;
- формирование сводок и других отчетных документов по созданным на этапе проектирования шаблонам.

К интерфейсу, созданному на базе программного обеспечения SCADA, предъявляется несколько фундаментальных требований:

- он должен быть интуитивно понятен и удобен для оператора/диспетчера;
- единичная ошибка оператора не должна вызывать выдачу ложной команды управления на объект.

5.1.2 Архитектурное построение SCADA-систем

На начальном этапе развития (80-е годы) 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 90-х) программные средства для операторских станций становятся самостоятельным продуктом.

Одной из первых задач, поставленных перед разработчиками SCADA, стала задача организации многопользовательских систем управления, то есть 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.

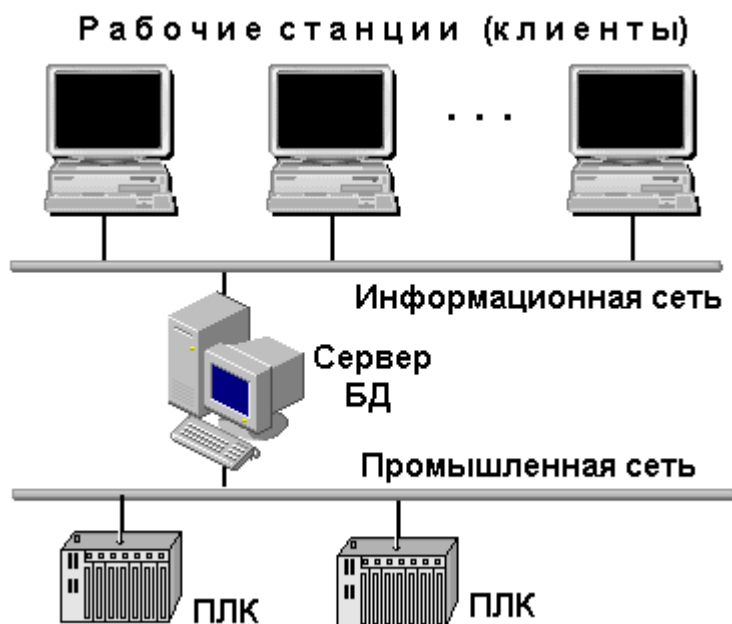


Рис.5.2. Клиент-серверная архитектура.

Клиент-серверная архитектура 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. ПО 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.

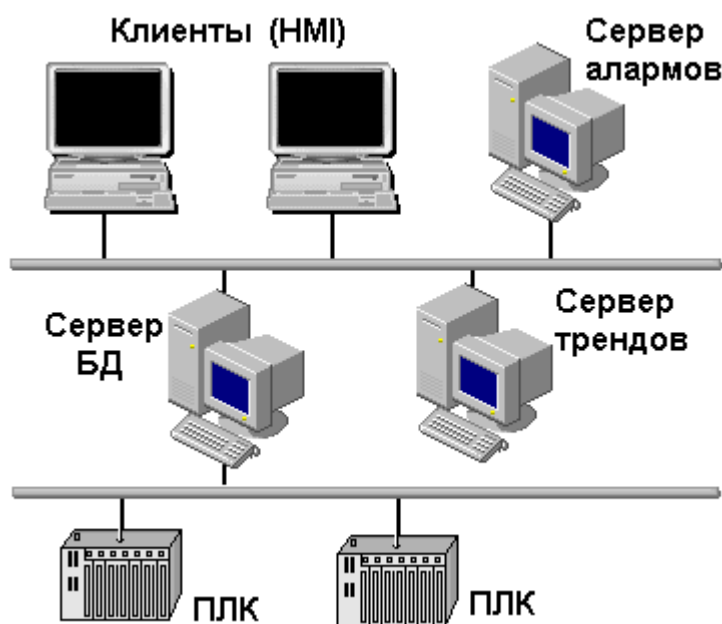


Рис. 5.3. Архитектура модульной SCADA.

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 рис. 5.3.

В клиент-серверной архитектуре системы управления, представленной на рис. 5.3, функции сбора и хранения данных, управления

алармами и трендами распределены между тремя серверами. Функция НМІ 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 истечении 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.

5.2 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.

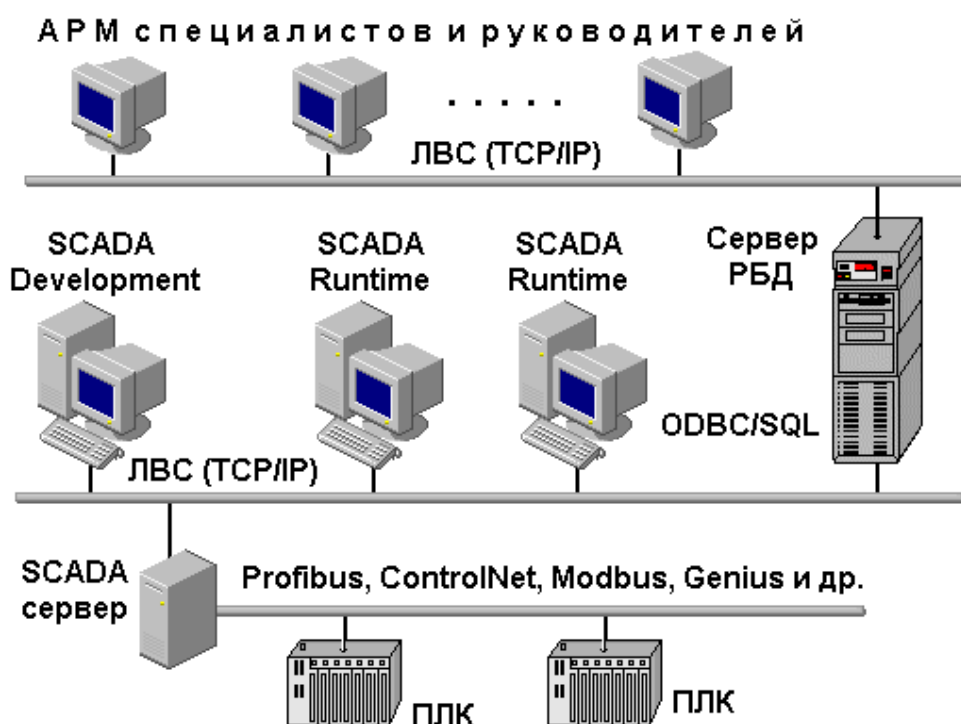


Рис.5.4. Интеграция SCADA в систему управления.

Концепция открытых систем предполагает свободное взаимодействие программных средств SCADA с программно-техническими средствами разных производителей. Это актуально, так как 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. (рис.5.4.). 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.

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:

- 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;
- SQL (Structured Query Language) – язык структурированных запросов.

Программное обеспечение SCADA должно взаимодействовать с контроллерами для обеспечения человеко-машинного интерфейса с системой управления. К контроллерам через модули ввода/вывода подключены датчики технологических параметров и исполнительные устройства.

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 (рис. 5.5):

- DDE (Dynamic Data Exchange - динамический обмен данными);
- обмен по собственным (известным только фирме-разработчику) протоколам.



Рис. 5.5. Обмен информацией с помощью DDE-протокола.

Взамен DDE компания Microsoft предложила более эффективное и надежное средство передачи данных между процессами – OLE (см. ниже). А вскоре на базе OLE появился новый стандарт OPC, ориентированный на рынок промышленной автоматизации.

5.3 OPC-интерфейс

OPC – это аббревиатура от OLE for Process Control (OLE для управления процессами). Технология OPC основана на разработанной компанией Microsoft 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, предложенный компанией Microsoft, 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.

Так что же такое OPC ? 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).

Более популярно 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. Идентификаторы элементов – свои у каждого сервера. По уникальному идентификатору сервер умеет находить нужное значение в соответствующем устройстве (например, контроллере). Для ПЛК идентификатор элемента обычно соответствует номеру регистра. Дополнительно сервер может снабжать полученные данные меткой времени.

Использование технологии OPC в настоящее время возможно лишь в операционных системах, построенных на технологии OLE/COM, т.е. в ОС Microsoft Windows 95/98 и Windows NT. Идут разработки поддержки этой технологии для операционной системы UNIX.

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

Развивающая стандарт OPC некоммерческая организация OPC Foundation (<http://www.opcfoundation.org>), насчитывает свыше 200 членов. В нее входят почти все ведущие мировые производители программно-аппаратных средств автоматизации.

Хотя стандарт OPC и основан на универсальном фундаменте - COM/DCOM, он разрабатывался специально для использования в промышленной автоматизации и поэтому имеет вполне содержательную концептуальную сторону.

Стандарт состоит из трех основных спецификаций:

- доступ к данным реального времени (Data Access);
- обработка тревог и событий (Alarms & Events);
- 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.



Рис.5.6. Схема взаимодействия с управляющими устройствами

Раньше разработчикам клиентских приложений приходилось писать множество драйверов (см. рис.5.6.) для взаимодействия с каждым из используемых управляющих устройств (контроллеров).



Рис.5.7. Стандарт OPC

Стандарт OPC позволяет написать лишь один-единственный драйвер (рис.5.7) для доступа к данным, поступающим в едином формате от самых различных источников.

OPC-интерфейс допускает различные варианты обмена: с физическими устройствами, с распределенными сетевыми системами управления и с любыми приложениями (рис.5.8). На рынке имеются и инструментальные пакеты для написания OPC-компонентов.

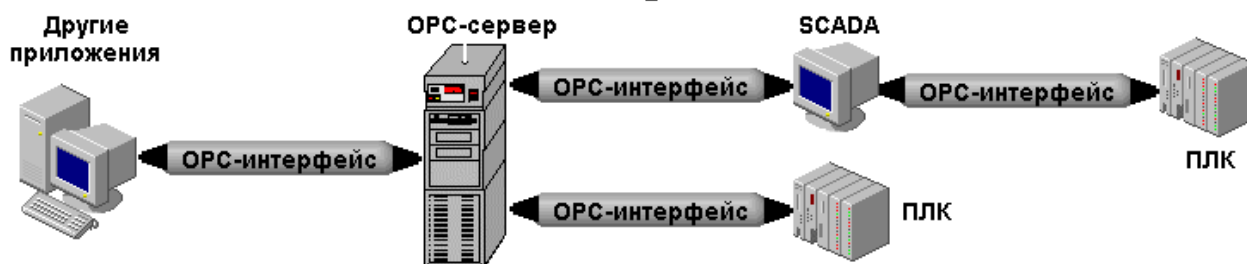


Рис.5.8. Обмен данными по OPC-интерфейсу.

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 языке типа C или Visual Basic уменьшается. Все это способствует расширению области применения SCADA-систем.

5.4 ActiveX-объекты

ActiveX – это технология MicroSoft, основанная на COM/DCOM (см. выше) и предназначенная для написания сетевых приложений. Она предоставляет программистам наборы стандартных библиотек, значительно облегчающих процесс кодирования.

Стандарт ActiveX позволяет программным компонентам взаимодействовать друг с другом по сети независимо от языка программирования, на котором они написаны (Visual Basic, Visual C++, Borland Delphi, Borland C++, любые средства разработки на Java).

ActiveX обеспечивает некий «склеивающий раствор», с помощью которого отдельные программные компоненты на разных компьютерах «склеиваются» в единую распределенную систему.

Технология ActiveX включает в себя клиентскую и серверную части.

Серверная часть технологии ActiveX реализована с помощью Microsoft Internet Information Server (IIS).

Клиентская технология ActiveX реализуется на машине-клиенте с помощью библиотек, поставляемых вместе с Microsoft Internet Explorer, являющимся полнофункциональным Web-браузером (WWW - World Wide Web) и контейнером для ActiveX-элементов. Сегодня технология ActiveX успешно внедряется в системы, функционирующие на Windows-платформе. Нет сомнения, что в ближайшее время эти технологии будут использоваться и на других платформах, так как информационные технологии развиваются очень высокими темпами.

Какое же отношение технология ActiveX имеет к SCADA-системам? Разработчики SCADA-программ на платформе WindowsNT/2000/XP воспользовались этой технологией Microsoft. Сейчас уже многие SCADA являются контейнерами для ActiveX-объектов. А это значит, что огромное количество готовых к многократному использованию ActiveX-объектов, создаваемых многочисленными производителями подобного программного

продукта, могут встраиваться с минимальным программированием в SCADA-приложения. И тогда процесс разработки человеко-машинного интерфейса будет напоминать работу с конструктором, заключающуюся в подборе и встраивании готовых компонентов.

Фильтры:							
Имя	Начальный момент	Конечный момент	Нижн. приоритет	Верхн. приоритет	Имя канала	К	
<div> <div>◀</div> <div></div> <div>▶</div> </div>							
События:							
Тип	Дата	Время	Имя канала	Кодировка	Сообщение	Время квитирования	Оператор
<div> <div>◀</div> <div></div> <div>▶</div> </div>							

Итак, открытость программного обеспечения SCADA обеспечивается целым рядом факторов, а именно:

- наличием специальных драйверов для связи SCADA с наиболее

- наличием специальных инструментальных средств для
создания новых

- возможностью их работы в типовых операционных системах;

- наличием типовых программных интерфейсов (DDE, OLE, OPC, ActiveX, ODBC, SQL и др.), связывающих ПО SCADA с другими программно-аппаратными средствами системы управления, включая и СУБД.

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

5.5 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.

В периодической прессе последних лет 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.

Но за услуги, как известно, надо платить. Поэтому весьма существенным критерием при организации клиентского узла (АРМ) является его стоимость (аппаратное и программное обеспечение).

В настоящее время существует несколько решений поставленной задачи, базирующихся на применении различных технологий. Но и стоимость предлагаемых решений тоже различна. Отсюда и появились такие понятия, как «бедные/богатые и тонкие/толстые клиенты».

Самыми простыми и распространенными клиентскими приложениями в настоящее время являются клиенты в локальной сети (рис. 5.10). Такие клиентские приложения в SCADA-системах традиционно объединяются с серверными приложениями протоколами локальных сетей. Часто таким протоколом является TCP/IP.

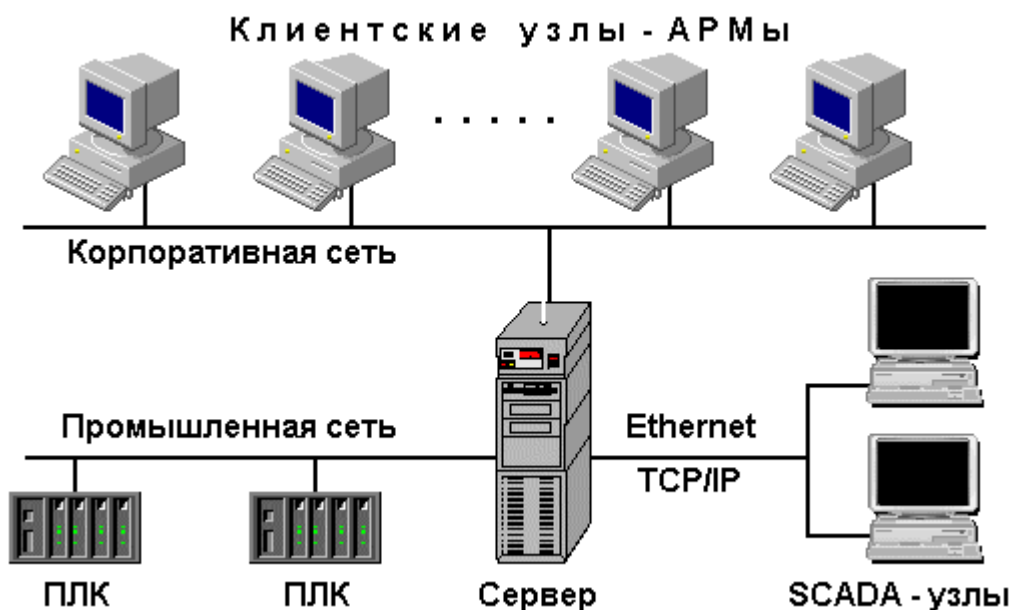


Рис. 5.10. Организация доступа к информации через локальную сеть.

Большинство современных SCADA-пакетов работает на платформах Windows 2000/NT/XP. Отсюда следует, что 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 («богатые» клиенты). 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.

5.6 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. Наиболее удачные решения предложили корпорация Microsoft (Windows 2000 Terminal Services) и компания Citrix (Metaframe). ПО Metaframe - это дополнение к Windows 2000 Terminal Services, которое дает возможность использовать на клиентских компьютерах операционные системы, отличные от Windows, например, Linux или Macintosh.

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). Таким образом, речь идет о технологиях терминального доступа с использованием так называемых «тонких» клиентов.

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 (рис. 5.11).

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).

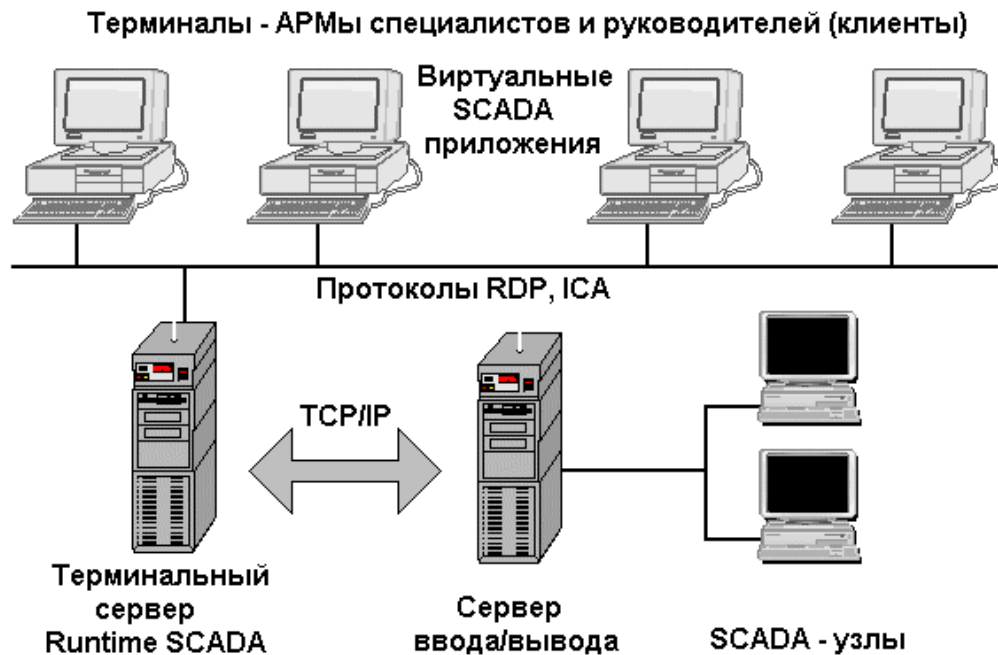


Рис. 5.11. Архитектура терминал-сервер.

Благодаря терминальным протоколам в качестве клиентов можно использовать рабочие станции, начиная с «супер-тонких» бездисковых, работающих на платформах Linux/CE, Windows 3.11/95/98, до станций, функционирующих под управлением Windows NT/2000.

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.

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. Компания Wonderware внедрила терминал-серверную технологию для SCADA-системы InTouch версии 7.1. Появление версий iFIX (Intellution/GE Fanuc), поддерживающих ОС Windows 2000, открыло возможность применения ПО iClient Terminal Server для поддержки многопользовательской работы «тонких» клиентов. Не отстали и другие ведущие производители SCADA-продуктов.

5.7 Internet/Intranet- технологии

Очевидным плюсом сети Internet является ее уникальная протяженность и распределенность, что позволяет передавать информацию через тысячи километров между любыми двумя точками земного шара. Кроме этого, сеть отличается уникальной стандартизацией передаваемых данных, что обеспечивает одинаковую читаемость, информативность и однозначность передаваемых данных вне зависимости от операционной системы, в которой работает компьютер – Windows 9x/NT/2000, Unix или OS/2. Эту возможность дает применение стандартного протокола передачи TCP/IP.

Однако наряду с достоинствами Internet следует отметить и основной недостаток - очень низкая скорость передачи данных. Сочетание различных физических сред передачи информации и таких свойств протокола TCP/IP как неопределенность времени получения ответа ведут к тому, что

передаваемая информация будет передана правильно и без потерь, но заранее сказать, какое время это займет, нельзя. Очевидно, что Internet -технологии мало подойдут для применения в системах с быстротекущими процессами, однако там, где время не является критичным, Internet является приемлемым решением по обеспечению своевременной и точной информацией оператора системы, инженера-технолога или руководителя.

Удобство и популярность Internet стали основной причиной того, что Web-технологии начали активно применяться во внутренних информационных системах предприятий. Каждое предприятие рано или поздно сталкивается с необходимостью автоматизации своей деятельности. Одной из первых ставится задача централизованного хранения информации и доступа к ней. Если раньше такие технологии использовались лишь на самом верхнем уровне управления - АСУП, то в последнее время все большее распространение они получают и в системах уровня АСУ ТП (в системах класса SCADA/HMI).

Внутренние информационные системы предприятия, построенные с использованием Web-технологий, получили собственное название – «Intranet» (интранет - внутренняя сеть). Интранет совсем не обязательно должна ограничиваться локальной сетью предприятия - она может объединять несколько предприятий, находящихся на значительных расстояниях. Отличие Intranet от Internet заключается в том, что ее информационные ресурсы и пользователи объединены общими задачами и принадлежностью одному коллективу.

Так какие же конкретно технологии и системы можно применить для совместной работы систем АСУ ТП на уровне HMI/SCADA и Интернет? Ниже предлагается краткий обзор уже существующих и на практике широко используемых технологий на базе Internet.

Самым простым, но очень действенным методом интеграции HMI/SCADA в Интернет является использование электронной почты в качестве средства оповещения при появлении новых записей в журнале

тревог. Этими возможностями обладают большинство SCADA-систем, имеющихся сейчас на рынке. Электронная почта, кроме прямой отправки письма адресату через Интернет, может использовать и различные «перевалочные пункты», например, шлюзы пейджинговых компаний для отправки сообщения непосредственно на пейджер адресата.

Гораздо более информативной является возможность генерирования отчетов о текущем положении дел на объекте в стандарте HTML. Для использования этого метода SCADA-система формирует отчет с диаграммами, графиками, таблицами в виде HTML-файла, который сохраняется на диске (локального или удаленного компьютера). Периодичность обновления отчета зависит только от настроек SCADA-системы и не очень влияет на производительность остальных компонентов системы управления. Сохраненный файл, в свою очередь, может использоваться Web-сервером для предоставления доступа к этим данным через сеть Интернет из любой точки земного шара, используя обыкновенный Web-браузер. Метод не предполагает возможности воздействовать на объект через систему автоматизации, доступны лишь функции мониторинга.

Большие возможности предоставляет супервизорное управление через Интернет. Для осуществления этого метода управления системой АСУ ТП необходима SCADA-система, поддерживающая функции управления по сети TCP/IP. При этом функционирующая на удаленном компьютере SCADA-система должна иметь в своем распоряжении копию проекта, включая описание используемых переменных, графические объекты, скрипты и т. п. («толстые» клиенты). В этом случае пересылаемые по сети Internet данные будут содержать только текущие значения параметров, считанных из контроллеров (сбор данных), и команды удаленного компьютера (управление). Примерами реализации таких систем могут служить программы WebCast (фирма Intellution, пакет iFix), NetLink (AdAstra, Trace Mode) и Scout (Wonderware, InTouch).

Другую концепцию предлагает метод связи через браузер (Web- browser). В этом случае используется технология так называемого «тонкого» клиента. При установке связи между Web-браузером и SCADA-сервером в локальный компьютер осуществляется загрузка данных о работающем в системе проекте (включая графические объекты). В этом случае вся математическая обработка данных происходит на удаленном сервере, на локальном же компьютере идет только представление данных, используя ActiveX или другую Web-технологию. Примером реализации могут служить наборы подключаемых модулей WebClient (US Data, FactoryLink/MonitorPro), WebActivator (AdAstra, Trace Mode).

Особое место в Web-технологиях занимает сбор данных через Интернет от удаленных контроллеров. Этот метод фактически соответствует традиционно принятой структуре построения АСУ ТП с использованием SCADA-систем, но в данном случае между самой системой и ПЛК может лежать не одна тысяча километров. В такой конфигурации может работать любая SCADA-система, умеющая посылать сообщения по протоколу TCP/IP (что могут делать практически все системы). Аналогично и ПЛК могут работать в такой системе, если они имеют Ethernet или последовательный порт с поддержкой TCP/IP. Практически все крупнейшие производители контроллеров имеют такие модели.

Совершенно новой технологией для управления через Интернет являются встраиваемые в ПЛК Web-серверы. Сейчас можно говорить лишь о наметившихся перспективах. Одна из главных особенностей этой «революционной» технологии (кроме универсальности связи с ПЛК) - отказ от использования SCADA-систем. Web-сервер находится в контроллере, который подключен непосредственно к сети Internet. Имеющийся в контроллере сопроцессор осуществляет формирование необходимых HTML-страниц и связывает их с данными, поступающими с объекта. Однако в данном случае основная тяжесть работы по обработке данных будет ложиться на плечи самого контроллера, который вынужден будет кроме

первичной обработки данных осуществлять и вторичную обработку, что может потребовать применения гораздо более мощного процессора ПЛК, чем в случае работы без Web-сервера.

Во всех Internet/Intranet-решениях по обмену данными кроме технологического сервера как поставщика данных и клиента как получателя информации задействован Web-сервер (рис. 3.12). Информация на сервере хранится в виде страниц, на которых, кроме текста, могут находиться разные объекты: графические изображения, аудио - и видеоролики, формы для ввода данных, интерактивные приложения и т.д.

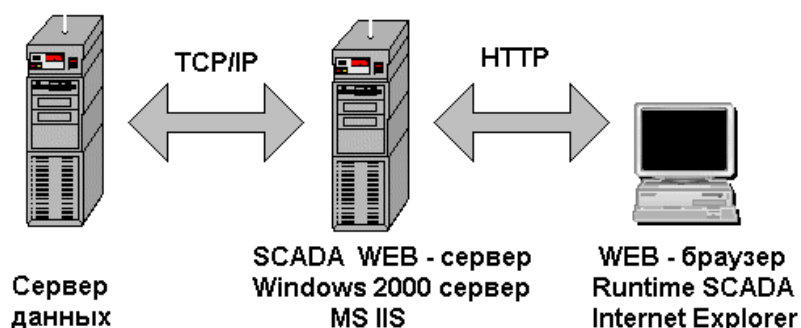


Рис. 5.12. Интеграция SCADA и Internet.

Взаимодействие между Web-сервером и клиентами осуществляется на основе протокола HTTP (HyperText Transfer Protocol - протокол передачи гипертекста).

Для просмотра приложений Web-клиентом могут использоваться навигатор Microsoft Internet Explorer соответствующей версии или SCADA-система в режиме Runtime.

Web-сервер работает на базе Microsoft Internet Information Server (IIS) и связывает установленные на нем приложения с Internet.

Практически все ведущие фирмы-разработчики SCADA-систем занимаются созданием программных продуктов с использованием Internet-технологий, в том числе и технологий с использованием «тонких» клиентов.

5.8 Интегрированные SCADA-системы

Одним из наиболее значимых факторов развития SCADA-систем становится то, что некоторые ведущие производители расширяют функции SCADA «по вертикали» иерархии многоуровневой системы управления.

С одной стороны, идет расширение функций в сторону непосредственного управления технологическими процессами (автоматическое регулирование и программно-логическое управление). Функции непосредственного управления реализуются в пакетах прикладных программ как для контроллеров, построенных на основе PC-совместимых контроллеров (SoftPLC), так и для компьютерной реализации функций непосредственного управления (SoftControl).

Широкое использование IBM PC платформы в контроллерах (softlogic) началось в 90-х годах XX века и было обусловлено многими факторами, один из которых – лучшее соотношение «производительность - цена». А для России того времени это было определяющим. И вот отечественная фирма AdAstrA интегрирует свою SCADA-систему с системой программирования PC-контроллеров. Так появилась новая технология сквозного программирования компонентов нижнего и верхнего уровней АСУТП.

Говоря о компьютерной реализации функций непосредственного автоматического управления технологическим процессом, следует отметить, что практически все известные инструментальные SCADA-системы обеспечивают эту возможность. Хотя такое совмещение и позволяет экономить на аппаратных средствах, оно может иметь ряд негативных последствий. Во-первых, операционная система операторской станции может не обеспечить необходимую для конкретного технологического процесса скорость реакции SCADA-системы. Во-вторых, никто не гарантирован от «зависания» системы, хотя для некоторых объектов (инерционных) это может быть и не критично.

С другой стороны, первичная информация, собранная SCADA-системами от технологических установок и производств для принятия оперативных (тактических) решений на уровне операторов/диспетчеров, должна быть доступна в реальном времени всем уровням управления с целью ее анализа и принятия управленческих (стратегических) решений. Но до недавнего времени в силу различных причин (подробнее см. главу 4) эта задача не была решена, что являлось существенным тормозом повышения эффективности предприятий.

Для ее решения на рынке программных продуктов стал появляться новый класс программного обеспечения – интегрированные пакеты промышленной автоматизации. В этих пакетах SCADA/HMI является лишь одним из компонентов. Другой важнейший компонент таких систем – базы данных реального времени или архивы исторических данных, предназначенные для хранения огромных массивов информации с возможностью доступа к ней с различных АРМ. Сюда же можно отнести специализированные пакеты для управления периодическими процессами, для выявления и минимизации простоев оборудования, для просмотра производственной информации с помощью Интернет-технологий и т. п.

К классу интегрированных систем можно отнести такие программные продукты ведущих производителей SCADA, как FIX Dynamics (Intellution/GE Fanuc), FactorySuite 2000 (Wonderware) и другие. Эти системы представляют собой мощные программные комплексы, обеспечивающие интеграцию системы управления производством в целом. Использование в системах разных уровней единого стиля оформления, единой терминологии, инструментария, служебных средств и т. д. значительно облегчают разработчикам проектирование систем, а предприятиям – их освоение и эксплуатацию.

5.9 Надежность SCADA-систем

Понятие надежности SCADA-пакетов включает в себя два аспекта: надежность самого программного продукта SCADA и возможность программного резервирования компонентов системы в различных вариантах.

Надежность SCADA-пакета определяется несколькими характеристиками: надежностью операционной системы, наличием средств сохранения данных и конфигурации при сбоях, наличием средств автоматического перезапуска системы.

По надежности современные SCADA-продукты, также как и по функциональности, незначительно отличаются друг от друга. Тем не менее, при выборе пакета можно обратить внимание на список его внедрений. Наличие в таком списке проектов для опасных и ответственных производств, проектов с большим числом параметров, территориально и функционально распределенных АРМов говорит о достаточно высокой надежности SCADA-пакета.

Но система управления может полностью выйти из строя не только по причине отказа программного обеспечения, но и оборудования.

Получившая наиболее широкое распространение распределенная система управления, представленная на рис. 5.13, выйдет из строя, если всего лишь в одном компоненте (сервере) возникнет неисправность.

Реализация SCADA-пакетами функций резервирования позволяет устранять отказы в системе без потери ее функциональных возможностей и производительности. Программное обеспечение SCADA поддерживает реализацию резервирования различных компонентов системы управления как вследствие особенности архитектуры, так и наличия встроенных механизмов.

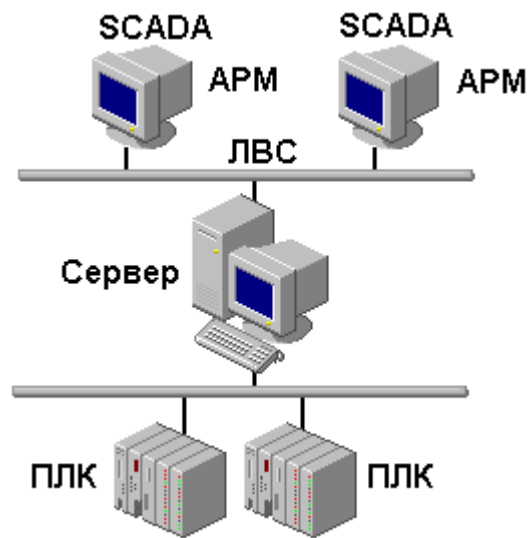


Рис. 5.13. Сетевая архитектура SCADA.

Дублирование сервера ввода/вывода. Для повышения надежности системы управления достаточно явно просматривается вариант с резервированием сервера (рис.5.14). Здесь возможны два варианта. В одном случае оба сервера (основной и резервный) взаимодействуют с устройствами ввода/вывода, удваивая нагрузку на промышленную сеть и снижая производительность системы. В штатном режиме клиенты взаимодействуют с основным сервером. При выходе его из строя они направляют свои запросы к резервному серверу.

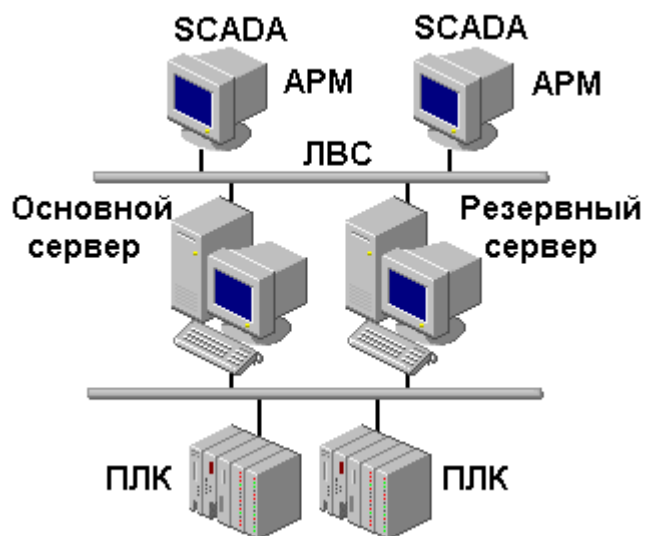


Рис.5.14. Резервирование сервера.

В распределенной клиент-серверной архитектуре SCADA-систем лишь один (основной) сервер взаимодействует с контроллерами. При этом основной сервер постоянно обновляет базу данных резервного сервера, обеспечивая его постоянную готовность.

Резервирование сети и контроллеров.

Структура, приведенная на рис. 5.14, увеличивает надежность системы, устраняя одно из основных «слабых» мест – отказ сервера. Другим «слабым» местом распределенной системы управления может быть сама сеть. Выход ее из строя нарушает управление, так как станции операторов/диспетчеров в этом случае оказываются отрезанными от системы. Повышение надежности системы управления обеспечивается дополнительной сетью (рис. 5.15).

Большинство контроллеров может поддерживать дополнительную (резервную) связь с сервером ввода/вывода. При отказе основного канала гарантируется обмен данными между контроллером и сервером.

Достичь полного резервирования можно путем дублирования контроллеров (рис. 5.15).

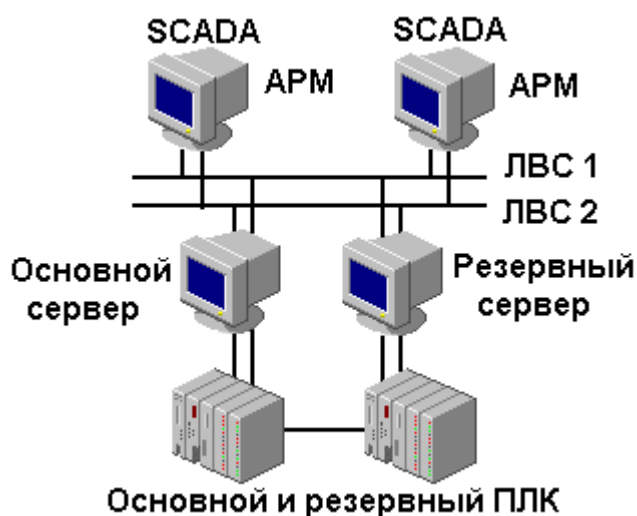


Рис. 5.15. Варианты резервирования.

Рассмотренные выше способы повышения надежности системы управления хорошо известны. Важным здесь является то, что именно встроенные в SCADA-систему механизмы позволяют конфигурировать распределенную клиент-серверную архитектуру, определяя на стадии проектирования основные и резервные устройства системы управления. А в режиме исполнения именно SCADA-система определяет неисправность того или иного компонента системы и автоматически производит переключение на резервное оборудование, предупреждая об этом оперативный персонал.

5.10 Программно-аппаратная платформа

К этой группе можно отнести следующие характеристики: компьютерная платформа, операционная система, конфигурация компьютера (частота процессора, требуемые ресурсы оперативной и дисковой памяти), возможность переноса приложений в другую операционную систему.

Анализ платформ и операционных систем необходим, поскольку они определяют возможность распространения SCADA-системы на имеющиеся вычислительные средства и стоимость системы.

Программное обеспечение SCADA, как и любое другое ПО, выполняется под управлением той или иной операционной системы. Какая же операционная система наиболее приемлема для программного обеспечения верхнего уровня? Обязательно применение ОСРВ или достаточно операционной системы общего назначения? Этот вопрос обсуждался на протяжении нескольких лет в различных периодических изданиях, посвященных автоматизации технологических процессов. В итоге, компромисс найден: требования к параметрам операционной системы должны определяться автоматизируемым объектом и прикладной задачей.

С одной стороны, в нефтегазовой отрасли существует довольно широкий класс инерционных объектов. Нельзя также и забывать, что

неотъемлемой частью верхнего уровня АСУ ТП является человек, время реакции которого на события недетерминировано и зачастую достаточно велико. И, наконец, нельзя не учитывать тенденции развития мирового рынка программного обеспечения.

В результате, подавляющее большинство SCADA-систем реализовано на MS Windows-платформах (Windows NT/2000). Это и InTouch, и FIX, и Genesis, и российский Трейс Моуд. Из четырнадцати систем, приведенных выше, двенадцать предназначены для работы в различных вариантах ОС MS Windows. Здесь, безусловно, сказались позиции компании Microsoft на рынке операционных систем. Известно, что именно компания Microsoft была и остается «законодателем моды» в этом классе программного обеспечения.

А вот такие популярные SCADA-системы, как RealFlex, Sitex, RTWin функционируют под управлением операционной системы реального времени QNX. Эта ОСРВ для IBM PC является одной из наиболее широко используемых при построении систем управления и сбора данных прежде всего за счет того, что гарантирует время реакции системы в пределах от нескольких десятков микросекунд до нескольких миллисекунд (в зависимости от быстродействия ПЭВМ и версии QNX).

Широко известная SCADA FactoryLink имеет целый список поддерживаемых ей программно-аппаратных платформ: OS/2 (IBM PC), UNIX (IBM PC), VMS (VAX), HP-UX (HP 9000) и MS Windows (IBM PC).

Компьютерные ресурсы, требуемые для установки и нормального функционирования различных компонентов SCADA-систем, определяются многими факторами, в том числе, назначением сетевого компьютера (рабочая станция оператора, сервер БД, АРМ специалиста и т. п.), количеством обрабатываемых переменных, используемой операционной системой (Windows 95/98/NT/2000, QNX) и т. п.

В качестве клиентских компьютеров наибольшее распространение в настоящее время находят IBM-совместимые ПК (от 486 до Pentium II 500/800 МГц).

Оперативная память, требуемая для SCADA-пакетов различных производителей, колеблется от 32 до 128/256 Мб.

Требования к свободному объему памяти на жестком диске также достаточно минимальны (100 – 200 Мб).

Могут накладываться также ограничения на качество и объем памяти видеокарты, разрешение экрана монитора, размеры монитора.

Требования к аппаратным средствам, призванным поддерживать серверные функции, могут быть существенно более высокими. Это относится и к объему оперативной памяти, и к объему жесткого диска, который может измеряться уже десятками и сотнями Гб.

С другой стороны, многие клиентские компьютеры при использовании современных сетевых технологий, таких, как архитектура Server/Terminal, Internet-технологий (WEB-сервер), могут быть достаточно слабыми конфигурациями (IBM 286/386) с минимальными требованиями как к оперативной, так и к дисковой памяти, а то и вовсе бездисковыми.

Масштабируемость - это способность ПО SCADA наращивать размеры системы управления, обеспечивая при этом преемственность по отношению ко всем ранее установленным программно-аппаратным средствам.

С ростом мощности компьютеров и соответствующим ростом информационной мощности операторских станций SCADA-системы становятся масштабируемыми. Они выпускаются в различных вариантах, которые при сохранении в целом функционального профиля поддерживают от нескольких десятков или сотен до десятков тысяч входов/выходов (лицензируемых точек).

Естественно, стоимость таких пакетов различна: чем больше переменных поддерживает SCADA-пакет, тем он дороже. Но это удобно потребителю - можно приобрести пакет под проект практически любого масштаба.

Градация количества лицензируемых точек в различных SCADA-пакетах различна. В ряде пакетов она более равномерна, чем в других. Например, на рынке программных продуктов можно найти SCADA-пакеты на 75, 150, 500, 1 500, 5 000, 15 000, 50 000, 150 000 и 450 000 переменных. При этом учитываются только внешние переменные, считываемые с устройств ввода/вывода. Внутренние переменные, которые будут определены разработчиком при проектировании, не являются лицензируемыми (бесплатны), хотя и будут храниться в памяти компьютера или на жестком диске. Другие фирмы-производители SCADA в общее количество лицензируемых точек включают и внутренние переменные. Например, приобретение такого пакета на 500 лицензируемых точек означает следующее. Если в соответствии с проектом разработчику потребуется создать 100 внутренних переменных, то система способна будет обрабатывать лишь 400 переменных ввода/вывода. Но и о возможном расширении системы не надо забывать.

При расширении системы управления, например, увеличении количества обрабатываемых переменных, создании новых станций для перераспределения вычислительной нагрузки между компьютерами в системе SCADA-пакеты снабжаются встроенными механизмами, которые позволяют разработчикам реализовать такие возможности. С точки зрения удобства использования этих механизмов все SCADA-пакеты различны. Многие фирмы предлагают системы, в которых основная работа по конфигурированию компьютеров клиент-серверной архитектуры хорошо автоматизирована.

5.11 Эксплуатационные характеристики

К этой группе можно отнести:

- удобство интерфейса среды разработки (это качество обеспечивается применением Windows –подобных интерфейсов), полнота и наглядность представления функций системы на экране, удобство и информативность контекстных и оперативных подсказок, справочной системы;

- качество документации - полнота, ясность и наглядность описания системы, применение установившейся терминологии, русификация, уровень русификации (экраны, подсказки, справочная система, системные сообщения, документация);

- полнота/недостаточность средств диагностики состояния системы при сбоях и отказах, нарушениях внешних связей; трудоемкость и уровень автоматизации работ при инсталляции и конфигурировании системы; возможности внесения изменений в систему без ее остановки и т.д.

- положение программного продукта на рынке: дилерская сеть, консультационная поддержка, наличие «горячей линии», обучение, условия обновления версий (upgrade), количество инсталляций и т. д.

Опыт работы авторов на факультете повышения квалификации специалистов в области автоматизации показывает, что на местах специалисты часто испытывают трудности в освоении SCADA из-за отсутствия качественной документации на приобретенные программные продукты. Учитывая далеко не поголовное знание английского языка программистами и, тем более, технологами, подробная и качественная документация на русском языке просто необходима.

Эксплуатационные характеристики в значительной мере носят субъективный характер и не могут быть оценены количественно. О них

можно судить только по результатам практического использования программного продукта: тестирования, апробирования, анализа, опыта промышленного внедрения. Косвенной характеристикой качества и отработанности крупнотиражного программного продукта служит его положение на рынке, поскольку большое число реализаций продукта свидетельствует о солидном опыте применений, учтенном при обновлениях продукта. Количество инсталляций SCADA-пакетов крупнейших производителей, таких как Wonderware и Intellution (GE Fanuc), перешагнуло уже за 200 тысяч.

5.12 Основные подсистемы SCADA-пакетов

Создание современной системы управления потребует от разработчика некоторого набора знаний применяемого в проекте SCADA-пакета. Что же надо знать о SCADA разработчику, приступая к созданию проекта?

Для реализации базовых функций SCADA-системы разработчику потребуется, как минимум:

- организовать взаимодействие SCADA-пакета с аппаратными средствами автоматизации (контроллерами);
- создать графический интерфейс для диспетчера/оператора, т.е. отображение технологического процесса и значений параметров на динамизированных мнемосхемах;
- обеспечить оперативный персонал информацией о ситуациях, связанных с отклонением технологических параметров от заданных значений, о предаварийном состоянии оборудования и т.п.;
- настроить систему регистрации и архивирования данных и их представление на мониторе в виде трендов, что позволит оператору и специалистам проводить анализ состояния процесса и оборудования.

Можно перечислить еще ряд типовых задач, решаемых в процессе разработки системы управления (шаблоны отчетов, статистическая обработка данных, взаимодействие с РБД и др.). Более того, практически каждый производитель SCADA предлагает свои специализированные механизмы, направленные на повышение информативности операторского интерфейса, удобства работы с ним. Безусловно, все они не могут быть рассмотрены в рамках данного учебного пособия.

Для разработки качественного операторского интерфейса разработчику также необходимо владеть встроенным в SCADA-пакет языком программирования. С его помощью создаются так называемые сценарии (скрипты) – фрагменты программ, обеспечивающие оперативный персонал своевременной информацией и облегчающие управление процессом.

Таким образом, SCADA – это набор инструментов (подсистем) для решения перечисленных выше задач.

Взаимодействие SCADA-пакетов с контроллерами.

Сбор данных и управление предполагают перемещение информации между объектом и станцией оператора. Обязательным промежуточным звеном в этой цепочке является контроллер. Взаимодействие контроллера, как поставщика и приемника информации, со SCADA-системой обеспечивается драйверами. Какие драйверы поставляются с тем или иным SCADA-пакетом, как установить драйвер, какие диалоги при этом должны быть заполнены, какая информация потребуется разработчику, имеется ли инструментарий для разработки собственных драйверов? На эти и многие другие вопросы еще предстоит ответить.

Кроме этого, система управления включает, как правило, еще ряд компонентов: серверы данных, рабочие станции специалистов и т.п. Все компоненты системы управления объединены между собой промышленной (управляющей) сетью. Системы управления отдельными технологическими процессами (АСУТП) и другие подразделения предприятия объединены между собой в локальную вычислительную сеть (ЛВС). И здесь возникает

еще целый ряд вопросов: какие популярные промышленные сети поддерживает SCADA-пакет, какие протоколы обмена с типовыми реляционными базами данных могут быть использованы?

Графический интерфейс.

Качество отображения информации на мнемосхемах определяется характеристиками графических возможностей пакетов. К ним можно отнести графический редактор, возможность создания объемных изображений, наличие библиотек и разнообразие графических заготовок и готовых объектов, богатство инструментария, многообразие динамических свойств элементов мнемосхем, форматы импортируемых изображений, наличие инструментария для создания растровых рисунков, наличие и возможности многооконных режимов и т. п.

При создании компонентов операторских интерфейсов (например, мнемосхем) разработчику приходится использовать графические объекты, представляющие собой технологические аппараты (колонны, емкости, теплообменники и т. д.), участки трубопровода и такие устройства, как клапаны, насосы, электродвигатели, контроллеры, компьютеры и т. д. Как правило, это сложные объекты, полученные объединением множества простых объектов или рисунки типа Bitmap.

Создание каждого из этих объектов требует большого времени и может значительно затянуть разработку проекта. Для ускорения работы над проектом практически все SCADA-пакеты предлагает разработчику библиотеки готовых объектов, включающие сотни и тысячи графических компонентов.

6. Методы доступа и защиты данных

Защита данных должна обеспечить их безопасность и секретность. Эти две функции тесно связаны между собой, и для их реализации используются одни и те же ресурсы, но между ними существует и принципиальное различие. Под функцией безопасности понимается защита

данных от непреднамеренного доступа к данным и искажению их со стороны пользователя, а также при сбоях аппаратно-программных средств. Под функцией секретности понимается защита данных от преднамеренного доступа. В таких случаях обеспечение секретности требует деления всей БД на разделы:

- общедоступные;
- конфиденциальные.

После деления, Администраторы БД должны принять меры по обеспечению секретности, их, в свою очередь, можно разделить:

- технические;
- организационные.

В дальнейшем будем рассматривать вопросы, связанные с техническими аспектами обеспечения секретности и безопасности, при этом СУБД не может разрешить выполнение каких-либо операций над БД, пользователю, не получившему на это соответствующего права.

При решении вопросов защиты данных в обязанности администратора БД входит:

- классификация данных в соответствии с их использованием;
- определение прав доступа отдельных пользователей к определенным группам данных и определение ограничений на характер операций, выполняемых пользователями с этими данными;
- организация системы контроля доступа к БД;
- тестирование вновь создаваемых средств хранения данных;
- периодическое проведение проверок правильности функционирования системы защиты данных;
- исследование возникающих случаев нарушения правильности системы защиты БД;
- исследование современных технологий в области защиты данных.

Рассмотрим возможные уровни санкционированного доступа для различных типов пользователей:

1. неограниченный доступ ко всем отношениям БД и их поколениям;
2. неограниченный доступ к группе отношений и их поколениям;
3. ограниченный доступ к группе отношений и их поколениям.

Для отношений различают следующие уровни доступа:

1. неограниченный доступ ко всему отношению и для всех типов операций;
2. доступ к любой части отношения, но с правом изменения только части отношения;
3. неограниченный доступ к части атрибутов для всех типов операций;
4. доступ к любой части отношения, но без права изменения его содержимого;
5. доступ только к части атрибутов, с правом изменения значения части доступных атрибутов;
6. доступ только к части атрибутов, без права изменения их значения, и запрет доступа к остальным атрибутам;
7. неограниченный доступ только к одному кортежу отношения для всех типов операций;
8. доступ только к одному или группе кортежей, без права изменения этих кортежей;
9. доступ, в соответствии с п.1...п.8, но с ограничением по времени;
10. запрет на доступ к любым частям отношения для всех типов операций.

Данный список уровней доступа пользователей отражает как бы требуемый диапазон работы системы и ее необходимую гибкость при защите данных. Однако реально таких систем не существует вследствие больших временных, ресурсных, аппаратных затрат.

Различают следующие основные методы обеспечения защиты данных:

- идентификация пользователей;
- управление доступом.

Идентификация пользователя. Администратор БД, предоставляя пользователю права доступа, должен указать системе, какие права предоставлены пользователю, сформировать паспорт пользователя и определить средства идентификации пользователя. В свою очередь, перед началом сеанса работы с системой пользователь должен:

- идентифицировать себя;
- подтвердить подлинность своей идентификации.

Процесс идентификации пользователя выполняется с помощью либо с системным идентификационным номером, либо машиночитаемого идентификатора, либо номера терминала, запросившего сеанс работы.

Процесс подтверждения подлинности – обмен информацией между пользователем и системой.

Во многих системах вторая проверка пользователя не реализована.

Управление доступом. В паспорт пользователя должно входить: идентификационный номер, процедура определения подлинности, перечень разрешенных операций. Однако, в ряде случаев, информации о разрешенных операциях оказывается недостаточно, для решения вопроса о допустимости выполнения этих операций с конкретными данными, поэтому для каждой группы данных необходимо иметь связанный с ними набор ограничений доступа:

- условие, которому должны удовлетворять пользователь, чтобы иметь доступ к этой группе данных;

- пароль, предъявляемый при выборке некоторой комбинации данных;
- пароль, предъявляемый при модификации данных.

Для отслеживания этого набора ограничений доступа должна существовать специальная программа, и если условие или ограничение доступа были нарушены, то система должна:

- подавить выполнение операций, требуемых пользователем;
- выдать ему сигнал ошибки;
- зафиксировать в специальном журнале факт попытки нарушения безопасности.

Алгоритм функционирования подобной программы базируется на аксиоме безопасности, которая формулируется следующим образом: если комбинация атрибутов A_1, \dots, A_N доступна (запрещена) пользователю 'X', в зависимости от условия 'B', то и подкомбинация атрибутов также доступна (запрещена) 'X' по 'B'.

Состав проверок выполняемых специальной программой, обычно следующий:

1. проверяется, все ли отношения, упомянутые в запросе, доступны пользователю 'X';
2. проверяется, все ли поколения, упомянутые в запросе, доступны пользователю 'X';
3. проверяется, все ли комбинации, упомянутые в запросе, доступны пользователю 'X';
4. проверяется, заданы ли квалифицирующие выражения, которые ограничивают доступный пользователю 'X', диапазон атрибутов, и проверяется, лежат ли запрашиваемые атрибуты в этом диапазоне.