

**МИНИСТЕРСТВО ВЫСШЕГО СРЕДНЕГО И
СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ РЕСПУБЛИКА
УЗБЕКИСТАН
НАВОЙСКИЙ ГОСУДАРСТВЕННЫЙ ГОРНЫЙ
ИНСТИТУТ
КАФЕДРА «АВТОМАТИЗАЦИЯ И УПРАВЛЕНИЯ»**



УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС
по предмету

**ИНЖЕНЕРНОЕ
ПРОГРАММИРОВАНИЕ**

для бакалавров

Область знаний	300 000 – Производственно – техническая сфера
Область образования	310 000 – Инженерная дело
Направление образования	5 311 000 – Автоматизация и управление технологических процессов и производств (по отраслям)

НАВОЙ – 2017

**МИНИСТЕРСТВО ВЫСШЕГО СРЕДНЕГО И
СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ РЕСПУБЛИКА
УЗБЕКИСТАН
НАВОИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГОРНЫЙ
ИНСТИТУТ
КАФЕДРА «АВТОМАТИЗАЦИЯ И УПРАВЛЕНИЯ»**



«УТВЕРЖДАЮ»

Проректор по учебной работе:

Н.А. Абдуазизов Н.А. Абдуазизов

«30» 08 2017 г.

**УЧЕБНО-МЕТОДИЧЕСКИЙ
КОМПЛЕКС**

по предмету

**ИНЖЕНЕРНОЕ
ПРОГРАММИРОВАНИЕ**

Учебно-методический комплекс разработан на основе утвержденного программа «Инженерное программирование» в МВССО РУз (Приказ №3 от «08» 08 2016)

Составитель: к.т.н., доц. Уринов Ш.Р. – доц. каф. АУ НГГИ

Рецензенты: к.т.н., доц. Эшмуродов З.О. – доц. каф. АУ НГГИ
Пулатов В.Б. – начальник УАП НГМК

В учебно-методическом комплексе по дисциплины «Инженерное программирование» приведена материалы лекционные, практические и лабораторные занятия, материалы самостоятельных работ, глоссария, рабочая программа, образовательная технология и технологическая карта, роздаточные материалы и др.

Настоящий учебно-методический комплекс предназначен по направления бакалавриата 5311000 - «Автоматизация и управления технологическими процессами и производствами»

Учебно-методический комплекс могут быть полезен для докторантов и самостоятельных исследователей, а также для научных сотрудников изучающих этого курса.

Учебно-методический комплекс обсуждена на заседание кафедры “Автоматизация и управление” (протокол №1 от «25» августа 2017 года) и рекомендован на совет энерго-механического факультета.

Зав. кафедрой:  доц. Жумаев О.А.

Учебно-методический комплекс рассмотрен обсужден на заседание Энерго-механического факультета (протокол №1 от «26» августа 2017 года) и (сдан) рекомендован на утверждения учебно-методический Совет института

Председатель совета факультета:  проф. Базарова С.Ж.

Учебно-методический комплекс рекомендован на внедрения в учебный процесс учебно-методическим советом Навоийского государственного горного института (протокол №1 от «30» августа 2017 года)

Секретарь учебно-методического Совета  Норматова М.Ж.

Согласованно:

Начальник учебно-методического отдела:  Каримов И.А.

СОДЕРЖАНИЕ

УЧЕБНЫЕ МАТЕРИАЛЫ.....	7
ЛЕКЦИОННЫЕ МАТЕРИАЛЫ.....	7
Лекция 1. Обзор пакетов, применяемых в автоматизации. Введение. Автоматизация и потребность в автоматизации	8
Лекция 2. Логические элементы и их построение И, ИЛИ, НЕ и их комбинации	12
Лекция 3. Классификация пакетов, применяемых в автоматизации	19
Лекция 4. Программный пакет Matlab. Общие понятия и принципы.	24
Лекция 5. Элементы программирования в MATLAB.....	28
Лекция 6. Построение графиков функции в MATLAB.....	31
Лекции 7-8. Подпрограммы программного пакета Matlab. Элементы подпрограммы Simulink.	41
Лекция 9. Моделирование системы управления в подпрограмме Simulink.	49
Лекция 10-11. Введение в программу AutoCAD	56
Лекция 12-13. SCADA системы. Общие понятия о SCADA системе.	70
Лекция 14. Архитектура SCADA системы	77
Лекция 15-16. Особенности проектирования распределенной иерархической АСУТП на базе SCADA-системы Trace Mode.....	83
Лекция 17-18. Обмен данными в SCADA системе TRACE MODE	90
Лекция 19-20. Тренажерные системы. Общие понятия о тренажерных системах.	94
Лекция 21-22. Программный пакет Experion PKS.....	105
Лекция 23-24. Программный пакет UniSIM	115
Лекция 25, 26. Программируемые контроллеры Simatic Step 7. Общие сведения.	121
Лекция 27. Simatic WinCC. Общие сведения.....	135
ЛАБОРАТОРНЫЕ РАБОТЫ	147
Лабораторная работа № 1. Создание таблицы истинности и логической схемы на основе логических элементов	147
Лабораторная работа №2. Работа с массивами	150
Лабораторная работа 3-4. Двухмерные массивы и матрицы	152
Лабораторная работа №5-6. Построение графиков функции в системе Matlab	154
Лабораторная работа 7. Ознакомление с операционной средой Simulink и простейшие модели	159
Лабораторная работа . Изучение основ работы на AutoCAD	166
Лабораторная работа. Создание проекта в TRACE MODE 6	169
Лабораторная работа.....	181
Лабораторная работа . Построение статических объектов дисплея.....	185
Лабораторная работа. Разработка программы управления	189
Лабораторная работа №18. Разработка мнемосхемы технологического процесса.....	191
ПРАКТИЧЕСКИЕ РАБОТЫ.....	194
Практическое занятие № 1.	194
Практическое занятие №2	195
Практическое занятие	199
Практическое занятие.....	200
Практическая работа	201
Практическая работа . Создание статического и динамического изображения в Trace Mode	203
Практическое занятие.....	209
Практическое занятие	211
Практическое занятие.....	214
Практическая работа. Логические компоненты STEP7.....	219
Практическая работа № Знакомство с пакетом программ STEP 7 Win CC. Создание проекта.....	225
САМОСТОЯТЕЛЬНАЯ ОБРАЗОВАНИЯ.....	229
ТЕМЫ РЕФЕРАТОВ.....	229
СБОРНИК ЗАДАЧИ	230
ГЛОССАРИЙ.....	233
ПРИЛОЖЕНИЯ	237
ТИПОВАЯ ПРОГРАММА	237
РАБОЧАЯ УЧЕБНАЯ ПРОГРАММА	249
ПРОМЕЖУТОЧНЫЕ И ИТоговые КОНТРОЛЬНЫЕ ВОПРОСЫ	255
ИТоговая КОНТРОЛЬНАЯ РАБОТА.....	258
РАЗДАТОЧНЫЕ МАТЕРИАЛЫ	259

УЧЕБНЫЕ МАТЕРИАЛЫ

ЛЕКЦИОННЫЕ МАТЕРИАЛЫ

Тема 1	ОБЗОР ПАКЕТОВ, ПРИМЕНЯЕМЫХ В АВТОМАТИЗАЦИИ
Технология проведения лекционного занятия	
Время 2 часа	Количество студентов 40-60 человек
Вид занятия	Введение. Визуальная лекция
План лекционного занятия	1. Введение. Автоматизация и потребность в автоматизации 2. Пакеты, применяемые в автоматизации.
Цель учебного занятия: Формирование у студентов глубокого представления о целях и задачах пакетов, применяемых в автоматизации	
Педагогические задачи:	Результаты обучения:
- Ознакомление с основными задачами автоматизации - Ознакомление с тенденцией развития пакетов, применяемых в автоматизации; - Внедрение в сознание студентов основных функций основных пакетов автоматизации	Студент: 1. Имеет представление о пакетах, применяемых в автоматизации 2. Излагает тенденции развития автоматизации в стратегии развития современного производства; 3. Характеризует основные функции SCADA систем 4. Характеризует основные распространенные пакеты, применяемые в автоматизации: САПР, SCADA системы, тренажерные системы, MATLAB, Step 7 и др.
Методика и техника обучения	-Информативная лекция, визуальной опрос, презентации
Средства обучения	- Текст лекции, проектор, плакаты, доска, мел
Форма обучения	-поток, группа, работа в парах
Условия обучения	-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 ин)	1.1. Приводит список используемой литературы по пакетам, применяемым в автоматизации. Знакомит студентов с рейтинговой оценкой знаний студентов. Объявляет тему и план занятия, знакомит с достижениями тенденции развития современного общества и потребность в автоматизации современного производства	1.1. Слушает и записывает
2-этап Основной	2.1. Обращает внимание студентов на вопросы плана занятия и обращает внимание на основные понятия. Проводит опрос. 2.2. Продолжает излагать лекционный материал. На примерах приводит и объясняет основные тенденции применения пакетов в автоматизации. 2.3. Обращает внимание студентов на основные понятия автоматизации и пакетов.	2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают
3-этап Итоговый	3.1. Делает заключительные выводы лекционной темы и заостряет внимание студентов на основных вопросах 3.2. Отмечает активных студентов 3.3. Дает задание на самостоятельную подготовку: Дать определение каждого пакета, применяемого в автоматизации, привести примеры	3.1. Слушают и усваивают 3.2. Записывают задание

Лекция 1.

Обзор пакетов, применяемых в автоматизации.

Введение. Автоматизация и потребность в автоматизации

План:

1. Обзор пакетов, применяемых в автоматизации
2. Пакеты, применяемые в автоматизации.
3. SCADA системы
4. Тренажерные системы
5. Matlab

1. Обзор пакетов, применяемых в автоматизации

Стратегия развития современного производства предполагает существенное повышение уровня автоматизации, прежде всего, технологических процессов. На первых порах технологическими процессами полностью управлялся человек. Однако по мере усложнения технологических процессов возможности человека становились ограничивающим фактором.

Автоматизация – это применение в производстве технических средств, методов и систем управления, освобождающих человека от непосредственного участия в производстве. Целью автоматизации заключается в повышении производительности и эффективности труда, улучшение качества продукции и условий трудовой деятельности человека.

Автоматизация технологических процессов неизменно связана с созданием различных систем управления и контроля, которые и выполняют функции управления и контроля, заменяя человека.

Человек, обладая разумом, знаниями, навыками и опытом, способен создавать машины, которые будут выполнять за него эту работу, причем не только физическую, но и умственную. Особенно важно, в первую очередь, освобождение человека от непосредственного участия в процессах именно материального производства, т.е. в технологических процессах.

Повышение эффективности любого человеческого труда и обеспечивает автоматизация. Очевидно, что автоматизация технологических процессов представляет собой создание технологий с минимальным участием человека, точнее включают комплекс мероприятий, направленных на сокращении числа работающих, повышение эффективности процессов, повышение качества продукции и улучшение условий труда.

Прогресс в области информационных технологий также дал огромный толчок в сфере автоматизации, а именно, появилось большое количество программных пакетов, повышающих эффективность производства и обеспечивающих автоматизацию.

2. Пакеты, применяемые в автоматизации.

В общем смысле, в промышленности можно выделить три ветви пакетов, применяемых в автоматизации – это системы, применяемые для проектирования систем автоматического и автоматизированного управления, системы и программные пакеты, применяемые при моделировании технологических процессов и программировании его частей, и так называемые SCADA системы, но это разделение условно, так как из года в год происходит интенсивная интеграция программных пакетов.

САПР

Проектирование продукта является важнейшей функцией в системе производства. Качество продукта и его дизайн являются, пожалуй, самым важным фактором, определяющим коммерческий успех и общественную ценность продукта. Если дизайн продукции оставляет желать лучшего, независимо от того, насколько хорошо он изготовлен, продукт вероятнее всего обречен не будет способствовать богатству и благополучию фирмы, которая его породила. Если дизайн продукта хороший и имеет достаточно низкую себестоимость, то он принесет компании прибыль и успех. Один из факторов из жизни дизайн продукта является то, что очень значительная часть стоимости продукта определяется его конструкцией. Таким образом, Дизайн и производство не могут быть отделены друг от друга, они связаны между собой функционально, технологически и экономически и дизайн является неотъемлемой частью автоматизации. (с. 755 [Mikell P. Groover. Automation, Production Systems])

Также необходимо отметить, что в настоящее время в деятельность изыскательских и проектных организаций быстро проникает компьютеризация, поднимающая проектную работу на качественно новый уровень, при котором резко повышаются темпы и качество проектирования, более обоснованно решаются многие сложные инженерные задачи, которые раньше рассматривались лишь упрощенно. Во многом это происходит благодаря использованию эффективных специализированных программ, которые могут быть как самостоятельными, так и в виде приложений к общетехническим программам. Деятельность по созданию программных продуктов и технических средств для автоматизации проектных работ имеет общее название - САПР.

САПР (англ. CAD, Computer-Aided Design) - программный пакет, предназначенный для проектирования (разработки) объектов производства (или строительства), а также оформления конструкторской и/или технологической документации.

Компоненты многофункциональных систем САПР традиционно группируются в три основных блока CAD, CAM, CAE. Модули блока CAD (Computer Aided Design) предназначены в основном для выполнения графических работ, модули CAM (Computer Aided Manufacturing) - для решения задач технологической

подготовки производства, модули САЕ (Computer Aided Engineering) - для инженерных расчетов, анализа и проверки проектных решений.

Существует большое количество пакетов САПР разного уровня. Значительное распространение получили системы, в которых основное внимание сосредоточено на создании "открытых" (т.е. допускающих расширение) базовых графических модулей САД, а модули для выполнения расчетных или технологических задач (соответствующие блокам САМ и САЕ) остаются для разработки пользователям или организациям, специализированным на соответствующем программировании. Такие дополнительные модули могут использоваться и самостоятельно, без САД-систем, что очень часто практикуется в строительном проектировании. Они сами могут представлять крупные программные комплексы, для которых разрабатываются свои приложения, позволяющие решать более узкие задачи.

Крупнейшим в мире поставщиком программного обеспечения для промышленного и гражданского строительства, машиностроения, рынка средств информации является компания Autodesk, Inc. Начиная с 1982 года компанией Autodesk был разработан широкий спектр решений для архитекторов, инженеров, конструкторов, позволяющих им создавать цифровые модели. Технологии Autodesk используются для визуализации, моделирования и анализа поведения разрабатываемых конструкций на ранних стадиях проектирования и позволяют не просто увидеть модель на экране, но и испытать её.

Самыми распространенными САПР в Узбекистане в сфере автоматизации технологических процессов являются AutoCAD (один из самых первых проектировочных пакетов), КОМПАС, OrCAD, Proteus, WorkBench и другие.

3. SCADA системы

Следующей линейкой программных продуктов являются SCADA системы. В настоящее время SCADA (Supervisory Control And Data Acquisition— диспетчерское управление и сбор данных) является наиболее перспективной технологией автоматизированного управления во многих отраслях промышленности. SCADA системы сейчас, являются неотъемлемой частью современных автоматизированных систем управления процессами или как ее еще называют средой визуализации. Получив широкое распространение в начале 20-го века, сегодня уже трудно себе представить объект автоматизации, на который бы не была установлена SCADA система. Уже практически ушли в историю огромные шкафы с панелями вторичных приборов, на смену им пришли автоматизированные рабочие места (АРМ). Внедрение SCADA систем приводит к существенному удешевлению эксплуатации вторичного оборудования на крупных объектах, путем переноса индикации и накопления технологической информации на пульт диспетчера АРМ. Не смотря на большое разнообразие SCADA систем на рынке, большинство из них имеет примерно одинаковый набор функциональных возможностей, позволяющих выполнять основные требования, предъявляемые к верхнему уровню АСУ ТП. Набор стандартных функций в SCADA системах обусловлен общим кругом задач при разработке систем автоматизации. Определим состав основных функций, позволяющих выполнить полноценный проект по автоматизации:

1. То без чего не обходится ни одна SCADA система – это графический интерфейс, который позволяет упростить задачу построения и отображения технологического процесса (ТП). К графической части можно отнести возможность упрощенного или детализированного отображения объектов ТП, средств измерения физических параметров технологических объектов (ТО). Кроме того, позволяет отображать кнопки, индикаторы, панели стрелочных или цифровых индикаторов, регуляторов и других вторичных приборов, которые раньше располагались на панели шкафа автоматизации. Поддержка библиотек изображений и видео, позволяющая выводить графическую информацию сторонних разработчиков на графическую панель SCADA системы, такие как элементы мнемосхем, динамические объекты.

2. SCADA системы позволяют вести архив измерений, событий и аварийных ситуаций, происходящих на ТО, с отображением изменений информации в окне временного тренда.

3. Упрощенный язык составления алгоритмов управления ТП, математических вычислений.

4. Драйвера устройств и оборудования согласованной работы со SCADA системой, находящихся на нижнем и среднем уровнях АСУ ТП, такие как датчики, вторичное оборудование контроллеры.

5. Поддержка других языков программирования высокого уровня (Visual C++, VBA, VB).

6. И одна из важнейших функций SCADA систем – средства защиты от несанкционированного доступа к файлам и компонент

Самыми распространенными SCADA системами на сегодняшний день являются: TRACE MODE (первая интегрированная информационная система для управления промышленным производством, объединяющая в едином целом продукты класса SOFTLOGIC-SCADA / HMI-MES-EAM-HRM), SIMATIC WinCC (мощная среда разработки верхнего уровня управления АСУ ТП с централизованным контролем и сбором данных, система SIMATIC WinCC (Windows Control Center) - это компьютерная система человеко-машинного интерфейса, работающая под управлением операционных систем Windows и предоставляющая широкие функциональные возможности для построения систем управления различного назначения и уровней автоматизации), Citect SCADA (программный продукт, представляющий собой полнофункциональную систему визуализации и мониторинга, управления и сбора данных), SCADA система InTouch (это достаточно мощная среда разработки визуализации и управления для промышленной автоматизации технологических процессов и диспетчерского контроля, применяется для создания DCS (распределенных систем управления) и других АСУ ТП) и другие.

4. Тренажерные системы

В последние несколько десятилетий во всем мире резко возрос интерес к проблемам построения высокоэффективных и высоконадежных систем диспетчерского управления и сбора данных. С одной стороны, это связано со значительным прогрессом в области вычислительной техники, программного обеспечения и телекоммуникаций, что увеличивает возможности и расширяет сферу применения автоматизированных систем. С другой стороны, развитие информационных технологий, повышение степени автоматизации и перераспределение функций между человеком и аппаратурой обострило проблему взаимодействия человека-оператора с системой управления. Расследование и анализ большинства аварий и происшествий в промышленности и на транспорте, часть из которых привела к катастрофическим последствиям, показали, что, если в 1960-х гг. ошибка человека являлась первоначальной причиной лишь 20% инцидентов, то в 1990-х гг. доля «человеческого фактора» возросла до 80%, причем в связи с постоянным совершенствованием технологий и повышением надежности электронного оборудования и машин доля эта может еще возрасти.

Учитывая вышеизложенное, большое распространение на сегодняшний день получают компьютерные тренажеры, работающие с такими пакетами как Unisim Design, Experion и другими, являющимися усовершенствованной версией SCADA систем.

Например, программное обеспечение UniSimDesign позволяет повысить эффективность работы предприятия и улучшить результаты коммерческой деятельности. При разработке программного обеспечения системы UniSim использованы лучшие из имеющихся технологий моделирования. В UniSim предусмотрен набор унифицированных блоков моделирования, способствующих повышению эффективности работы в течение всего жизненного цикла предприятия; с ними можно работать как в автономном режиме — для создания статических моделей технологических процессов, отладки системы управления и обучения операторов, так и в интерактивном режиме — для управления, оптимизации, отслеживания рабочих характеристик и планирования коммерческой деятельности.

5. MATLAB

К программным пакетам, применяемым при моделировании технологических процессов и программировании его частей относится один из самых распространенных пакетов в сфере автоматизации промышленности – MATLAB, которая после 2000-х годов в научных и инженерно-технических кругах получила широкое распространение, более того, в настоящее время она принята в качестве официального средства оформления инженерной документации и научных публикаций.

MATLAB – это высокоуровневый технический вычислительный язык, а также интерактивная среда для разработки алгоритмов, визуализации данных и численных расчетов, а также анализа данных. С помощью продукта MATLAB, можно решить технические вычислительные задачи быстрее, чем при традиционном программировании на языках, такие как C, C++ и fortran. Можно использовать MATLAB в широком диапазоне, включая сигналы обработки изображений, связи, контроля проектирования, испытания и измерения, финансового моделирование и анализ, и вычислительной биологии.

Система MATLAB специально создана для проведения именно инженерных расчетов: математический аппарат, который используется в ней, предельно приближен к современному математическому аппарату инженера и ученого и опирается на вычисления, проводимые с матрицами, векторами и комплексными числами; графическое представление функциональных зависимостей просто в изложении и понимании.

Язык программирования MATLAB весьма прост, он содержит лишь несколько десятков операторов; незначительное количество операторов здесь компенсируется большим числом процедур и функций, содержание которых понятно пользователю.

В отличие от большинства математических систем, MATLAB является открытой системой: практически все ее процедуры и функции доступны не только для использования, но и для модификации.

MATLAB позволяет моделировать, анализировать и синтезировать системы любого уровня сложности с помощью различных подпрограмм и пакетов как Control System Toolbox, LTIViewer, Simulink, а также создавать программы и S-модели.

При программировании неотъемлемой части систем автоматизированного управления – логических контроллеров – применяют различные языки программирования. В республике большое распространение получил язык программирования логических контроллеров Step 7.

Таким образом, в настоящее время в автоматизации применяют множество программных пакетов различного назначения и профиля, которые существенно упрощают задачи, решаемые автоматизацией.

Контрольные вопросы:

1. Что такое автоматизация?
2. Что Вы понимаете под автоматизацией технологических процессов?
3. Какие пакеты, применяемые в автоматизации Вы знаете?
4. Что такое SCADA система?
5. Для чего служат CAD системы?
6. Какой программный пакет позволяет моделировать, анализировать и синтезировать системы любого уровня сложности в сфере промышленной автоматизации?

Тема 2	Логические элементы и их построение	
Технология проведения лекционного занятия		
Время 2 часа	Количество студентов 40-80 человек	
Вид занятия	Визуальная лекция	
План лекционного занятия	1. Элемент И. 2. Элемент ИЛИ. 3. Элемент НЕ. 4. Элемент И-НЕ.	
Цель учебного занятия: Формирование у студентов глубокого представления о логических элементах и их применение		
Педагогические задачи:		Результаты обучения:
- Ознакомление с основными логическими элементами - Ознакомление с структурой и таблицами истинности логических элементов - Внедрение в сознание студентов основных законов элементов логики		Студент: 1. Имеет представление о целях и задачах логических элементов 2. Излагает структурной обозначение логических элементов. 3. Характеризует связь логических элементов в цифровой электроники и её место в развитии науки 4. Характеризует основные законы логических элементов
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.1. Сообщает тему занятия и план, ожидаемые результаты	1.1. Слушает и записывает
2-этап Основной	2.1. Обращает внимание студентов на вопросы темы лекционного плана, основные понятия, правила, законы, опрос 2.2. Продолжает излагать лекционный материал. Комментирует понятия дизъюнкция, конъюнкция, инверсия. 2.3. Объясняет методы определения таблицы истинности и составления структурной схемы. 2.4. Подчеркивает студентам основные понятия и обращает внимание на основные моменты лекции для записи студентами и подчеркивает самостоятельное повторение	2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают
3-этап Итоговый	3.1. Делает заключительные выводы лекционной темы и заостряет внимание студентов на основные вопросы 3.2. Отмечает активных студентов 3.3. Даст задание на самостоятельную подготовку: Дать определение каждого закона, привести примеры	3.1. Слушают и усваивают 3.2. Записывают задание

Лекция 2. Логические элементы и их построение И, ИЛИ, НЕ и их комбинации

План:

1. Элемент И.
2. Элемент ИЛИ.
3. Элемент НЕ.
4. Элемент И-НЕ.

Современный этап промышленного развития характеризуется тем, что разработчики систем автоматики и вычислительной техники стремятся использовать функциональные модули, выполняющие определённые схемные задачи: логические преобразования, хранение информации и т.д. Конкретный вид электрической схемы, использованной для реализации заданной логической функции, как правило, не имеет существенного значения. Техническое устройство, реализующее логическую функцию, может рассматриваться просто как логический элемент, внутренняя структура которого не конкретизируется.

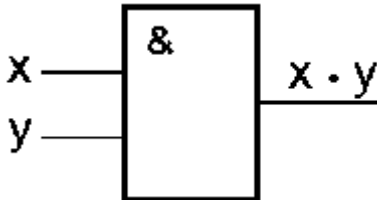
Основой цифровой электроники являются логические элементы. На их основе состоят различные триггеры, дешифраторы, счётчики и т.д. Вот, к примеру, говорят же в процессоре миллионы транзисторов, но как их так собрали, ничего не перепутав и всё упорядочив? Из транзисторов собраны логические элементы, из логических элементов собраны различные счётчики, дешифраторы, триггеры, а из триггеров ОЗУ память и т.д., а всё вместе в сборе получается, процессор.

Элемент И

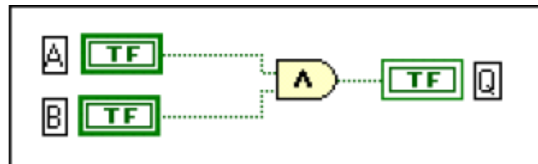
На рисунке представлена таблица истинности элемента "И" с двумя входами. Хорошо видно, что логическая единица появляется на выходе элемента только при наличии единицы на первом входе и на втором. В трёх остальных случаях на выходе будут нули.

Таблица истинности		
X1	X2	Y1
0	0	0
0	1	0
1	0	0
1	1	1

На принципиальных схемах логический элемент "И" обозначают так



На зарубежных схемах обозначение элемента "И" имеет другое начертание. Его кратко называют AND.



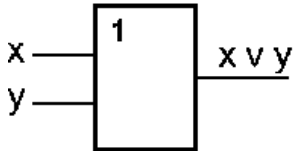
LabVIEW и функционирование проводной ввод / вывод клеммные коробки

Элемент ИЛИ

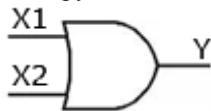
Элемент "ИЛИ" с двумя входами работает несколько по-другому. Достаточно логической единицы на первом входе или на втором как на выходе будет логическая единица. Две единицы так же дадут единицу на выходе.

Таблица истинности		
X1	X2	Y1
0	0	0
0	1	1
1	0	1
1	1	1

На схемах элемент "ИЛИ" изображают так



На зарубежных схемах его изображают чуть по-другому и называют элементом OR.

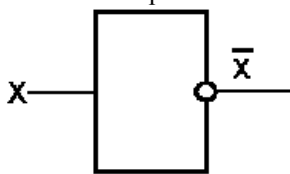


Элемент НЕ

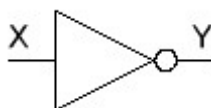
Элемент, выполняющий функцию инверсии «НЕ» имеет один вход и один выход. Он меняет уровень сигнала на противоположный. Низкий потенциал на входе даёт высокий потенциал на выходе и наоборот.

Таблица истинности	
X1	Y1
0	1
1	0

Вот таким образом его показывают на схемах.



В зарубежной документации элемент "НЕ" изображают следующим образом. Сокращённо называют его NOT.



Все эти элементы в интегральных микросхемах могут объединяться в различных сочетаниях. Это элементы: И-НЕ, ИЛИ-НЕ, и более сложные конфигурации.

Элемент И-НЕ

Рассмотрим несколько реальных логических элементов на примере серии транзисторно-транзисторной логики (ТТЛ) К155 с малой степенью интеграции. На рисунке когда-то очень популярная микросхема К155ЛА3, которая содержит четыре независимых элемента 2И – НЕ. Кстати, с помощью её можно собрать простейший маячок на микросхеме.

Цифра всегда обозначает число входов логического элемента. В данном случае это двухвходовой элемент «И» выходной сигнал которого инвертируется. Инвертируется, это значит "0" превращается в "1", а "1" превращается в "0". Обратим внимание на кружочек на выходах – это символ инверсии. В той же серии существуют элементы 3И-НЕ, 4И-НЕ, что означает элементы «И» с различным числом входов (3, 4 и т.д.).

Зарубежное обозначение элемента И-НЕ (в данном случае 2И-НЕ). Называется NAND.

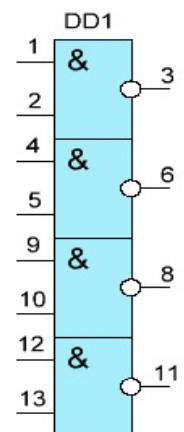
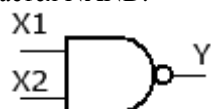


Таблица истинности		
X1	X2	Y1
0	0	1
0	1	1
1	0	1
1	1	0

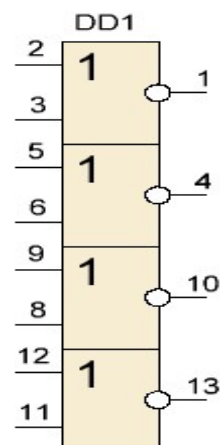
В таблице истинности элемента 2И – НЕ мы видим, что благодаря инвертору получается картина противоположная элементу «И». В отличие от трёх нулей и одной единицы мы имеем три единицы и ноль. Элемент «И – НЕ» часто называют элементом Шеффера.

Элемент 2ИЛИ-НЕ

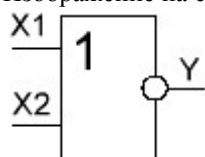
Логический элемент 2ИЛИ – НЕ представлен в серии К155 микросхемой 155ЛЕ1. Она содержит в одном корпусе четыре независимых элемента. Таблица истинности так же отличается от схемы "ИЛИ" применением инвертирования выходного сигнала.

Таблица истинности для логического элемента 2ИЛИ-НЕ.

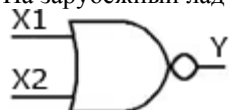
Таблица истинности		
X1	X2	Y1
0	0	1
0	1	0
1	0	0
1	1	0



Изображение на схеме.



На зарубежный лад изображается так. Называют как NOR.



Мы имеем только один высокий потенциал на выходе, обусловленный подачей на оба входа одновременно низкого потенциала. Здесь, как и на любых других принципиальных схемах, кружочек на выходе подразумевает инвертирование сигнала. Так как схемы И–НЕ и ИЛИ–НЕ встречаются очень часто, то для каждой функции имеется своё условное обозначение. Функция И – НЕ обозначается значком "&", а функция ИЛИ – НЕ значком "1".

Для отдельного инвертора таблица истинности уже приведена выше. Можно добавить, что количество инверторов в одном корпусе может достигать шести.

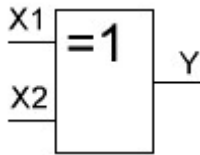
Элемент "исключающее ИЛИ"

К числу базовых логических элементов принято относить элемент реализующий функцию «исключающее ИЛИ». Иначе эта функция называется «неравнозначность».

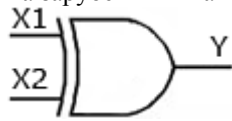
Высокий потенциал на выходе возникает только в том случае, если входные сигналы не равны. То есть на одном из входов должна быть единица, а на другом ноль. Если на выходе логического элемента имеется инвертор, то функция выполняется противоположная – «равнозначность». Высокий потенциал на выходе будет появляться при одинаковых сигналах на обоих входах.

Таблица истинности		
X1	X2	Y
0	0	0
1	0	1
0	1	1
1	1	0

Эти логические элементы находят своё применение в сумматорах. «Исключающее ИЛИ» изображается на схемах знаком равенства перед единицей "=1".



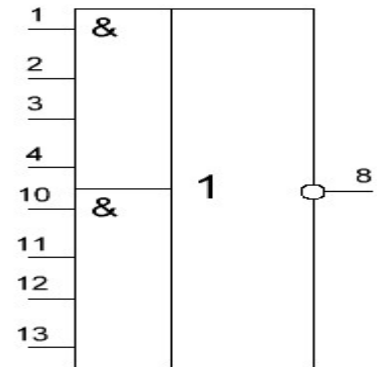
На зарубежный манер "исключающее ИЛИ" называют XOR и на схемах рисуют вот так.



Кроме вышеперечисленных логических элементов, которые выполняют базовые логические функции очень часто, используются элементы, объединённые в различных сочетаниях. Вот, например, К555ЛР4. Она называется очень серьёзно 2-4И-2ИЛИ-НЕ.

Её таблица истинности не приводится, так как микросхема не является базовым логическим элементом. Такие микросхемы выполняют специальные функции и бывают намного сложнее, чем приведённый пример. Так же в логический базис входят и простые элементы "И" и "ИЛИ". Но они используются гораздо реже. Может возникнуть вопрос, почему эта логика называется транзисторно-транзисторной.

Если посмотреть в справочной литературе схему, допустим, элемента 2И – НЕ из микросхемы К155ЛА3, то там можно увидеть несколько транзисторов и резисторов. На самом деле ни резисторов, ни диодов в этих микросхемах нет. На кристалл кремния через трафарет напыляются только транзисторы, а функции резисторов и диодов выполняют эмиттерные переходы транзисторов. Кроме того в ТТЛ логике широко используются много эмиттерные транзисторы. Например, на входе элемента 4И стоит четырёх эмиттерный транзистор.



Основные понятия булевой алгебры

Исходным понятием логики высказываний является простое высказывание. Это понятие не определяется через другие понятия, так как является базовым. Под высказыванием обычно понимают всякое повествовательное предположение, утверждающее что-либо о чем-либо. Если смысл, содержащийся в высказывании, соответствует действительности, то высказывание называют истинным. В противном случае – ложным.

Обычно элементарные высказывания обозначают строчными буквами латинского алфавита a, b, c, x, y, \dots , которые также являются логическими переменными. Истинные значения обозначаются буквой И или 1, а ложные – Л или 0.

Из элементарных высказываний можно составить более сложные с помощью логических связок $\neg, \wedge, \vee, \rightarrow, \equiv$, называемых соответственно отрицание, логическое и (конъюнкция), логическое или (дизъюнкция), логическое следствие (импликация), эквивалентность и круглых скобок $(,)$. Семантику логических связок можно представить с помощью таблицы истинности. В левой части этой таблицы перечисляются все возможные комбинации значений логических переменных. В правой части – соответствующие им значения новых выражений, полученных из переменных и связок.

X	y	$\neg x$	$x \wedge y$	$x \vee y$	$x \rightarrow y$	$x \equiv y$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Связки имеют следующий приоритет: $\neg \wedge \vee \rightarrow \equiv$. Приоритет операций, представленных логическими связками можно изменить с помощью скобок. Высказывания, построенные с помощью простых высказываний, связок и скобок, называют правильно построенными формулами или сокращённо формулами.

Замечательным свойством логики высказываний является то, что ее семантика близка к соответствующим высказываниям на естественном языке. Так, например семантика формул содержащих связки \neg и \wedge практически совпадает со смыслом фраз содержащих слова «не» и «и». Однако имеются и некоторые различия. Так формула $x \vee y$ несколько шире, чем русское « x или y ». Выражение « x или y » по смыслу ближе к формуле $x \wedge \neg y \vee \neg x \wedge y$. Еще больше различий между семантикой формулы $x \rightarrow y$ в логике высказываний и выражению «из x следует y ». В русском языке это выражение истинно, если истинны x и y , т.е. предложение русского языка по смыслу совпадает с формулой $x \wedge y$. Логическое следствие истинно также, если x и y ложны

или x ложна, а y истинна. Логическую формулу $x \rightarrow y$ следует интерпретировать на естественном языке так: «Если x истинна, то y тоже истинна, а остальное неизвестно».

Для любой формулы также можно построить таблицу истинности. Например, для формулы таблица истинности будет выглядеть следующим образом:

x	y	$\neg x$	$\neg x \vee y$	$\neg x \wedge (\neg x \vee y)$	$\neg x \wedge (\neg x \vee y) \rightarrow \neg x$
0	0	1	1	1	1
0	1	1	1	1	1
1	0	0	0	0	1
1	1	0	0	0	1

Очевидно, что если формула содержит n переменных, то в таблице истинности будет содержаться 2^n строк. В приведенном примере формула содержит 2 переменные и $2^2 = 4$ строки. Кроме того, данная формула истинна на любом наборе значений своих переменных. Такие формулы называются тождественно истинными или тавтологиями. В противоположной ситуации, формула является тождественно ложной или невыполнимой. Если две разные формулы принимают одинаковые значения на любом наборе значений переменных, то такие формулы называют равносильными. Равносильные формулы будем обозначать знаком равенства $=$.

Основные законы алгебры логики

В алгебре логики имеется ряд законов, позволяющих производить равносильные преобразования логических выражений.

В логике высказываний известно много общезначимых формул, которые также называются законами логики высказываний. Основными законами являются следующие:

- законы идемпотентности:
 - $x \wedge x = x$
 - $x \vee x = x$
- $x \wedge 1 = x$
- $x \vee 1 = 1$
- $x \wedge 0 = 0$
- $x \vee 0 = x$
- $x \wedge \neg x = 0$ – закон противоречия
- $x \vee \neg x = 1$ – закон исключения третьего
- $\neg \neg x = x$ – закон снятия двойного отрицания
- законы поглощения
 - $x \wedge (y \vee x) = x$
 - $x \vee (y \wedge x) = x$

Доказательство этих и последующих законов элементарно осуществляется с помощью построения таблиц истинности или простейших логических рассуждений.

Следующая группа законов представляет взаимосвязь между логическими операциями:

- $(x \equiv y) = (x \rightarrow y) \wedge (y \rightarrow x)$
- $x \rightarrow y = \neg x \vee y$
- законы Де Моргана
 - $\neg (y \vee x) = \neg y \wedge \neg x$
 - $\neg (y \wedge x) = \neg y \vee \neg x$

Замечательным следствием приведенных выше законов является следующий факт. Любую логическую формулу можно заменить равносильной ей, но содержащую только две логические операции: конъюнкцию или отрицание, или дизъюнкцию или отрицание. Дальнейшее исключение логических операций, очевидно, невозможно, то есть приведенные пары представляют минимальный базис для построения правильно построенных формул. Однако существует операция, с помощью которой можно представить любую логическую связку. Эта операция получила название «штрих Шеффера» и определяется следующим образом:

x	y	$x y$
0	0	1
0	1	1
1	0	1
1	1	0

На основании этого определения можно ввести следующие законы, выражающие взаимосвязь операции «штрих Шеффера» и других логических связок:

- $\neg x = x | x$
- $x \wedge y = (x | y) | (x | y)$

Также следует отметить, что $x | y = \neg (x \wedge y)$.

К основным законам алгебры логики также относятся следующие:

- коммутативные законы

- $x \wedge y = y \wedge x$
- $x \vee y = y \vee x$
- дистрибутивные законы
- $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
- $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$
- ассоциативные законы
- $x \wedge (y \wedge z) = (x \wedge y) \wedge z$
- $x \vee (y \vee z) = (x \vee y) \vee z$

Еще одним важным законом алгебры логики является закон двойственности. Пусть формула A содержит только операции конъюнкции, дизъюнкции и отрицания. Для операции конъюнкции двойственной считается дизъюнкция, а для дизъюнкции – конъюнкция. Тогда по определению формулы A и A^* называются двойственными, если формула A^* получается из A путем замены в ней каждой операции на двойственную. Например, для формулы $(x \vee y) \wedge z$ двойственной формулой будет $(x \wedge y) \vee z$. Для двойственных формул справедлива следующая теорема: если формулы A и B равносильны, то равносильны и двойственные им формулы, то есть $A^* = B^*$. Данную теорему оставим без доказательства.

С помощью законов логики можно осуществлять равносильные преобразования. Такие преобразования используются для доказательств, приведения формул к заданному виду, упрощения формул.

Под сложностью формул обычно понимается количество символов, используемых для ее записи. То есть формула α проще формулы β , если α содержит меньше букв и логических операций. Например, для формулы $(\neg(x \vee y) \rightarrow x \vee y) \wedge y$ можно записать следующую цепочку преобразований, приводящих ее к более простому виду:

$$(\neg\neg(x \vee y) \vee x \vee y) \wedge y = (x \vee y \vee x \vee y) \wedge y = (x \vee y) \wedge y = y.$$

Закон	Для ИЛИ	Для И
Переместительный	$A \vee B = B \vee A$	$A \cdot B = B \cdot A$
Сочетательный	$(A \vee B) \vee C = A \vee (B \vee C)$	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$
Распределительный	$(A \vee B) \cdot C = A \cdot C \vee B \cdot C$	$(A \cdot B) \vee C = (A \vee C) \cdot (B \vee C)$
Правила де Моргана	$A \vee B = \overline{\overline{A} \cdot \overline{B}}$	$A \cdot B = \overline{\overline{A} \vee \overline{B}}$
Идемпотентности	$A \vee A = A$	$A \cdot A = A$
Исключения третьего и противоречия	$A \vee \overline{A} = 1$	$A \cdot \overline{A} = 0$
Операции с константами	$A \vee 1 = 1$ $A \vee 0 = A$	$A \cdot 1 = A$ $A \cdot 0 = 0$
Поглощения	$A \vee (A \cdot B) = A$	$A \cdot (A \vee B) = A$
Склеивания	$(A \cdot B) \vee (\overline{A} \cdot B) = B$	$(A \vee B) \cdot (\overline{A} \vee B) = B$
Контрапозиции	$A \rightarrow B = \overline{B} \rightarrow \overline{A}$	
Двойного отрицания	$\overline{\overline{A}} = A$	
Формулы упрощения	$A \rightarrow B = \overline{A} \vee B$ $A \leftrightarrow B = (A \rightarrow B) \cdot (B \rightarrow A) = (\overline{A} \vee B) \cdot (\overline{B} \vee A)$	

Контрольные вопросы.

1. Опишите логику И.
2. Опишите логику ИЛИ.
3. Опишите логику НЕ.
4. Опишите логику И-НЕ.
5. Что такое равносильные выражение в логике.

Тема 3	КЛАССИФИКАЦИЯ ПАКЕТОВ, ПРИМЕНЯЕМЫХ В АВТОМАТИЗАЦИИ	
Технология проведения лекционного занятия		
Время 2 часа	Количество студентов 40-60 человек	
Вид занятия	Введение. Визуальная лекция	
План лекционного занятия	1. Программное обеспечение современных ЭВМ 2. Прикладные программные пакеты 3. Классификация пакетов, применяемых в автоматизации	
Цель учебного занятия: Формирование у студентов глубокого представления о классификации пакетов, применяемых в автоматизации		
Педагогические задачи:		Результаты обучения:
- Ознакомление с основными понятиями программного обеспечения современных ЭВМ - Ознакомление с тенденцией развития прикладных программных пакетов; - Внедрение в сознание студентов классификации пакетов, применяемых в автоматизации, по выполняемым задачам		Студент: 5. Имеет представление о программном обеспечении современных ЭВМ 6. Излагает тенденции развития прикладных программных пакетов; 7. Характеризует основные функции программного обеспечения современных ЭВМ; 8. Характеризует основные классы пакетов, служащие для автоматизации систем управления технологическими процессами.
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.1. Сообщает тему занятия и план, ожидаемые результаты	1.1. Слушает и записывает
2-этап Основной	2.1. Обращает внимание студентов на вопросы темы лекционного плана, основные понятия, правила, законы, опрос 2.2. Продолжает излагать лекционный материал. Комментирует понятия системного, инструментального и прикладного программного обеспечения. 2.3. Объясняет классификацию прикладного программного обеспечения. 2.4. Объясняет классификацию средств, создающих обеспечение прикладных программ для специалистов. 2.5. Подчеркивает студентам основные понятия и обращает внимание на основные моменты лекции для записи студентами и подчеркивает самостоятельное повторения	2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают 2.3. Записывают и запоминают 2.4. Записывают и запоминают 2.5. Записывают основные понятия и приводят примеры
3-этап Итоговый	3.1. Делает выводы по лекции и обращает внимание студентов на основные вопросы 3.2. Отмечает активных студентов и поощряет их	3.1. Слушают и усваивают 3.2. Записывают задание на самоподготовку

Лекция 3.

Классификация пакетов, применяемых в автоматизации

План:

1. Программное обеспечение современных ЭВМ
2. Прикладные программные пакеты
3. Классификация пакетов, применяемых в автоматизации

1. Программное обеспечение современных ЭВМ

Как известно, в отличие от других вычислительных устройств, осуществляющих только заранее заложенные в них функции, ЭВМ могут выполнять любые действия по обработке информации. Для этого необходимо составить для компьютера на понятном ему языке точную и подробную последовательность инструкций, то есть программу, как надо обрабатывать информацию. Сборник таких программ составляет **программное обеспечение ЭВМ** и является одним из средств, демонстрирующих их возможности.

По назначению программное обеспечение (ПО) можно разделить на системное (System), инструментальное (Tools) и прикладное (ApplicationSoftware) программное обеспечение (рис. 1). По способам распространения и применения ПО делят на коммерческое (Commercial Software, Shareware), свободно распространяемое (Freeware, FreeSoftware) и открытое (Open Source Software) программное обеспечение.

В соответствии с современным законодательством большинства стран, программный продукт и его код защищаются законом об авторском праве. Авторы вправе предоставлять пользователям право использовать, изменять и распространять программный продукт. Это осуществляется при выпуске программного кода с условием свободной лицензии.

Свободно распространяемое ПО используется без ограничений в офисах, дома, образовательных и государственных учреждениях и т.д. Программный код открытого ПО можно без ограничений открыть и изменить. Открытое лицензированное ПО не обязательно предоставляется в бесплатном виде.

Системное ПО – совокупность программ, обеспечивающих общее управление функционированием вычислительной системы и выполнение функций по ее обслуживанию. К системному ПО относятся операционные системы и оболочки, программы-драйверы, базовые настроечные программы (BIOS и т.д.), файл-менеджеры, а также сервисные программы (утилиты: средства архивирования и форматирования, диагностирования и оптимизации, управления памятью и антивирусы).

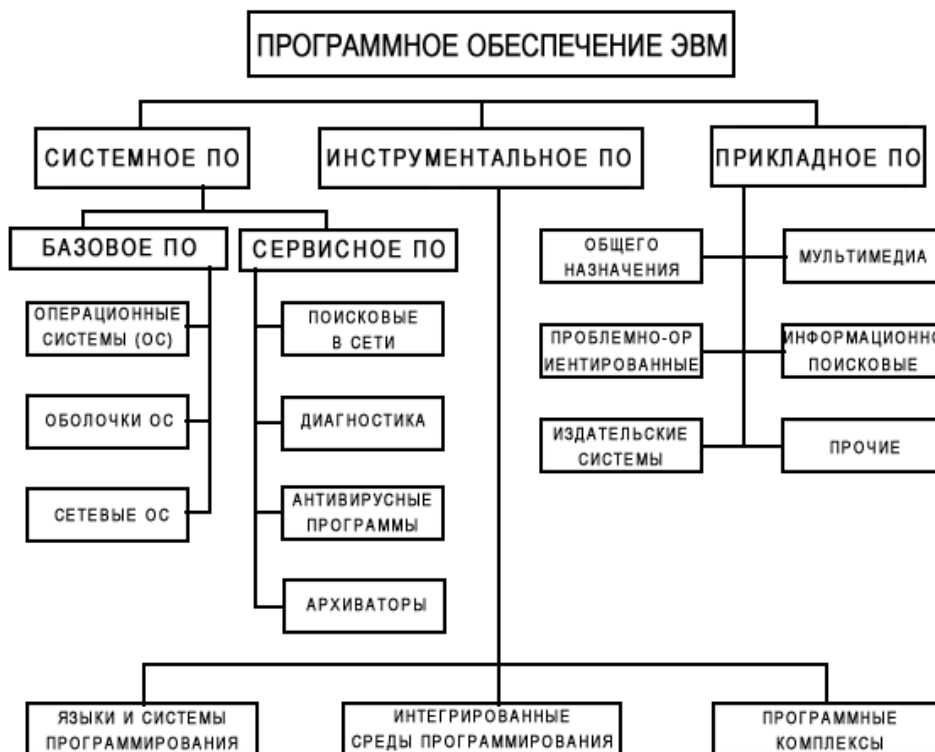


Рис. 1. Классификация программного обеспечения ЭВМ по назначению

Инструментальное ПО - совокупность средств, обеспечивающих процесс функционирования прикладных программ, т.е. обеспечивающих перевод кода программ, написанных пользователем с помощью высокоуровневого языка программирования на языки более низкого уровня – в машинный код. Это трансляторы, трансляторы-ассемблеры, трансляторы-интеграторы, трансляторы-компиляторы.

Прикладное ПО – совокупность программ, обеспечивающих использование ЭВМ при решении частных задач, необходимых конкретным группам пользователей.

Прикладное ПО делится на:

- универсальные (общецелевые) программы;
- специализированные (проблемно-ориентированные) программы.

К универсальным (общецелевым) программам относятся следующие:

- текстовые редакторы;
- процессоры электронных таблиц;
- презентационные программы;
- программы работы с базами данных (СУБД);
- графические редакторы;
- мультимедиа, аудио, видео и анимационные программы;
- системы поиска данных;
- программы электронного документооборота;
- языки программирования;
- искусственный интеллект и экспертные системы;
- программы разработки веб-страниц;
- программы работы в локальных и глобальных сетях и т.д.;

К специализированным (проблемно-ориентированным) программам относятся следующие:

- научные программы (программные средства, предназначенные для решения задач математики, физики, химии, биологии, медицины и др. Например, системы компьютерной алгебры и т.д.);
- системы автоматизации процессов проектирования;
- издательские системы и переводчики;
- экономическое управление и системы банковской информации;
- программы для решения задач сложных систем и процессов и т.д.

2. Прикладные программные пакеты

Прикладные программы непосредственно обеспечивают выполнение необходимых пользователям работ: редактирование текстов, рисование картинок, обработка информации и т.д. До недавнего времени для решения научной задачи человеку было необходимо не только хорошо знать математику, но и уметь работать на компьютер, знать как минимум один язык программирования, а также иметь навыки использования одного из сложнейших методов решения. Сейчас же для тех, кто не хочет или не умеет программировать существуют программные средства, которые содержат готовые программы, электронные руководства и средства, выполняемые расчет типичных вычислений - прикладные программные пакеты.

Эти пакеты выполняют все работу или основную часть работы, необходимой пользователю: исследование проблемы (также в аналитической форме); анализ данных, выявление наличия решения; моделирование; оптимизация; построение графиков; документальное оформление и формирование результатов; создание презентации. Умело используя эти пакеты, специалисты могут: подогнать пакеты под свою проблематику, модифицировать программу, сэкономить время расчетов и т.д.

Начиная с конца XX века и до сегодняшнего времени интенсивное развитие технологий применяемых в ПК, привело к тому, что алгебраическая система компьютеров рассматривается с новой точки зрения. Применяемый в расчетных системах системам компьютерной алгебры «компьютерный интеллект» открывает новые возможности применения компьютерной техники. Эти системы для специалистов, создающих обеспечение прикладных программ создает следующие средства:

- высокоуровневые системы программирования;
- редакторы, позволяющие редактировать и создавать программы;
- современные многооконные интерфейсы, позволяющие пользователю работать в режиме непосредственной беседы;
- высокоуровневые справочные системы;
- комплекс правил и алгоритмов, повторно решающих математические выражения;
- программные процессоры, выполняющие числовые и аналитические операции;
- система диагностики, указывающая на ошибки при работе в режиме беседы;
- библиотека функций, вложенная непосредственно в ядро системы;
- комплекс пакетов, необходимых для внедрения и применения математических методов.

Эти средства позволяют полностью автоматизировать процессы вывода математической модели проблемы, выбора метода расчета, проведения расчетных экспериментов и анализа результатов при создании прикладного программного обеспечения. А это в свою очередь в корне меняет рамки традиционных технологий при построении прикладного программного обеспечения и решении задач на ЭВМ.

3. Классификация пакетов, применяемых в автоматизации

Среди прикладных программных пакетов значительную часть занимают пакеты, предназначенные для процессов вычисления, моделирования и проектирования, а также служащих для автоматизации систем управления. В общем случае эти разновидности прикладных программных пакетов можно назвать пакетами,

применяемыми в автоматизации (ППА), которые можно классифицировать и по строению, выполняемым задачам (в каких процессах применяется) и по другим свойствам.

Таким образом, по выполняемым задачам (в каких процессах применяется) ППА можно классифицировать следующим образом:

- пакеты, служащие для автоматизации процессов вычисления;
- пакеты, служащие для автоматизации процессов проектирования;
- пакеты, служащие для автоматизации процессов моделирования технических систем;
- пакеты, служащие для автоматизации систем управления технологическими процессами и т.д.

Примером пакетов, служащих для автоматизации процессов вычисления могут служить универсальные пакеты (системы компьютерной алгебры: Mathematica, Maple, MathCAD, Matlab, Derive, Reduce, Eureka, Statistica и другие), пакеты, предназначенные для решения прикладных задач различных отраслей (например, для бухгалтерского учета: 1С, 2С и другие), а также пакеты, направленные на решение сложных научно-исследовательских проблем.

Примером пакетов, служащих для автоматизации процессов проектирования могут служить CAD (Computer aided design – проектирование с помощью компьютера) пакеты (например, AutoCAD, ArchiCAD, OrCAD, PCAD и т.д.), CAE (Computer aided engineering – компьютерная инженерия) пакеты. Более подробная классификация САПР (система автоматизированного проектирования) представлена на рис. 2.

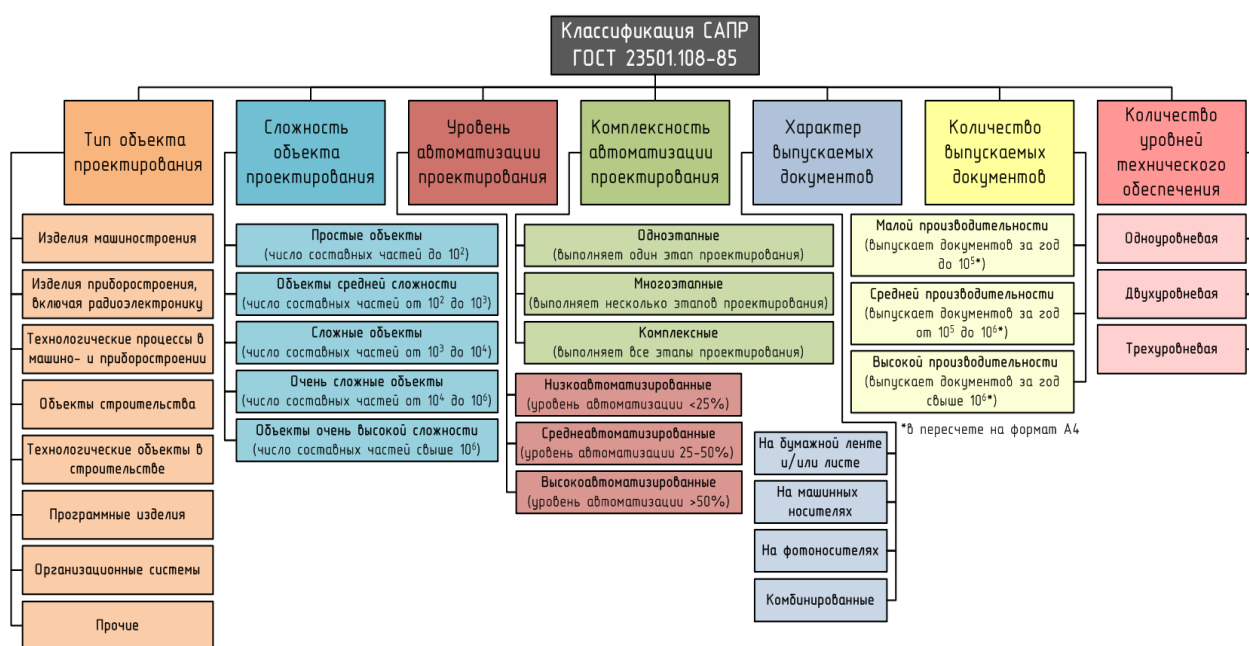


Рис. 2. Классификация САПР

Примером пакетов, служащих для автоматизации процессов моделирования технических систем могут быть структурные (или блочные) пакеты моделирования (например, Matlab/Simulink – все расширения, входящие в пакет Matlab, VisSim, LabView, Classic, MBTU (Моделирование в технических устройствах) и т.д.), физические мультидоменные пакеты моделирования (например, Matlab/SimMechanics, Matlab/SimPower Systems, Matlab/SimPower Electronics, MultiSim, Dymola, DYNASTи т.д.), а также пакеты моделирования сложных гибридных (смешанных) систем (например: Model Vision Studium и т.д.).

Пакеты, служащие для автоматизации систем управления технологическими процессами можно разделить на три класса:

- А класс: SCADA пакеты, служащие для автоматизации систем управления технологическими процессами (например: TraceModeи т.д.);
- В класс: пакеты, служащие для автоматизации систем управления ресурсами производства (например, DRCS (Distributed Resource Control System – система управления распределение ресурсов) и т.д.);
- С класс: пакеты, служащие для автоматизации систем планирования предприятий производства (например, MPCS (Manufactoriing Planning Control System – система управления планированием производства) и т.д.).

Программные продукты класса SCADA (Supervisory Control And Data Acquisition – диспетчерское управление и сбор данных) получили широкое распространение на мировом рынке. Существуют десятки пакетов SCADA, многие из которых используются и в Узбекистане. Самые распространенные из них – это:

- In Touch (Wonderware) - США;
- Sitect (CI Technologi) - Австралия;
- FIX (Intelection) - США;

- Genesis (Iconics Co) - США;
- Factory Link (United States Data Co) - США;
- Real Flex (BJ Software Systems) - США;
- Sitex (Jade Software) - Великобритания;
- Trace Mode (Ad Astra) - Россия;
- Simplicity (GE Fanuc) - США;
- SARGON (NTV- Avtomatika) - Россия.

Контрольные вопросы:

1. Укажите основные принципы классификации ППА.
2. Расскажите основные понятия объекта проектирования.
3. Расскажите о степенях в структуре технического обеспечения.
4. Опишите принципы взаимодействия ППА с другими автоматизированными системами.

Тема 4	ПРОГРАММНЫЙ ПАКЕТ MATLAB. ОБЩИЕ ПОНЯТИЯ И ПРИНЦИПЫ	
Технология проведения лекционного занятия		
Время 2 часа	Количество студентов 40-60 человек	
Вид занятия	Введение. Визуальная лекция	
План лекционного занятия	<ol style="list-style-type: none"> 1. Окружающая среда «Matlab»; 2. Командное окно программы; 3. Стандартные функции в программе «Matlab»; 4. Математические вычисления; 	
Цель учебного занятия: Ознакомление студентов с программными пакетами Matlab, общие понятия и принципы. Применяемые отрасли программного пакета Matlab		
Педагогические задачи:		Результаты обучения:
<ul style="list-style-type: none"> - Ознакомление рабочее пространство системы MATLAB и её командное окно - Ознакомление с основными частями программного пакета Matlab - давать сведения об учёных и педагогах, занимающихся данной темой; -объяснить предмет дисциплины 		Студент: <ol style="list-style-type: none"> 1. Студент записывает в конспект и запоминает: общие понятия и принципы среда «Matlab». 2. Имеет представление о целях и задачах исследования в области автоматического управления. 3. Изучает анализ данных и визуализацию результатов технологических процессов
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.1. Объявляет тему и план занятия, знакомит с целью занятия	1.1. Слушает и записывает
2-этап Основной	<ol style="list-style-type: none"> 2.1. Привлекая внимание студентов к вопросам и плана лекции и основным понятиям темы. Проводит блиц-опрос беглый опрос по основным понятиям. 2.2. Продолжает изложение лекции основываясь на визуальных материалах. 2.3. Учат окружающую среду «Matlab». 2.4. Излагает представление о целях изучения среда «Matlab». 2.5.Объясняет решить математические вычисления в программе «Matlab». 	<ol style="list-style-type: none"> 2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают 2.3.Записывают и запоминают 2.4. Слушают и запоминают
3-этап Итоговый	<ol style="list-style-type: none"> 3.1. Делает выводы по лекции. 3.2. Дает оценку студентам. 3.3. Дает задание на самостоятельную подготовку. 	<ol style="list-style-type: none"> 3.1. Слушают и усваивают 3.2. Записывают задание

Лекция 4. Программный пакет Matlab. Общие понятия и принципы.

План:

1. Окружающая среда «Matlab»;
2. Командное окно программы;
3. Стандартные функции в программе «Matlab»;
4. Математические вычисления;

1. Система Matlab

Система MATLAB (Matrix Laboratory) разработана фирмой The MathWorks, Inc. (США) и является системой инженерных и научных расчётов для различных областей науки и техники. Среди них:

- математика и вычисления;
- разработка алгоритмов;
- вычислительный эксперимент;
- имитационное моделирование;
- анализ данных, исследование и визуализация результатов;
- исследования в области автоматического управления;
- статистическая обработка сигналов и процессов и др.

Система MATLAB – это:

1. Интерактивная система, которая позволяет производить вычисления с помощью непосредственного ввода команд с клавиатуры;
2. Огромная библиотека готовых функций и алгоритмов, реализующих наиболее распространённые методы вычислений, которые для облегчения их поиска разбиты на специализированные разделы (Toolboxes);
3. Язык программирования, позволяющий пользователю создавать собственные функции, а также законченные программные приложения, в том числе и с использованием библиотечных функций системы.

Система MATLAB состоит из этих главных частей:

Настольные Инструменты и Среда проектирования - Эта часть MATLAB - набор инструментов и средств, которые помогают нам использовать и стать более продуктивными с функциями MATLAB и файлами. Многие из этих инструментов - графические интерфейсы пользователя. Это включает: рабочий стол MATLAB и Окно Команды, редакторы, коды анализаторов и браузеры, рабочее пространство и папки.

Математическая Функциональная Библиотека - Эта библиотека - обширная коллекция вычислительных алгоритмов в пределах от элементарных функций, как сумма, синус, косинус и сложная арифметика, к более сложным функциям как матричная инверсия, матричные собственные значения, и быстрый Фурье преобразовывает.

Язык - язык MATLAB - язык матрицы/множества высокого уровня, проверяют утверждения потока, функции, структуры данных, ввод/вывод и особенности объектно-ориентированного программирования. Это позволяет “маленькая программирования”, быстро создавать быстрые программы, которые Вы не намереваетесь снова использовать. Вы можете также сделать “большое программирование”, чтобы создать сложные приложения, предназначенные для повторного использования.

2. Рабочее пространство системы MATLAB и её командное окно

Когда Вы запускаете MATLAB и начинаете производить вычисления, в командном окне показываются вводимые с клавиатуры числа, переменные (через их имена), результаты вычислений. Обычно вычисления повторяются вновь и вновь: вводятся с клавиатуры новые числовые данные и новые символьные выражения. В результате в окне MATLABа не хватает свободного места и производится "скроллинг" ("протяжка"; "прокрутка") — все строки сдвигаются на одну позицию вверх, так что самая верхняя строка покидает область видимости, а в самом низу окна появляется свободная строка для ввода новых данных. Естественно, эта строка содержит знак приглашения>>.

Та информация, что покинула видимую часть окна, никуда не исчезает. Её всегда можно просмотреть снова, если осуществить прокрутку содержимого окна стандартным графическим средством управления — полосой прокрутки (на английском— Scrollbar). Для этого нужно щелкнуть мышью на этой полосе, или протянуть с помощью мыши ползунок полосы прокрутки в нужном направлении (вверх или вниз).

Можно также осуществлять прокрутку содержимого командного окна системы MATLAB с помощью следующих клавиш клавиатуры: PageUp, PageDown, Ctrl+Home (одновременное нажатие клавиш Ctrl и Home) и Ctrl+End.

Клавиши "Стрелка вверх" и "Стрелка вниз", в любом текстовом редакторе осуществляющие перемещение курсора вверх-вниз и прокрутку содержимого окна, в системе MATLAB работают по-другому. Эти клавиши позволяют вернуть в строку ввода ранее введённые с клавиатуры команды и другую входную информацию, то есть вся эта информация запоминается в специальной области памяти. Эту область памяти называют *стеком команд*, так самая последняя входная информация при её прокрутке клавишей "Стрелка вверх" появится первой. Затем появится предпоследняя команда и так далее. Клавиша "Стрелка вниз" осуществляет прокрутку команд в противоположном направлении.

В итоге можно сказать, что вся видимая информация в окне системы MATLAB располагается в двух принципиально разных зонах: *зоне просмотра* и *зоне редактирования*.

Примером простейшей команды служит ввод числовой константы. Результатом выполнения команды станет сама числовая константа, сохраненная во временной переменной *ans* (англ. *answer* - ответ). Переменная *ans* автоматически создается системой для сохранения результата любой команды, введенной в командном окне. После выполнения команды результат помещается в *ans* и отображается в рабочем пространстве.

```
>> ans = 1
ans =
1
```

Примером более сложного действия служит выполнение математических операций над числовыми константами.

```
>> 2 + 4
ans =
6
```

Как и в любом другом языке программирования в *MATLAB* можно создавать переменные и различные объекты в памяти. Например, результат выполнения выше приведенных команд можно поместить в отдельные переменные.

```
>> x = 1
x =
1
>> y = 2 + 4
y =
6
```

Чтобы не «загромождать» командное окно выводом результатов следует использовать символ «;» (точку с запятой) после введенной команды.

```
>> y = 2 + 4;
```

Кроме того, можно использовать команду *clc* для очистки командного окна от содержащегося текста. Если необходимо удалить созданный ранее объект или переменную из памяти используется команда *clear* или *clear all*. Также, можно удалять отдельные объекты и переменные с помощью команды *clear* [имя_объекта]. Например,

```
>> clear y;
```

Как упоминалось выше, основным типом данных в *MATLAB* является массив (матрица). Даже в рассмотренных выше примерах числовые константы и переменные являются массивами, состоящими из одного элемента.

Для того чтобы задать массив используются квадратные скобки, элементы массива разделяются пробелами.

```
>> A = [1 2 3]
A =
1 2 3
```

Для транспонирования после имени или описания массива вводится знак апострофа «'».

```
>> A = [1 2 3]'
A =
1
2
3
>> A'
ans =
1 2 3
```

Многомерные массивы задаются следующим образом:

```
>> B = [1 2 3; 4 5 6]
B =
1 2 3
4 5 6
```

Все арифметические операции, определенные для чисел также определены и для массивов.

```
>> A = [10 20 30; 40 50 60];
>> B = [1 2 3; 4 5 6];
>> C = A + B
C =
11 22 33
44 55 66
>> C = A - B
C =
9 18 27
36 45 54
>> C = A * B'
C =
```

```

140 320
320 770
>> C = A / B
C =
10.0000 0.0000 0.0000 10.0000

```

Такие операции как умножение и деление также имеют свои аналоги для поэлементного умножения и деления.

```

>> C = A .* B
C =
10 40 90
160 250 360
>> C = A ./ B
C =
10 10 10
10 10 10

```

Полное описание всех матричных операций можно найти в электронной документации

Все операции в Matlab ориентированы прежде всего на работу с матрицами, но могут быть использованы при работе с векторами и скалярными переменными.

3. Элементарные функции Matlab.

К ним относятся следующие функции:

Тригонометрические:

```

sin(x) sinh(x) asin(x) asinh(x)
cos(x) cosh(x) acos(x) acosh(x)
tan(x) tanh(x) atan(x) atan2(x,y) atanh(x)
cot(x) coth(x) acot(x) acoth(x)

```

Экспоненциальные:

```

exp(x) log(x) log10(x) log2(x) pow2(x) sqrt(x)

```

Функции комплексного аргумента:

```

abs(z) – модуль комплексного числа; angle(z) – аргумент;
complex(x, y) – формирование комплексного числа x+у*і;
conj(z) – возвращает комплексно-сопряженное (по отношению к z) число;
real(z) – возвращает действительную часть; imag(z) – мнимую часть.

```

Функции округления и нецелочисленного деления:

```

fix(x) – округление в сторону к нулю, floor(x) – к -∞, ceil(x) – к +∞,
round(x) – до ближайшего целого;
mod(x, y) – остаток от деления x/y с учетом знака,
rem(x, y) – то же, без учета знака;
sign(x) – знак числа (+1/-1).

```

Элементарные функции, как и всякие другие, имеют один результат (массив). Элементарные функции могут быть использованы с одинаковым синтаксисом как для чисел, так и для массивов.

Пример 1. Разный тип параметров

```

>> x=-8; X=[5 -1 -2]; A=[1 2 -3; -5 6 7]; z=3-4i;
>> abs(x)
ans =
5
>> abs(X)
ans =
5 1 2
>> abs(A)
ans =
1 2 3
5 6 7
>> abs(z)
ans =
5

```

Контрольные вопросы:

1. Для чего предназначен Matlab.
2. Какие возможности имеет Matlab
3. Опишите командное окно Matlab

Тема 5	ЭЛЕМЕНТЫ ПРОГРАММИРОВАНИЯ В MATLAB	
Технология проведения лекционного занятия		
Время 2 часа	Количество студентов 40-60 человек	
Вид занятия	Введение. Визуальная лекция	
План лекционного занятия	1. Основные понятия программирования; 2. Виды программирования; 3. М-ФАЙЛЫ; 4. Примеры элементов программирования;	
Цель учебного занятия: Ознакомление студентов с типами данных Matlab, виды программирования и стандартные функции. Понятие М- файлов сценарий и функций		
Педагогические задачи:		Результаты обучения:
<ul style="list-style-type: none"> - ознакомление с основными понятиями, средствами, видами программирования; - ознакомление с основными типами данных системы Matlab; - ознакомить с различной литературой, связанной с данной дисциплиной; - объяснить предмет дисциплины 		Студент: <ul style="list-style-type: none"> 4. Студент конспектирует и запоминает: основные понятия программирования среды «Matlab». 5. Имеет представление о целях и задачах программирования в области автоматического управления. 6. Изучает анализ данных и визуализацию результатов технологических процессов
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.1. Объявляет тему и план занятия, знакомит с целью занятия	1.1. Слушает и записывает
2-этап Основной	2.1. Привлекая внимание студентов к вопросам и плана лекции и основным понятиям темы. Проводит блиц-опрос беглый опрос по основным понятиям. 2.2. Продолжает изложение лекции основываясь на визуальных материалах. 2.3. Учитель понятия и виды программирования в среде «Matlab». 2.4. Излагает цели и задачи изучения среды «Matlab». 2.5. Объясняет примеры программирования в среде «Matlab».	2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают 2.3. Записывают и запоминают 2.4. Слушают и запоминают
3-этап Итоговый	3.1. Делает выводы по лекции. 3.2. Дает оценку студентам. 3.3. Дает задание на самостоятельную подготовку.	3.1. Слушают и усваивают 3.2. Записывают задание

Лекция 5. Элементы программирования в MATLAB.

План:

1. Основные понятия программирования;
2. Виды программирования;
3. М-файлы;
4. Примеры элементов программирования;

1. Основные понятия программирования

До сих пор мы в основном использовали систему MATLAB в режиме непосредственного счета — в командном режиме. Однако при решении серьезных задач возникает необходимость сохранения используемых последовательностей вычислений, а также их дальнейшей модификации. Иными словами, существует необходимость *программирования* решения задач. Это может показаться отходом от важной цели, которая преследуется разработчиками большинства математических систем, — выполнения математических вычислений без использования традиционного программирования. Однако это не так. Выше было показано, что множество математических задач решается в системе MATLAB без программирования. С использованием языков высокого уровня для их решения потребовалось бы написать и оттестировать сотни программ.

Практически невозможно предусмотреть в одной, даже самой большой и мощной, математической системе возможность решения всех задач, которые могут интересовать пользователя. Программирование в системе MATLAB является эффективным средством ее расширения и адаптации к решению специфических проблем. Оно реализуется с помощью *языка программирования* системы.

Большинство объектов этого языка, в частности все команды, операторы и функции, одновременно являются объектами *входного языка* общения с системой в командном режиме работы. Так что фактически мы приступили к описанию языка программирования системы MATLAB с первых строк данной книги.

Так в чем же отличие входного языка от языка программирования? В основном — в способе фиксации создаваемых ими кодов. Сессии в командном режиме работы не сохраняются в памяти компьютера (ведение дневника не в счет). Хранятся только определения созданных в ходе их выполнения переменных и функций. А вот программы на языке программирования MATLAB сохраняются в виде текстовых m-файлов. При этом могут сохраняться как целые программы в виде файлов-сценариев, так и отдельные *программные модули* — функции. Кроме того, важно, что программа может менять структуру алгоритмов вычислений в зависимости от входных данных и данных, создаваемых в ходе вычислений.

С позиций программиста язык программирования системы является типичным *проблемно-ориентированным* языком программирования высокого уровня. Точнее говоря, это даже язык *сверхвысокого* уровня, содержащий сложные операторы и функции, реализация которых на обычных языках (например, Бейсике, Паскале или Си) потребовала бы много усилий и времени. К таким функциям относятся матричные функции, функции быстрого преобразования Фурье (БПФ) и др., а к операторам — операторы построения разнообразных графиков, генерации матриц определенного вида и т. д.

2. Основные средства программирования

Итак, программами в системе MATLAB являются m-файлы текстового формата, содержащие запись программ в виде программных кодов. Язык программирования системы MATLAB имеет следующие средства:

- данные различного типа;
- константы и переменные;
- операторы, включая операторы математических выражений;
- встроенные команды и функции;
- функции пользователя;
- управляющие структуры;
- системные операторы и функции;
- средства расширения языка.

Коды программ в системе MATLAB пишутся на языке высокого уровня, достаточно понятном для пользователей умеренной квалификации в области программирования. Язык программирования MATLAB является типичным *интерпретатором*. Это означает, что каждая инструкция программы распознается и тут же исполняется, что облегчает обеспечение диалогового режима общения с системой. Этап компиляции всех инструкций, т. е. полной программы, отсутствует. Высокая скорость выполнения программ обеспечена наличием заведомо откомпилированного ядра, хранящего в себе критичные к скорости выполнения инструкции, такие как базовые математические и иные функции, а также тщательной обработкой системы контроля синтаксиса программ в режиме интерпретации.

3. Основные типы данных

Структура типов данных системы MATLAB представлена ниже:

Типы данных `agau` и `numeric` являются *виртуальными* («кажущимися»), поскольку к ним нельзя отнести какие-либо переменные. Они служат для определения и комплектования некоторых типов данных. Таким образом, в MATLAB определены следующие основные типы данных, в общем случае представляющих собой многомерные массивы:

1. `single` — числовые массивы с числами одинарной точности;
2. `double` — числовые массивы с числами удвоенной точности;
3. `char` — строчные массивы с элементами-символами;
4. `sparse` — наследует свойства `double`, разреженные матрицы с элементами-числами удвоенной точности;
5. `cell` — массивы ячеек; ячейки, в свою очередь, тоже могут быть массивами;
6. `struct` — массивы структур с полями, которые также могут содержать массивы;
7. `function_handle` — дескрипторы функций;
8. `int32`, `uint32` — массивы 32-разрядных чисел со знаком и без знаков;
9. `int16`, `uint16` — массивы 16-разрядных целых чисел со знаком и без знаков;
10. `int8`, `uint8` — массивы 8-разрядных целых чисел со знаками и без знаков.

Кроме того, предусмотрен еще один тип данных — `UserObject`, который относится к типам данных (объектом), определяемым пользователем. Типы данных `double`, `char` и `sparse` были рассмотрены ранее, так что в этой главе будут детально рассмотрены оставшиеся типы. Что касается чисел класса `uint8`, то они представляют значения от 0 до 255 и занимают в памяти 1/8 часть от размера одного числа с двойной точностью. В основном этот тип данных применяется в служебных целях.

Каждому типу данных можно соотнести некоторые характерные для него операции, называемые *методами*. Дочерние типы данных, расположенные на приведенной диаграмме ниже родительских типов, наследуют от последних их методы, что является признаком наследования объектов. Поскольку в иерархии типов данных сверху находятся данные типа `array`, это значит, что все виды данных в MATLAB являются массивами.

4. Виды программирования

На рынке программного обеспечения система MATLAB позиционируется как язык высокого уровня для научно-технических расчетов. Таким образом, возможность программирования относится к важным достоинствам данного языка, несмотря на обилие средств прямого решения задач. И действительно, именно возможность программирования сложных задач и практически неограниченного расширения системы сделала MATLAB столь почитаемой системой в университетах и крупных научных учреждениях. MATLAB открывает широчайшие возможности реализации новых алгоритмов вычислений, численных методов и методик расчета и проектирования различных систем и устройств.

Язык программирования системы MATLAB вобрал в себя все средства, необходимые для реализации различных видов программирования:

- процедурного;
- операторного;
- функционального;
- логического;
- структурного (модульного);
- объектно-ориентированного;
- визуально-ориентированного.

В основе *процедурного*, *операторного* и *функционального* типов программирования лежат процедуры, операторы и функции, используемые как основные объекты языка. Эти типы объектов присутствуют в MATLAB. *Логическое* программирование реализуется в MATLAB с помощью логических операторов и функций. Это позволяет реализовать основные идеи логического программирования, хотя на выдающуюся роль в этом классе языков программирования MATLAB не претендует.

Тема 6	ПОСТРОЕНИЕ ГРАФИКОВ ФУНКЦИИ В MATLAB	
Технология проведения лекционного занятия		
Время 2 часа	Количество студентов 40-60 человек	
Вид занятия	Введение. Визуальная лекция	
План лекционного занятия	<ol style="list-style-type: none"> 1. Графические команды и функции 2. Элементарные графические функции 3. Двумерные и Трехмерные график 4. Надписи и пояснения к графикам 5. Специальная графика 	
Цель учебного занятия: Формирование у студентов глубокого представления о целях и задачах математические пакеты в решении инженерных задач и путях её дальнейшего развития с помощью MATLAB.		
Педагогические задачи:		Результаты обучения:
<ul style="list-style-type: none"> - Ознакомление с основными задачами состав системы MATLAB - Ознакомление с историей развития MATLAB и выдающимся её представителями - Внедрение в сознание студентов основных графические команды 		Студент: <ol style="list-style-type: none"> 1. Имеет представление о целях и задачах системы MATLAB 2. Излагает историю развития системы MATLAB и выдающихся её представителей 3. Будет знать, как строить на экране и вывести на печатающее устройство типы графиков 4. Получат навыки решения следующих задач: нахождение нулей и точек экстремума функции, вычисление определенных интегралов, решение дифференциальных уравнений
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.1. Приводит список используемой литературы по системе MATLAB. Знакомит студентов с рейтинговой оценкой знаний студентов. Объявляет тему и план занятия, знакомит с достижениями химической науки	1.1. Слушает и записывает
2-этап Основной	<ol style="list-style-type: none"> 2.1. Обращает внимание студентов на вопросы плана занятия и обращает внимание на основные понятия. Проводит опрос. 2.2. Продолжает излагать лекционный материал. На примерах приводит и объясняет основные двумерные и трехмерные графики. 2.3. Обеспечивает студентом внимание и запись основных понятий и законов химии. 	<ol style="list-style-type: none"> 2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают
3-этап Итоговый	<ol style="list-style-type: none"> 3.1. Делает заключительные выводы лекционной темы и заостряет внимание студентов на основные вопросы 3.2. Отмечает активных студентов 3.3. Даст задание на самостоятельную подготовку: Дать определение каждого команды, привести примеры 	<ol style="list-style-type: none"> 3.1. Слушают и усваивают 3.2. Записывают задание

Лекция 6. Построение графиков функции в MATLAB

План:

1. Графические команды и функции
2. Элементарные графические функции
3. Двумерные и Трехмерные график
4. Надписи и пояснения к графикам
5. Специальная графика

1. Графические команды и функции

Начиная с версии 4.0 в состав системы MATLAB входит мощная графическая подсистема, которая поддерживает как средства визуализации двумерной и трехмерной графики на экран терминала, так и средства презентационной графики. Следует выделить несколько уровней работы с графическими объектами. В первую очередь это команды и функции, ориентированные на конечного пользователя и предназначенные для построения графиков в прямоугольных и полярных координатах, гистограмм и столбчатых диаграмм, трехмерных поверхностей и линий уровня, анимации. Графические команды высокого уровня автоматически контролируют масштаб, выбор цветов, не требуя манипуляций со свойствами графических объектов. Соответствующий низкоуровневый интерфейс обеспечивается дескрипторной графикой, когда каждому графическому объекту ставится в соответствие графическая поддержка (дескриптор), на который можно ссылаться при обращении к этому объекту. Используя дескрипторную графику, можно создавать меню, кнопки вызова, текстовые панели и другие объекты графического интерфейса.

Из-за ограниченного объема данного справочного пособия в него включены только графические команды и функции с минимальными элементами дескрипторной графики. Заинтересованному читателю следует обратиться к документации по системе MATLAB, и в первую очередь к только что вышедшей из печати книге "Using MATLAB Graphics" (Natick, 1996).

2. Элементарные графические функции

Элементарные графические функции системы MATLAB позволяют построить на экране и вывести на печатающее устройство следующие типы графиков: линейный, логарифмический, полулогарифмический, полярный.

Для каждого графика можно задать заголовок, нанести обозначение осей и масштабную сетку.

Двумерные графики

- [PLOT](#) - график в линейном масштабе
- [LOGLOG](#) - график в логарифмическом масштабе
- [SEMILOGX](#), [SEMILOGY](#) - график в полулогарифмическом масштабе
- [POLAR](#) - график в полярных координатах

Трехмерные графики

В системе MATLAB предусмотрено несколько команд и функций для построения трехмерных графиков. Значения элементов числового массива рассматриваются как z-координаты точек над плоскостью, определяемой координатами x и y. Возможно несколько способов соединения этих точек. Первый из них - это соединение точек в сечении (функция plot3), второй - построение сетчатых поверхностей (функции mesh и surf). Поверхность, построенная с помощью функции mesh, - это сетчатая поверхность, ячейки которой имеют цвет фона, а их границы могут иметь цвет, который определяется свойством EdgeColor графического объекта surface. Поверхность, построенная с помощью функции surf, - это сетчатая поверхность, у которой может быть задан цвет не только границы, но и ячейки; последнее управляется свойством FaceColor графического объекта surface. Уровень изложения данной книги не требует от читателя знания объектно-ориентированного программирования. Ее объем не позволяет в полной мере описать графическую подсистему, которая построена на таком подходе. Заинтересованному читателю рекомендуем обратиться к документации по системе MATLAB, и в первую очередь к только что вышедшей из печати книге Using MATLAB Graphics (Natick, 1996).

1. [PLOT3](#) - построение линий и точек в трехмерном пространстве
2. [MESHGRID](#) - формирование двумерных массивов X и Y
3. [MESH](#), [MESHZ](#), [MESHZ](#) - трехмерная сетчатая поверхность
4. [SURF](#), [SURFC](#) - затененная сетчатая поверхность
5. [SURFL](#) - затененная поверхность с подсветкой
6. [AXIS](#) - масштабирование осей и вывод на экран
7. [GRID](#) - нанесение сетки
8. [HOLD](#) - управление режимом сохранения текущего графического окна
9. [SUBPLOT](#) - разбиение графического окна
10. [ZOOM](#) - управление масштабом графика
11. [COLORMAP](#) - палитра цветов
12. [CAXIS](#) - установление соответствия между палитрой цветов и масштабированием осей

13. [SHADING](#) - затенение поверхностей
14. [CONTOURC](#) - формирование массива описания линий уровня
15. [CONTOUR](#) - изображение линий уровня для трехмерной поверхности
16. [CONTOUR3](#) - изображение трехмерных линий уровня

3. Надписи и пояснения к графикам

- [TITLE](#) - заголовки для двух- и трехмерных графиков
- [XLABEL, YLABEL, ZLABEL](#) - обозначение осей
- [CLABEL](#) - маркировка линий уровня
- [TEXT](#) - добавление к текущему графику текста
- [GTEXT](#) - размещает заданный текст на графике с использованием мыши
- [LEGEND](#) - пояснение к графику
- [COLORBAR](#) - шкала палитры

Специальная графика

Раздел специальной графики включает графические команды и функции для построения столбцовых диаграмм, гистограмм, средств отображения векторов и комплексных элементов, вывода дискретных последовательностей данных, а также движущихся траекторий как для двумерной, так и для трехмерной графики. Этот раздел получил свое дальнейшее развитие в версии системы MATLAB 5.0, где специальные графические средства улучшены и существенно расширены.

[BAR](#) - столбцовые диаграммы

[ERRORBAR](#) - график с указанием интервала погрешности

[HIST](#) - построение гистограммы

[STEM](#) - дискретные графики

[STAIRS](#) - ступенчатый график

[ROSE](#) - гистограмма в полярных координатах

[COMPASS, FEATHER](#) - графики векторов

[QUIVER](#) - поле градиентов функции

[COMET](#) - движение точки по траектории

[FILL](#) - закраска многоугольника

[COMET3](#) - движение точки по пространственной траектории

[SLICE](#) - сечения функции от трех переменных

[WATERFALL](#) - трехмерная поверхность

[FILL3](#) - закраска многоугольника в трехмерном пространстве

[VIEWMTX](#) - вычисление матрицы управления углом просмотра

[VIEW](#) - управление положением точки просмотра

Синтаксис:

```
plot(y)
plot(x, y)
plot(x, y, s)
plot(x1, y1, s1, x2, y2, s2, ...)
```

Описание:

Команда `plot(y)` строит график элементов одномерного массива `y` в зависимости от номера элемента; если элементы массива `y` комплексные, то строится график `plot(real(y), imag(y))`. Если `Y` - двумерный действительный массив, то строятся графики для столбцов; в случае комплексных элементов их мнимые части игнорируются.

Команда `plot(x, y)` соответствует построению обычной функции, когда одномерный массив `x` соответствует значениям аргумента, а одномерный массив `y` - значениям функции. Когда один из массивов `X` или `Y` либо оба двумерные, реализуются следующие построения:

- если массив `Y` двумерный, а массив `x` одномерный, то строятся графики для столбцов массива `Y` в зависимости от элементов вектора `x`;
- если двумерным является массив `X`, а массив `y` одномерный, то строятся графики столбцов массива `X` в зависимости от элементов вектора `y`;
- если оба массива `X` и `Y` двумерные, то строятся зависимости столбцов массива `Y` от столбцов массива `X`.

Команда `plot(x, y, s)` позволяет выделить график функции, указав способ отображения линии, способ отображения точек, цвет линий и точек с помощью строковой переменной `s`, которая может включать до трех символов из следующей таблицы:

Тип линии	Тип точки	Цвет
-----------	-----------	------

Непрерывная	-	Точка	.	Желтый	y
Штриховая	--	Плюс	+	Фиолетовый	m
Двойной пунктир	:	Звездочка	*	Голубой	c
Штрих-пунктирная	-.	Кружок	o	Красный	r
		Крестик	x	Зеленый	g
				Синий	b
				Белый	w
				Черный	k

Если цвет линии не указан, он выбирается по умолчанию из шести первых цветов, с желтого до синего, повторяясь циклически.

Команда `plot(x1, y1, s1, x2, y2, s2, ...)` позволяет объединить на одном графике несколько функций $y_1(x_1)$, $y_2(x_2)$, ..., определив для каждой из них свой способ отображения.

Обращение к командам `plot` вида `plot(x, y, s1, x, y, s2)` позволяет для графика $y(x)$ определить дополнительные свойства, для указания которых применения одной строковой переменной `s1` недостаточно, например при задании разных цветов для линии и для точек на ней.

4. Примеры:

Построим график функции $y = \sin(x)$ на отрезке $[-\pi; \pi]$ с шагом $\pi/500$:

```
x = -pi:pi/500:pi;
```

```
y = sin(x);
```

```
plot(y) % рис. а      plot(x, y) % рис. б
```

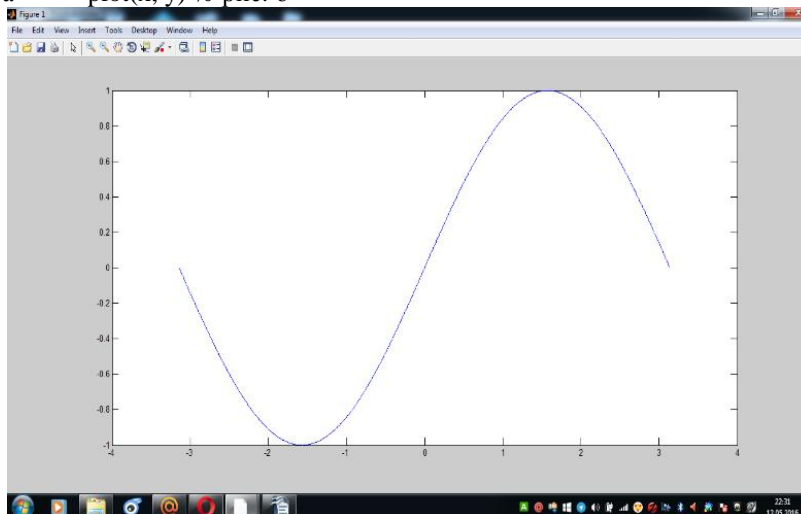


График на рис. а отображает значения одномерного массива y , состоящего из 1001 элемента, как функцию от номера элемента; график на рис. б отображает значения того же массива как функцию элементов массива x .

Рассмотрим различные способы применения функции `plot(x, y)` на примере графиков двух функций $y_1 = \sin(x)$ и $y_2 = x \sin(x)$:

```
x1 = -pi:pi/500:pi;
```

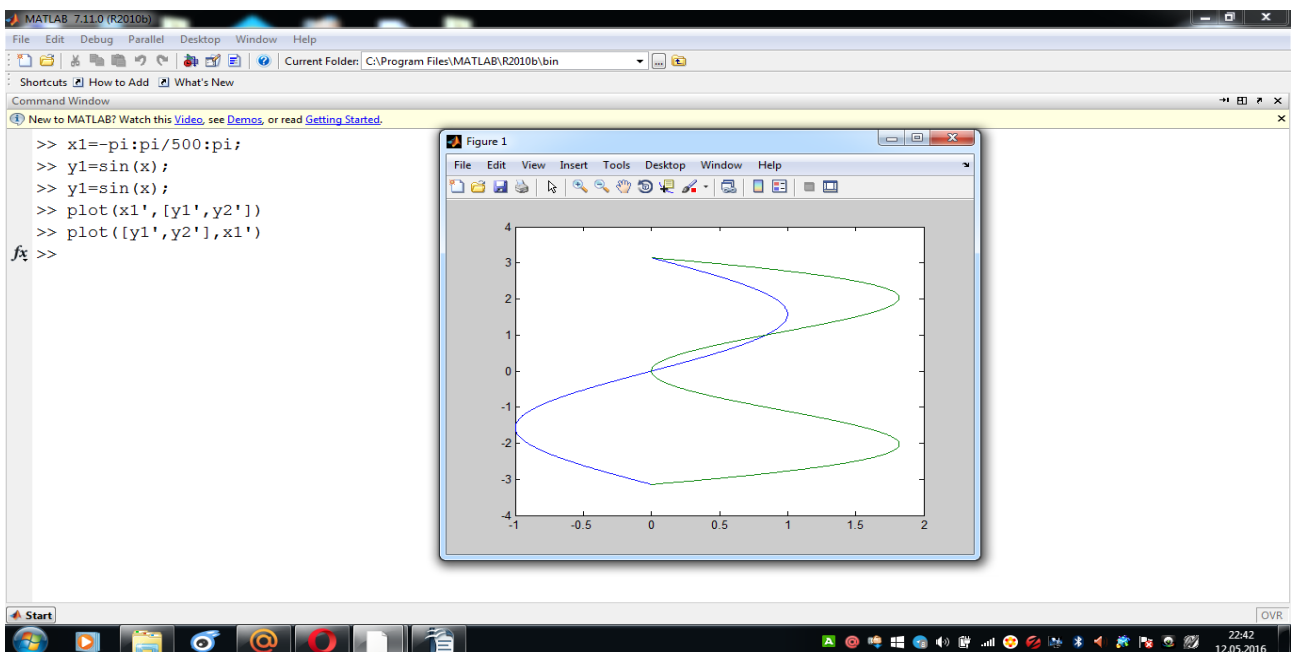
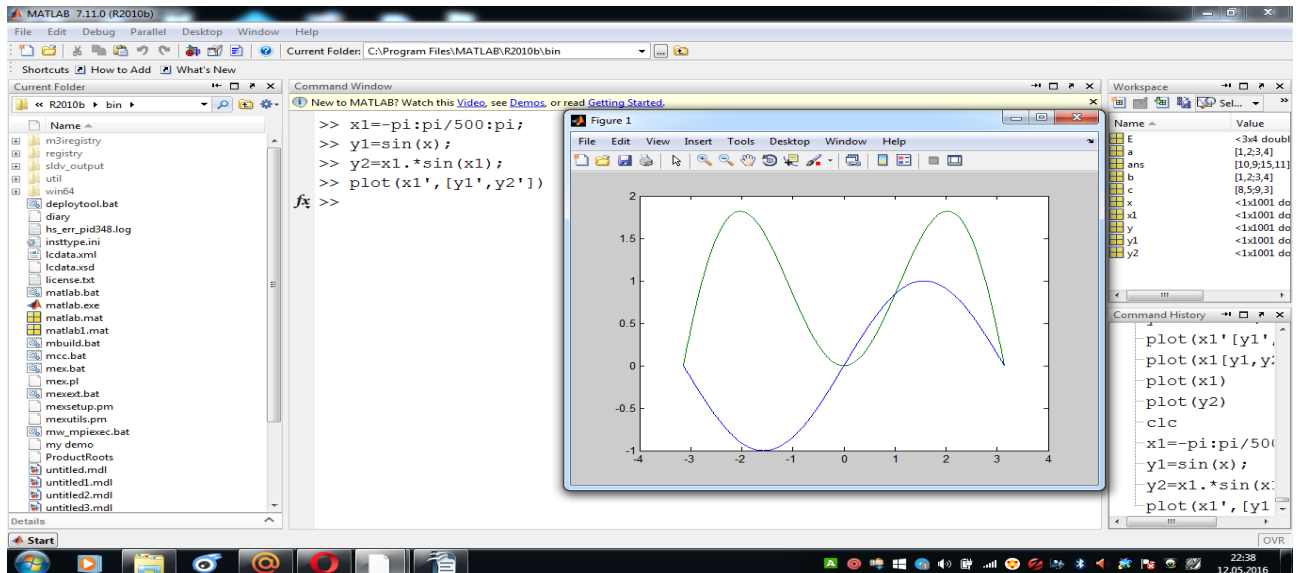
```
y1 = sin(x1);
```

```
y2 = x1.*sin(x1);
```

```
plot(x1',[y1' y2']) % рис. В
```

```
plot([y1' y2'], x1') % рис. Г= x1/2;
```

```
y2 = x2.*sin(x2);
```



Сопутствующие функции и команды: [LOGLOG](#), [SEMILOGX](#), [SEMILOGY](#), [POLAR](#).

Рассмотрение возможностей MatLab по визуализации данных начнем с двумерных графиков, которые обычно строятся с помощью функции plot(). Множество вариантов работы данной функции лучше всего рассмотреть на конкретных примерах.

Предположим, что требуется вывести график функции синуса в диапазоне от 0 до π . Для этого зададим вектор(множество) точек по оси Ox, в которых будут отображаться значения функции синуса:

$$x = 0:0.01:\pi;$$

В результате получится вектор столбец со множеством значений от 0 до π и с шагом 0,01. Затем, вычислим множество значений функции синуса в этих точках:

$$y = \sin(x);$$

и выведем результат на экран

$$\text{plot}(x,y);$$

В результате получим график, представленный на рис. 6.1.

Представленная запись функции plot() показывает, что сначала записывается аргумент со множеством точек оси Ox, а затем, аргумент со множеством точек оси Oy. Зная эти значения, функция plot() имеет возможность построить точки на плоскости и линейно их интерполировать для придания непрерывного вида графика.

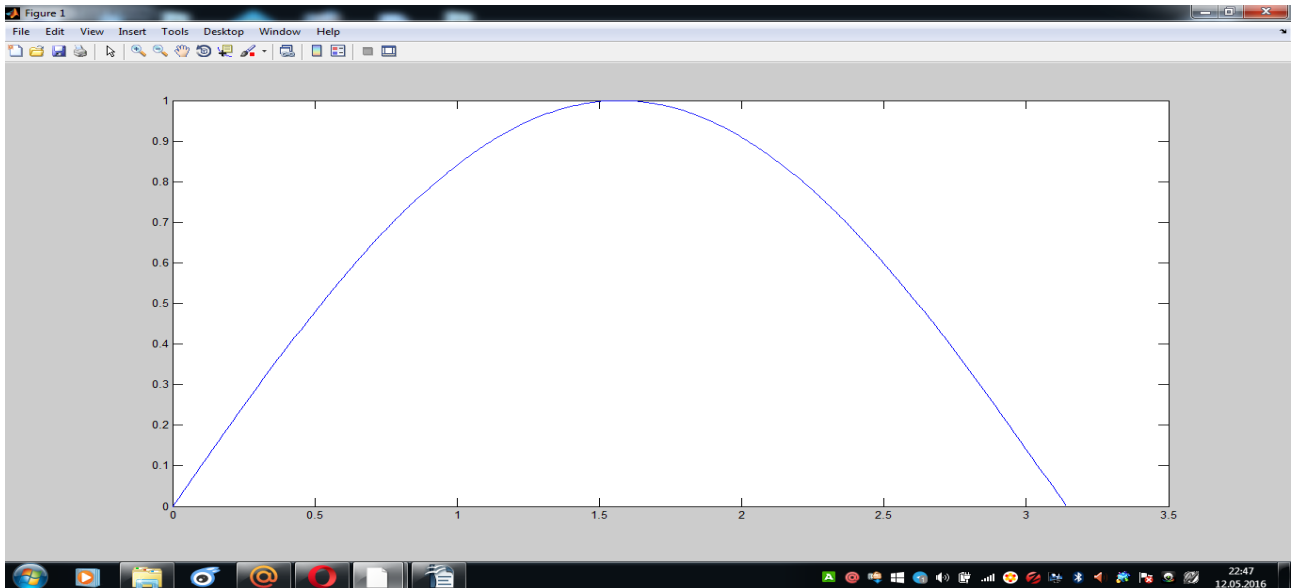
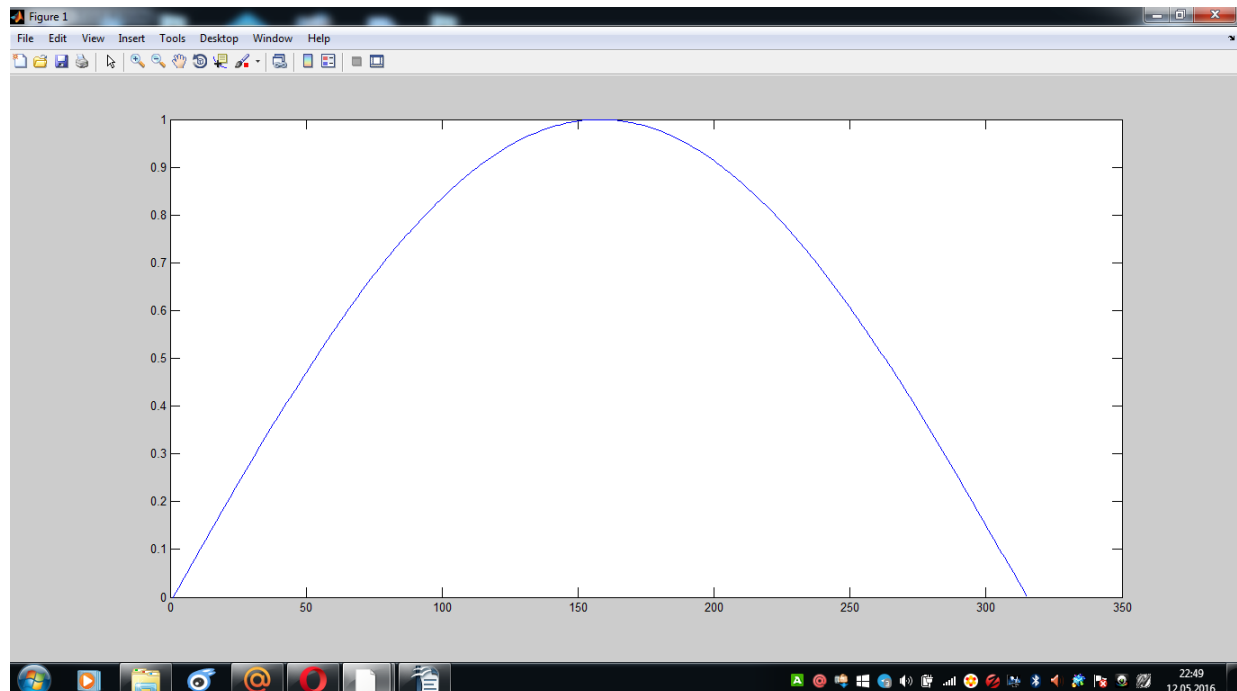


Рис. 6.1. Отображение функции синуса с помощью функции plot().



Функцию plot() можно записать и с одним аргументом x или y:

```
plot(x);
```

```
plot(y);
```

в результате получим два разных графика, представленные на рис. 6.2.

Анализ рис. 3.2 показывает, что в случае одного аргумента функция plot() отображает множество точек по оси Oy, а по оси Ox происходит автоматическая генерация множества точек с единичным шагом. Следовательно, для простой визуализации вектора в виде двумерного графика достаточно воспользоваться функцией plot() с одним аргументом.

Для построения нескольких графиков в одних и тех же координатных осях, функция plot() записывается следующим образом:

```
x = 0:0.01:pi;
```

```
y1 = sin(x);
```

```
y2 = cos(x);
```

```
plot(x,y1,x,y2);
```

Результат работы данного фрагмента программы представлен на рис. 6.3.

б)

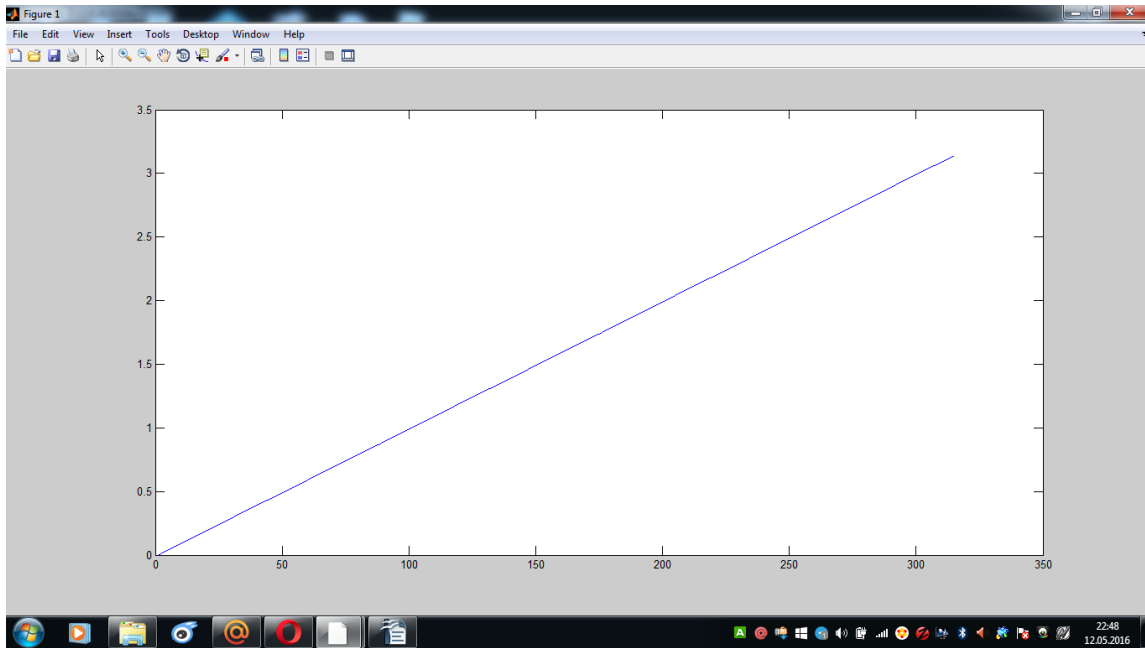


Рис. 6.2. Результаты работы функции plot() с одним аргументом:

а – plot(x); б – plot(y).

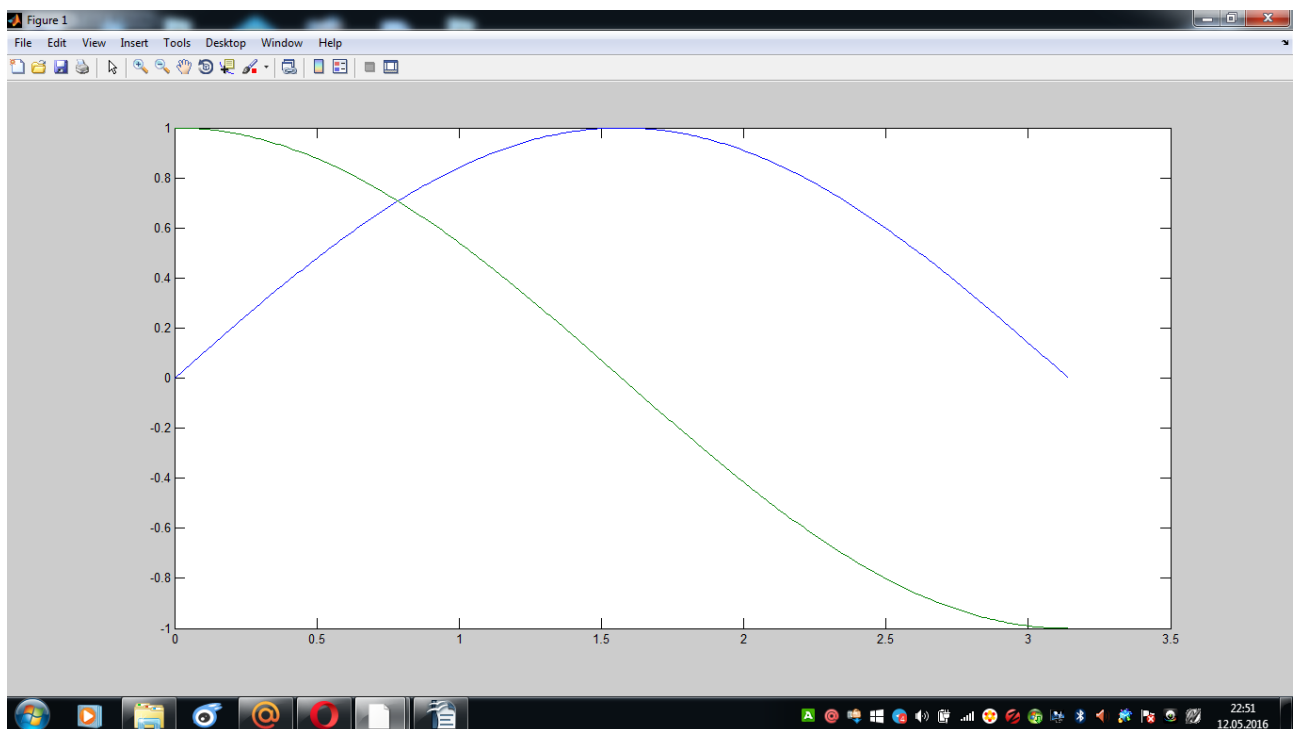


Рис. 6.3. Отображение двух графиков в одних координатных осях.

Аналогичным образом можно построить два графика, используя один аргумент функции plot(). Предположим, что есть два вектора значений

$$y1 = \sin(x);$$

$$y2 = \cos(x);$$

которые требуется отобразить на экране. Для этого объединим их в двумерную матрицу

$$[y1' y2'] = \begin{bmatrix} y1_1 & y2_1 \\ y1_2 & y2_2 \\ \dots & \dots \\ y1_N & y2_N \end{bmatrix},$$

в которой столбцы составлены из векторов y1 и y2 соответственно. Такая матрица будет отображена функцией plot([y1' y2']); апострофы переводят вектор-строку в вектор-столбец в виде двух графиков (рис. 3.4).

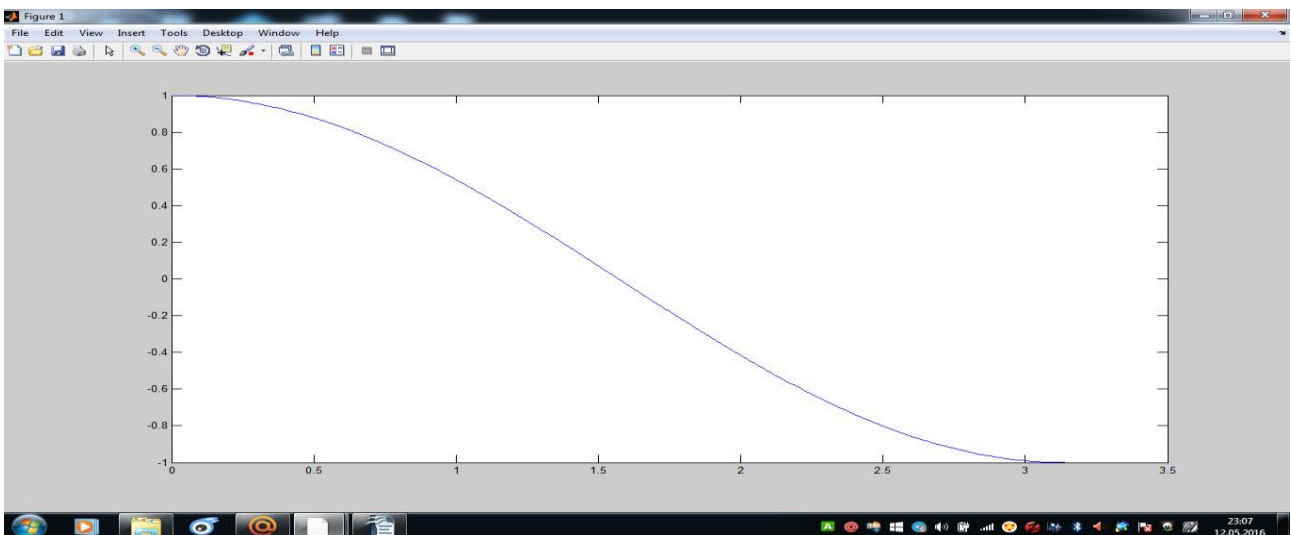
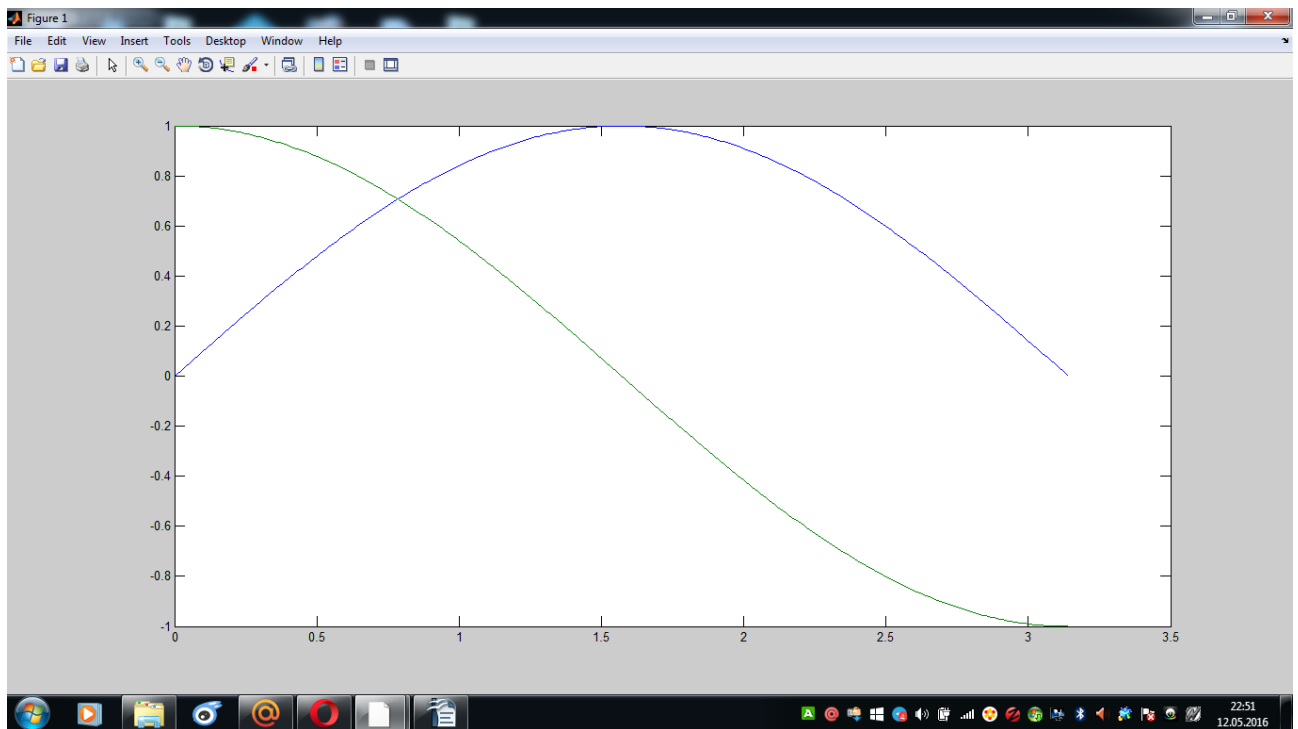
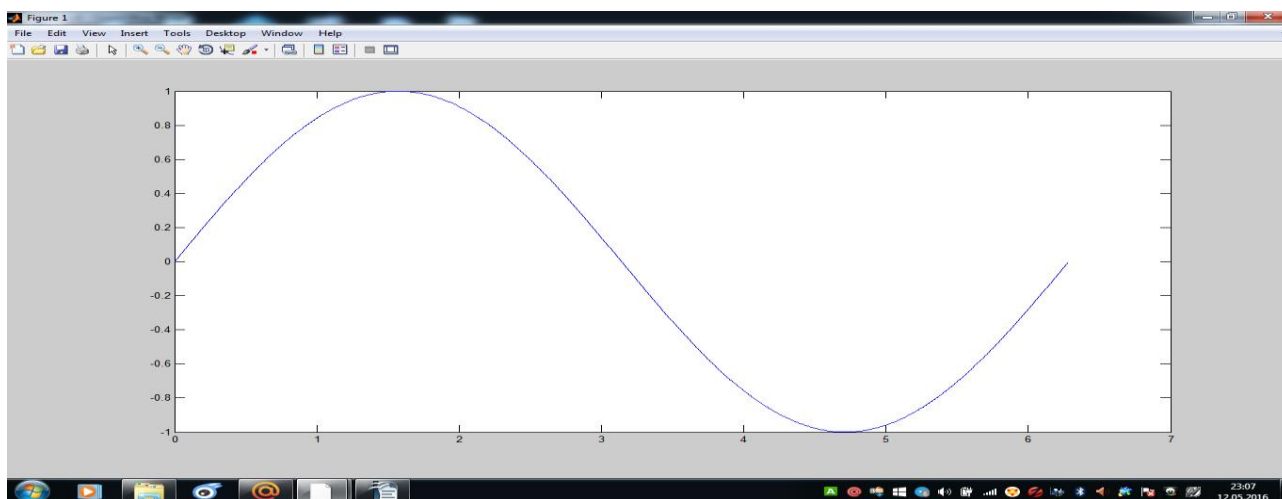


Рис.6.4. Отображение двумерной матрицы в виде двух графиков.

Два вектора в одних осях можно отобразить только в том случае, если их размерности совпадают. Когда же выполняется работа с векторами разных размерностей, то они либо должны быть приведены друг к другу по числу элементов, либо отображены на разных графиках. Отобразить графики в разных координатных осях можно несколькими способами. В самом простом случае можно создать два графических окна и в них отобразить нужные графики. Это делается следующим образом:

```
x1 = 0:0.01:2*pi;
y1 = sin(x1);
x2 = 0:0.01:pi;
y2 = cos(x2);
plot(x1, y1);      рисование первого графика figure;
создание 2-го графического окна plot(x2, y2);
рисование 2-го графика во 2-м окне
```



Функция `figure`, используемая в данной программе, создает новое графическое окно и делает его активным. Функция `plot()`, вызываемая сразу после функции `figure`, отобразит график в текущем активном графическом окне. В результате на экране будут показаны два окна с двумя графиками.

Неудобство работы приведенного фрагмента программы заключается в том, что повторный вызов функции `figure` отобразит на экране еще одно новое окно и если программа будет выполнена дважды, то на экране окажется три графических окна, но только в двух из них будут актуальные данные. В этом случае было бы лучше построить программу так, чтобы на экране всегда отображалось два окна с нужными графиками. Этого можно достичь, если при вызове функции `figure` в качестве аргумента указывать номер графического окна, которое необходимо создать или сделать активным, если оно уже создано. Таким образом, вышеприведенную программу можно записать так.

```
x1 = 0:0.01:2*pi;
y1 = sin(x1);
x2 = 0:0.01:pi;
y2 = cos(x2);
figure(1); %создание окна с номером 1
plot(x1, y1); % рисование первого графика
figure(2); % создание графического окна с номером 2
plot(x2, y2); % рисование 2-го графика во 2-м окне
```

При выполнении данной программы на экране всегда будут отображены только два графических окна с номерами 1 и 2, и в них показаны графики функций синуса и косинуса соответственно.

В некоторых случаях большего удобства представления информации можно достичь, отображая два графика в одном графическом окне. Это достигается путем использования функции `subplot()`, имеющая следующий синтаксис:

```
subplot(<число строк>, <число столбцов>, <номер координатной оси>)
```

Рассмотрим пример отображения двух графиков друг под другом вышеприведенных функций синуса и косинуса.

```
x1 = 0:0.01:2*pi;
y1 = sin(x1);
x2 = 0:0.01:pi;
y2 = cos(x2);
figure(1);
subplot(2,1,1); % делим окно на 2 строки и один столбец
plot(x1,y1); % отображение первого графика
subplot(2,1,2); % строим 2-ю координатную ось
plot(x2,y2); % отображаем 2-й график в новых осях
```

Результат работы программы показан на рис. 3.5.

Аналогичным образом можно выводить два и более графиков в столбец, в виде таблицы и т.п. Кроме того, можно указывать точные координаты расположения графика в графическом окне. Для этого используется параметр `position` в функции `subplot()`:

```
subplot('position', [left bottom width height]);
```

где `left` – смещение от левой стороны окна; `bottom` – смещение от нижней стороны окна; `width`, `height` – ширина и высота графика в окне. Все эти переменные изменяются в пределах от 0 до 1.

Ниже представлен фрагмент программы отображения графика функции синуса в центре графического окна. Результат работы показан на рис. 3.5.

```
x1 = 0:0.01:2*pi;  
y1 = sin(x1);  
subplot('position', [0.33 0.33 0.33 0.33]);  
plot(x1,y1);
```

В данном примере функция subplot() смещает график на треть от левой и нижней границ окна и рисует график с шириной и высотой в треть графического окна. В результате, получается эффект рисования функции синуса по центру основного окна.

Таким образом, используя параметр position можно произвольно размещать графические элементы в плоскости окна.

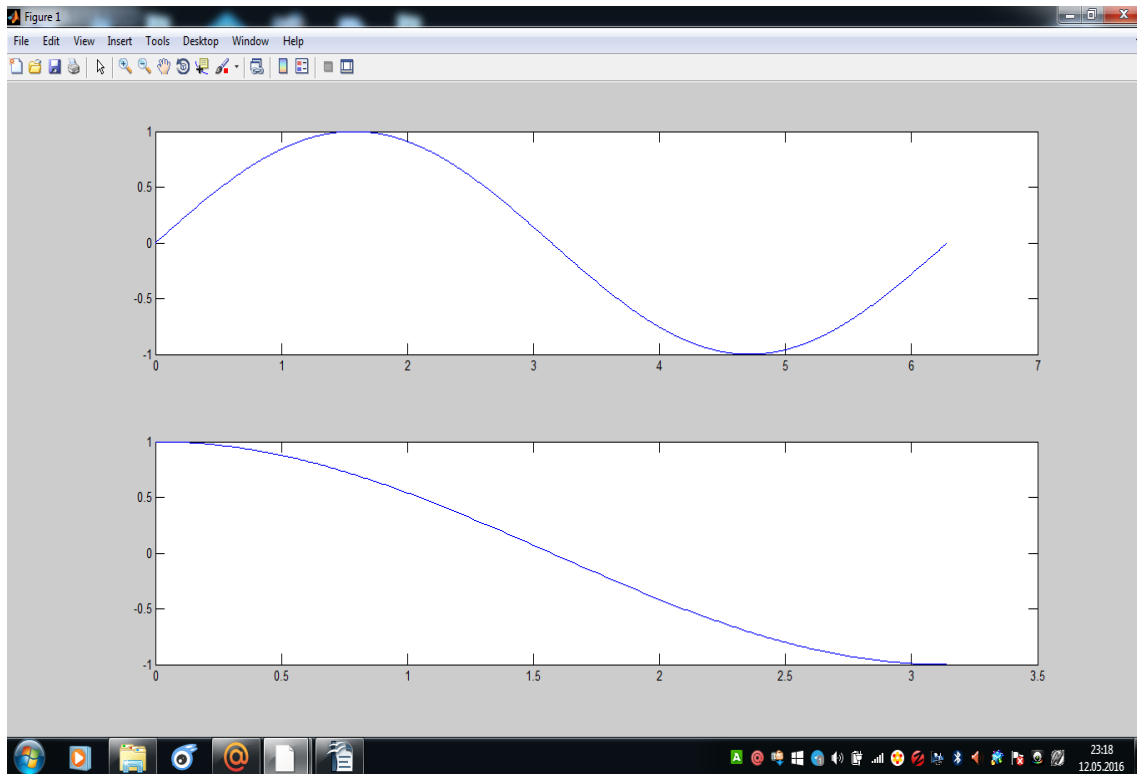


Рис. 3.5. Пример работы функции subplot с параметром position.

Контрольные вопросы:

1. Какими командами создаются двумерные графики?
2. Какими командами создаются трехмерные графики?

Тема 7-8	Подпрограммы программного пакета MATLAB. Элементы подпрограммы SIMULINK.	
Технология проведения лекционного занятия		
Время 2 часа	Количество студентов 40-60 человек	
Вид занятия	Введение. Визуальная лекция	
План лекционного занятия	1. Подпрограммы программного пакета MATLAB. 2. Ознакомление с подпрограммным пакетом SIMULINK. 3. Блоки математических операций в SIMULINK. 4. Решение системы нелинейных уравнений.	
Цель учебного занятия: Формирование у студентов глубокого представления о целях и задачах MATLAB и Элементы подпрограммы SIMULINK и путей её дальнейшего развития		
Педагогические задачи:	Результаты обучения:	
- Ознакомление с основными задачами MATLAB и Элементы подпрограммы SIMULINK - Ознакомление с историей развития MATLAB и Элементы подпрограммы SIMULINK и выдающимися её представителями - Внедрение в сознание студентов основных MATLAB и Элементы подпрограммы SIMULINK	Студент: 1. Имеет представление о целях и задачах MATLAB и Элементы подпрограммы SIMULINK 2. Излагает историю развития MATLAB и Элементы подпрограммы SIMULINK и выдающихся её представителей 3. Характеризует связь MATLAB и Элементы подпрограммы SIMULINK с другими предметами и её место в развитии науки 4. Характеризует основные законы химии и приводит примеры по ним	
Методика и техника обучения	- Информативная лекция, визуальной опрос, презентации	
Средства обучения	- Текст лекции, проектор, плакаты, доска, мел	
Форма обучения	- поток, группа, работа в парах	
Условия обучения	- оснащенная аудитория	

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.1. Приводит список используемой литературы по MATLAB и Элементы подпрограммы SIMULINK. Знакомит студентов с рейтинговой оценкой знаний студентов. Объявляет тему и план занятия, знакомит с достижениями MATLAB и Элементы подпрограммы SIMULINK	1.1. Слушает и записывает
2-этап Основной	2.1. Обращает внимание студентов на вопросы плана занятия и обращает внимание на основные понятия. Проводит опрос 2.2. Продолжает излагать лекционный материал. На примерах приводит и объясняет MATLAB и Элементы подпрограммы SIMULINK. 2.3. Обеспечивает студентом внимание и запись основных понятий и MATLAB и Элементы подпрограммы SIMULINK.	2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают
3-этап Итоговый	3.1. Делает заключительные выводы лекционной темы и заостряет внимание студентов на основные вопросы 3.2. Отмечает активных студентов 3.3. Даст задание на самостоятельную подготовку: Дать определение каждого закона, привести примеры	3.1. Слушают и усваивают 3.2. Записывают задание

Лекции 7-8.

Подпрограммы программного пакета Matlab. Элементы подпрограммы Simulink.

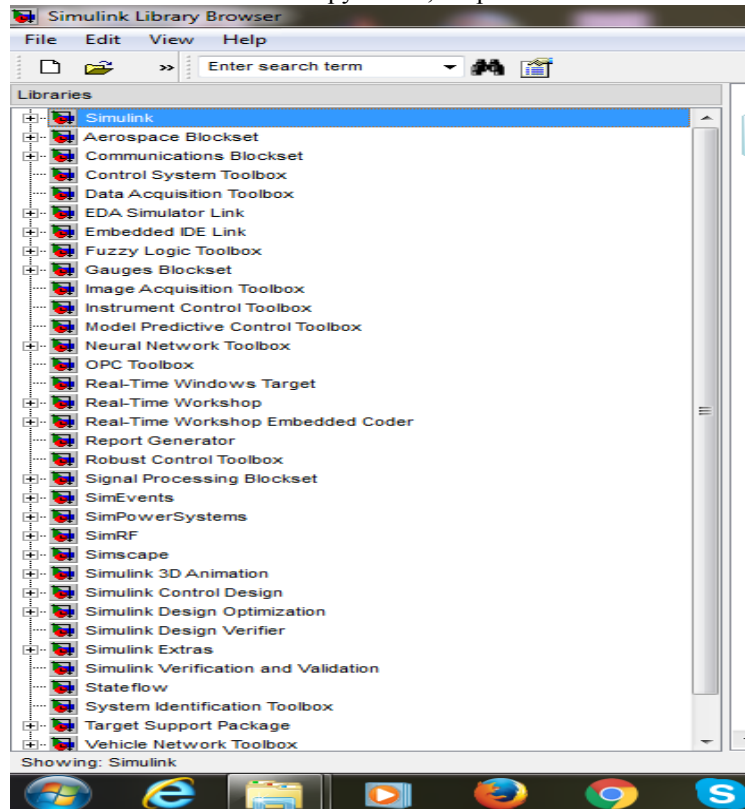
План:

1. Подпрограммы программного пакета MATLAB.
2. Ознакомление с подпрограммным пакетом SIMULINK.
3. Блоки математических операций в SIMULINK.
4. Решение системы нелинейных уравнений.

1. Подпрограммы программного пакета MATLAB.

Основные компоненты библиотеки:

Часть входящих в состав пакетов имеет инструменты, встраиваемые в Simulink:



Ниже указаны описания этих компонентов:

- MATLAB: Examples: Mathematics: гиперссылки в правом окне в разделе Polynomials and Interpolation – приближение многомерных данных.
- MATLAB: Examples: Mathematics: гиперссылки в правом окне в разделе Data Analysis and Statistics – приближение полиномами в смысле наименьших квадратов, приближение экспоненциальными моделями, использование средства Basic Fitting из графического окна.
- MATLAB: Mathematics: Polynomials and Interpolation: Polynomials: Polynomial Curve Fitting – приближение полиномами по методу на и меньших квадратов.
- MATLAB: Mathematics: Polynomials and Interpolation: Interpolation - обзор функций MATLAB для интерполяции данных, интерполяция одномерных данных при помощи сплайнов и дискретного преобразования Фурье, интерполяция двумерных данных и сравнение различных способов (по ближайшему соседу, билинейная и бикубическая интерполяция), интерполяция разбросанных данных.
- MATLAB: Mathematics: Data Analysis and Statistics: Data Preprocessing - проверка отсутствующих данных и их удаление, удаление выбросов из данных.
- MATLAB: Mathematics: Data Analysis and Statistics: Regression and Curve Fitting - полиномиальная регрессия, множественная регрессия (без применения специальных функций MATLAB).
- MATLAB: Mathematics: Data Analysis and Statistics: Case Study: Curve Fitting - приближение полиномиальной и экспоненциальной моделями в смысле наименьших квадратов.
- MATLAB: Functions – Categorical List: Mathematics: Interpolation and Computational Geometry - интерполяция одномерных, двумерных и многомерных табличных данных, построение выпуклой оболочки множества точек на плоскости и в N-мерном пространстве, построение триангуляции Делоне и диаграммы Вороного на плоскости и в N-мерном пространстве, поиск ближайшей точки и симплекса, содержащего заданную точку, интерполяция разбросанных данных.
- Spline Toolbox - интерполяция и сглаживание сплайнами в кусочно-полиномиальной и B-форме, переход от одной формы к другой, выбор "хороших" (это не термин, подробнее написано в разделе, посвященном Spline Toolbox) узлов и полюсов интерполяции, приближение сплайнами по методу наименьших квадратов,

построение сплайнов типа тонких пластин, рациональных сплайнов, визуализация сплайнов, их интегрирование и дифференцирование, нахождение их корней и экстремумов, описание графических оболочек, входящих в состав Spline ToolBox. Подробнее про возможности Spline ToolBox написано в разделе Обзор возможностей и средств Spline Toolbox 3.2.

- Curve Fitting Toolbox - предварительная обработка и сглаживание данных, приближение данных линейными и нелинейными параметрическими и непараметрическими (интерполяционные и сглаживающие сплайны) моделями, анализ качества приближений, вычисление их характеристик, визуализация данных и полученных моделей. Описана работа как в приложении с графическим интерфейсом, так и использование функций Curve Fitting Toolbox.
- Statistic Toolbox-Также полезно посмотреть демонстрации (вкладка Demos в окне справочной системы):
- MATLAB: Mathematics: Optimal Fit of a Non-linear Function - объяснено решение задачи о нахождении параметров, линейно и нелинейно входящих в параметрическую модель, при помощи построения целевой функции ошибки приближения и ее минимизации с использованием функции fminsearch пакета MATLAB.
- MATLAB: Mathematics: Tesselation and Interpolation of Scattered Data - рассмотрено построение выпуклой оболочки в трехмерном случае, триангуляция Делоне, построение диаграммы Вороного и интерполяция трехмерных разбросанных данных, включая визуализацию.
- MATLAB: Mathematics: Splines in Two Dimensions - разбирается построение двумерных сплайновых кривых при помощи базовых функций MATLAB.
- Toolboxes: Curve Fitting - два фильма, демонстрирующих использование графической оболочки Curve Fitting Tool для импорта и сглаживания данных, а также подбор параметров в заданной пользователем модели.
- Toolboxes: Spline - пятнадцать демонстраций, включая простые примеры, объяснение кусочно-полиномиальной и В-формы сплайнов, интерполирование и сглаживание кубическими сплайнами, построение сплайновых кривых, аппроксимация двумерных данных тензорным произведением сплайнов, применение сплайнов для решения граничных задач для обыкновенных дифференциальных уравнений, эксперименты с приближениями различными сплайнами и эксперименты с узлами в-сплайнов.

Ознакомление с подпрограммным пакетом SIMULINK.

Программа Simulink является приложением к пакету MATLAB. При моделировании с использованием Simulink реализуется принцип визуального программирования, в соответствии с которым, пользователь на экране из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты. При этом, в отличие от классических способов моделирования, пользователю не нужно досконально изучать язык программирования и численные методы математики, а достаточно общих знаний, требующихся при работе на компьютере и, естественно, знаний той предметной области, в которой он работает.

Simulink является достаточно самостоятельным инструментом MATLAB и при работе с ним совсем не требуется знать сам MATLAB и остальные его приложения. С другой стороны, доступ к функциям MATLAB и другим его инструментам остается открытым и их можно использовать в Simulink. Часть входящих в состав пакетов имеет инструменты, встраиваемые в Simulink (например, LTI-Viewer приложения Control System Toolbox – пакета для разработки систем управления). Имеются также дополнительные библиотеки блоков для разных областей применения (например, Power System Blockset – моделирование электротехнических устройств, Digital Signal Processing Blockset – набор блоков для разработки цифровых устройств и так далее).

Одной из самых сложных проблем в реализации математического моделирования в среде MATLAB является подготовка модели моделируемой системы или устройства. Модель обычно представляется в форме графического, табличного или таблично-топологического описания. При этом необходимо предусмотреть организацию связей между компонентами и установку их параметров. После этого нужно запустить модель на исполнение, т.е. задать решение автоматически составленной системы уравнений состояния и вывод результатов решения, что зачастую представляет собой достаточно сложную задачу.

SIMULINK – важная составная часть системы MATLAB. Это расширение реализует визуально-ориентированное программирование задач автоматического составления графической модели системы или устройства, составления и решения уравнений состояния и наглядного представления результатов моделирования. Пакет SIMULINK позволяет выполнять симуляцию работы моделируемых систем и устройств, т. е. осуществлять имитационное моделирование.

Пакет SIMULINK является ядром интерактивного программного комплекса, предназначенного для математического моделирования линейных и нелинейных динамических систем, и устройств, представленных своей функциональной блок-схемой, именуемой S-моделью или просто моделью. При этом возможны различные варианты моделирования: во временной области, в частотной области, с событийным управлением, на основе спектральных преобразований, с использованием метода Монте-Карло (реакция на воздействия случайного характера) и т. п.

Для построения функциональной блок-схемы моделируемых устройств SIMULINK имеет обширную библиотеку блочных компонентов и удобный редактор блок-схем. Он основан на графическом интерфейсе пользователя и по существу является типичным средством визуально-ориентированного программирования. Используя палитры компонентов (наборы), пользователь с помощью мыши переносит нужные блоки с палитр

на рабочий стол пакета SIMULINK и соединяет линиями входы и выходы блоков. Таким образом, создается блок-схема системы или устройства, то есть модель.

SIMULINK автоматизирует следующий, наиболее трудоёмкий этап моделирования: он составляет и решает сложные системы алгебраических и дифференциальных уравнений, описывающих заданную функциональную схему (модель), обеспечивая удобный и наглядный визуальный контроль за поведением созданного пользователем виртуального устройства – достаточно уточнить (если нужно) вид анализа и запустить SIMULINK в режиме симуляции созданной модели системы или устройства.

Ценность пакета SIMULINK заключается и в обширной, открытой для изучения и модификации библиотеке компонентов (блоков). Она включает источники сигналов с практически любыми временными зависимостями, масштабированные, линейные и нелинейные преобразователи с разнообразными формами передаточных характеристик, квантующее устройство, интегрирующие и дифференцирующие блоки и т. д. Кроме этого пакет SIMULINK включает в себя отдельные специализированные библиотеки, наиболее полезными из которых являются пакет для моделирования систем передачи дискретных сообщений (Communications block set) и пакет для моделирования систем цифровой обработки сигналов (DSP blockset).

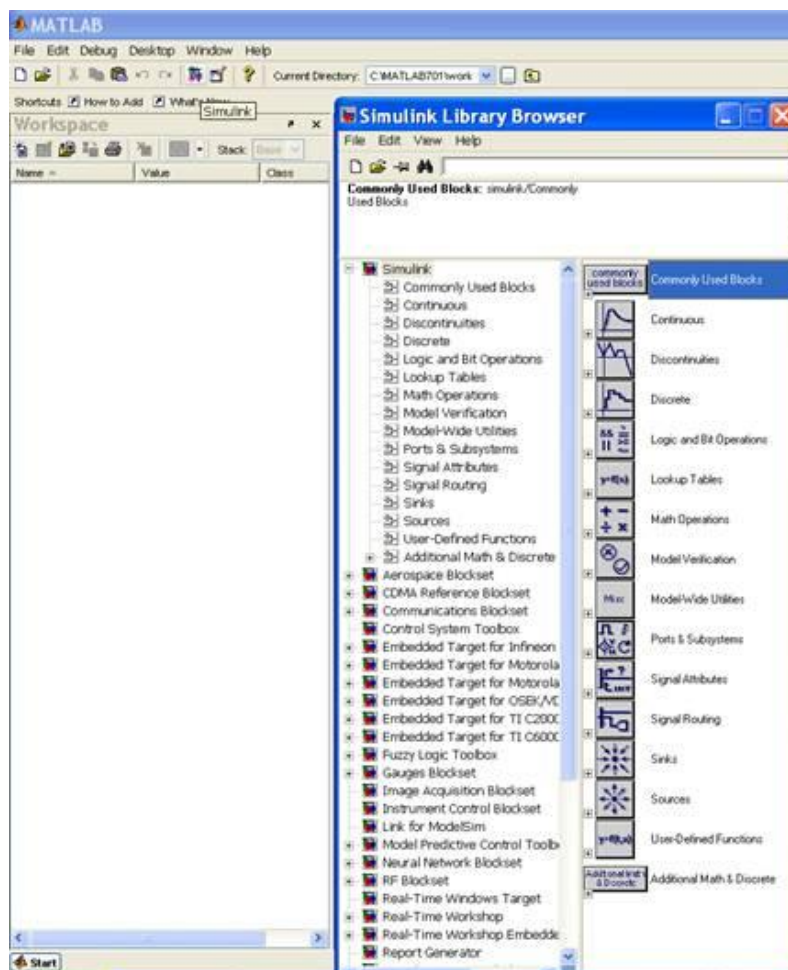
Программные средства моделирования динамических систем известны давно, к ним относятся, например, программы Tutsim и LabVIEW ForIndustrial Automation. Однако для эффективного применения таких средств необходимы высокоскоростные решающие устройства. Интеграция системы MATLAB с пакетом SIMULINK открывает новые возможности использования самых современных математических методов для решения задач динамического и ситуационного моделирования сложных систем и устройств. Средства графической анимации SIMULINK позволяют строить виртуальные физические лаборатории с наглядным представлением результатов моделирования. Возможности SIMULINK охватывают задачи математического моделирования сложных динамических систем в физике, электро- и радиотехнике, биологии и других областях науки и техники. Этим объясняется популярность данного пакета как в вузах, так и в научных лабораториях.

Важным достоинством пакета SIMULINK является возможность задания в блоках произвольных математических выражений, что позволяет решать типовые задачи, пользуясь примерами пакета SIMULINK или же просто задавая новые выражения, описывающие работу моделируемых пользователем систем и устройств. Важным свойством пакета является возможность задания системных функций (S-функций) с включением их в состав библиотек SIMULINK. Необходимо также отметить возможность моделирования устройств и систем в реальном масштабе времени.

Как программное средство SIMULINK – типичный представитель визуально-ориентированных языков программирования. На всех этапах работы, особенно при подготовке моделей систем, пользователь практически не имеет дела с обычным программированием. Программа в кодах автоматически генерируется в процессе ввода выбранных блоков компонентов, их соединений и задания параметров компонентов.

Важное преимущество SIMULINK – это интеграция не только с системой MATLAB, но и с рядом других пакетов расширения, что обеспечивает, по существу, неограниченные возможности применения SIMULINK для решения практически любых задач имитационного и событийного моделирования. При работе с Simulink пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков. При моделировании пользователь может выбирать метод решения дифференциальных уравнений, а также способ изменения модельного времени (с фиксированным или переменным шагом). В ходе моделирования имеется возможность следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения, входящие в состав библиотеки Simulink. Результаты моделирования могут быть представлены в виде графиков или таблиц. Для запуска программы необходимо предварительно запустить пакет MATLAB. Основное окно пакета MATLAB показано на рисунке. Там же показана подсказка, появляющаяся в окне при наведении указателя мыши на ярлык Simulink в панели инструментов. Нажатие на ярлык Simulink в панели инструментов приводит к открытию окна обозревателя разделов библиотеки Simulink. Чтобы создать свою программу необходимо создать новый файл модели, расположить блоки в окне модели. Для этого необходимо открыть соответствующий раздел библиотеки. Далее, указав курсором на требуемый блок и нажав на левую клавишу “мыши” - “перетащить” блок в созданное окно. Далее, если это требуется, нужно изменить параметры блока, установленные программой “по умолчанию”. Для этого необходимо дважды щелкнуть левой клавишей “мыши”, указав курсором на изображение блока. Откроется окно редактирования параметров данного блока. При задании численных параметров следует иметь в виду, что в качестве десятичного разделителя должна использоваться точка, а не запятая. После внесения изменений нужно закрыть окно кнопкой ОК.

После установки на схеме всех блоков из требуемых библиотек нужно выполнить соединение элементов схемы. Для соединения блоков необходимо указать курсором на “выход” блока, а затем, нажав и, не отпуская левую клавишу “мыши”, провести линию к входу другого блока. После чего отпустить клавишу. В случае правильного соединения изображение стрелки на входе блока изменит цвет. Для создания точки разветвления в соединительной линии нужно подвести курсор к предполагаемому узлу и, нажав правую клавишу “мыши”, протянуть линию. Для удаления линии требуется выбрать линию (также как это выполняется для блока), а затем нажать клавишу Delete на клавиатуре.



Окно обозревателя библиотеки блоков содержит следующие элементы:

1. Заголовок, с названием окна – Simulink Library Browser.
2. Меню с командами File, Edit, View, Help.
3. Панель инструментов, с ярлыками наиболее часто используемых команд.
4. Окно комментария для вывода поясняющего сообщения о выбранном блоке.
5. Список разделов библиотеки, реализованный в виде дерева.
6. Окно содержимого раздела библиотеки (список вложенных разделов библиотеки или блоков)
7. Строка состояния, содержащая подсказку по выполняемому действию.

Библиотека Simulink содержит следующие основные разделы:

1. Continuous – библиотека непрерывных элементов (интегратор, дифференциатор, линейная система ОДУ и т.д.);
2. Discrete – библиотека дискретных элементов (интегратор с дискретным временем, дискретный фильтр и т.д.);
3. Functions&Tables – функции и таблицы (вызов функций Matlab, S-функции, блок, отображающий входной вектор в выходной (с использованием линейной интерполяции значений, определенных в параметрах блока) и т.д.);
4. Math – математические функции (абсолютное значение, комбинаторная логика, выделение вещественной и мнимой составляющей комплексного числа и т.д.);
5. Nonlinear – нелинейные элементы (релейное звено, переключатель и т.д.);
6. Signals&Systems – сигналы и системы (составной блок, входной сигнал, выходной сигнал, мультиплексер, демультимплексер и т.д.);
7. Sinks – средства отображения (временная диаграмма, вывод результатов в файл, остановка выполнения модели и т.д.);
8. Sources – источники сигналов (генератор импульсных/синусоидальных сигналов, генератор случайных чисел, генератор пилообразных сигналов, часы и т.д.).
9. Subsystems – блокиподсистем.

Список разделов библиотеки Simulink представлен в виде дерева, и правила работы с ним являются общими для списков такого вида:

Рассмотрим создание нескольких примеров

Графопостроитель XY Graph

Назначение:

Строит график одного сигнала в функции другого (график вида $Y(X)$).

Параметры:

x-min – Минимальное значение сигнала по оси X.

x-max – Максимальное значение сигнала по оси X

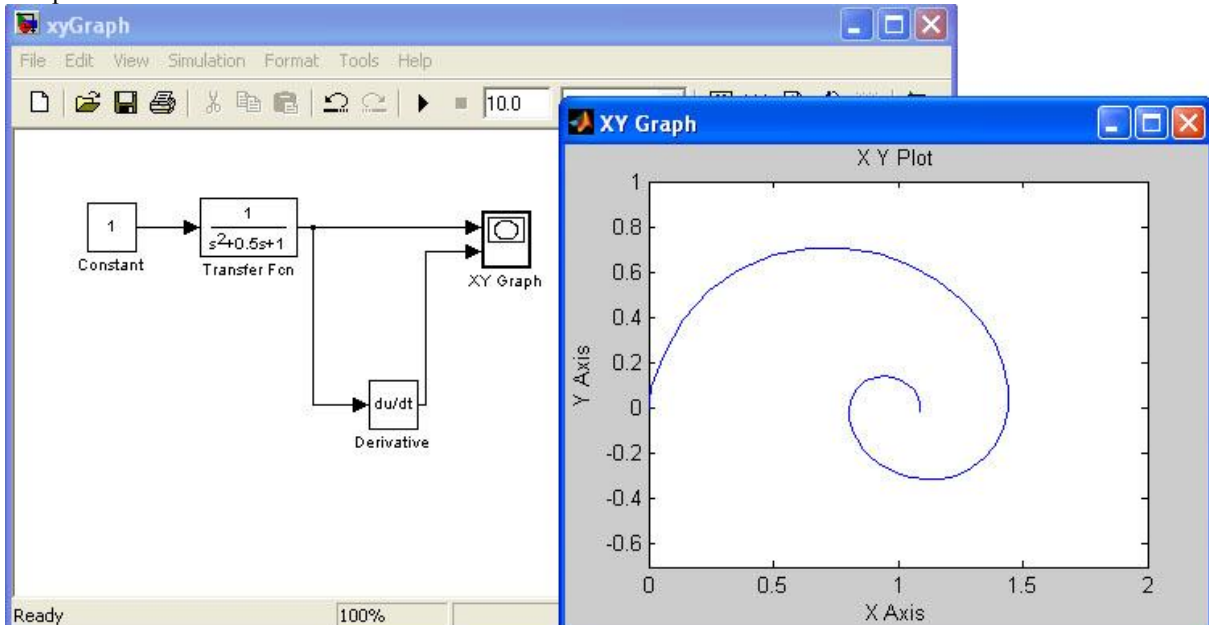
y-min – Минимальное значение сигнала по оси Y.

y-max – Максимальное значение сигнала по оси Y

Sampletime – шаг модельного времени.

Блок имеет два входа. Верхний вход предназначен для подачи сигнала, который является аргументом (X), нижний – для подачи значений функции (Y).

На рисунке, в качестве примера использования графопостроителя, показано построение фазовой траектории колебательного звена.



Constant – из Commonly Used Blocks

Чтобы установить необходимые параметры блока необходимо дважды щелкнуть на блоке, в параметрах задать значение константы =1.

Transfer Function – Continuous. Представляет собой отношение степенных полиномов. В параметрах для нашего случая необходимо задать числитель: Numerator – [1], и знаменатель: Denominator – [1 0.5 1].

Derivative - Continuous. Блок вычисляет значение производной.

XYGraph – Sinks. Вывод результата на график в виде зависимости X от Y.

Блок алгебраического контура Algebraic Constraint

Назначение: Выполняет поиск корней алгебраических уравнений.

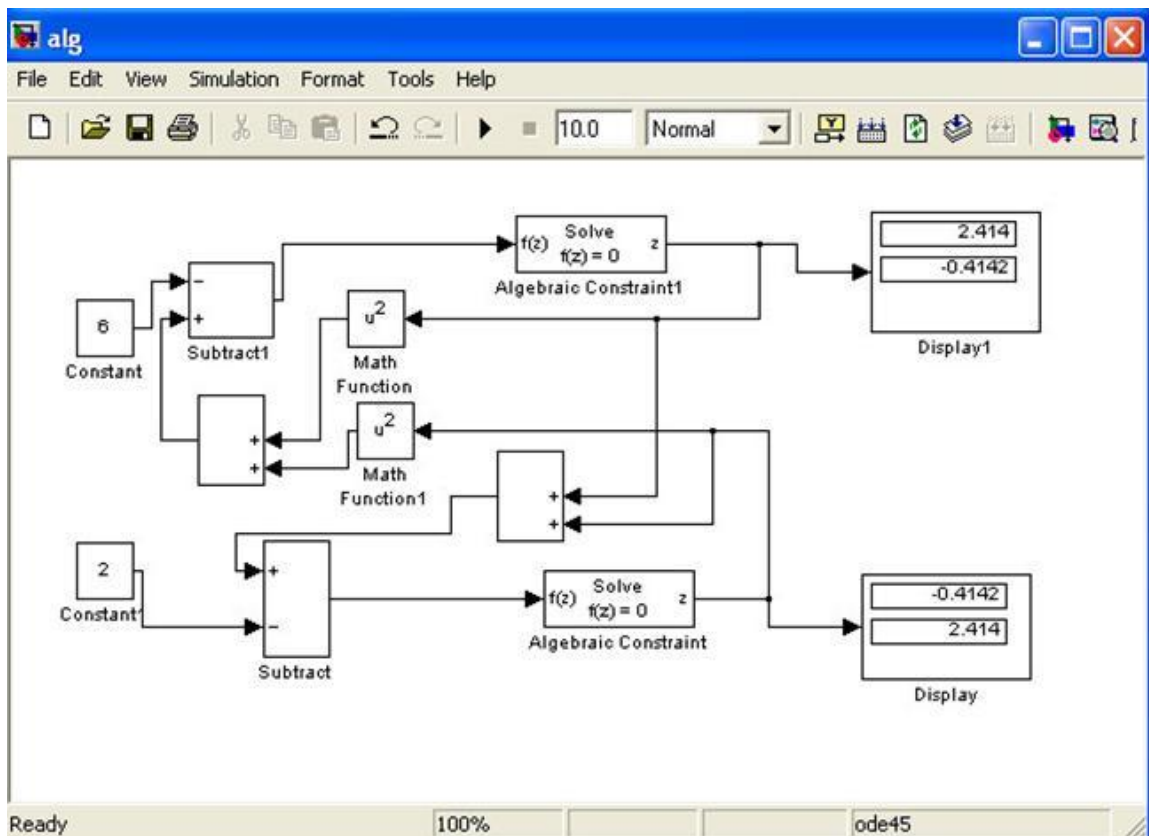
Параметры: Initialguess – Начальное значение выходного сигнала.

Блок находит такое значение выходного сигнала, при котором значение входного сигнала становится равным нулю. При этом входной сигнал должен быть прямо или опосредованно связан с входным сигналом.

На рисунке показан пример решения системы нелинейных уравнений вида:

$$\begin{cases} x^2 + y^2 = 6 \\ x + y = 2 \end{cases}$$

Поскольку данная система уравнений имеет два решения, то начальные значения блоков Algebraic Constraint заданы в виде векторов. Для первого (верхнего) блока начальное значение задано вектором [1 -1], а для второго (нижнего) блока – вектором [-1 1].



Рассмотрим блоки, которые не использовались в предыдущем примере.

Subtract – Math Operations. В параметрах в окошке Listof Signs надо поставить те операции, которые необходимы(- +).

MathFunction – Math Operations. В параметрах в окошке Function надо выбрать square(чтобы был квадрат значения).

AlgebraicConstant – Math Operations. В параметрах в окошке Initialguess задать начальное значение вектором [1 -1] или [-1 1].

Display – Sinks. Выводит результат.

Непрерывные линейные системы (Continuous)

TransferFunction– передаточная функция, в параметрах задаются числитель (**Numerator**) и знаменатель (**Denominator**) в виде полиномов.

StateSpace – модель в пространстве состояний, в параметрах задаётся четвёрка матриц, определяющих модель, и начальные условия для вектора состояния (**Initialconditions**).

Zero-Pole– модель в форме «нули-полюса», в параметрах задаются массивы нулей (**Zeros**), полюсов (**Poles**), а также коэффициент усиления (**Gain**).

Integrator – интегратор с возможностью установки начальных условий (**InitialCondition**), а также пределов насыщения (**Lower saturation limit** и **Upper saturation limit**). Когда сигнал выхода выходит за границы, определяемые этими пределами, интегрирование прекращается.

TransportDelay – блок фиксированной задержки сигнала.

Нелинейные элементы (Discontinuities)

Relay– блок реализует релейную характеристику с гистерезисом.

Saturation – блок выполняет ограничение величины сигнала.

Backlash– блок моделирует нелинейность типа «люфт».

RateLimiter– блок обеспечивает ограничение скорости изменения сигнала (первой производной).

DeadZone– блок реализует нелинейную зависимость типа «зона нечувствительности (мертвая зона)».

Основные источники сигналов (Sources)

Constant– сигнал постоянной величины.

Step– ступенчатый сигнал, меняется время скачка (**StepTime**), начальное (**InitialValue**) и конечное значения (**FinalValue**).

Ramp– линейно возрастающий сигнал с заданным наклоном (**Slope**). Можно задать также время начала изменения сигнала (**StartTime**) и начальное значение (**InitialValue**).

PulseGenerator– генератор прямоугольных импульсов, задаются амплитуда (**Amplitude**), период (**Period**), ширина (**PulseWidth** в процентах от периода), фаза (**PhaseDelay**).

Repeating Sequence – последовательность импульсов, их форма задаётся в виде пар чисел (время; величина сигнала).

Sine Wave– синусоидальный сигнал, задаётся амплитуда (**Amplitude**), частота (**Frequency**), фаза (**Phase**) и среднее значение (**Bias**).

Signal Builder– построитель сигналов, позволяющий задавать форму сигнала, перетаскивая мышью опорные точки.

Random Number – случайные числа с нормальным (гауссовым)распределением. Можно задать среднее значение (**Mean Value**), дисперсию (**Variance**), период изменения сигнала (**SampleTime**).

Uniform Random Number– случайные числа с равномерным распределением в заданном интервале от **Minimum** до **Maximum**.

BandLimitedWhiteNoise– случайный сигнал, ограниченный по полосе белый шум (имеющий равномерный спектр до некоторой частоты). Блок используется как источник белого шума для моделей непрерывных систем. Задаётся интенсивность (**NoisePower**) и интервал дискретизации (**SampleTime**), в течение которого удерживается постоянное значение сигнала. Чем меньше интервал, тем точнее моделирование, однако больше вычислительные затраты.

Основные устройства вывода (Sinks)

Display– цифровой дисплей, показывает изменение входного сигнала в цифровом виде.

Scope – осциллограф, показывает изменение сигнала в виде графика, позволяет передавать данные в рабочую область MATLAB для последующей обработки и оформления.

Другие часто используемые блоки

MathOperations

Gain– усилитель, задаётся коэффициент усиления (**Gain**).

Sum– сумматор, используется для сложения и вычитания входов. Параметр **Listofsigns**-задаёт количество входов, их знаки («+» для сложения и «-» для вычитания). Промежутки между входами (обозначаются знаком).

Trigonometric Function– тригонометрическая функция.

SignalRouting

Manual Switch– ручной переключатель, позволяет двойным щелчком переключать выход на один из двух входных сигналов.

Mux– мультиплексор, объединяет несколько сигналов в один «жгут» (векторный сигнал), в параметрах задаётся число входов (**Number of Inputs**).

Demux– демультиплексор, позволяет «разбить» векторный сигнал на несколько скалярных выражений.

Контрольные вопросы:

1. Опишите подпрограммы пакета MATLAB.
2. Опишите блоки SIMULINK.
3. Часто используемые блоки SIMULINK.
4. Алгебраические блоки в пакете SIMULINK.

Тема 9	Моделирование системы управления в подпрограмме Simulink	
Технология проведения лекционного занятия		
Время 2 часа	Количество студентов 40-90 человек	
Вид занятия	Лекция	
План лекционного занятия	1. Моделирование в Simulink; 2. Анализ Simulink моделей; 3. Маскирование подсистем в Simulink; 4. Управление Simulink - моделью из MATLAB.	
Цель учебного занятия: Формирование у студентов представления о моделировании систем управления в подпрограмме Simulink		
Педагогические задачи:		Результаты обучения:
- Ознакомление с Моделированием в Simulink - Ознакомление с Анализом Simulink моделей - Внедрение в сознание студентов управления Simulink - моделью из MATLAB		Студент: 1. Имеет представление о Моделировании в Simulink 2. Усваивать анализ Simulink моделей 3. Даст понятие и определение о Маскировании подсистем в Simulink 4. Может управлять Simulink - моделью из MATLAB
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа,
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.2. Объявляет тему и план занятия, знакомит с достижениями MATLAB, в частности подпрограммы СИМУЛИНК	1.1. Слушает и записывает
2-этап Основной	2.1. Обращает внимание студентов на вопросы плана занятия и обращает внимание на основные понятия. Проводит опрос 2.2. Продолжает излагать лекционный материал. На примерах приводит и объясняет особенности среды Симулинк. 2.3. Обеспечивает студентом внимание и запись основных элементов и инструментов среды Симулинк.	2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают
3-этап Итоговый	3.1. Делает заключительные выводы лекционной темы и заостряет внимание студентов на основные вопросы 3.2. Отмечает активных студентов 3.3. Даст задание на самостоятельную подготовку: Дать определение каждого инструмента, привести примеры	3.1. Слушают и усваивают 3.2. Записывают задание

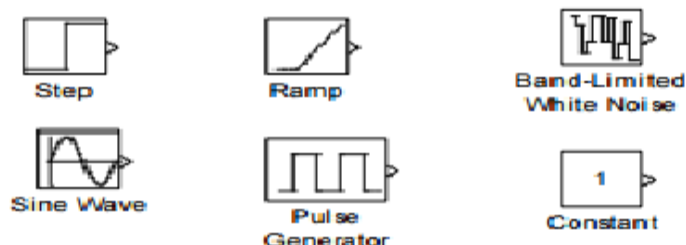
Лекция 9.
Моделирование системы управления в подпрограмме Simulink.

План:

1. Моделирование в Simulink;
2. Анализ Simulink моделей;
3. Маскирование подсистем в Simulink;
4. Управление Simulink - моделью из MATLAB.

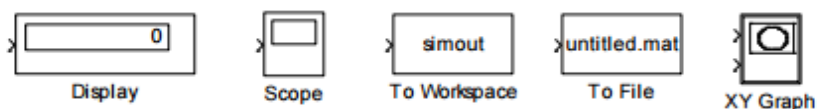
1. Моделирования в SIMULINK

Для моделирования входных и управляющих воздействий (синусоидальных, ступенчатых, импульсных и других) используются генераторы сигналов. Все они сосредоточены в библиотеке (группе блоков) Sources (источники).



Название блока	Тип сигнала
Step	Ступенька
Sine wave	Гармонический сигнал $y = A_0 \sin(\omega t + \varphi) + b$
Ramp	Линейно изменяющийся сигнал
Pulse generator	Последовательность прямоугольных импульсов
Band-limited white noise	Белый шум, ограниченный по полосе
Constant From File	Константа Загрузка из файла
From Workspace	Загрузка из рабочего пространства MATLAB

Для представления результатов моделирования используются блоки, расположенные в группе Sinks. Наиболее важными из них являются блоки Scope, Graph (осциллографы), Display, To File и To Workspace.

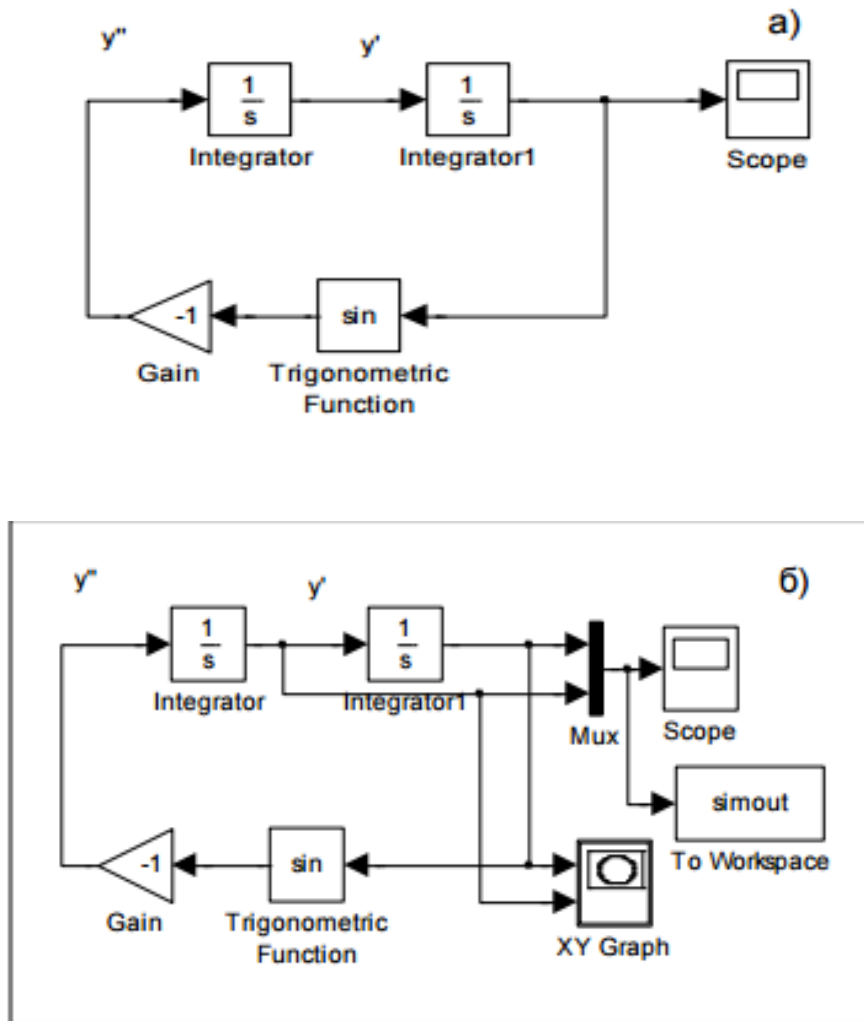


Блок Scope предназначен для вывода графиков сигналов. Блок XY Graph необходим для изображения параметрических графиков, например, фазовых траекторий динамических систем. Блок Display выводит в окне числовое значение сигнала. Блок To File предназначен для сохранения результатов моделирования в файле. Пользователь может выбрать имя файла и имя переменной. После выполнения моделирования содержимое файла можно загрузить в рабочее пространство командой load. Результаты моделирования можно передать в рабочее пространство MATLAB при помощи блока To Workspace. Пользователь должен ввести имя переменной (по умолчанию simout). Имеется три варианта представления результатов – массив (array), структура (structure), структура со временем (structure with time). В первом случае в рабочем пространстве MATLAB появляется массив значений переменной simout и соответствующий массив времени timeout. Во втором и третьем случаях результат возвращается в виде структуры и помещается в рабочее пространство MATLAB. Извлечение переменных из полей структуры производится с помощью команд типа simout.time и simout.signal.

1.1. Пример моделирования в SIMULINK

Основной способ исследования систем в SIMULINK – составление структурной схемы на усилителях, сумматорах, интеграторах и других блоках, последующее моделирование и наблюдение результатов на имитаторах осциллографов и других регистрирующих приборов. Рассмотрим процедуру моделирования на примере дифференциального уравнения $y' = \sin y$, $y'(0) = 0$, $y(0) = 1/2$. Оно описывает свободное колебание математического маятника, который отклонили от положения равновесия на угол около 300 и отпустили без начальной скорости. Соответствующая схема моделирования приведена на рис.3.7, а. Она содержит два

интегратора, тригонометрический блок для получения функции $\sin u$ и инвертирующий усилитель. Начальные условия на интеграторах устанавливаются при задании внутренних параметров блока. Внесем в эту схему дополнительные блоки, позволяющие: Наблюдать фазовые траектории, т.е. графики зависимости y от y' . Получать два графика $y(t)$ и $y'(t)$ на одном осциллографе. Передавать результаты моделирования в MATLAB. Измененная схема показана на рис.3.7, б. Для наблюдения фазовых траекторий введен блок XY Graph. Для совмещения двух графиков на вход осциллографа Scope подан «сдвоенный» сигнал. Он получен с помощью блока «мультиплексор» (Mux) из группы блоков «маршрутизация сигналов» (Signal Routing). Для передачи данных в MATLAB использован блок To Workspace с форматом вывода structure with time. Теперь после прогона модели графики $y(t)$ и $y'(t)$ можно будет построить из командного окна MATLAB, набрав: `>> plot(simout.time,simout.signals.values)` Такой способ построения полезен, если необходимо изменять вид графика, поскольку возможности осциллографа Scope значительно ниже, чем у команды plot.



В пункте меню «Simulation/Simulation parameters» можно задать ряд параметров моделирования. Часть из них расположены на панелях Solver (решатель) и Workspace I/O (интерфейс с рабочим пространством). На панели Solver в группе элементов управления Simulation Time (время моделирования) можно определить время начала (Start time) и конца моделирования (Stop time). В группе опций решателя (Solver) можно выбрать используемый алгоритм решения дифференциальных уравнений с фиксированным шагом (Fixed-step) или с изменяемым шагом (Variable step). Можно также задать минимальный и максимальный размер шага, а также относительную и абсолютную точность интегрирования. На панели Workspace I/O можно определить имена переменных, в которых хранятся результаты моделирования и которые используются как источники сигналов. Чтобы проиллюстрировать их применение, усложним нашу модель, введя в нее управляющий сигнал u и e ; $y \sin u$; $y(0) \square 1$, $y(0) \square 1$. Теперь это уравнение описывает вынужденное движение маятника под действием управляющего воздействия $u(t)$. Управляющий сигнал можно получить либо непосредственно в SIMULINK, либо сформировать его в рабочем пространстве MATLAB и оттуда передать в SIMULINK.

Зададим время моделирования $Stop\ time=20$. На панели Workspace I/O в графе Input запишем $[t, u]$, в графе Initial State запишем $xInitial$, в графе Time – tout, в графе Output – youut, и установим соответствующие чек-боксы. В рабочем пространстве MATLAB подготовим управляющий сигнал t и e и установим начальные условия $y(0) \square 1$, $y(0) \square 1$: `>> xInitial=[1; 1]; % столбец начальных условий >> t=linspace(0,20)'; % время`

моделирования `>> u=exp(-t); % управление u` Запустим процесс моделирования в SIMULINK. По окончании моделирования в рабочем пространстве появятся переменные tout и yout. Набрав `plot(tout,yout,t,u,r')`, можно посмотреть три графика – $y(t)$, $y'(t)$, $u(t)$

В MATLAB имеется возможность моделировать линейные системы с дискретным временем. Для описания таких систем используются разностные уравнения различных порядков, модели в пространстве состояний и дискретные передаточные функции. Линейное разностное уравнение n -го порядка имеет вид
$$y(k) + a_1 y(k-1) + \dots + a_n y(k-n) = b_0 u(k) + b_1 u(k-1) + \dots + b_n u(k-n)$$
 Здесь k – номер такта дискретного времени, $u(k)$ – входной сигнал, $y(k)$ – выходной сигнал, a_i , b_i – постоянные коэффициенты. Такими уравнениями, в частности, описываются цифровые фильтры. В ядре MATLAB имеется команда filter, позволяющая рассчитать выходной сигнал фильтра по известному входному сигналу, заданному массивом своих значений. Ее входными аргументами являются векторы a и b коэффициентов разностного уравнения, а также массив значений сигнала u : `y=filter(b,a,u)`. С помощью четвертого входного аргумента можно задавать начальные условия фильтра. Гораздо больше возможностей по синтезу и моделированию цифровых фильтров предоставляет тулбокс SIGNAL, подробнее об этом говорится в конце данного раздела. Описание дискретной системы в пространстве состояний, как и в случае непрерывных систем, задается четверкой матриц:

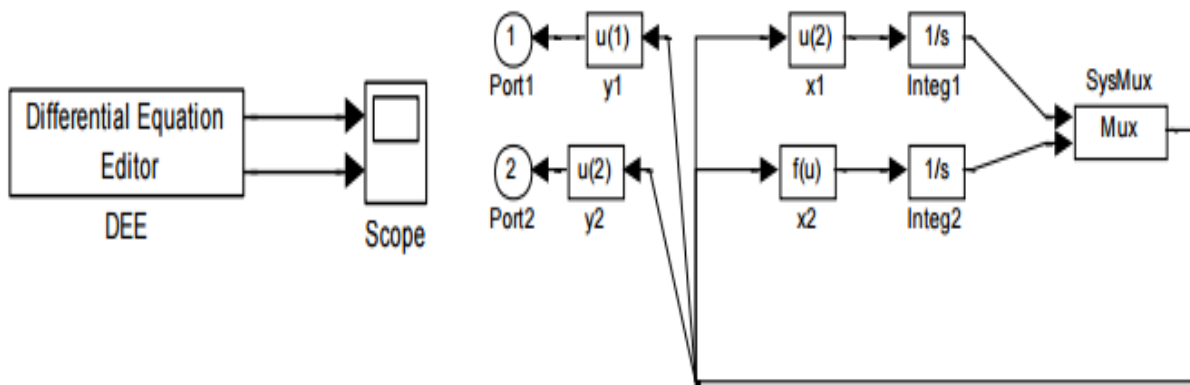
$$X_{k+1} = AX_k + B u_k, \quad y = CX_k + D u_k.$$

Редактор дифференциальных уравнений DEE Эффективным средством для решения нелинейных дифференциальных уравнений с известными начальными условиями является моделирование в SIMULINK. Оно требует построения эквивалентной структурной схемы и ее реализации на стандартных линейных и нелинейных блоках. Определенную помощь в этом процессе может оказать редактор дифференциальных уравнений DEE (Differential Equation Editor). Этот редактор сам строит схему моделирования для SIMULINK по системе дифференциальных уравнений, записанной в форме Коши. Это немного эффективней, чем "ручное" построение схемы на интеграторах, и несколько проще, чем написание блока «с нуля». Редактор вызывается командой `dee`. Появляется окно с несколькими блоками, один из них называется Differential Equation Editor, его надо скопировать (перетащить мышкой) на рабочую страницу SIMULINK, войти в него двойным щелчком мыши и ввести параметры дифференциального уравнения (левые части, начальные условия и др.). Приведем пример заполнения параметров для моделирования дифференциального уравнения математического маятника

$$\begin{aligned} \dot{y}_1 &= y_2, & y_1(0) &= 0, & y_2(0) &= 0, \\ \dot{y}_2 &= -y_1, \end{aligned}$$

В дальнейшем входить в блок для изменения параметров можно с помощью команды `diffeqed`.

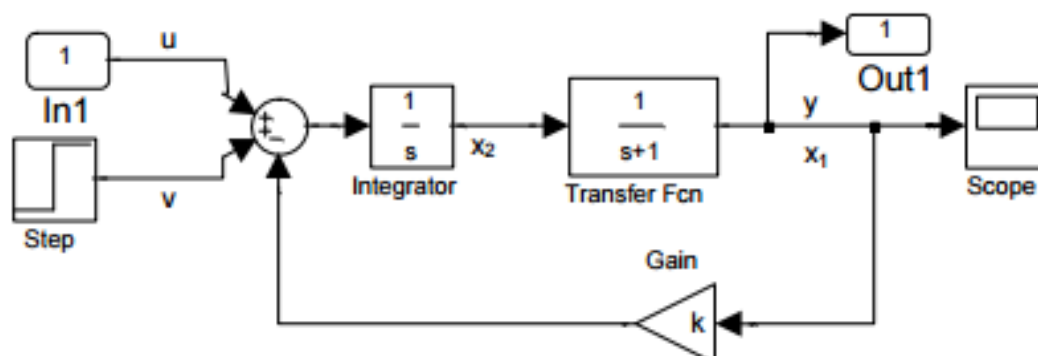
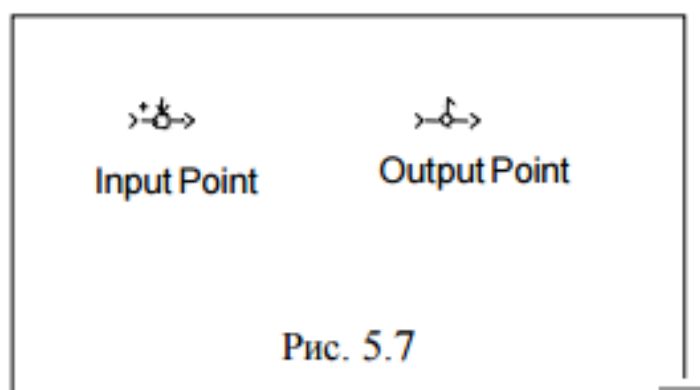
Чтобы одновременно наблюдать графики двух сигналов в блоке Scope в параметрах осциллографа устанавливаем поле «Number of axes» равным двум. Параметры осциллографа можно редактировать, нажав кнопку. Если есть желание взглянуть непосредственно на схему моделирования, соответствующую заданному уравнению, надо, выделив блок, нажать правую кнопку мыши и выбрать пункт меню `look under mask`, при этом появится схема, показанная справа на рис.



2. Анализ Simulink-моделей

Для получения графиков временных или частотных характеристик схемы, нарисованной в SIMULINK, и построения диаграммы нулей и полюсов ее передаточной функции, используется команда Linear Analysis из пункта Tools меню, расположенного вверху рабочей страницы SIMULINK. Ответ выдается в виде диаграмм средства LTIview. Предварительно на схеме надо пометить вход и выход, пользуясь метками входных и выходных точек Input Point, Output Point из библиотеки Control_System_Toolbox (в ранних версиях – Model_Inputs_and_Outputs) (см. рис.).

Для модели (в общем случае – нелинейной), реализованной в SIMULINK, можно получить описание в пространстве состояний. Это делается при помощи команды linmod. Поясним ее применение на примере схемы следящей системы, показанной на следующем рис., которой мы присвоим имя test_sys.mdl. Предварительно на схеме надо пометить вход и выход, используя блоки In и Out. Найдем



описание схемы в пространстве состояний при $k=1$. Набирая в командной строке код `[A,B,C,D]=linmod('test_sys')`, получим ответ: $A = -1 \ 1 \ -1 \ 0$ $B = 0 \ 1$ $C = 1 \ 0$ $D = 0$. Число выходных параметров команды linmod можно брать любым от 0 до 4. Применяя эту команду для нелинейных систем, получим описание в пространстве состояний линеаризованной системы. В теории управления часто встает задача определения стационарных точек системы, т.е. положений равновесия, при которых все производные равны нулю $\dot{X} = 0$. Для Simulink-моделей эту задачу можно решить при помощи команды trim (от trim – приводить в порядок; уравнивать). Входным параметром функции является имя mdl-файла, а выходными – координаты точки равновесия в фазовом пространстве X , а также установившиеся значения входного и выходного сигналов u и y . Возвращаемое значение dX в случае успеха должно быть равно нулю.

3. Маскирование подсистем в SIMULINK

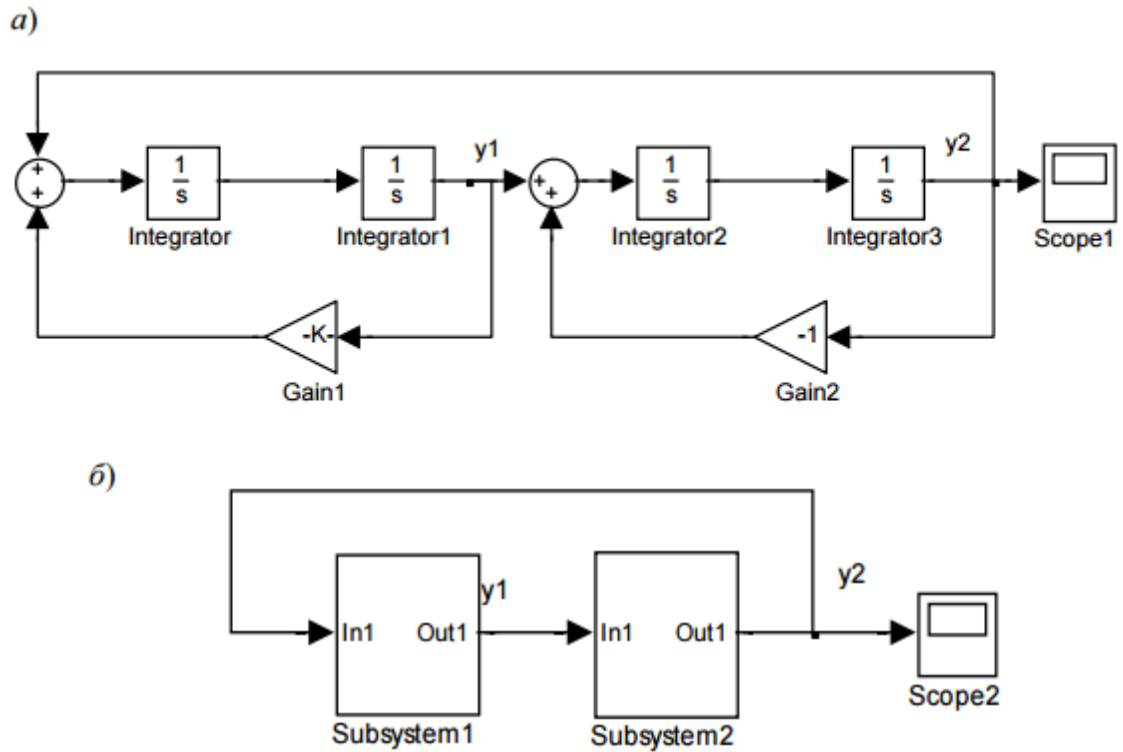
При моделировании сложных систем, когда Simulink-схема становится слишком большой, бывает полезно перейти к иерархической модели, разбив систему на подсистемы. Выделив мышью группу блоков, можно изобразить их в виде одного блока подсистемы, используя пункт меню Edit/Create subsystem. В дальнейшем пользователь может, щелкнув мышью по блоку подсистемы, войти внутрь и редактировать эту подсистему. Иногда такую возможность следует запретить, произведя так называемое «маскирование» системы (Edit/Mask subsystem). Содержимое маскированной системы можно просмотреть по команде меню Edit/Look under mask. По команде Edit/Edit mask открывается редактор масок. Он содержит кнопку «Unmask», позволяющую удалить маску, а также управлять параметрами маскированной подсистемы и процедурой отрисовки блока в редакторе SIMULINK.

Пример. Моделирование двойного маятника. Консервативная колебательная система из двух взаимосвязанных маятников описывается уравнениями

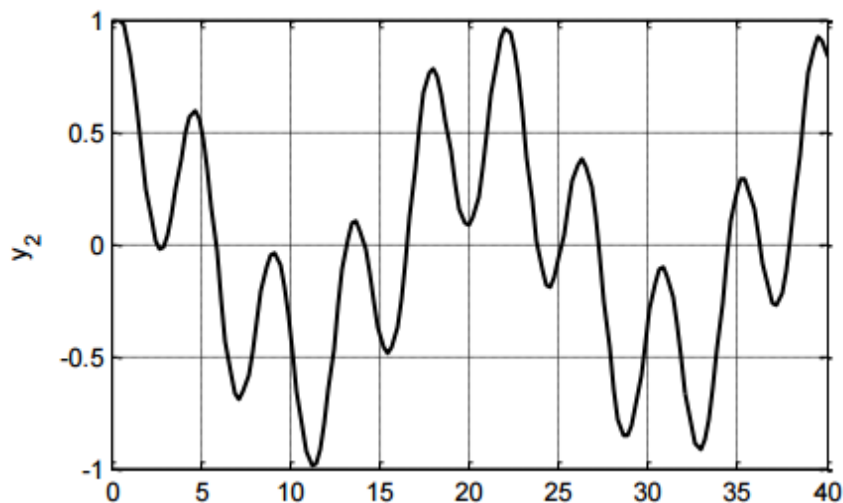
$$\ddot{y}_1 + ky_1 + y_2 = 0,$$

$$\ddot{y}_2 + y_1 + y_2 = 0.$$

Схема моделирования этих уравнений в SIMULINK приведена на рис., а. Ее левая часть отвечает первому маятнику, правая – второму. Выделяя поочередно эти части и выполняя их маскирование, получаем схему, изображенную на рис., б. Она занимает меньше места и наглядно показывает структуру взаимосвязи маятников. Чтобы войти внутрь подсистемы и изменить ее параметры, нужен двойной щелчок мышью.



На следующем рис. приведен результат моделирования для начальных условий $y_1(0) = 0$, $y_2(0) = 1$ и значения $k=1.2$.



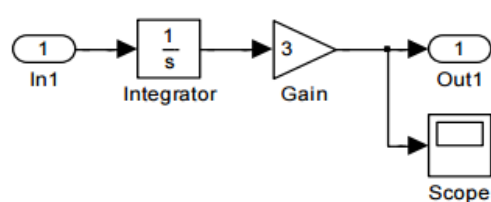
4. Управление Simulink-моделью из MATLAB

При организации численных экспериментов в SIMULINK может возникнуть необходимость многократно запускать одну и ту же модель с различными параметрами и входными сигналами. Это можно обеспечить, управляя процессом моделирования из командного окна.

Прежде всего, необходимо открыть SIMULINK-модель вручную или при помощи команды **open**. Так, команда **open('vdp')** открывает модель vdp.mdl (схема моделирования уравнения Ван дер Поля). Чтобы запустить

процесс моделирования из командной строки используется команда `sim`, например, `sim('vdp')`. В качестве параметров этой команде можно передать время моделирования, входной сигнал (при наличии входных портов у модели), начальные условия и опции решателя. В этом случае синтаксис ее вызова будет иметь вид `y=sim('model',timespan,options,ut)`, где `timespan` – интервал моделирования, `ut` – массив, состоящий из отсчетов времени и значений входного сигнала.

Пример. Создадим модель, показанную на рис. и сохраним ее под именем `simple_model`. Найдем ее реакцию на синусоиду на интервале 30с. Опций моделирования использовать не будем – в качестве аргумента возьмем пустой массив.



```
>>open('simple_model');
>>t=0:0.1:30;
>>sim('simple_model',[0 30],[t' sin(t)])

>> [t1,y1]=sim('simple_model',[0 30],[t' sin(t)]);
>>plot(t1,y1)
```

После выполнения этих команд можно открыть блок `Scope` в схеме и увидеть результаты моделирования. Поместить результаты моделирования в рабочее пространство и затем посмотреть график с помощью команды `plot` можно, используя вызов `sim` с выходными параметрами. При вызове команды `sim` с опциями решателя структура опций задается командой `simset`. Для примера изменим тип решателя на `ode23`: `>> [t2,y2]=sim('simple_model',[0 30],simset ('Solver','ode23'),[t' sin(t)]);` С полным списком опций моделирования можно ознакомиться, используя команду `simget`, например, `o=simget ('simple_model')`. Помимо задания опций моделирования, существует возможность программно изменять параметры блоков. Для их чтения и записи используют команды `get_param` и `set_param`. Поскольку названия параметров блоков не всегда совпадают с текстами пояснений в окне параметров, бывает полезна конструкция `get_param(gcb, 'objectparameters')`. Мышью в окне модели выбирается нужный блок. После этого команда `gcb` возвращает имя текущего блока. Так, если будет выделен блок `Gain`, команда `gcb` вернет строку `'simple_model/Gain'`. При помощи той же функции `get_param` можно получить любой конкретный параметр

Контрольные вопросы:

1. Создайте простую модель в пакете SIMULINK.
2. Приёмники сигналов в SIMULINK.
3. Источники в SIMULINK

Тема 10-11	ВВЕДЕНИЕ в программу AutoCAD	
Технология проведения лекционного занятия		
Время 4 часа	Количество студентов 40-60 человек	
Вид занятия	Введение. Визуальная лекция	
План лекционного занятия	<ol style="list-style-type: none"> 1. Графические возможности . 2. Методы задания координат . 3. Организация и работа пакета AUTOCAD. 4. Простое и сложное редактирование 5. Вывод информации 	
Цель учебного занятия: Формирование у студентов глубокого представления о целях и задачах пакета AUTOCAD и путях её дальнейшего развития		
Педагогические задачи:		Результаты обучения:
<ul style="list-style-type: none"> - Ознакомление с основными задачами инженерной науки - Ознакомление графических е возможности - Внедрение в сознание студентов основных законов инженерной науки 		Студент: <ol style="list-style-type: none"> 1. Имеет представление о целях и задачах инженерной науки 2. Излагает историю развития программирования и выдающихся её представителях 3. Характеризует связь инженерной программирования с другими предметами и её место в развитии науки 4. Характеризует основные законы инженерной науки и приводит примеры по ним
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.3. Приводит список используемой литературы по инженерной программирования . Знакомит студентов с рейтинговой оценкой знаний студентов. Объявляет тему и план занятия, знакомит с достижениями инженерной науки	1.1. Слушает и записывает
2-этап Основной	<ol style="list-style-type: none"> 2.1. Обращает внимание студентов на вопросы плана занятия и обращает внимание на основные понятия. Проводит опрос 2.2. Продолжает излагать лекционный материал. На примерах приводит и объясняет основные законы инженерной науки . 2.3. Обеспечивает студентом внимание и запись основных понятий и законов программирование. 	<ol style="list-style-type: none"> 2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают
3-этап Итоговый	<ol style="list-style-type: none"> 3.1. Делает заключительные выводы лекционной темы и заостряет внимание студентов на основные вопросы 3.2. Отмечает активных студентов 3.3. Даст задание на самостоятельную подготовку: Дать определение каждого закона, привести примеры 	<ol style="list-style-type: none"> 3.1. Слушают и усваивают 3.2. Записывают задание

Лекция 10-11. Введение в программу AutoCAD

План:

1. Графические возможности .
2. Методы задания координат .
3. Организация и работа пакета AUTOCAD.
4. Простое и сложное редактирование
5. Вывод информации

1. Графические возможности

Программа AutoCAD представляет собой систему автоматизированного проектирования, относящуюся к классу так называемых CAD-систем (Computer Aided Design System) – систем, предназначенных для проектирования моделей объектов и разработки конструкторской документации.

Произведя установку всех необходимых компонентов и выполнив запуск системы AutoCAD, пользователю открывается основное окно программы. Начиная с версии 2010, рабочее пространство имеет характерный ленточный интерфейс. На рис. 1.1., представлено рабочее пространство AutoCAD 2013.

В зависимости от целей использования рабочая среда имеет различную компоновку. Всего в системе AutoCAD реализовано четыре вида рабочих пространств:

2D Drafting & Annotation – двумерное моделирование;

AutoCAD Classic – классический AutoCAD;

3D Modeling – трехмерное моделирование;

Initial Setup Workspaces – рабочее пространство начальной настройки.

Переход между рабочими пространствами осуществляется выбором из раскрывающегося списка на панели инструментов **Workspaces (Рабочие пространства)**

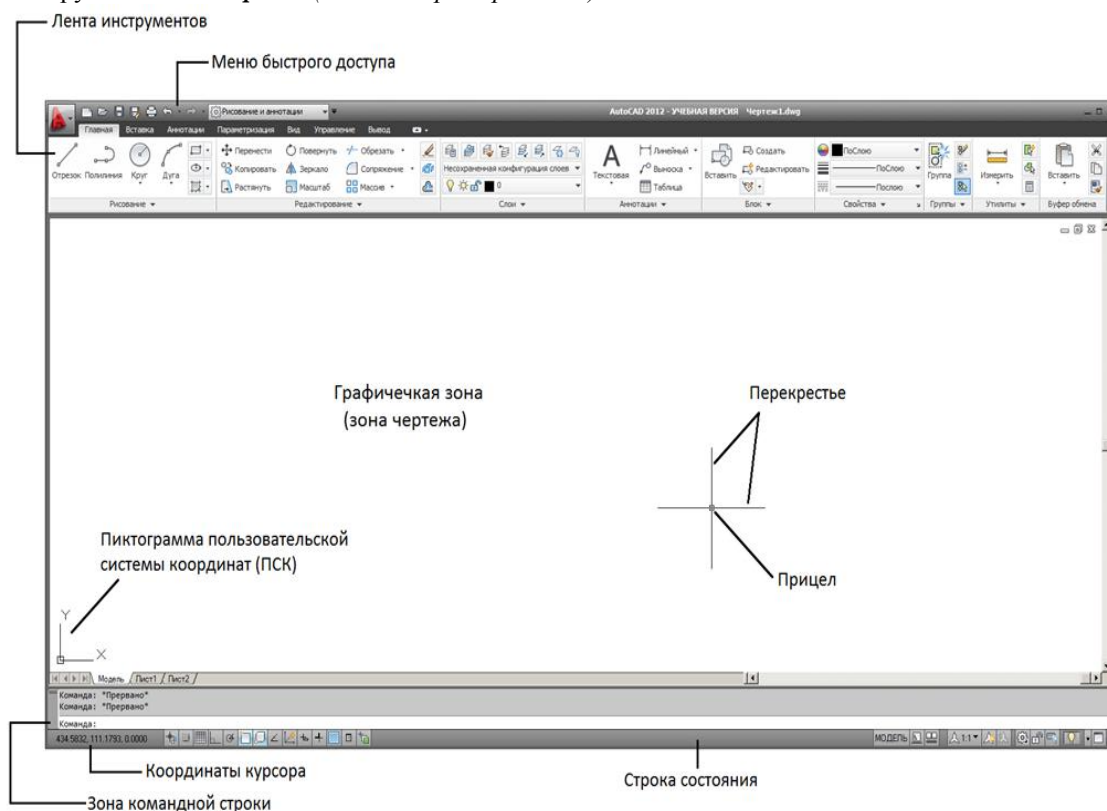


Рис. 1.1 Рабочее пространство AutoCAD 2012 по умолчанию.

Графическая зона – пространство (по умолчанию черного цвета) в середине рабочего окна, в котором производятся все построения.

Ниже ПСК располагаются текущие значения координат курсора. В AutoCAD графическая зона представляет собой пространственную систему координат, в которой пользователь может задавать координаты точек как произвольно с помощью мыши, так и с заданной точностью с помощью клавиатуры.

В самом низу рабочего пространства AutoCAD располагается строка состояния, в левой части которой отображаются текущие координаты курсора, а также размещены кнопки, задающие режимы черчения (рис 1.2)

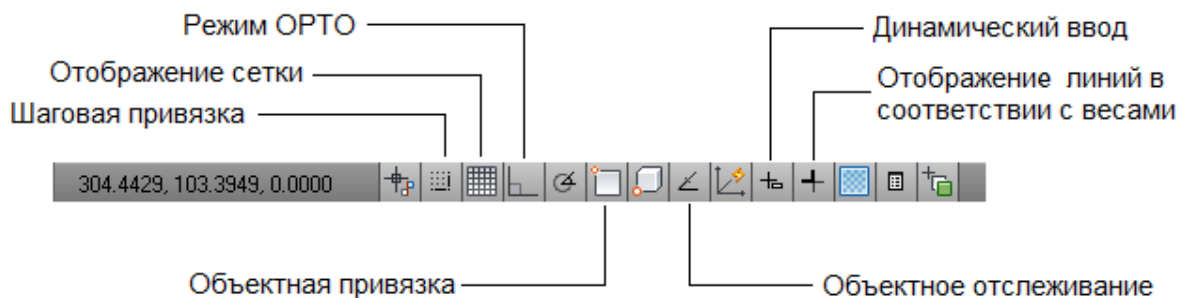


Рис.2. Строка состояния системы AutoCAD

Режим ОРТО – включение режима ортогональных построений, что упрощает начертание прямых горизонтальных и вертикальных линий.

Отображение сетки – отображение вспомогательной сетки из точек с заданным шагом.

Шаговая привязка – включение режима привязки к определенным точкам, равномерно отстоящим друг от друга с некоторым интервалом. Точки шаговой привязки и вспомогательной сетки могут не совпадать.

Объектная привязка – включение режима объектной привязки к характерным точкам объектов на чертеже.

Отображение линий в соответствии с весами – включение режима отображения толщины линий на чертеже.

Зона командной строки – это небольшое встроенное прямоугольное окно внизу рабочего пространства.

Также начинающему пользователю следует помнить следующее:

1. Все действия в AutoCAD выполняются с помощью команд;
2. Каждая команда может быть вызвана тремя способами:
 - щелчком левой кнопки мыши по соответствующей кнопке на панели инструментов;
 - выбором из строки меню;
 - вводом ее имени в командную строку;
3. Использование каждой последующей команды возможно только после завершения предыдущей.

Настройка параметров рабочего пространства в системе AutoCAD осуществляется в диалоговом окне **Options (Настройки)** (рис. 1.5), вызов которого можно осуществить следующими способами:

- из строки меню **Tools (Инструменты)** -> **Options (Настройки)**;
- вводом в командную строку `_options (НАСТРОЙКА)`;

Как и в любой программе Windows, панели инструментов AutoCAD 2006 предоставляют быстрый доступ к наиболее часто используемым командам. Программа AutoCAD поставляется с несколькими панелями инструментов.

-Standard (Стандартная). Расположена непосредственно под главным меню.

С ее помощью выполняется управление файлами и другие стандартные операции Windows. Кроме того, на стандартной панели инструментов расположены кнопки некоторых специфичных для AutoCAD операций, таких, как зумирование и панорамирование.

-S Styles (Стили). Расположена справа от стандартной панели инструментов. Во многом аналогична панели инструментов форматирования в программах Microsoft, однако здесь она предназначена для форматирования текстов AutoCAD размерных объектов и табличных стилей.

-S Layers (Слои). Расположена под стандартной панелью. Содержит команды и раскрывающийся список, предназначенные для управления слоями чертежа.

Это наиболее фундаментальная панель инструментов AutoCAD.

-S Properties (Свойства). Расположена справа от панели инструментов Layers и немного напоминает панели форматирования в других программах Microsoft, однако здесь она предназначена для форматирования таких свойств объектов AutoCAD, как цвет, тип и толщина линий.

-f Draw (Рисование). Расположена вертикально с левой стороны экрана и содержит наиболее часто используемые команды раскрывающегося меню Draw.

-S Modify (Редактирование). Расположена вертикально в правой части экрана и содержит команды раскрывающегося меню Modify.

S Draw Order (Последовательность вывода). Находится под панелью Modify и содержит кнопки управления расположением __ объектов один поверх другого.

Как и в других приложениях Windows, вы можете переместить, открыть или скрыть любую панель инструментов.

- Чтобы переместить панель инструментов, наведите указатель мыши на ее *бордюр* (двойная линия в торце панели) и, удерживая нажатой кнопку мыши, перетащите панель в нужное место.

-S Чтобы открыть или скрыть панель инструментов, щелкните правой кнопкой мыши на любой панели и в раскрывшемся списке установите или снимите флажок рядом с ее именем (см. рис. 1.4).

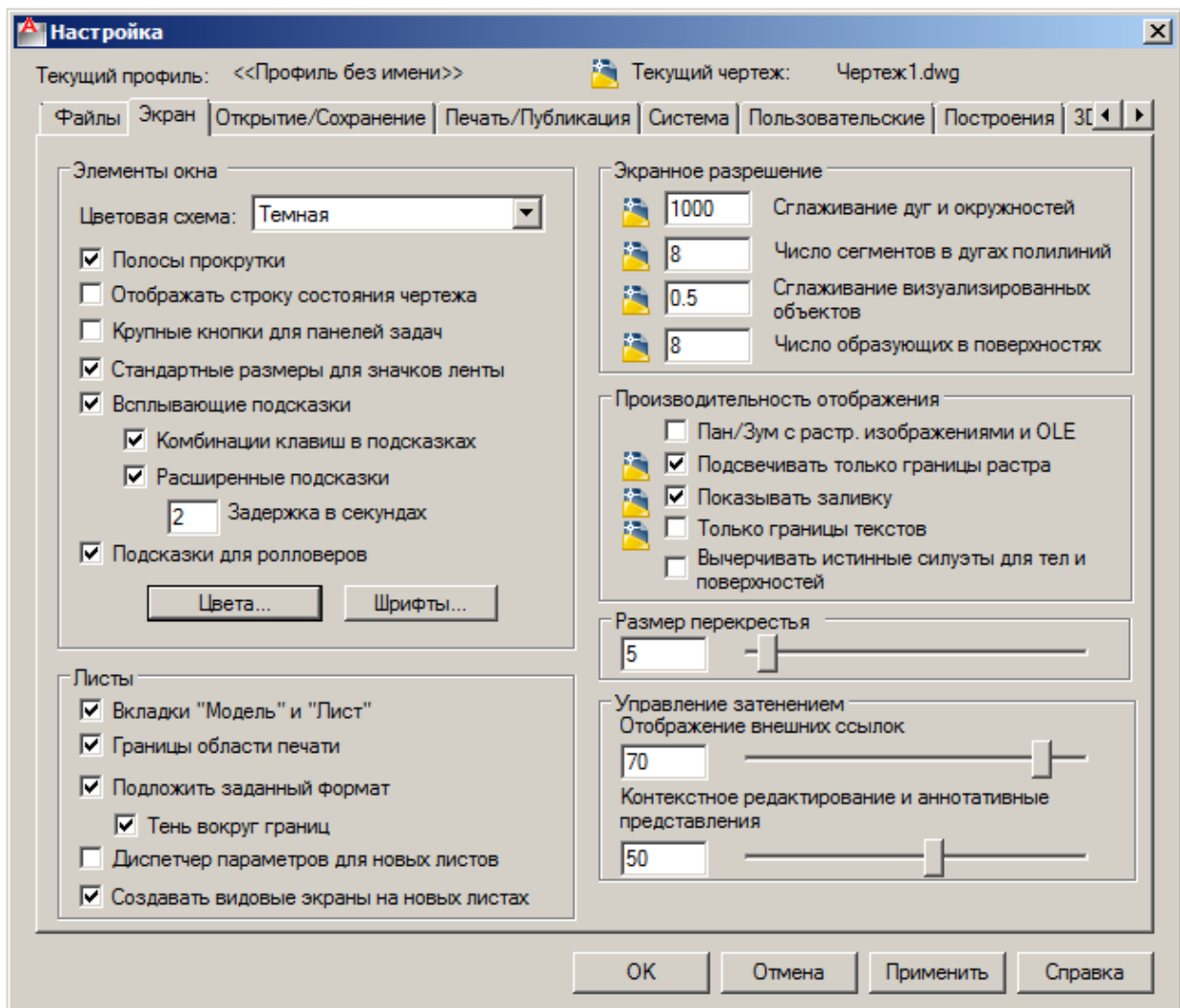


Рис.1.3. Диалоговое окно *Options* (*Настройка*)

Как уже отмечалось ранее, графическая зона в системе AutoCAD представляет собой пространственную систему координат, в которой пользователь может задавать координаты точек с помощью клавиатуры.

Всего в AutoCAD предусмотрено пять способов задания координат:

- интерактивный метод;
- метод абсолютных координат;
- метод относительных прямоугольных координат;
- метод относительных полярных координат;
- задание направления и расстояния;

В интерактивном методе пользователь осуществляет задание координат щелчками мыши в пространстве чертежа. Недостаток такого метода – недостаточная точность.

Метод абсолютных координат заключается в непосредственном вводе координат X и Y в командную строку через запятую. В основе данного метода лежит стандартная прямоугольная система координат.

Метод относительных прямоугольных координат отличается от метода абсолютных координат тем, что координаты (X,Y) задаются относительно последней заданной точки (рис. 1.5). При вводе относительных прямоугольных координат используется символ @.

Метод полярных координат подразумевает указание двух параметров:

- расстояния от исходной точки;
- угла между нулевым направлением полярной системы отсчета и вектором, направленным от начала координат к искомой точке.

При задании относительных полярных координат используется два специальных символа: @ и <.

На рис. 1.6 показан отрезок, построенный в относительной полярной системе координат.

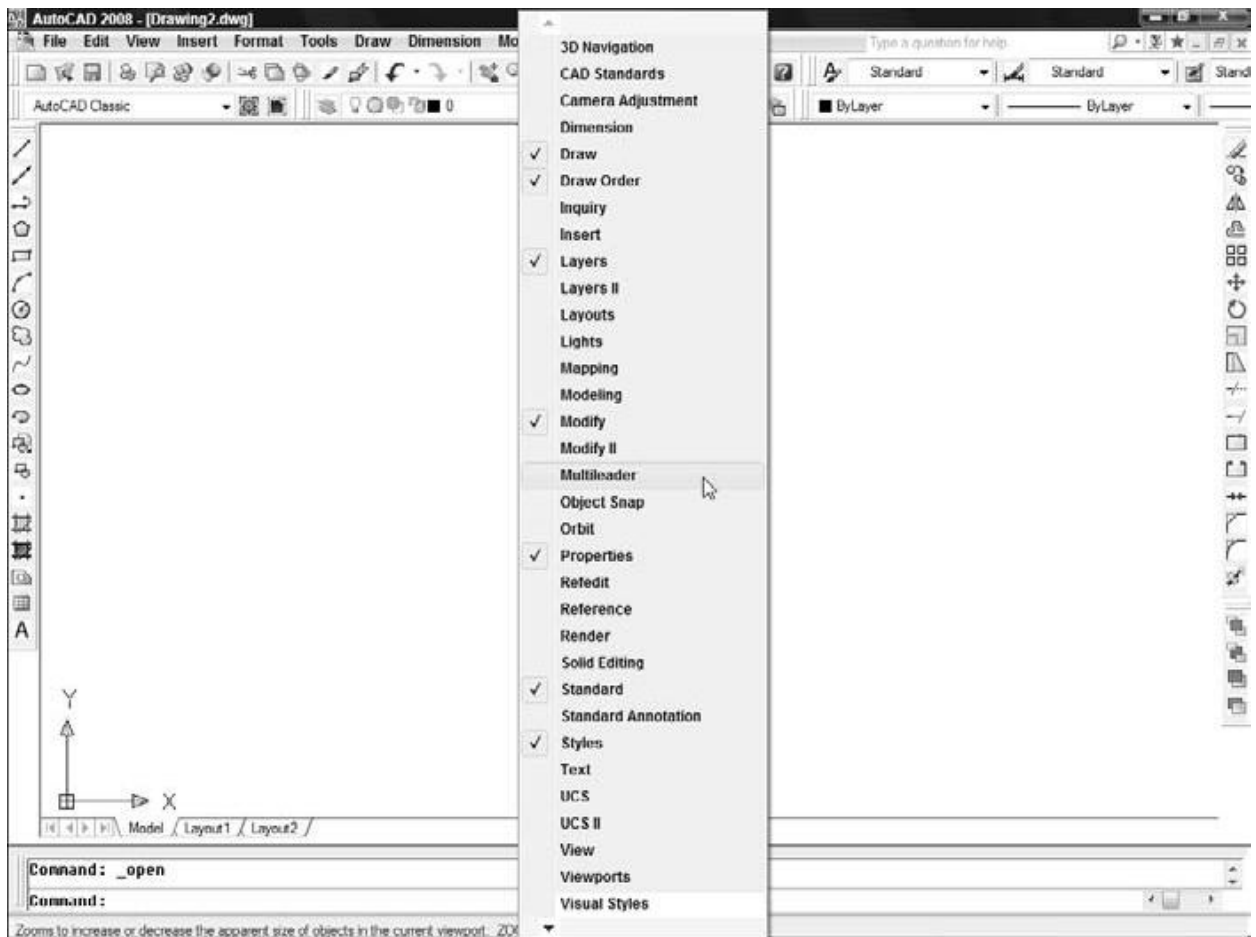


Рис. 1.4. Панели инструментов

David Byrnes <<AutoCAD for Dummies>> 26-с

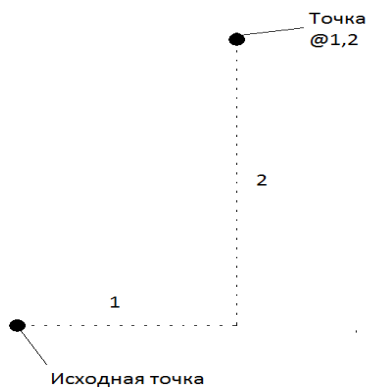


Рис. 1.5. Задание относительных координат

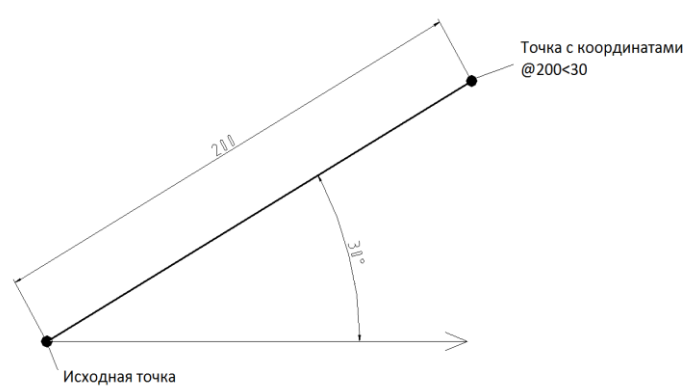


Рис. 1.6. Задание относительных полярных координат

Шаговая привязка – режим рисования, являющийся вспомогательным средством черчения. В режиме шаговой привязки курсор будет перемещаться только между узлами прямоугольной координатной сетки с заданным шагом.

В системе AutoCAD простановка размеров автоматизирована. Это означает, что пользователю нужно указать только тип размера и точки, по которым размер будет построен. При этом системой будет определено и отражено значение размера. В дальнейшем при изменении объекта, для которого проставлен размер, будет происходить изменение значения размера.

Очевидно, что размер в AutoCAD имеет сложную структуру и воспринимается как единый объект, состоящий из выносных линий, размерной линии со стрелками и численным значением размера (рис. 1.7).

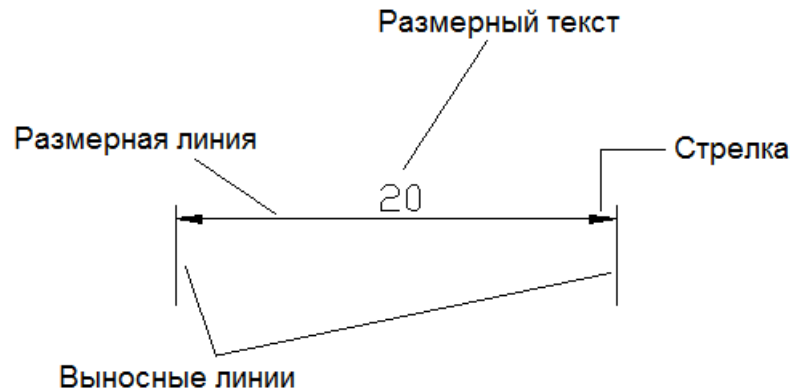


Рис 1.7. Структура размера в AutoCAD

Нанесение размеров и управление ими производится через раздел меню **Dimension (Размеры)**. Основные используемые размеры – это:

- **Linear (Линейный)** – размер параллелен одной из осей координат;
- **Aligned (Параллельный)** – размер параллелен измеряемому участку;
- **Arc Length (Длина дуги)** – размер определяющий длину дуги;
- **Radius (Радиус)** и **Diameter (Диаметр)** для измерения радиусов и диаметров кругов и дуг;
- **Angular (Угловой)** – для измерения углов;
- **Continue (продолжение)** – нанесение цепочки размеров;
- **Leader** – выносной текст, выноска.

Наиболее часто применяемые размеры, **Linear** и **Aligned**, проставляются тремя щелчками мыши (начальная точка измерения, конечная точка измеряемого участка, положение размерной линии).

Рассмотрим основные правила нанесения размеров, регламентированные по ГОСТ 2.307-68:

1. Первая выносная линия должна находиться на расстоянии 10 мм от контура объекта.
2. Расстояние между параллельными размерными линиями должно составлять 7-10 мм.
3. Выносные линии должны выходить за концы стрелок размерной линии на 1-5 мм.
4. Размеры следует наносить таким образом, чтобы ближе к изображению детали был расположен меньший размер.
5. Размерный текст наносится над размерной линией как можно ближе к ее середине. Для величин, размерная линия которых расположена вертикально, размерный текст пишется и читается сле-ва.
6. В том случае, если на чертеже имеется несколько одинаковых элементов, размер рекомендуется наносить лишь для одного из них, при чем с указанием общего количества таких элементов.
7. При вычерчивании плоской детали в одной проекции ее длину можно указывать с помощью английской буквы *l*, а толщину – с помощью буквы *s*.
8. Осевая линия должна выходить за контур детали на 2-3 мм.
9. Если окружность изображена полностью, то для нее наносят диаметральный размер. Для дуг наносят радиальный размер.
10. При нанесении размера радиуса перед размерным числом помещают прописную букву *R*.
11. Размерные линии и сами размеры предпочтительно располагать за контуром изображения.
12. Необходимо избегать пересечения размерных и выносных линий, а также пересечения размерных линий между собой.
13. Каждый размер наносят на чертеже только один раз.
14. Размерный текст не разрешается пересекать или разделять другими линиями чертежа. В месте нанесения размерного числа осевые, центровые линии и линии штриховки прерываются.

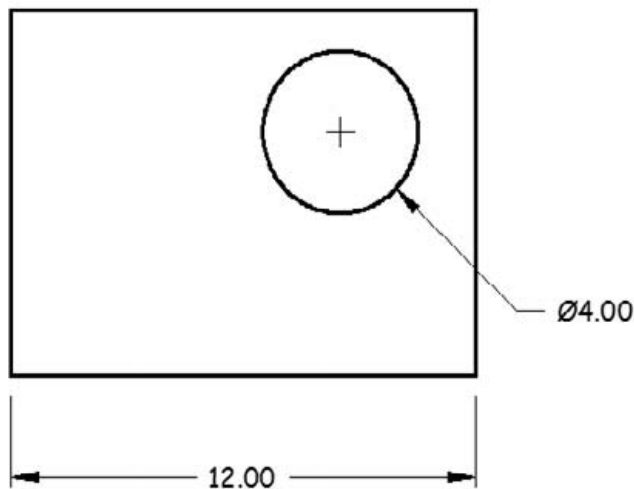
Названия всех элементов размерного объекта показаны а описание каждого из них приведено ниже.

S Размерное число (часто его называют *размерным текстом*). Представляет собой значение того или иного пространственного параметра моделируемого объекта (например, значение реального расстояния или угла). Однако размерное число может сопровождаться вспомогательной текстовой информацией, которая дополняет, а в некоторых случаях замещает его. Например, можно добавить суффикс вроде ТИП, чтобы указать, что размер является типичным для аналогичных объектов чертежа, или же можно вставить описание наподобие такого: Подробности на 3/A2.

• **S Размерная линия** (Dimension line). Пересекает размерное число (проходит параллельно краю измеряемого объекта) и предназначена для указания протяженности измеряемого объекта. В соответствии с параметрами размерного стиля, заданными в AutoCAD по умолчанию, размерное число располагается в центре размерной линии как по горизонтали, так и по вертикали.

Однако эти параметры можно изменить, например расположить размерное число над размерной линией, не разрывая ее, как показано на

Подробнее об этом речь идет несколько позже в этой главе.



Размерный текст Стрелочка
Выносная линия Размерная линия
Определяющая точка выносной линии
Рис. 1.8. Элементы размерного объекта •

•/ **Размерные** стрелки (Dimension arrowhead). Находятся на концах размерной линии и позволяют более четко обозначить пределы измеряемой длины. По умолчанию размерная линия оканчивается заполненными (закрытыми) стрелками, которые показаны на рис. 1.8. Однако можно выбрать в качестве "стрелок", обозначающих концы размерной линии, и другие символы, например засечки. (Вы уж не обессудьте, но любые символы, которыми оканчиваются размерные линии, в AutoCAD называются *размерными стрелками*, даже если на стрелку они совсем не похожи.)

•S **Выносная линия** (Extension line). Служит для указания определяющих точек объекта (часто заданных с помощью объектной привязки), расстояние между которыми обозначается размерной линией и размерным числом. По негласной договоренности на чертежах между определяющей точкой и началом выносной линии принято оставлять небольшой зазор. Кроме того, обычно выносную линию продолжают чуть дальше той точки, в которой она пересекается с размерной линией.

Назначение системы

Система AutoCAD разработана американской фирмой Autodesk в начале 80-х годов и была первоначально ориентирована на существовавшие в то время персональные компьютеры (PC XT, PC AT без сопроцессора и т. п.). Уже эти, по нынешним меркам слабые, версии вызвали интерес у конструкторов и чертежников, желавших автоматизировать свой труд хотя бы в части рисования на листе бумаги.

Широкое распространение системы в России началось с десятой версии, которая работала в операционной системе MS DOS, существовала, как в английском, так и русском вариантах (как, впрочем, и в других национальных модификациях). Эта версия была уже достаточно развита, поскольку команды можно было вводить из командной строки или экранных, падающих и графических меню.

Одиннадцатая версия прошла сравнительно незаметно. Следующей популярной в России версией стала двенадцатая, которая обладала диалоговыми окнами даже в варианте для MS DOS (вариант для Windows 3.1 и Windows 95 тоже существовал, но не переводился на русский язык). В ней было появившееся еще в 11-й версии "пространство листа", окончательно утвердившее AutoCAD как пространственную графическую систему, в которой, построив трехмерный объект, можно было вывести его виды (проекции) в расположенные на поле листа окна (видовые экраны) в необходимом масштабе.

Тринадцатая версия существовала сразу в двух вариантах (для MS DOS и Windows 95), причем на стадии инсталляции (установки на компьютер) можно было выбрать один вариант системы или установить сразу оба. Четырнадцатая версия, вобрав в себя новшества тринадцатой, была сделана более компактной и быстрой, чем предыдущая. Она была рассчитана только на операционную систему Windows (Windows 95 или Windows NT), поскольку эта система де-факто уже стала общеиспользуемой и устанавливалась на все новые персональные компьютеры.

В 1999 году началось внедрение 15-й версии, которой, отдавая дань моде, присвоили номер 2000. Эта версия стала очередным шагом вперед, как в простом двумерном рисовании, так и в трехмерном моделировании. Заметные изменения претерпели средства управления выводом на плоттер (графопостроитель) и принтер (устройство печати).

Первые варианты системы содержали в основном инструменты для простого двумерного рисования, которые постепенно, от версии к версии, дополнялись и развивались. В результате AutoCAD стал очень удобным "электронным кульманом".

Большим преимуществом такого средства рисования является возможность формирования электронного архива чертежей. Каждый из созданных в системе AutoCAD чертежей легко редактируется, что позволяет быстро разрабатывать чертежи-аналоги по чертежам-прототипам. Для облегчения процесса выпуска чертежной документации можно разрабатывать "библиотеки стандартных элементов". Эта идея стала хорошим стимулом для создания на базе системы локальных рабочих мест по различным конструкторским, архитектурным и другим направлениям, а также для разработки новых специализированных систем.

Десятая версия позволяла выполнить достаточно сложные трехмерные построения в любой плоскости пространства и отобразить их на разных видовых экранах с различными точками зрения. Поэтому она уже была инструментом трехмерного (и тем более двумерного) моделирования.

Двенадцатая версия стала переломной. Она позволила работать с расширенной памятью, ввела диалоговые окна, а через появившийся в 11-й версии механизм пространства листа и видовых экранов дала возможность получать чертеж с проекциями трехмерного объекта или сооружения. В AutoCAD 2000 эта идея получила дальнейшее развитие: по одной модели можно получать несколько листов чертежа. Таким образом, система AutoCAD 2000 предназначена не только для черчения, но и для формирования трехмерных моделей.

Требования к компьютеру

Персональный компьютер, на котором может быть установлена система AutoCAD 2000, должен удовлетворять определенным требованиям. Это должна быть машина класса Pentium 133 или выше, с оперативной памятью 32 Мбайта (лучше 64 Мбайта), винчестером (жестким диском) 1 Гбайт — на винчестере надо иметь свободными 200 Мбайт под программное обеспечение и не менее 100 Мбайт для временных файлов, которые система образует во время сеансов работы. И конечно, на винчестере должно оставаться место для хранения создаваемых файлов чертежей. На компьютере должна быть установлена операционная система Windows NT 4.0 или Windows 95 или Windows 98.

Итак, требования к компьютеру, на котором планируется установить современную систему AutoCAD, достаточно высоки. Похожи требования и для 14-й версии (в ней можно ограничиться емкостью оперативной памяти в 32 Мбайта), а для 13-й версии размер оперативной памяти может быть 24 Мбайта.

Эти цифры заметно превосходят требования, предъявлявшиеся к AutoCAD 12, предназначенной для MS DOS: PC 486DX 66 с оперативной памятью 8 Мбайт и свободными 100 Мбайт на жестком диске. А 10-я версия вообще могла работать на 286 компьютере с сопроцессором, оперативной памятью 1-2 Мбайт и с винчестером 40 Мбайт.

AutoCAD является гибкой системой, легко встраиваемой в вычислительные системы предприятий. Она не только адаптируется к особенностям информационных потоков, но и позволяет конструкторам и архитекторам разрабатывать свои графические базы и программные продукты и интегрировать их в среду AutoCAD.

Известный пакет AutoCAD компании Autodesk является постоянно развивающейся средой проектирования и пользуется у специалистов по САПР заслуженным авторитетом. Это высокопроизводительное программное обеспечение с Windows-интерфейсом позволяет одинаково хорошо решать задачи, касающиеся различных областей проектирования, в том числе машиностроения, строительства, архитектуры, электроники, экономических оценок в проектировании, управления инфраструктурой и другие. На основе этой системы или, как принято говорить, в среде AutoCAD созданы различные приложения, позволяющие проектировать дома, печатные платы, станки и роботы, одежду и т.д. Данное средство предоставляет пользователю возможности, которые ранее могли быть реализованы только на больших и дорогих вычислительных системах. С помощью AutoCAD может быть построен любой рисунок, если только его можно нарисовать вручную. Другими словами, AutoCAD способен выполнять практически любые виды графических работ.

В основу структуры AutoCAD был положен принцип открытой архитектуры, позволяющий адаптировать и развивать многие функции AutoCAD применительно к конкретным задачам и требованиям. Это позволяет создавать собственные:

- экранные, планшетные, кнопочные, падающие и графические меню для автоматизации часто выполняемых действий;
- пакетные файлы для автоматизации выполнения длинных последовательностей команд;
- шрифты;
- типы линий;
- образцы штриховок;
- библиотеки символов и фрагментов рисунков;
- образцы штриховок формата PostScript;
- рисунки-прототипы с заданными значениями параметров по умолчанию.

Пакет поддерживает множество различных устройств ввода-вывода, включая графические мониторы с высоким разрешением, дигитайзеры и манипуляторы, графопостроители, принтеры и даже фотонаборные

машины. К достоинствам пакета во всех версиях можно отнести хорошую его настройку на работу с плоттерами - основными аппаратными средствами распечатки чертежей компьютерного производства.

Основные понятия AutoCAD

Примитивы

В отличие от "художественных" графических редакторов, AutoCAD работает не с изображением как таковым, а с геометрическим описанием объектов, составляющих изображение. Так, например, отрезок во внутреннем представлении графического редактора AutoCAD описывается двумя точками, круг описывается центром и радиусом.

Все примитивы AutoCAD обладают рядом свойств (принадлежность слою, цвет, тип линии, ширина). Некоторые из этих свойств (например, цвет) присущи всем примитивам.

Системы координат

Используется традиционная декартова система координат. Можно ввести пользовательские системы координат с помощью команды USC. В определенный момент времени пользователь работает только с одной предварительно выбранной системой координат, которая называется текущей. Вся работа с изображением проводится в текущей системе координат.

Единицы измерения и масштаб

Расстояния между точками на рисунке измеряются в условных единицах. Конкретный формат представления размеров (дюймы, футы, миллиметры и др.) не имеет значения для AutoCAD. Иначе говоря, при создании объектов в чертеже AutoCAD "измеряет" все расстояния в относительных единицах. В AutoCAD нет масштаба в обычном понимании конструктора, конструктор задает все расстояния и координаты в реальных единицах - мы как бы работаем в масштабе 1:1. Масштабирование различных частей изображения в соответствии с желаемым форматом документа может осуществляться в момент компоновки чертежа (команда SCALE (МАСШТАБ)) или при выводе чертежа или его части на плоттер (принтер).

Вид

Когда вы создаете чертеж, то работаете с изображением части чертежа, выводимой на дисплей. Будем называть эту часть изображения видом. Эту видимую часть чертежа (окно зрения) можно увеличивать (при этом изображение чертежа будет уменьшаться), уменьшать (изображение будет увеличиваться) или перемещать по полю чертежа без изменения масштаба отображения (панорамирование). Изменение вида осуществляется командой ZOOM.

Слой

AutoCAD дает возможность распределять выбранные фрагменты чертежа по различным слоям. Работая за кульманом, конструктор имеет дело с одним листом бумаги и располагает изображения объектов на нем и только на нем. В среде AutoCAD располагать изображения можно как бы на нескольких совмещенных в пространстве носителях (это можно сравнить с наложенными друг на друга прозрачными кальками). Например, чертеж может содержать на одном слое построение призмы со всеми вспомогательными линиями, на другом - сам чертеж в окончательном исполнении, а на третьем - ход лучей в призме. Количество слоёв не ограничивается. Слои можно делать видимыми и невидимыми. С каждым слоем чертежа связывается цвет и тип линий. По мере создания чертежа вы можете вводить новые слои, менять свойства существующих. Подобно системе координат, в любом чертеже AutoCAD всегда существует по крайней мере один слой с именем "0".

Чертеж

Чертеж - это файл, содержащий некую графическую и вспомогательную информацию, полностью описывающую графический объект. AutoCAD предоставляет в распоряжение конструктора все вычислительные ресурсы микроЭВМ и новые средства создания и редактирования чертежа.

Вместе с каждым рисунком AutoCAD содержит так называемые *системные переменные*, в которые заносится определенная информация: о текущих установках рисования (т. е. установках слоя, цвета, типа линий и т. п.), о последнем выполненном действии (т. е. имя последней команды, последняя точка, последний радиус и т. п.), о настройках некоторых команд (т. е. длины фаски, радиус сопряжения и т. п.) и многое другое. Пользователь может вывести на экран перечень и значения системных переменных и большую часть из них изменить. Остальные изменяются самой системой в процессе работы.

Система AutoCAD позволяет настраивать многие элементы пользовательского интерфейса. Параметры настройки формируются уже на стадии установки AutoCAD на ваш компьютер — большая часть по умолчанию, а что-то (например, размещение папок для программного обеспечения) задает пользователь. Для изменения установок нужно воспользоваться либо командой НАСТРОЙКА (OPTIONS), либо пунктом **Настройка...** (Options...) падающего меню **Сервис** (Tools), либо пунктом **Настройка...** (Options...) контекстного меню, вызываемого по щелчку правой кнопкой мыши в зоне командных строк.

Типы примитивов

Примитивы могут быть простыми и сложными. К простым примитивам относятся следующие объекты: точка, отрезок, круг (окружность), дуга, прямая, луч, эллипс, сплайн, текст.

К сложным примитивам относятся: полилиния, мультилиния, мультитекст, размер, выноска, допуск, штриховка, вхождение блока или внешней ссылки, атрибут и растровое изображение.

Способы ввода координат точек

Можно задавать конечные точки отрезка с помощью мыши. Но этот способ ввода (указания) точек не является единственным. Больше распространен **второй способ** — ввод координат точки с клавиатуры, например: **65,113.24**

В данном примере введена точка с двумя координатами: $X=65$ мм, $Y=113.24$ мм. При вводе координат с клавиатуры запятая является разделителем между абсциссой и ординатой, а точка используется как разделитель между целой и дробной частью числа. Вводимые координаты могут быть целыми или вещественными. При вводе координат следует учитывать, где вы выбрали точку с координатами 0,0. Чаще всего это точка левого угла графического экрана (хотя в процессе работы вы перемещаетесь по рисунку, и точка 0,0 может оказаться в любом месте, даже уйти в невидимую часть чертежа).

Третий способ ввода точек — это относительный ввод в декартовых координатах с клавиатуры, например: @50,25

Данная запись означает, что новая точка задается относительно предыдущей (что определяет символ "@"), со сдвигом по оси X на +50 мм (т. е. вправо на 50 мм) и сдвигом по оси Y на +25 мм (т. е. вверх на 25 мм). Здесь запятая также является разделителем координат. Вводимые числа могут быть целыми и вещественными, положительными, нулевыми и отрицательными.

Четвертый способ ввода точек — это относительный ввод в полярных координатах с клавиатуры, например: @33.5<45

В этой форме записи уже нет запятых, зато появился символ "<", который интерпретируется как знак угла. В данном примере новая точка задается относительно предыдущей, причем расстояние между ними в плоскости равно 33,5 мм (т. е. числу влево от символа угла), а вектор из предыдущей точки в новую образует угол 45 градусов с положительным направлением оси абсцисс (угол измеряется в тех угловых единицах, которые вы задали в настройке). Расстояние должно обязательно быть положительным, а угол может быть любым числом.

Пятый способ ввода точек — это указание с помощью функций объектной привязки. Доступ к функциям объектной привязки осуществляется либо через групповую кнопку панели **Стандартная** (Standard), либо через панель **Объектная привязка** (Object Snap). *Групповой* называется кнопка, у которой в правом нижнем углу имеется черный треугольник. Если выбрать указателем мыши такую кнопку и нажать (не отпуская!) левую кнопку мыши, то раскроется набор кнопок инструментов, которые входят в данную группу. Нужно опуститься по появившимся кнопкам до той, которая вам нужна, и только тогда отпустить нажатую левую кнопку мыши, однако объектной привязкой лучше пользоваться, имея на экране одноименную панель.

Получение справок

В процессе работы очень полезными оказываются команды получения справочной информации о создаваемых объектах. Групповая кнопка справочных операций позволяет получить информацию об:

- **Расстоянии** (Distance)
- **Площади** (Area)
- **Массе** (Mass Properties)
- **Списке** (List)
- **Координатах** (Locate Point).

Команды общего редактирования

Кнопки команд общего редактирования объектов (копирование, перенос, удлинение и т. п.) расположены в панели **Редактирование** (Modify). Каждую из этих команд можно ввести по имени с клавиатуры, а также вызвать с помощью падающего меню **Редакт** (Modify). Многие команды данной группы работают либо с набором предварительно выбранных объектов, либо при отсутствии такого набора выдают запрос **Выберите объекты:** (Select objects:). Остальные команды запрашивают редактируемые объекты в соответствующий момент времени.

Свойства.

У каждого примитива могут быть свои цвет, слой, тип линии, масштаб типа линии, стиль печати, вес линии, гиперссылка и высота — все это в AutoCAD отнесено к **свойствам**. Напомню, что определить текущие значения свойств объекта можно, например, с помощью команды СПИСОК (LIST).

При создании сложных рисунков возникает необходимость присвоения имен отдельным объектам или группам объектов, чтобы ими можно было удобнее оперировать в дальнейшей работе. Особенно это важно при разработке своих собственных приложений, функционирующих в среде AutoCAD. Данной цели служит еще одно свойство примитивов — слой. Более того, слой обладает неоценимой возможностью замораживания (выключения), когда ряд второстепенных в данный момент объектов можно, не удаляя, сделать невидимыми, что позволит успешнее работать с главными объектами.

Вес линии — совершенно новое свойство примитивов, которое отсутствовало в предыдущих версиях системы AutoCAD — это толщина, с которой объект будет выводиться на устройство печати (или графопостроитель). Вы можете нарисовать объекты тонкой линией, но задать ненулевой вес и получить при этом жирные линии на листе бумаги.

Высота — это свойство примитива, применяемое в трехмерных построениях. Оно задает величину выдавливания вдоль оси Z, расположенной перпендикулярно осям X и Y. Например, чтобы круг преобразовать в цилиндр, его нужно выдавить на ненулевую высоту.

В рисунках системы AutoCAD могут присутствовать описания стилей некоторых объектов, что, конечно, облегчает оформление чертежа. К таким стилям относятся: текстовые, размерные и стили мультитиний.

Вывод на плоттер.

Под понятием "плоттер" будем иметь в виду не только графопостроитель, но и любое другое устройство вывода, в том числе и принтер. Стоит заметить, что для современных устройств, использующих струйную и лазерную технологию, практически отсутствует грань между принтерами и плоттерами. Поэтому принтерами часто называют плоттеры небольшого формата (не превышающего А2). Поскольку в меню и в документации системы AutoCAD чаще используется термин плоттер, то мы будем применять именно его для обозначения любого устройства вывода.

Любое устройство (локальное или сетевое), к которому вы обращаетесь для вывода чертежа из AutoCAD, должно быть специальным образом конфигурировано (описано) в системе AutoCAD. Операция по установке плоттеров или редактированию их настроек требует специальных знаний. Ее лучше выполнять опытным пользователям или системным программистам, обслуживающим вычислительные комплексы, на которых функционирует AutoCAD. Дополнительную информацию можно найти либо в справочной системе AutoCAD, либо в документации, поставляемой вместе с системой. Автономная настройка (т. е. не зависящая от связи с AutoCAD) самого устройства выполняется с помощью документации, поставляемой вместе с плоттером.

Для того чтобы определить, настроена ли ваша версия AutoCAD, а если настроена, то на плоттеры каких марок, следует воспользоваться командой НАСТРОЙКА (OPTIONS). Эту команду можно вызвать либо с помощью пункта **Настройка** (Options) падающего меню **Сервис** (Tools), либо с помощью контекстного меню, появляющегося при нажатии правой кнопки мыши, если ее указатель расположен в этот момент в зоне командных строк. Команда НАСТРОЙКА (OPTIONS) вызывает диалоговое окно **Настройка** (Options).

Для работы с наиболее распространенными плоттерами и форматами графических файлов в системе AutoCAD присутствуют специальные программы (драйверы), обеспечивающие передачу данных на соответствующие устройства или в соответствующие форматы. AutoCAD 2000 в стандартной поставке поддерживает большое количество перьевых и струйных типов плоттеров таких фирм, как Hewlett-Packard, Xerox, Ose, CalComp и Houston Instruments, а также наиболее распространенные форматы растровых файлов (JPEG, BMP, PNG, TGA и др.) и форматы PostScript, применяемые в лазерных устройствах печати.

Стили печати

Стили печати — это новое свойство, отсутствовавшее в предыдущих версиях AutoCAD, которое отображает графические объекты при выводе на плоттер специальным образом. Таким образом, примитив в рисунке может на экране выглядеть совсем не так, как он будет нарисован плоттером на бумаге. Изменяться может цвет, тип, а также вес линии. Можно также задать специальное оформление концов и заливки линии. Все такие установки заносятся в таблицы стилей. Система AutoCAD при установке создает ряд стандартных таблиц стилей печати, которые доступны пользователю.

Стили печати могут быть двух видов: именованные и цвето-зависимые. Именованный стиль печати может быть назначен любому объекту, а цвето-зависимый стиль используется в зависимости от цвета примитива.

Цвето-зависимые стили, которых в каждой таблице 255 (по количеству цветов системы AutoCAD), описывают, каким образом нужно выводить на плоттер объекты, имеющие данный цвет. Такие стили удобны для вывода на перьевой плоттер, который имеет ограниченное количество цветов и размеров перьев. По умолчанию, когда имя таблицы действующего цвето-зависимого стиля не задано, действует стиль, который выводит объекты в том виде, в каком они созданы в рисунке.

Блоки и внешние ссылки.

Важным инструментом автоматизации процесса разработки чертежей является использование блоков и внешних ссылок. **Блок**. — это сложный именованный объект, для которого создается описание, состоящее из любого количества примитивов системы AutoCAD текущего рисунка. Блок имеет базовую точку и может применяться для вставки в любое место чертежа, причем в процессе вставки возможен его поворот и масштабирование с различными коэффициентами по разным осям. Примитив, который образуется от операции вставки блока, называется **ВХОЖДЕНИЕ БЛОКА (BLOCK REFERENCE)**. В рисунке может быть любое количество вхождений одного и того же блока.

Внешняя ссылка — это изображение внешнего файла вместе с элементами текущего рисунка, причем файл, на который Вы таким образом ссылаетесь, не переносится в основной рисунок. В результате текущий рисунок может быть насыщен большим количеством внешних изображений новых объектов, но размер текущего файла от этого практически не увеличится. Примитив, образующийся от операции вставки внешней ссылки, будем называть **вхождением внешней ссылки** или просто **внешней ссылкой**.

Оба упомянутых инструмента являются средством автоматизации труда конструктора и чертежника. С помощью блоков можно строить однотипные объекты, описывая полностью только один их них. Внешние ссылки дают возможность пользоваться ранее созданными файлами стандартных графических элементов.

Первый шаг к использованию блока — создать описание этого блока. Для этого нужно определиться, из каких примитивов будет состоять блок и где у него будет базовая точка. Объекты, которые были включены в блок при его описании, сохраняют свои основные свойства (слои, цвет, тип линии, вес) и во вставленном блоке. Исключением является специальное значение ПОБЛОКУ, которое может быть дано цвету, типу линии и весу.

Таким образом, любые части рисунков могут сохраняться в виде отдельных файлов, а любые созданные файлы могут вставляться в текущий рисунок с образованием (или без образования) блоков.

Часто возникает необходимость вместе с блоком держать и надписи, которые могли бы менять свои значения после вставки блока. Например, если Вы рисуете схему с использованием заранее подготовленных блоков условных элементов, тогда номера или наименования вставленных графических элементов Вам нужно будет оформить в виде текстовых надписей. Однако в системе AutoCAD есть специальный примитив, называемый **ОПИСАНИЕ АТРИБУТА (ATTRIBUTE DEFINITION)**, который может быть включен в описание блока, а при операции вставки этого блока будет запрошено его значение и создан атрибут (текстовая строка), входящий в состав блока.

Атрибуты могут содержать текстовую информацию, которая дополняет графические примитивы рисунка. Извлечение значений атрибутов может быть сделано с помощью специальной команды **АТЭКСП (ATTTEXT)**, которая выводит извлекаемые данные в текстовый файл. Эта операция полезна при создании систем автоматизированного проектирования на базе AutoCAD.

Внешние ссылки

Вставка с помощью команды **ВСТАВИТЬ (INSERT)** одного файла рисунка в другой рисунок, который является текущим, увеличивает его объем, т. к. в него переносятся примитивы вставляемого файла. Но есть еще один способ добавить к текущему рисунку изображение другого рисунка — вставить файл с помощью внешней ссылки. При этом вставляемый файл в текущий рисунок не переносится, а только запоминается его полное имя (обычно вместе с путем). В дальнейшем, когда AutoCAD открывает рисунок, имеющий внешнюю ссылку, то загружается сначала открываемый файл, а затем — содержимое дополнительного файла-ссылки. Таким образом, файл-ссылка не хранится вместе с основным рисунком. Разумеется, при таком варианте основной файл имеет меньший размер по сравнению с вариантом вставки файла с помощью команды **ВСТАВИТЬ (INSERT)**, но попадает в зависимость от дополнительного файла, т. к. тот должен всегда обнаруживаться на своем привычном месте и не менять своего имени. Возможны вложенные ссылки, когда ссылка выполняется на вставляемый файл, который сам содержит внешнюю ссылку на другой файл. Команда **ССЫЛКА (XREF)** управляет в текущем рисунке внешними ссылками на другие файлы.

Вставка объектов, созданных другими системами.

AutoCAD может читать ряд других графических форматов и вставлять объекты, созданные другими известными приложениями (например, Microsoft Office).

Вставка и редактирование растровых изображений

Растровое изображение — это изображение, состоящее из точек (растров), которые благодаря цветам формируют рисунок. AutoCAD может прочитать файл с растровой картинкой и вставить его в текущий рисунок в виде цветного прямоугольника (аналогично внешней ссылке). Редактировать вставленное изображение на точечном уровне AutoCAD не может, но может выполнять подрезку, масштабирование, перенос и другие простые операции редактирования. При наложении одного растрового изображения на другое можно управлять порядком их следования (переносить на передний план или убирать на задний).

Для операций с растровыми изображениями используется команда **ИЗОБ (IMAGE)**, которой соответствует кнопка **Изображение (Image)** панели **Ссылки (Reference)**, а также кнопка **Изображение (Image)** панели **Вставка (Insert)** и пункт **Диспетчер изображений... (Image Manager...)** падающего меню **Вставка (Insert)**.

Трехмерные построения.

AutoCAD может строить примитивы не только в плоскости XY, но в любой плоскости трехмерного пространства. Кроме того, в системе AutoCAD существует большой набор пространственных примитивов (поверхностей, тел и др.), которые позволяют выполнять построения трехмерных моделей зданий, сооружений и различных машиностроительных изделий.

Можно не только строить трехмерные объекты, но и рассматривать их в разных видах и проекциях, используя новые системы координат. Имеются такие возможности AutoCAD, как скрытие невидимых линий, тонирование и назначение объектам тех или иных материалов. Все построенные модели можно с помощью пространства листа, оформлять красиво и удобно в виде чертежей.

Работа по модернизации чертежа

Общеизвестно, что в процессе проектирования чертежа конструктор много времени тратит на редактирование. Несмотря на то, что мы с вами, возможно, традиционным способом сможем начертить новый чертеж быстрее, чем с помощью системы AutoCAD, наверняка для исправления его нужно будет потратить немалые усилия, т.е. извести массу стирательной резинки, пожертвовать товарным видом чертежа в пользу его правильности или долго и нудно перечерчивать чертеж. С использованием редакторских возможностей AutoCAD мы можем значительно облегчить свою жизнь.

Функции редактирования позволяют:

- удалять фрагменты изображения;
- восстанавливать случайно удаленные фрагменты;
- перемещать или поворачивать фрагменты или отдельные изображения относительно других;
- копировать созданные фрагменты и располагать их в указанном месте;
- увеличивать или уменьшать объекты;

- создавать зеркальносимметричное изображение;
- изменять свойства (принадлежность к слою, цвет и тип линии) созданных объектов;
- сопрягать линии и строить фаски;
- делить объекты на равные части или размечать на сегменты с заданным интервалом;
- расчленять блоки или полилинии на составные части;
- редактировать полилинии (сглаживать, изменять свойства и т. д.);
- растягивать части рисунка;
- проводить линии, расположенные на заданном (постоянном) расстоянии относительно других.

Средства реализации систем автоматизации конструкторской документации предоставляет компьютерная графика, обеспечивающая создание, хранение и обработку моделей геометрических объектов и их графических изображений с помощью компьютера. Автоматизация особенно эффективна при разработке устройств на базе параметрически управляемых унифицированных и типовых элементов конструкций, обеспечивающих многовариантность конструирования. Модель геометрических объектов, содержащая информацию о геометрии объекта, используется как для получения двумерной геометрической модели, так и для расчета различных характеристик объекта и технологических параметров его изготовления. Отсюда следует, что геометрическое моделирование является ядром автоматизированного конструирования и технологической подготовки производства.

Система автоматизации разработки и выполнения конструкторской документации выполняет ввод, хранение, обработку и вывод графической информации в виде конструкторских документов. Для реализации системы необходимы: документы, регламентирующие работу системы, исходная информация для формирования информационной базы, информационная база, содержащая модели геометрических объектов и графических изображений, элементы оформления чертежа по ГОСТу, технические и программные средства создания моделей геометрических объектов и графических изображений и их вывода, интерфейс пользователя в виде графического диалога с компьютером.

Построение таких систем значительно упрощается, если они создаются на базе универсальной, открытой среды проектирования для реализации графических возможностей САПР. Примером данной среды является AutoCad – универсальная графическая система, в основу структуры которой положен принцип открытой архитектуры, позволяющей адаптировать и развивать многие функции AutoCad применительно конкретным задачам и требованиям.

Можно выделить два подхода к конструированию на основе компьютерных технологий. Первый подход базируется на двумерной геометрической модели графического изображения и использовании компьютера как электронного кульмана, позволяющего значительно ускорить процесс конструирования и улучшить качество оформления конструкторской документации. Центральное место при таком подходе занимает чертеж, который служит средством представления изделия и содержит информацию для решения графических задач, а также для изготовления изделия. В таком случае получение графического изображения за компьютером будет рациональным и достаточно эффективным, если созданное графическое изображение используется многократно.

В основе второго подхода лежит пространственная геометрическая модель изделия, которая является более наглядным способом представления оригинала и более мощным и удобным инструментом для решения геометрических задач. Чертеж в этом случае играет вспомогательную роль, а способы его создания основаны на методах компьютерной графики и отображения пространственной модели.

При первом подходе обмен информацией осуществляется на основе конструкторской, нормативно-справочной и технологической документации, при втором подходе – на основе компьютерного представления геометрических объектов, общей базы данных, что способствует эффективному функционированию программного обеспечения САПР конкретного изделия.

Можно выделить два основных вида геометрических объектов:

- постоянный, с постоянными размерами и геометрической формой. Например: графические изображения условных графических обозначений радио изделий электрических схем или печатных плат.
- параметрически заданный, с переменными размерами и геометрической формой. Например: радио изделие, зависящее от типоразмеров, типовые и унифицированные несущие конструкции радио электронных устройств, конструктивные элементы типовых деталей.

Постоянные геометрические объекты могут быть сформированы с использованием графического редактора AutoCad. Методы описания параметрически заданных геометрических объектов характеризуются большими затратами на формирование компьютерного представления. Чтобы сократить эти затраты, при описании некоторых групп технических объектов можно пользоваться одним из двух принципиально различных методов: вариантным или генерирующим.

Вариантный метод основан на том, что для определенного класса изделий выявляется модель-образец, с помощью которой можно получить все геометрические формы этого класса изделий. Исполнение изделия определяется заданными параметрами, обнуление которых приводит к исключению составных элементов геометрического объекта. В простейшем случае изменяют только размеры, а конструкция отдельных вариантов класса изделий остается неизменной. Такой вид конструирования называют принципиальным. В этом методе данные технологической документации не подготавливаются каждый раз заново, а закреплены за уже

имеющимися соответствующими чертежами. Применение такого метода предполагает, что выбор геометрии для проектируемого изделия уже сделан. Затраты на описание типовой модели превышают затраты на получение вариантов, поэтому во многих системах используется принцип вложенности моделей: один раз описанные типовые модели используются для формирования других типовых моделей в качестве макрокоманд.

В противоположность вариантному методу при генерирующем определяются различные сочетания конструктивных и технологических элементов и выбирается наилучшее решение. Принцип работы системы, использующей генерирующий метод, основан на разделении геометрического объекта на элементы и создании новых геометрических объектов из имеющихся элементов. Различают следующие группы элементов: основные (функциональные), вспомогательные (конструктивные геометрические и элементы формы) и технологические. С помощью основных элементов создается геометрическая форма детали (наружные и внутренние поверхности), проточки (внутренние и наружные). Это дает прежде всего общее описание детали. С помощью вспомогательных элементов, которые непосредственно связаны с основными, осуществляется более подробное описание детали, что позволяет полностью передать ее геометрическую форму. Технологические элементы или характеристики относятся и к основным, и к вспомогательным элементам. Они также влияют на простановку размеров. САПР, работающие по генерирующему принципу, обладают высокой гибкостью и пригодны для решения множества задач. Использование данного метода эффективно, так как большинство конструкторских разработок, называемых новыми конструкциями, создается путем ранее неиспользовавшегося сочетания элементов, давно известных как по принципу формирования, так и по исполнению.

Контрольные вопросы:

1. Для чего необходима данная программа?
2. Какие имеются методы задания координат?
3. Какие операционные системы поддерживают формат данной программы?

Тема 12-13	SCADA системы. Общие понятия о SCADA системе	
Технология проведения лекционного занятия		
Время 4 часа	Количество студентов 40-60 человек	
Вид занятия	Визуальная лекция	
План лекционного занятия	1. Обзор SCADA системы 2. Сеть связи 3. Центральный Компьютера 4. Рабочие станции оператора и компоненты программного обеспечения	
ель учебного занятия: Формирование у студентов глубокого представления о целях и задачах SCADA систем и путях её дальнейшего развития		
Педагогические задачи:		Результаты обучения:
- Ознакомление с основными задачами SCADA системы - Ознакомление с историей развития SCADA системы и выдающимися её представителями - Внедрение в сознание студентов основных законов SCADA системы		Студент: Имеет представление о целях и задачах SCADA системы Излагает историю развития SCADA системы и выдающихся её представителей Характеризует связь SCADA системы с другими предметами и её место в развитии науки Характеризует основные SCADA систем и приводит примеры по ним
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.4. Приводит список используемой литературы по SCADA систем. Знакомит студентов с рейтинговой оценкой знаний студентов. Объявляет тему и план занятия, знакомит с достижениями SCADA систем	1.1. Слушает и записывает
2-этап Основной	2.1. Обращает внимание студентов на вопросы плана занятия и обращает внимание на основные понятия. Проводит опрос 2.2. Продолжает излагать лекционный материал. На примерах приводит и объясняет основные SCADA систем. 2.3. Обеспечивает студентом внимание и запись основных понятий и SCADA систем.	2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают
3-этап Итоговый	3.1. Делает заключительные выводы лекционной темы и заостряет внимание студентов на основные вопросы 3.2. Отмечает активных студентов 3.3. Даст задание на самостоятельную подготовку: Дать определение каждого закона, привести примеры	3.1. Слушают и усваивают 3.2. Записывают задание

Лекция 12-13. SCADA системы. Общие понятия о SCADA системе.

План:

1. Обзор SCADA системы
2. Сеть связи
3. Центральный Компьютера
4. Рабочие станции оператора и компоненты программного обеспечения

1. Обзор SCADA систем

Национальная система связи (NCS) был создан на основе президентского меморандума, подписанного президентом Джоном Кеннеди 21 августа 1963 г. В меморандуме назначен NCS ответственность за обеспечение необходимых коммуникаций для федерального правительства в рамках национальных чрезвычайных условиях, связывая вместе, совершенствование и расширение коммуникационные возможности различных учреждений. В апреле 1984 года президент Рональд Рейган подписал Указ (Е.О.) 12472, Назначение национальной безопасности и готовности к чрезвычайным ситуациям (NS / EP) Функции Телекоммуникации, который расширил миссию и направленность национальной системы связи. С того времени, NCS оказывает содействие Председателю и исполнительного аппарата президента (ИАП) в осуществлении военного времени и не в военное время электросвязи в чрезвычайных ситуациях и в координации планирования для, и снабжение, NS / EP связи для федерального правительства в рамках все обстоятельства. В связи с этим, Управление диспетчера, NCS (OMNCS), в частности, его технологии и программы Отдел (N2), всегда стремится улучшить способность федерального правительства реагировать на национальной безопасности и готовности к чрезвычайным ситуациям ситуаций. В рамках этой миссии, разделение N2 идентифицирует новые технологии, повышающие NS / EP коммуникационные возможности и обеспечивает основные функции NS / EP, такие как приоритет, функциональной совместимости, надежности, доступности и безопасности поддерживаются новыми стандартами. В согласии с этим подходом, N2 управляет федеральной программой стандартизации в области связи. Кроме того, разделение N2 направляет усилия в обоих NS / EP управления и прикладных услуг.

Требования к национальной безопасности и Готовность к чрезвычайным ситуациям попадают в районы как показано в таблице 1.1, и определены в докладе Целевой группы по конвергенции Цель данного технического бюллетеня (ИПТ) заключается в следующем:

- Изучить диспетчерского управления и (SCADA) систем сбора данных;
- Опишите, как системы SCADA развивались с разворачивается в 1960-е годы;
- Изучить, как протоколы SCADA эволюционировали от строго патентованный к разработке открытых протоколов, которые позволяют оборудованию от различных производителей работать вместе;
- Адреса аспекты безопасности систем SCADA;
- Рассматривает стандарты, которые в настоящее время существуют или разрабатывается, чтобы помочь поддержать рост этих систем;
- Наблюдения, выводы и рекомендации о том, как эти технологии могут поддержать NCS и их NS / EP и SIP миссии

SCADA является аббревиатурой для управления наблюдательным и сбора данных. SCADA-системы используются для мониторинга и контроля завода или оборудования в таких отраслях, как телекоммуникации, водоснабжение и управление отходами, энергетики, переработки нефти и газа и транспортировки. Эти системы включают передачу данных между SCADA центральным компьютером хоста и нескольких удаленных терминалов (RTU) и / или программируемых логических контроллеров (ПЛК), а также центральным узлом и терминалами оператора. Система SCADA собирает информацию (например, где произошла утечка на трубопроводе), передает информацию на центральный сайт, а затем предупреждает домашнюю станцию, которая произошла утечка, проведение необходимого анализа и контроля, таких как определение, если утечка имеет решающее значение, и отображения информации в логическом и организовано. Эти системы могут быть относительно простыми, такими как один, который отслеживает условия окружающей среды небольшого офисного здания, или очень сложными, например, как система, которая контролирует всю деятельность в атомной электростанции или деятельности муниципальной системы водоснабжения. Традиционно, SCADA системы сделали использование коммутируемой сети общего пользования (PSN) для целей мониторинга. Сегодня многие системы контролируются с использованием инфраструктуры корпоративной локальной сети (LAN) / глобальной сети (WAN). Беспроводные технологии в настоящее время широко используются для целей мониторинга.

Промышленная система управления (ICS) представляет собой общий термин, который включает в себя несколько типов систем управления, в том числе диспетчерского управления и сбора данных (SCADA) систем, распределенных систем управления (DCS), и другие конфигурации системы меньше контроля, таких как трелевочных установленные программируемые логические контроллеры (PLC) часто встречается в промышленных секторах и критических инфраструктур. Ингаляционных КСП, как правило, используются в таких отраслях, как электротехника, воды, нефти и газа, химической, транспортировке, фармацевтической, целлюлозно-бумажной промышленности, продуктов питания и напитков, а также дискретного производства (например, автомобильной, аэрокосмической и товаров длительного пользования.) Эти системы управления имеют решающее значение для функционирования критических инфраструктур США, которые зачастую тесно

взаимосвязаны и взаимозависимых систем. Важно отметить, что примерно 90 процентов критической инфраструктуры страны находятся в частной собственности и эксплуатации. Федеральные ведомства также работают многие из промышленных процессов, упомянутых выше; Другие примеры включают в себя управление воздушным движением и обработки материалов (например, Почтовая служба обработки почты.) В этом разделе представлен обзор SCADA, DCS, и системы PLC, в том числе типовых архитектур и компонентов. Несколько диаграмм представлены изобразить сетевые соединения и компоненты, обычно встречаются в каждой системе, чтобы облегчить понимание этих систем. Диаграммы в данном разделе, не решать проблемы безопасности и диаграммы в данном разделе, не представляют собой безопасную архитектуру. контроль безопасности Архитектура и безопасности рассматриваются в разделе 5 и разделе 6 настоящего документа, соответственно.

SCADA-системы весьма распределенные системы, используемые для управления географически распределенных активов, часто разбросанные тысячи квадратных километров, где централизованное сбора данных и управления имеют решающее значение для функционирования системы. Они используются в системах распределения, таких как распределение воды и сбора сточных вод, систем нефте- и газопроводов, электрических сетей электропередач и железнодорожные транспортные системы. Центр управления SCADA осуществляет централизованный мониторинг и управление для полевых участков на сетях дальней связи, включая мониторинг аварийных сигналов и данных о состоянии обработки. На основе информации, полученной от удаленных станций, автоматизированных или команды оператора приводом надзора могут быть прижаты к удаленным устройствам управления станцией, которые часто называют полевыми устройствами. Полевые устройства контроля локальных операций, таких как открытие и закрытие клапанов и выключателей, сбор данных от сенсорных систем, а также мониторинг локальной среды для условий сигнализации. DCSS используются для контроля промышленных процессов, таких как производство электроэнергии, нефти и газа, нефтеперерабатывающих заводов, воды и очистки сточных вод, а также химической, пищевой и автомобильной промышленности. DCSS интегрированы в архитектуру управления, содержащую наблюдательный уровень контроля, осуществляющего надзор с несколькими встроенными subsystems, которые отвечают за контроль детали локализованного процесса. Продукт и управления технологическим процессом, как правило, достигается за счет развертывания обратной связи или прямой подачи контура управления при этом ключевые условия продукта и / или процесса автоматически сохраняется вокруг желаемой уставки. Для достижения желаемого продукта и / или допуск процесса вокруг заданной уставки, определенные программируемые контроллеры (PLC) используются в полевых условиях и пропорциональной, интегральной и / или дифференциальных настроек на ПЛК настроены для обеспечения желаемой толерантности, а также скорость самокоррекции во время сбоев процесса. DCSS широко используются в технологических отраслей, основанных.

ПЛК являются компьютерные твердотельные устройства, которые контролируют промышленное оборудование и процессы. В то время как ПЛК являются компонентами системы управления, используемые в системах SCADA и DCS, они часто являются основными компонентами в конфигурации системы управления меньше, используемых для обеспечения регулирующего контроля дискретных процессов, таких как автомобильные сборочных линий и электростанций управления копоти воздухоудовок. ПЛК широко используются практически во всех промышленных процессах.

Процесс на основе обрабатывающей промышленности, как правило, используют два основных процесса [1]:

Непрерывные производственных процессов. Эти процессы происходят непрерывно, часто с переходами, чтобы сделать разные сорта продукта. Типичные непрерывные производственные процессы включают топлива или потока пара в электростанции, нефть на нефтеперерабатывающих заводах и перегонку на химическом заводе.

■ Пакетная производственных процессов. Эти процессы имеют различные этапы обработки, проводимых на количестве материала. Существует четкое начало и конец шаг периодического процесса с возможностью кратковременных стационарных операций во время промежуточных шагов.

Дискретные на основе отраслей обрабатывающей промышленности, как правило, проводят ряд шагов, на одном устройстве для создания конечного продукта. Электронные и механические детали сборки и механообработка деталей являются типичными примерами такого рода промышленности.

Оба процесса на основе и на основе дискретных отраслей используют одни и те же типы систем управления, датчиков и сетей. Некоторые объекты представляют собой гибрид дискретного и на основе процессов производства.

В то время как системы управления, используемые в распределительных и обрабатывающей промышленности очень похожи в эксплуатации, они отличаются в некоторых аспектах. Одним из основных отличий является то, что DCS или PLC контролируемых subsystems, как правило, расположены в пределах более ограниченного пространства завода или завода-ориентированной области, по сравнению с географически рассредоточенных полевых сайтов SCADA. DCS и PLC связи, как правило, осуществляется с использованием локальной сети (LAN) технологии, которые, как правило, более надежны и высокая скорость по сравнению с системами дальней связи, используемых системами SCADA. На самом деле, системы SCADA, специально предназначенные для обработки вызовов дальней связи, такие как задержки и потери данных, создаваемых в различных средствах массовой коммуникации используются. DCS и системы ПЛК обычно используют большую степень управления с обратной связью по сравнению с системами SCADA, поскольку контроль

промышленных процессов, как правило, сложнее, чем диспетчерского управления процессами распределения. Эти различия можно считать тонким для рамки этого документа, в котором основное внимание на интеграции безопасности информационных технологий (IT) в этих системах. На протяжении оставшейся части этого документа, SCADA системы, системы DCSS и PLC будет называться ингаляционных КСП, если конкретная ссылка не делается на один (например, полевого устройства, используемого в системе SCADA).

SCADA-системы состоят из:

- Один или несколько полевых устройств интерфейса данных, как правило, RTUs или ПЛК, которые взаимодействуют с полем сенсорных устройств и местные управления и используют переключатели приводами клапанов;
- система А связи, используемый для передачи данных между устройствами интерфейса полевых данных и блоков управления и компьютеров в SCADA центральном узле. Система может быть радио, телефон, кабель, спутник и т.д., или любую их комбинацию;
- Центральный хост-компьютер-сервер или серверы (иногда называемый SCADA-центр, мастер станции или блок Master Terminal (MTU));
- Набор стандартных и / или программного обеспечения на заказ [иногда называемый человеко-машинный интерфейс (HMI), программное обеспечение или человеко-машинный интерфейс (MMI) Программное обеспечение] системы, используемые для обеспечения SCADA центрального узла и приложения терминала оператора, поддерживает систему связи, а также контролировать и дистанционного управления, расположенные интерфейсные устройства полевых данных.

Рисунок 1.1 показывает очень базовую систему SCADA, а на рисунке 2.2 показана типичная система SCADA. Каждый из указанных выше компонентов системы, будут подробно рассмотрены в следующих разделах.

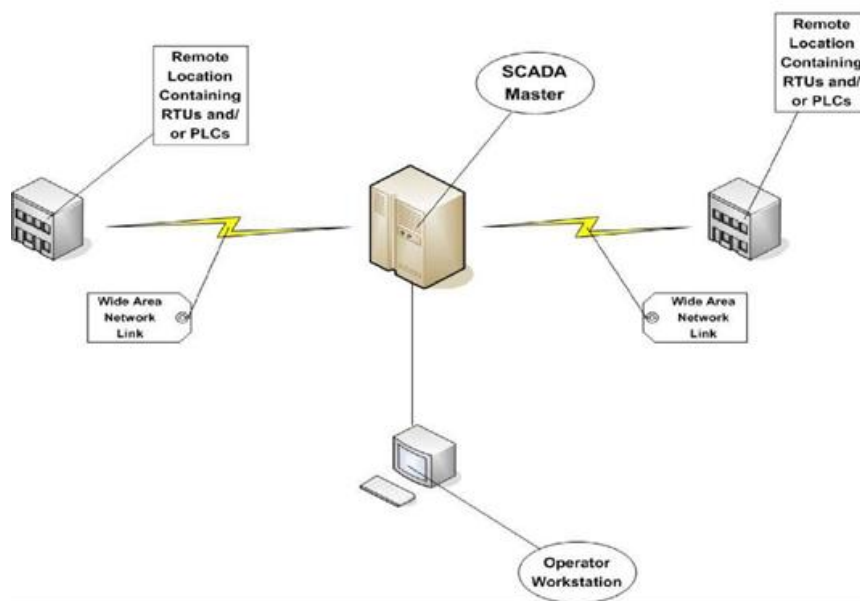


Рисунок 1.1: Типичная система SCADA [4]

Полевые устройства.

Интерфейс передачи данных интерфейсные устройства Полевые данные формируют "глаза и уши" системы SCADA. Такие устройства, как измерители уровня водохранилища, расходомеры воды, датчиков положения клапана, датчики температуры, измерители расхода энергии, а также датчиков давления все предоставляют информацию, которая может сказать, опытный оператор, насколько хорошо выполняет система распределения воды. Кроме того, оборудование, такие как электрические приводы клапанов, распределительных щитов управления двигателем и электронным дозированием химических объектов могут быть использованы для формирования "руки" системы SCADA и помощь в автоматизации процесса распределения воды.

Тем не менее, до того, как автоматизация или удаленного мониторинга может быть достигнуто, информация, которая передается и из интерфейса данных полевых устройств должны быть преобразованы в форму, совместимую с языком системы SCADA. Для достижения этой цели, некоторые формы интерфейса данных электронного поля требуется. RTUs, также известный как дистанционных блоков телеметрических, обеспечивают этот интерфейс. Они в основном используются для преобразования электронных сигналов,

принятых от интерфейса полевой устройств на язык (известный как коммуникационный протокол), используемый для передачи данных по каналу связи.

Инструкции для автоматизации устройств интерфейса полевых данных, таких как логика управления насосом, как правило, хранятся локально. Это в значительной степени из-за ограниченной пропускной способности, типичной для линий связи между SCADA центральным компьютером хоста и устройств интерфейса полевых данных. Такие инструкции традиционно проводятся в ПЛК, которые имеют в прошлом были физически отделены друг от удаленных терминалов. ПЛК представляет собой устройство, которое используется для автоматизации мониторинга и контроля промышленных объектов. Он может быть использован в качестве автономного или в сочетании с SCADA или другой системы. ПЛК подключаются непосредственно к устройствам интерфейса полевых данных и включать запрограммированную интеллект в виде логических процедур, которые будут выполняться в случае некоторых полевых условиях.

ПЛК имеют свои истоки в индустрии автоматизации, и поэтому часто используются в производстве и технологических установок приложений. Потребность в ПЛК, для подключения к каналам связи не было большим в этих приложениях, так как они часто были только необходимы для замены традиционных реле логических систем или пневматические контроллеры. SCADA-системы, с другой стороны, имеют корни в ранних телеметрических приложений, где это было необходимо только знать основную информацию из удаленного источника. RTUs подключены к этим системам не было никакой потребности в программировании управления, поскольку алгоритм локального управления был проведен в логике переключения реле.

Интерфейсные устройства. Полевые данные формируют "глаза и уши" системы SCADA. Такие устройства, как измерители уровня водохранилища, расходомеры воды, датчиков положения клапана, датчики температуры, измерители расхода энергии, а также датчиков давления все предоставляют информацию, которая может сказать, опытный оператор, насколько хорошо выполняет система распределения воды. Кроме того, оборудование, такие как электрические приводы клапанов, распределительных щитов управления двигателем и электронным дозирования химических объектов могут быть использованы для формирования "руки" системы SCADA и помочь в автоматизации процесса распределения воды.

Полевые устройства Интерфейс передачи данных

Тем не менее, до того, как автоматизация или удаленного мониторинга может быть достигнуто, информация, которая передается и из интерфейса данных полевых устройств должны быть преобразованы в форму, совместимую с языком системы SCADA. Для достижения этой цели, некоторые формы интерфейса данных электронного поля требуется. RTUs, также известный как дистанционных блоков телеметрических, обеспечивают этот интерфейс. Они в основном используются для преобразования электронных сигналов, принятых от интерфейса полевой устройств на язык (известный как коммуникационный протокол), используемый для передачи данных по каналу связи.

Инструкции для автоматизации устройств интерфейса полевых данных, таких как логика управления насосом, как правило, хранятся локально. Это в значительной степени из-за ограниченной пропускной способности, типичной для линий связи между SCADA центральным компьютером хоста и устройств интерфейса полевых данных. Такие инструкции традиционно проводятся в ПЛК, которые имеют в прошлом были физически отделены друг от удаленных терминалов. ПЛК представляет собой устройство, которое используется для автоматизации мониторинга и контроля промышленных объектов. Он может быть использован в качестве автономного или в сочетании с SCADA или другой системы. ПЛК подключаются непосредственно к устройствам интерфейса полевых данных и включать запрограммированную интеллект в виде логических процедур, которые будут выполняться в случае некоторых полевых условиях.

ПЛК имеют свои истоки в индустрии автоматизации, и поэтому часто используются в производстве и технологических установок приложений. Потребность в ПЛК, для подключения к каналам связи не было большим в этих приложениях, так как они часто были только необходимы для замены традиционных реле логических систем или пневматические контроллеры. SCADA-системы, с другой стороны, имеют корни в ранних телеметрических приложений, где это было необходимо только знать основную информацию из удаленного источника. RTUs подключены к этим системам не было никакой потребности в программировании управления, поскольку алгоритм локального управления был проведен в логике переключения реле.

В ПЛК использовались чаще для замены реле переключения системы управления логики, телеметрия использовали все больше и больше с ПЛК на удаленных объектах. Это стало желательным, чтобы влиять на программу в ПЛК посредством использования удаленного сигнала. Это по сути является "диспетчерское управление" часть аббревиатурой SCADA. Где требуется только простая локальная программа управления, это стало возможным хранить эту программу в RTU и осуществлять контроль внутри этого устройства. В то же время, традиционные ПЛК включены коммуникационные модули, которые позволили бы ПЛК, чтобы сообщить о состоянии управляющей программы на компьютер, подключенный к ПЛК или к удаленному компьютеру по телефонной линии. Поэтому PLC и RTU производители конкурируют за тот же рынок.

В результате этих событий, линия между ПЛК и удаленными терминалами размыла и терминология практически взаимозаменяемы. Ради простоты термин RTU будет использоваться для обозначения интерфейса данных устройства дистанционного поля; Тем не менее, такое устройство может включать в себя программирование автоматизации, которые традиционно были бы классифицированы как ПЛК.

1.2. Сеть связи

Сеть связи призвана обеспечить средства, с помощью которых данные могут передаваться между центральным хост-компьютером и серверами поля на основе удаленных терминалов. Сеть связи относится к оборудованию, необходимому для передачи данных и с разных сайтов. Среда, используемая может быть либо кабель, телефон или радио.

Использование кабеля обычно реализуется на заводе. Это не практично для систем, охватывающих большие географические области из-за высокой стоимости кабелей, трубопроводов и обширной рабочей силы в их установке. Использование телефонных линий (т.е., в аренду или коммутируемый) является более экономичным решением для систем с большим охватом. Выделенная линия используется для систем, требующих он-лайн связь с удаленными станциями. Это дорого, так как одна телефонная линия будет необходимо на сайте. Коммутируемые линии могут быть использованы в системах, требующих обновления через регулярные промежутки времени (например, ежечасные обновления). Здесь обычные телефонные линии могут быть использованы. Хост может набрать определенное количество удаленного сайта, чтобы получить показания и отправлять команды.

Удаленные узлы, как правило, не доступны по телефонным линиям. Использование радио предлагает экономичное решение. Радиомодемы используются для подключения удаленных объектов к хозяину. Операция по линии также могут быть реализованы в системе радиосвязи. Для тех мест, где не может быть установлена связь прямой радио, радио ретранслятор используется для связывания этих сайтов.

Исторически сложилось так, SCADA сети были посвящены сети; Тем не менее, с увеличением развертывания офисных локальные и глобальные сети в качестве решения для межофисных компьютерных сетей, существует возможность интеграции SCADA локальных сетей в обычных офисных компьютерных сетей.

Передовой преимуществ этой системы является то, что нет необходимости вкладывать средства в отдельную компьютерную сеть для терминалов оператора SCADA. Кроме того, существует простой путь к интеграции данных SCADA с существующими офисными приложениями, такими как электронные таблицы, системы управления базами данных, работа по истории данных, географической информационной системы (ГИС), системы распределения воды и систем моделирования.

1.3. Центральный Компьютера

Центральный хост-компьютер или главная станция чаще всего один компьютер или сеть компьютерных серверов, которые предоставляют интерфейс оператора человеко-машинный к системе SCADA. Компьютеры обрабатывают информацию, полученную от и отправляется на сайты РТУ и представить его человеческих операторов в форме, что операторы могут работать. Терминалы управления подключаются к центральному главному компьютеру с помощью LAN / WAN так, чтобы окна просмотра и связанные с ними данные могут быть отображены для операторов. Последние системы SCADA могут предложить компьютерной графики с высоким разрешением для отображения графического интерфейса пользователя или имитирующих экран сайта или воды сети водоснабжения в вопросе. Исторически сложилось, что производители SCADA предложили собственные аппаратные средства, операционные системы и программное обеспечение, которое в значительной степени несовместимы с SCADA других производителей системы. Расширение системы требуется дополнительный контракт с оригинальным SCADA поставщика. Принимающие компьютерные платформы характерно используют архитектуру UNIX на базе, и хост-компьютер сети был физически удален из любых офисных вычислений домена.

Однако, с увеличением использования персонального компьютера, компьютерных сетей стало обычным делом в офисе, и в результате, SCADA-системы теперь доступны, которые могут создавать сети с офисными на базе персональных компьютеров. В самом деле, многие из сегодняшних систем SCADA могут находиться на компьютерных серверах, которые идентичны с этими серверами и компьютерами, используемых для традиционных офисных приложений. Это открыло ряд возможностей для увязки SCADA систем на основе офисных приложений, таких как ГИС-систем, гидравлические программного обеспечения для моделирования, рисования систем управления, систем планирования работы и информационных баз данных.

1.4. Рабочие станции оператора и компоненты программного обеспечения

Операторские рабочие станции чаще всего компьютерные терминалы, которые объединены в сеть с центральным хост-компьютером SCADA. Центральный хост-компьютер выступает в качестве сервера для приложения SCADA, и терминалы оператора являются клиенты, запрос и отправить информацию на центральный хост-компьютер на основе запроса и действия операторов.

Важным аспектом любой системы SCADA является компьютерная программа, используемая в системе. Наиболее очевидным компонентом программного обеспечения является интерфейс оператора или (/ HMI MMI) пакет Man Machine Interface / человеко-машинный интерфейс; Тем не менее, программное обеспечение той или иной форме охватывает все уровни системы SCADA. В зависимости от размера и характера применения SCADA, программное обеспечение может быть важным пунктом затрат при разработке, поддержании и расширении системы SCADA. Когда программное обеспечение хорошо определены, разработаны, написаны, проверены и испытаны, успешная система SCADA, вероятно, будет выпускаться. Плохие выступления в любом из этих этапов проекта будет очень легко вызвать проект SCADA на провал.

Многие системы SCADA используют коммерческое проприетарное программное обеспечение, на котором разрабатывается система SCADA. Проприетарное программное обеспечение часто конфигурируется для конкретной аппаратной платформы и может не взаимодействовать с программным или аппаратным обеспечением производства конкурирующих производителей. Широкий спектр коммерческих вне-полки

(COTS) программные продукты также доступны, некоторые из которых могут удовлетворить требуемое приложение. Программное обеспечение КРОВАТКИ обычно является более гибким, и будет взаимодействовать с различными типами аппаратных средств и программного обеспечения. Как правило, в центре внимания патентованного программного обеспечения на процессы и функциональные возможности управления, в то время как программное обеспечение COTS подчеркивает совместимость с различными оборудованием и приборами. Поэтому важно, чтобы гарантировать, что адекватное планирование осуществляется для выбора программных систем, соответствующих любой новой системе SCADA.

Программные продукты, как правило, используемые в системе SCADA следующие:

- **Центральный хост-компьютер** Операционная система: Программное обеспечение для управления центральным узлом компьютерной техники. Программное обеспечение может быть основано на UNIX или других популярных операционных систем.

- **Оператор терминала** Операционная система: Программное обеспечение для управления центральным узлом компьютерной техники. Программное обеспечение, как правило, такой же, как центральный хост-компьютера под управлением операционной системы. Это программное обеспечение, а также, что для центрального хост-компьютера, как правило, способствует организации сети центрального хоста и операторских терминалов.

Центральный хост-компьютер приложений: Программное обеспечение, которое обрабатывает передачу, так и прием данных и от удаленных терминалов и центрального хоста. Программное обеспечение также предоставляет графический пользовательский интерфейс, который предлагает сайт мимических экраны, страницы сигнализации, трендов страниц и функции управления.

- **Терминал оператора приложения:** приложение, которое позволяет пользователям получать доступ к информации, имеющейся на центральном хост-приложения на компьютере. Это, как правило, является подмножеством программного обеспечения, используемого на центральном компьютере, на хост.

- **Драйверы протокола передачи данных:** Программное обеспечение, которое обычно базируется в пределах центрального узла и удаленных терминалов, и требуется контролировать перевод и интерпретацию данных между концами линий связи в системе. Драйверы протокола подготовки данных для использования либо в полевых устройствах или центрального конца хост-системы.

- **Программное обеспечение для управления сетью связи:** Программное обеспечение, необходимое для управления сетью связи и позволить самим сетям связи для мониторинга производительности и сбоев.

- **RTU программное обеспечение для автоматизации:** Программное обеспечение, которое позволяет инженерно-технического персонала настраивать и поддерживать приложения размещенную в удаленных терминалов (или ПЛК). Чаще всего это включает в себя локальное приложение автоматизации и любые задачи по обработке данных, которые выполняются в РТУ.

Предшествующие программные продукты обеспечивают строительные блоки для специфичных для приложения программное обеспечение, которое должно быть определено, разработаны, написаны, проверены и развернуты для каждого SCADA системы.

Контрольные вопросы:

1. Какие функции выполняет SCADA системы.
2. Сетевые возможности SCADA систем.
3. Применения SCADA систем.

Тема 14	Архитектура SCADA системы	
Технология проведения лекционного занятия		
Время 2 часа	Количество студентов 40-60 человек	
Вид занятия	Введение. Визуальная лекция	
План лекционного занятия	1. Первое поколение – Монолитный 2. Второе поколение – Распределенная 3. Третье поколение - подключенный к сети	
Цель учебного занятия: Формирование у студентов глубокого представления о целях и задачах химической науки и путях её дальнейшего развития		
Педагогические задачи:		Результаты обучения:
- Ознакомление с основными задачами системы SCADA - Ознакомление с историей развития автоматизации и поколений систем SCADA - основные задачи SCADA систем при управлении технологическими процессами.		Студент: 1. Имеет представление о целях и задачах системы SCADA 2. Излагает историю развития автоматизации и системы SCADA 3. Основные задачи контроллеров верхнего и нижнего уровня. 4. управления базами данных и составление отчетов по аварийным ситуациям.
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.1. Объявляет тему и план занятия, знакомит с достижениями развития автоматизации и системы SCADA	1.1. Слушает и записывает
2-этап Основной	2.1. Обращает внимание студентов на вопросы плана занятия и обращает внимание на основные понятия. Проводит опрос 2.2. Продолжает излагать лекционный материал. На примерах приводит и объясняет развития автоматизации и основные возможности системы SCADA 2.3. Обеспечивает студентом внимание и запись основных понятий и преимущества системы SCADA	2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают
3-этап Итоговый	3.1. Делает заключительные выводы лекционной темы и заостряет внимание студентов на основные вопросы 3.2. Отмечает активных студентов 3.3. Даст задание на самостоятельную подготовку.	3.1. Слушают и усваивают 3.2. Записывают задание

Лекция 14. Архитектура SCADA системы

План:

1. Первое поколение – Монолитный
2. Второе поколение – Распределенная
3. Третье поколение - подключенный к сети

SCADA системы развивались параллельно с ростом и усложнением современной вычислительной техники. В следующих разделах будет содержаться описание следующих трех поколений систем SCADA:

- Первое поколение – Монолитный
- Второе поколение – Распределенная
- Третье поколение - подключенный к сети

1. Монолитные SCADA системы

Когда системы SCADA были впервые разработаны, концепция вычислений в целом сосредоточена на системах "мэйнфреймов". Сети, как правило, не существует, и каждая централизованная система осталась одна. В результате, SCADA системы были автономные системы с практически без подключения к другим системам.

В глобальных сетях (WAN), которые были осуществлены для связи с удаленными оконечными устройствами (RTU) были разработаны с единственной целью в виду, что-общения с удаленными терминалами в поле и больше ничего. Кроме того, протоколы WAN используются сегодня в основном были неизвестны в то время.

Коммуникационные протоколы используемые на сетях SCADA были разработаны поставщиками оборудования RTU и часто были частной собственностью. Кроме того, эти протоколы, как правило, очень "тощий", не поддерживает практически никаких функциональных возможностей за пределами этого необходимого сканирования и контроля указывает в удаленном устройстве. Кроме того, как правило, не представляется возможным перемешивать других типов трафика данных с RTU сообщений по сети.

Возможность подключения к самой главной SCADA станции была очень ограничена поставщика системы. Подключение к мастеру, как правило, были сделаны на уровне шины с помощью патентованного адаптера или контроллера, подключенного к Центральный процессор (CPU) объединительной платы.

Избыточность в этих системах первого поколения, было достигнуто за счет использования двух одинаково оборудованных мэйнфреймовых систем, первичного и резервного копирования, соединенных на уровне шины. Основная функция резервной системы в том, чтобы следить за первичным и взять на себя в случае обнаруженной ошибки. Этот тип резервного режима означает, что мало или вообще не обработка была сделана на резервной системе. На рисунке 3.1 показана типичная SCADA архитектура первого поколения.

2. Второе поколение – Распределенная

DCSS используются для управления производственными системами в пределах того же географического положения для таких отраслей, как нефте- и газоперерабатывающих заводов, воды и очистки сточных вод, генерирующих электростанций, химических производственных заводов и перерабатывающих предприятий фармацевтической. Эти системы, как правило, управления технологическими процессами или дискретные системы управления частью. PCU использует централизованную цикл диспетчерского управления посредничать группу локализованных контроллеров, которые разделяют общие задачи проведения весь производственный процесс [6]. По модульности системы производства, PCU уменьшает влияние одной неисправности на систему в целом. В большинстве систем DCS сопряжена с корпоративной сетью, чтобы дать бизнес-операций вид продукции.

Этот DCS охватывает весь объект от производства нижнего уровня процессов до корпоративного или корпоративного уровня. В этом примере, задающий контроллер (управляющий сервер) осуществляет связь с его подчиненными через сеть управления. Надзорный орган направляет заданные точки, чтобы и запрашивает данные из распределенных контроллеров на местах. Распределенные контроллеры управляют их исполнительные механизмы процесса на основе команд сервера управления и датчиком обратной связи от датчиков процесса.

Устройства управления поля показали включают контроллер, контроллер процесса, контроллер одного цикла и контроллер машины. Одиночные контроллер контура интерфейсы датчиков и исполнительных механизмов с использованием точка-точка проводки, в то время как остальные три полевые устройства включают полевой шины сети для взаимодействия с датчиками и исполнительными механизмами процесса. Fieldbus сети устраняют необходимость точка-точка проводного соединения между контроллером и индивидуальными полевых датчиков и исполнительных механизмов. Кроме того, полевая шина обеспечивает большую функциональность по не зависящим, в том числе диагностики устройств на местах, и может выполнять алгоритмы управления в полевой шине, что позволяет избежать маршрутизации сигнала обратно в ПЛК для каждой операции управления. Стандартные протоколы промышленной коммуникации, разработанные. В дополнение к контурам управления супервизорного уровня и на местном уровне, промежуточные уровни управления также могут существовать. Например, в случае DCS, контролирующей

дискретную часть производственного предприятия, там может быть промежуточным супервизор уровня для каждой ячейки в растении. Этот руководитель будет включать в себя производственную ячейку, содержащую контроллер машины, который обрабатывает часть и контроллер робота, который обрабатывает сырьё и готовой продукции. Там может быть несколько из этих клеток, которые управляют контроллерами на уровне полей под основной DCS контура диспетчерского управления

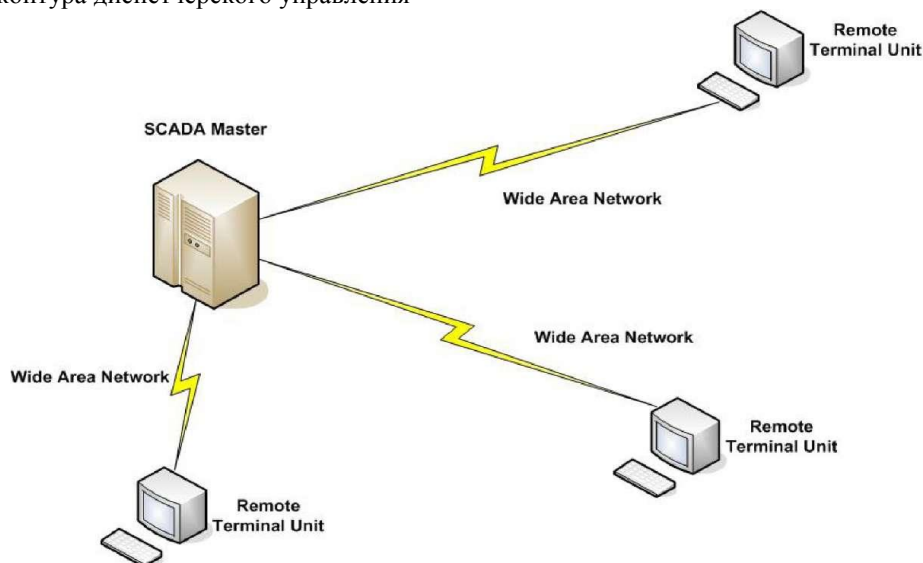
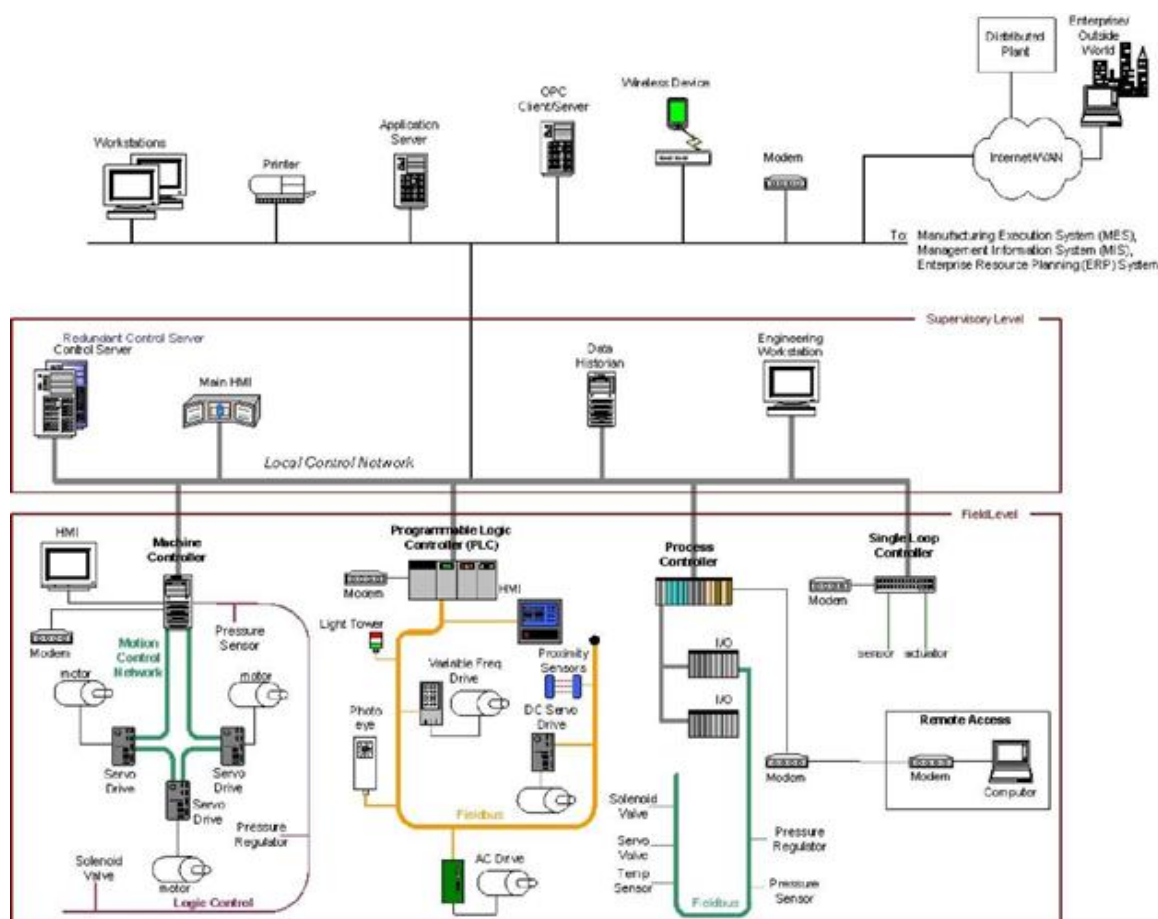


Рисунок 11.1. Первое поколение SCADA Архитектура [5]



DCS системы

Следующее поколение систем SCADA воспользовались разработками и улучшение системы миниатюризации и технологии локальной сети (LAN) к распределить обработку по нескольким системам. Несколько станций, каждая из которых имеет определенную функцию, были подключены к локальной сети и

обмениваются информацией друг с другом в режиме реального времени. Эти станции должны были, как правило, из мини-компьютерного класса, меньше и дешевле, чем их первые процессоры поколения.

Некоторые из этих распределенных станций служил в качестве коммуникационных процессоров, в первую очередь связи с полевыми устройствами, такими как удаленным терминалам. Некоторые служили в качестве интерфейсов оператора, обеспечивая интерфейс человек-машина (HMI) для системных операторов. Тем не менее другие служили в качестве процессоров расчета или серверов баз данных. Распределение индивидуальной системы SCADA функции по нескольким системам при условии большей вычислительной мощности для системы в целом, чем были бы доступны в одном процессоре. Сети, которые соединили эти отдельные системы, как правило, основаны на сетевых протоколов и не были способны достичь за пределы местной окружающей среды.

Некоторые из сетевых протоколов, которые были использованы были собственной природы, где поставщик создал свой собственный сетевой протокол или ее разновидность, а не вытягивать существующий с полки. Это позволило производителю оптимизировать свой протокол локальной сети для трафика в реальном времени, но оно ограничено (или эффективно устранены) соединение сети от других поставщиков к SCADA LAN. На рисунке 3.2 изображена типичная SCADA архитектуры второго поколения.

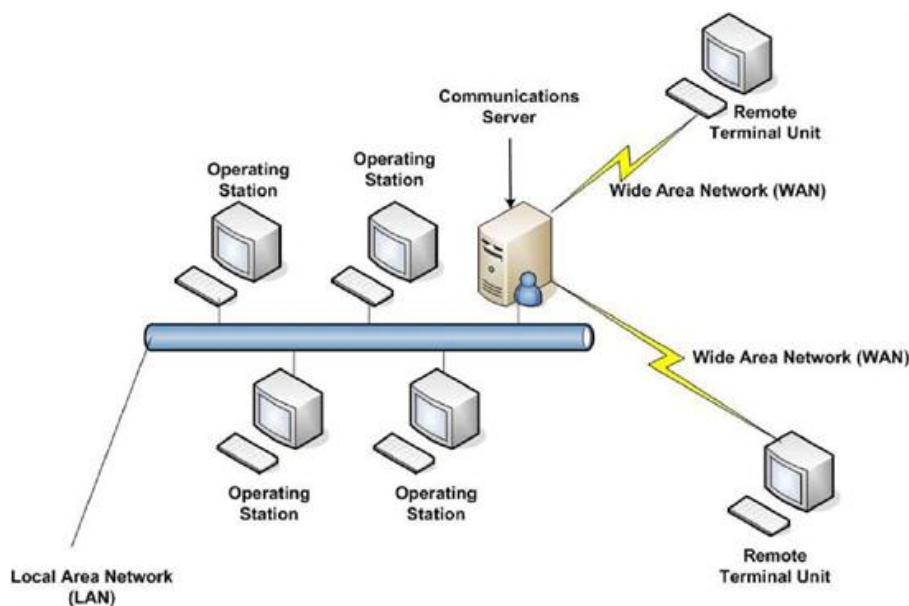


Рисунок 11.2: Второе поколение SCADA Архитектура [5]

Распределение функциональных возможностей системы через подключенных к сети систем служили не только для увеличения вычислительной мощности, но и повысить избыточность и надежность системы в целом. Вместо простой первичной / резервной схемы аварийного переключения, которая была использована во многих системах первого поколения, распределенная архитектура часто сохраняются все станции в локальной сети в оперативном режиме все время. Например, если станция HMI выйдет из строя, другая станция HMI может быть использована для работы системы, не дожидаясь отказа от первичной системы на вторичной обмотке.

WAN используется для связи с устройствами в этой области в основном не изменились развитием подключения к локальной сети между местными станциями на мастер-SCADA. Эти внешние сети связи по-прежнему ограничены протоколами RTU и не были доступны для других типов сетевого трафика.

Как и в случае с первым поколением систем, второе поколение систем SCADA также ограничено аппаратными средствами, программным обеспечением и периферийными устройствами, которые были предоставлены или, по крайней мере, выбранным поставщиком.

3. Третье поколение - подключенный к сети

Нынешнее поколение SCADA архитектуры ведущей станции тесно связана с проблемой второго поколения, причем основное различие в том, что открытой архитектуры системы, а не поставщика, контролируемой собственной среде. Есть еще несколько сетевых систем, разделяя функции мастер станции. Есть еще RTUs, использующие протоколы, которые поставщик несобственнической. Основное улучшение в третьем поколении является то, что открытия архитектуры системы, используя открытые стандарты и протоколы и делает возможным распределить функциональные возможности SCADA через глобальную сеть, а не только по локальной сети.

Открытые стандарты устраняют ряд ограничений предыдущих поколений систем SCADA. Использование отходящих готовых систем облегчает пользователю подключать сторонних периферийных

устройств (например, мониторы, принтеры, дисковые накопители, ленточные накопители и т.д.) к системе и / или сети.

Когда они переехали в "открытый" или "вне готовых" систем, производители SCADA постепенно вышла из девелоперского бизнеса аппаратного обеспечения. Эти производители смотрели поставщикам системы, таких как Compaq, Hewlett-Packard и Sun Microsystems для их опыт в разработке основных компьютерных платформ и операционных системное программное обеспечение. Это позволяет поставщикам SCADA сконцентрировать свое развитие в области, где они могут добавить конкретное значение в системе, то программное обеспечение SCADA мастер станции.

Terminal Unit Рисунок 11.3: Третье поколение SCADA системы подключенный к сети дистанционный терминал. Значительное улучшение в системах SCADA третьего поколения происходит от использования протоколов WAN, таких как Интернет-протокола (IP) для связи между ведущей станцией и коммуникационного оборудования. Это позволяет часть главной станции, который отвечает за связь с полевыми устройствами, чтобы быть отделены друг от ведущей станции "правильной" через глобальную сеть. Продавцы теперь производят RTU, которые могут взаимодействовать с главной станцией с помощью соединения Ethernet. Рисунок 3.3 представляет собой сетевую систему

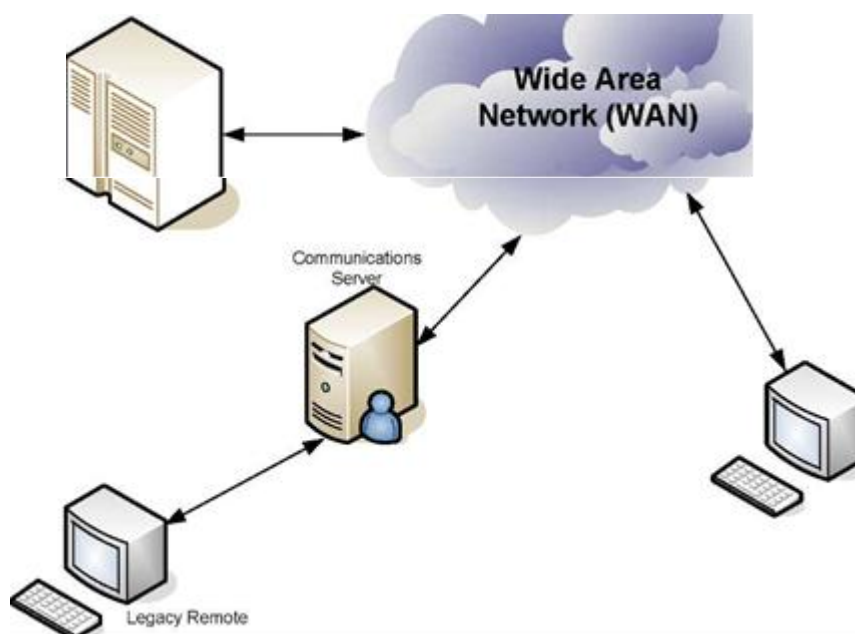


Рисунок 11.3: Третье поколение SCADA системы

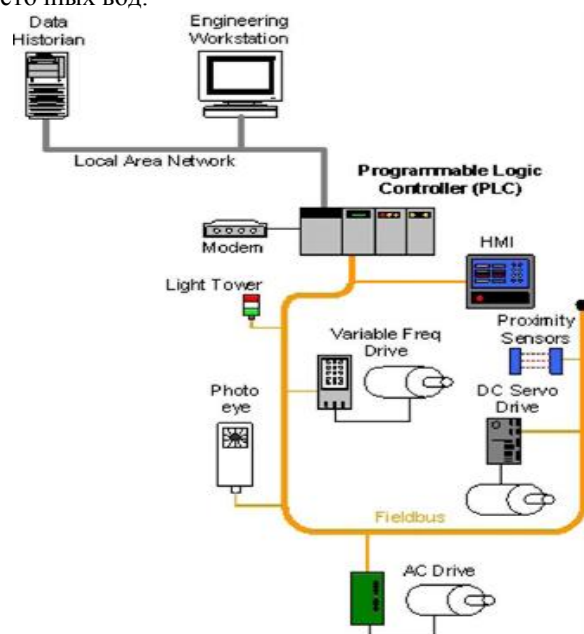
Еще одно преимущество вызвано распределением функциональных возможностей SCADA через глобальную сеть является то, что в случае стихийных бедствий живучести. Распределение обработки SCADA по локальной сети в системах второго поколения повышает надежность, но в случае полной потери объекта жилищного строительства мастер SCADA, вся система может быть потеряна, а также. Распределяя обработку через физические отдельные места, становится возможным построить систему SCADA, которая может выдержать полную потерю любого одного места. Для некоторых организаций, которые рассматривают SCADA как сверхкритическом функции, это реальная выгода. 2.6 Программируемые логические контроллеры

ПЛК используются в обоих SCADA и DCS систем как компоненты управления от общей иерархической системы, чтобы обеспечить локальное управление процессами с помощью управления с обратной связью, как описано в разделах выше. В случае систем SCADA, они обеспечивают такую же функциональность удаленных терминалов. При использовании в DCSS, ПЛК реализуются в виде локальных контроллеров в рамках надзорной схемы управления. ПЛК также реализованы в качестве основных компонентов в небольших конфигураций системы управления. ПЛК имеют настраиваемые пользователем память для хранения команд с целью реализации конкретных функций, таких как управление вводом / выводом, логики, времени, подсчета, три режима пропорционально-интегрально-дифференциального (ПИД) управления, связи, арифметика, и данных и файл обработка. На рисунке 2-8 показан контроль производственного процесса выполняется с помощью ПЛК по сети полевой шины. ПЛК доступен через программный интерфейс, расположенный на инженерной рабочей станции, а данные

Система управления PLC Пример реализации

Обе линии электропередачи и распределительные сетевые отрасли промышленности используют территориально распределенной технологии управления SCADA для управления тесно взаимосвязаны и динамические системы, состоящие из тысяч государственных и частных коммунальных и сельских кооперативов для подачи электроэнергии конечным потребителям. SCADA системы мониторинга и управления

распределением электроэнергии путем сбора данных и выдачи команд географически дистанционного управления полевых станций от централизованно. Системы SCADA также используются для мониторинга и контроля воды, нефти и распределение газа, в том числе трубопроводов, судов, грузовиков и железнодорожных систем, а также систем сбора сточных вод.



SCADA-системы и DCS часто связаны между собой. Это тот случай, для электрических центров управления мощностью и объектов генерации электрической энергии. Хотя операция выработка электроэнергии объект управляется DCS, ДКС должна взаимодействовать с системой SCADA для координации объемов производства с требованиями передачи и распределения электроэнергии.

Критическая инфраструктура США часто называют как "системы систем" из-за взаимозависимостей, которые существуют между ее различными отраслями промышленности, а также взаимосвязей между деловыми партнерами. Критические инфраструктуры тесно взаимосвязаны и зависят друг от друга в сложных отношениях, как физически, так и через множество информационных и коммуникационных технологий. Инцидент в одной инфраструктуре могут прямо или косвенно повлиять на другие инфраструктуры за счет каскадирования и эскалацией неудач.

Электричество часто считается одним из самых распространенных источников сбоев взаимозависимых критических инфраструктур. В качестве примера, ступенчатое разрушение может быть инициировано срыву сети радиорелейной связи, используемой для электрической системы SCADA передачи мощности. Отсутствие возможностей для проведения мониторинга и контроля может привести к большой энергоблок в автономный режим, событие, которое привело бы к потере мощности на подстанции передачи. Эта потеря может привести к серьёзной дисбаланс, вызвав собой каскадный через энергосистемы. Это может привести к значительным отключениям области, которые влияют на нефти и природного газа, нефтеперерабатывающих операций, системы очистки воды, системы сбора сточных вод, а также транспортных трубопроводных систем, которые полагаются на сетке для власти.

Контрольные вопросы

1. Особенности проектирования интегрированных систем проектирования и управления. SCADA- системы, их функции.
2. Программируемые логические контроллеры, PLC и PC-base контроллеры. Структура и технические характеристики.
3. Программно-аппаратные платформы SCADA-систем. Средства сетевой поддержки, системы визуализации ввода-вывода, графические возможности, отображение трендов, алармов, отчетов.
4. Этапы создания системы диспетчерского управления и контроля на основе SCADA-систем.
5. Примеры SCADA-систем: SCADA-система In-Touch "Wonderware" США;
6. Сравнительный анализ и тестирование SCADA-систем.
7. SCADA системы в ИСПУ мониторинга и диагностики технологических процессов и оборудования.
8. Разработка распределенной АСУТП в SCADA-системе TraceMode 6 как единого проекта.

Тема 15-16	ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ РАСПРЕДЕЛЕННОЙ ИЕРАРХИЧЕСКОЙ АСУТП НА БАЗЕ SCADA-СИСТЕМЫ TRACE MODE	
Технология проведения лекционного занятия		
Время 4 часа	Количество студентов 40-90 человек	
Вид занятия	Введения. Визуальная лекция	
План лекционного занятия	1 Идеология распределенных комплексов с применением SCADA-систем 1.1. Уровень контроллеров. 1.2. Оперативный уровень. 1.3 Административный уровень. 2. Архитектура TRACE MODE 3. Основные понятия системы TRACE MODE	
Цели лекционного занятия: изучение Идеология распределенных комплексов с применением SCADA-систем и изучение уровней.		
Педагогические задачи:		Результаты обучения:
- ознакомить студентов с архитектурой TRACE MODE, основных понятий системы TRACE MODE		Студент записывает в конспект и запоминает следующие: - Уровень контроллеров; - Оперативный уровень; - Административный уровень; - Архитектура TRACE MODE - Основные понятия системы TRACE MODE
Методика и техника обучения		-визуальная лекция
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-коллективно, группами
Условия обучения		-оборудованная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.1. Излагаются теме и план лекции, а также ожидаемые результаты	1.1. Слушает и записывает
2-этап Основной	2.1. Привлекает внимание студентов к вопросам плана, основным понятиям. 2.2. Продолжает излагать лекционный материал. 2.3. Рассказывает о распределенных комплексов с применением SCADA-систем 2.4. Привлекает внимание студентов к основным понятиям темы.	2.1. Отвечают на вопросы 2.2. Слушает и записывает
3-этап Итоговый	3.1. Делает вывод а теме,привлекает внимание студентов на основные вопросы; стимулирует активных студентов 3.2. Даёт задание на самостоятельную работе. Оценивает выполненные работы.	3.1. Слушают и усваивают 3.2. Записывают задание

Лекция 15-16.
Особенности проектирования распределенной
иерархической АСУТП на базе SCADA-системы Trace Mode

План:

1. Идеология распределенных комплексов с применением SCADA-систем.
2. Архитектура TRACE MODE.
3. Основные понятия системы TRACE MODE.

1. Идеология распределенных комплексов с применением SCADA-систем

SCADA-система (Supervisory Control And Data Acquisition) предназначена для **проектирования и эксплуатации** распределенных АСУ. Дословный перевод названия - диспетчерское управление и сбор данных. Однако в последних версиях её предназначение значительно расширилось. В частности, отечественная компания AdAstra выпустила 6-ю версию SCADA-системы **TRACE MODE**, объединяющую все уровни управления производством (рис. 1).

SCADA-система TRACE MODE (ТРЕЙС МОУД) имеет мощные средства для создания **распределенных иерархических АСУТП** (см рис. 47), включающих в себя до трех уровней иерархии: уровень контроллеров – нижний уровень; уровень операторских станций - верхний уровень; административный уровень. Деление на уровни иногда может быть весьма условным. В малых системах функции всех уровней часто реализуются на одной операторской станции. В больших системах на каждом уровне может быть выделена собственная иерархия. Тем не менее, в большинстве случаев такое деление правомерно. Необходимо заметить, что при разработке крупных сетевых систем, включающих в себя десятки узлов, лимитирующим элементом становятся не характеристики пакета по количеству одновременно работающих в сети узлов, а пропускная способность линий связи. Исполнительная система ТРЕЙС МОУД включает в себя мониторы (управляющие операционные системы), предназначенные для работы на всех уровнях систем управления (см. рис. 48).

1.1. Уровень контроллеров.

На этом уровне реализуется сбор данных от датчиков, а также непосредственное цифровое управление (НЦУ) в соответствии с принятыми законами (ПИД-, ПДД- и ШИМ- регуляторы, позиционный, нечеткий регуляторы и т.д.).

Для создания этого уровня предусмотрены мониторы: Микро МРВ (монитор реального времени), Микро МРВ Модем+, Микро МРВ GSM+. Первый из них предназначен для запуска в контроллерах, связанных с верхним уровнем по локальной сети или последовательному интерфейсу, второй – при связи по коммутируемым линиям, а третий – по GSM-сети. При использовании выделенных телефонных линий или радиоканалов следует применять первый монитор.

Эти мониторы не имеют графического интерфейса. Однако по математическим функциям они идентичны мониторам верхнего уровня, а также имеют ряд функций, необходимых для работы в контроллерах (например, поддержка сторожевого таймера).

1.2. Оперативный уровень.

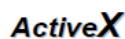
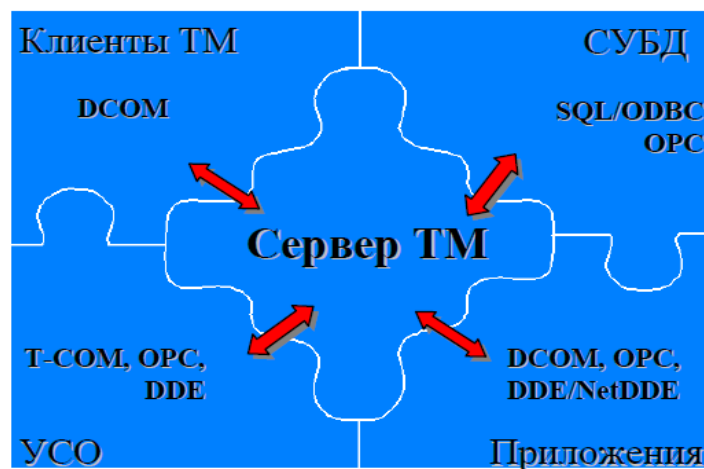
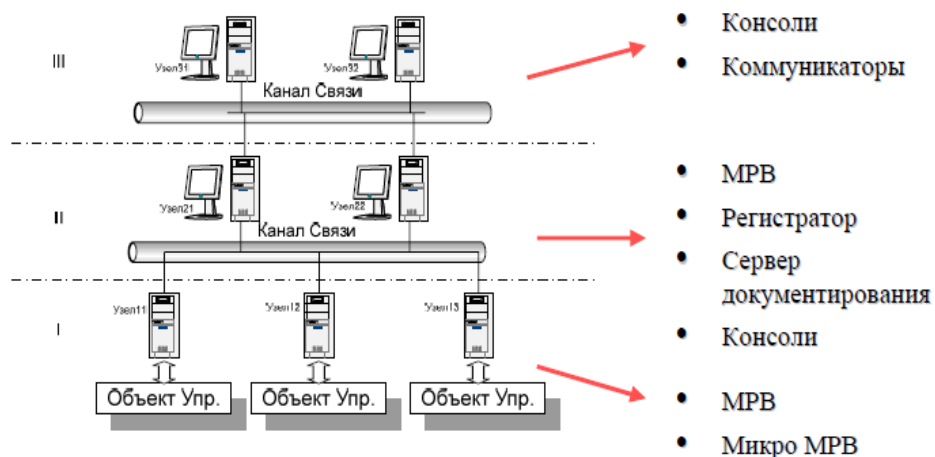
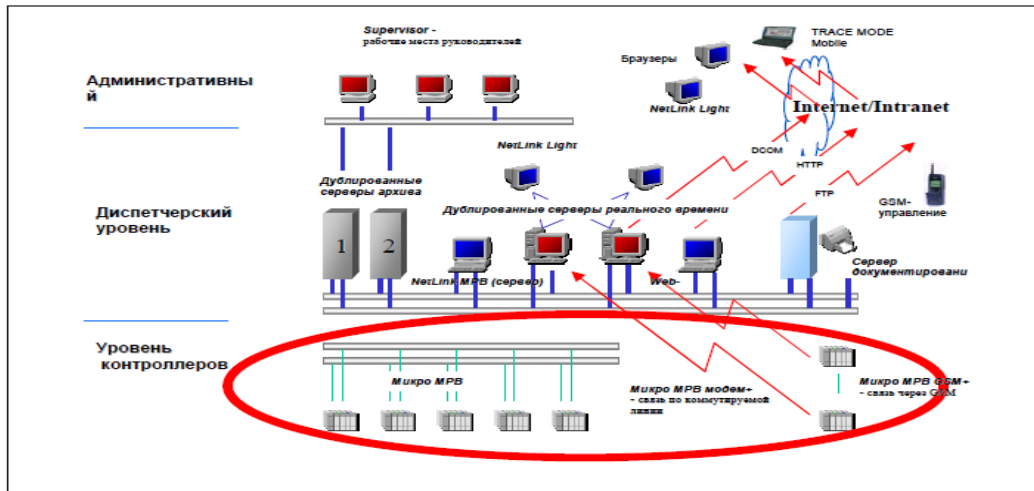
Для верхнего уровня АСУТП предусмотрены такие мониторы, как МРВ, NetLink МРВ, NetLink Light. Они позволяют создавать рабочие станции оперативного управляющего персонала.

МРВ может обмениваться данными с другими мониторами ТРЕЙС МОУД, а также с любыми контроллерами через встроенные протоколы или драйвер. Он запрашивает данные у нижнего уровня и передает ему команды управления. Полученные данные могут отображаться, архивироваться и передаваться другим приложениям WINDOWS по протоколам ODBC, OPC и DDE (см. рис. 49). NetLink МРВ – это сетевая рабочая станция. Этот монитор может обмениваться данными с операторскими станциями (по последовательному интерфейсу или локальной сети), а также с Микро МРВ, работающими в PC-based контроллерах. По функциям визуализации, архивирования, связи с базами данных и документирования NetLink МРВ аналогичен МРВ. В отличие от МРВ, в нем блокированы поддержка плат УСО, обмен с драйвером, обмен по встроенным протоколам MODBUS и DCS, а также клиентские функции OPC и DDE. NetLink Light – это сетевой графический терминал. Он не имеет своего сервера математической обработки, а связывается с сервером МРВ или NetLink МРВ, запущенным на другом компьютере. NetLink Light позволяет создавать дополнительные рабочие места оператора.

1.3. Административный уровень.

Задачей данного уровня управления является контроль текущего состояния производственных процессов и анализ функционирования производства по архивным данным. Для решения задач данного уровня предусмотрен монитор SUPERVISOR. Он является специализированной графической консолью, которая может подключаться к серверу матобработки МРВ, NetLink МРВ или ГР. В первых двух случаях просматривается локальный СПАД архив, а в последнем - глобальный архив. Кроме того, SUPERVISOR можно переключить в режим реального времени. В этом случае он работает как консоль NetLink Light, и может использоваться для управления процессом. При работе с архивами SUPERVISOR реализует следующие функции: отображение последних изменений значений каналов; просмотр архивов в режиме PLAYBACK; просмотр на заданное архивное время с пошаговым переходом по времени.

До тех пор, пока речь идет о связи между компонентами одного узла, не возникает вопрос об аппаратно/программном интерфейсе, который должен быть задействован для обеспечения связи. В этом случае достаточно выполнить конфигурирование свойств связи/вызовов компонентов. Если взаимодействующие компоненты относятся к разным узлам, интерфейс связи, как правило, должен быть указан и сконфигурирован.



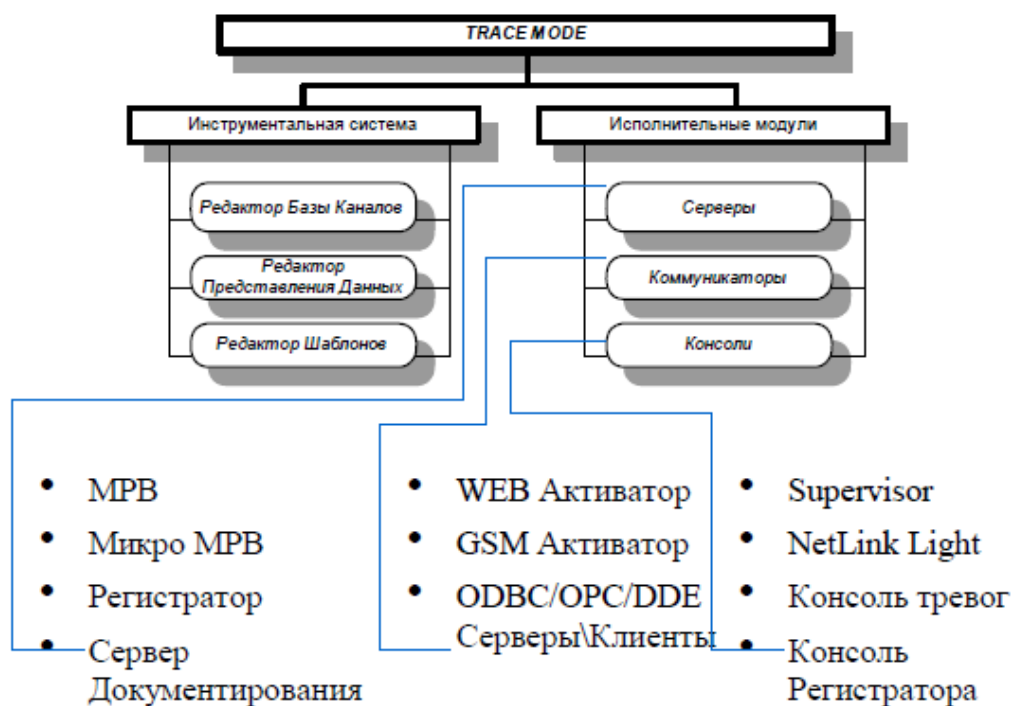
Последняя разработка TRACE MODE версии 6 является интегрированной системой, позволяющей решать задачи автоматизации ТП и управления производством АСУТП. Теперь специалисты по АСУТП могут решать задачи АСУТП на основе привычного и понятийного аппарата. Благодаря использованию в своей оболочке T-FACTORY MES-системы управления производством стало возможным автоматизировать работы по планированию, контролю исполнения, сбору статистики и анализу производственного цикла предприятия.

2. Архитектура TRACE MODE

Все программы, входящие в ТРЕЙС МОУД, подразделяются на две группы (см. рис. 50): инструментальную систему разработки и исполнительные модули (runtime). Как видно на рис. 50, инструментальная система разработки содержит три редактора [76]: редактор базы каналов, редактор представления данных, редактор шаблонов.

В редакторе базы каналов создается математическая основа системы управления: описываются конфигурации всех рабочих станций, контроллеров и УСО, настраиваются информационные потоки между ними. Здесь же описываются входные и выходные сигналы и их связь с устройствами сбора данных и управления; задаются периоды опроса или формирования сигналов, настраиваются законы первичной обработки и управления, технологические границы, программы обработки данных и управления, осуществляется архивирование технологических параметров, сетевой обмен, а также решаются некоторые другие задачи.

Результатами работы в этом редакторе являются математическая и информационная структуры проекта АСУТП, которые включают в себя набор баз каналов и файлов конфигурации для всех контроллеров и операторских станций (узлов) проекта, а также файл конфигурации всего проекта с расширением *cmf*. Все остальные файлы проекта хранятся в рабочей директории в каталоге, имя которого совпадает с именем файла конфигурации.



В редакторе представления данных разрабатывается графическая часть проекта системы управления. Сначала создается статичный рисунок технологического объекта, а затем поверх него размещаются динамические формы отображения и управления. Среди этих форм присутствуют такие, как поля вывода числовых значений, графики, гистограммы, кнопки, области ввода значений и перехода к другим графическим фрагментам и т.д.

Кроме стандартных форм отображения, ТРЕЙС МОУД позволяет вставлять в проекты графические формы представления данных или управления, разработанные пользователями. Для этого можно использовать стандартный механизм ActiveX (см. рис. 49).

Все формы отображения информации, управления и анимационные эффекты связываются с информационной структурой, разработанной в редакторе базы каналов.

Для разработки шаблонов документов в состав инструментальной системы включен редактор шаблонов.

Исполнительная система ТРЕЙС МОУД включает в себя исполнительные модули (мониторы, MPB) - программные модули различного назначения, под управлением которых в реальном времени выполняются составные части проекта, размещаемые на отдельных компьютерах или в контроллерах, предназначенные для работы на всех уровнях систем управления, о которых говорилось выше.

Существует ряд программных модулей, назначение которых четко не привязано к функциям одного из перечисленных уровней систем управления. К таким модулям относятся (см. рис. 50): глобальный регистратор; сервер документирования; Web-активатор; GSM-активатор. Они могут использоваться для создания как оперативного, так и административного уровней систем управления [70].

Глобальный регистратор служит для обеспечения надежного хранения архивов ТП. Он архивирует данные, посылаемые ему по сети мониторами реального времени (64 000 параметров с дискретностью 0,001 с), обеспечивает автоматическое восстановление данных после сбоя, а также может передавать архивные данные для просмотра мониторами SUPERVISOR. Глобальный регистратор может также выступать как OPC- и DDE-сервер и поддерживает обмен с базами данных через ODBC.

Для документирования технологической информации в ТРЕЙС МОУД предусмотрен модуль - сервер документирования. Документирование осуществляется по шаблонам, которые создаются в редакторе шаблонов. Время или условие генерирования документа, имя файла шаблона, а также направление вывода документа описываются в программах документирования - сценариях.

Подготовка отчетов (документов) чаще всего привязывается к астрономическому времени. Например, они могут генерироваться один раз в час, один раз в сутки, один раз в месяц или один раз в десять минут. Кроме того, можно установить режим подготовки документа один раз в смену и затем описать разбивку суток на смены.

Сервер документирования NetLink Light используется для решения задачи документирования технологической информации. Он по команде МРВ, собственному сценарию или по команде оператора интерпретирует созданные заранее шаблоны, запрашивает у МРВ необходимые данные и формирует по ним документы. Эти документы могут быть распечатаны на принтере, отправлены по E-mail или опубликованы на Web-сервере.

Утилита консоль тревог позволяет просматривать отчет тревог разных МРВ одного проекта. Для каждого просматриваемого отчета тревог создается отдельное окно. В него можно выводить информацию из файла отчета тревог или сообщения, формируемые МРВ. Любая рабочая станция системы ТРЕЙС МОУД может выступать в качестве Web-сервера, что позволяет управлять технологическим процессом через Интернет (Internet) [70]. На удаленном компьютере необходимо иметь только доступ к Интернет и Web-браузер. Для реализации данного режима предназначен модуль Web-активатор, который используется в качестве www-шлюза для локальных систем АСУ ТП на базе ТРЕЙС МОУД или для придания функций Web-сервера мониторам реального времени. Использование Web-активатора позволяет быстро превратить существующие АСУ ТП и АСУП в Internet/Intranet-системы без переделки баз данных реального времени (баз каналов).

Доступ к данным реального времени через Web-активатор осуществляется при помощи обыкновенного браузера, работающего под любой операционной системой, позволяющей запуск виртуальной Java-машины. Информация о технологическом процессе представляется пользователю в виде анимированных мнемосхем, трендов и таблиц.

Связь с серверами реального времени ТРЕЙС МОУД может осуществляться практически любыми доступными средствами, например через сотовую сеть стандарта GSM, инфракрасный порт, сеть на основе интерфейса RS-232/485 или модем с использованием высоконадежного протокола TCP/IP. Можно осуществлять подключение и непосредственно через Internet. Для этого достаточно войти в Internet и набрать IP-адрес сервера ТРЕЙС МОУД - подключение произойдет автоматически.

Для доступа к данным пользователю достаточно набрать Web-адрес активатора и ввести пароль, тогда весь проект загружается в удаленный компьютер в виде Java-апплета [70]. Использование стандартного языка Java при написании апплетов позволяет реализовать на удаленных компьютерах не только Windows, но и другие операционные системы, например Unix, Linux, Mac OS и т.д., а также ОС, использующиеся в карманных РС. Проект ТРЕЙС МОУД поступает к пользователю в виде Java-апплета, объем которого не превышает 300 Кбайт, что дает возможность использовать Web-активатор в сетях с низким качеством связи. Достоинством технологии Java является также повышенная безопасность.

При использовании Web-активатора не требуется установка Web-серверов других производителей (например, MS IE), что выгодно отличает эту программу от решений, примененных в других SCADA. Для обеспечения мобильных пользователей АСУ оперативной информацией в режиме реального времени на базе ТРЕЙС МОУД разработан программный продукт - GSM-активатор. Он предназначен для дистанционного мониторинга и управления технологическими процессами, а также для получения оперативной технико-экономической информации при помощи сверхпортативных компьютеров handheld PC.

В реальном времени GSM-активатор может принимать информацию от 64 000 датчиков, осуществлять супервизорное управление, получать технико-экономическую информацию из баз данных через сервер, использующий стандартные интерфейсы SQL/ODBC. OPC, DDE (см. рис. 49) и т.д. Вся поступающая информация отображается графически в виде анимированных мнемосхем и трендов.

GSM-активатор относится к новому классу систем оперативного управления, отражающих тенденцию к миниатюризации и автономизации компьютерных систем. GSM-активатор может быть использован в качестве персональной информационной системы руководителя. К GSM-активатору проявляют интерес нефтяные компании, электрические и тепловые сети PAO ЕЭС и PAO ГАЗПРОМ, коммунальные и другие службы, управляющие пространственно распределенными объектами.

GSM-активатор пригоден также к применению в охранных службах: получение в реальном времени информации о состоянии охраняемого объекта может стать основой успеха операции группы быстрого реагирования, вызванной по тревоге. Отметим, что в последней версии TRACE MODE 6 все редакторы системы вызываются из одной программы - Интегрированной среды разработки (ИС). ИС – единая программная оболочка, содержащая все необходимые средства для разработки проекта.

Все переменные проекта, к чему бы они не относились – к контроллеру, к операторской станции, к управлению техобслуживанием или производством **хранятся в единой базе данных проекта**. Единая база проекта устраняет лишнюю работу проектировщика по созданию, поддержке и взаимной увязке во многом одинаковых баз переменных контроллеров и ПК, характерную для систем предыдущего поколения.

Логическая структура проекта полностью отделена от аппаратной части. Благодаря **единому пространству распределенных переменных**, переменные из разных узлов могут связываться между собой также легко, как и в пределах одного узла, любые изменения, вносимые в объект, **автоматически применяются везде**, где он был задействован. В целях пояснения особенностей и принципов работы SCADA-системы воспользуемся некоторыми справочными материалами предыдущих версий.

3. Основные понятия системы TRACE MODE

Определения. ПРОЕКТ системы управления – это совокупность всех математических и графических элементов системы, функционирующих на различных операторских станциях и контроллерах одной АСУ ТП, объединенных информационными связями и единой системой архивирования. Проект может быть масштабным (сотни узлов), а может включать в себя только **один** контроллер или одну операторскую станцию. Под **проектом в TRACE MODE 6** понимается вся совокупность данных и алгоритмов

функционирования распределенной АСУ (АСУТП и/или TFACTORY), заданных средствами TRACE MODE.

Итогом разработки проекта является создание файлов, содержащих информацию об алгоритмах работы АСУ. Эти файлы размещаются на аппаратных средствах (компьютерах и контроллерах) и выполняются под управлением **исполнительных модулей TRACE MODE**. Составная часть проекта, размещаемая на отдельном компьютере или в контроллере и выполняемая под управлением одного или нескольких исполнительных модулей TRACE MODE, называется **узлом проекта**.

УЗЕЛ – любое устройство в рамках проекта, в котором запущено программное обеспечение **TRACE MODE**, реализующее серверные функции. Это может быть контроллер, операторская станция или архивная станция. В проекте **не может быть более 128** узлов. В общем случае размещение узла на том же аппаратном средстве, на котором он должен исполняться под управлением монитора, не является обязательным – мониторы могут загружать узлы с удаленных аппаратных средств.

БАЗА КАНАЛОВ – совокупность **всех** каналов, математических объектов, **FBD** – программ и **PL** – программ, созданных для каждого конкретного узла. **ОБЪЕКТ БАЗЫ КАНАЛОВ** – совокупность любых каналов, которой приписан определенный набор свойств и атрибутов. Среди последних можно назвать имя, графический идентификатор, флаг подчинения: родитель, потомок. Оформленные группы каналов могут быть подчинены друг другу и создавать таким образом иерархические структуры.

ДРАЙВЕРЫ обмена – драйверы, используемые мониторами TRACE MODE для взаимодействия с устройствами, протоколы обмена с которыми не встроены в мониторы.

Каналы. КАНАЛ (базовое понятие системы) – это структура, состоящая из набора **переменных и процедур**, имеющая **настройки** на внешние данные, **идентификаторы** и **период** пересчета ее переменных. Идентификаторами канала являются: **имя, комментарий** и **кодировка**. Например, имя канала, связанного с пятым каналом платы аналогового ввода, расположенной в первом посадочном месте контроллера, будет **AI-pp01-0005**. Кроме того, каждый канал имеет числовой идентификатор, используемый внутри системы для ссылок на этот канал. Среди переменных канала выделяются четыре основных значения: **входное (In)**, **аппаратное (A)**, **реальное (R)** и **выходное (Q)**. С помощью **настроек** входное значение канала связывается с источником данных, а выходное – с приемником.

В зависимости от направления движения информации, т.е. от внешних источников (данные с контроллеров, УСО или системные переменные) в канал или наоборот, каналы подразделяются на входные (тип **INPUT**) (см. рис. 51) и выходные (тип **OUTPUT**) (см. рис. 52).

Входной канал (см. рис. 51) запрашивает данные у внешнего источника (контроллер, другой МРВ и пр.) или значение системных переменных (счетчик ошибок, длина архива и пр.).

Полученное значение поступает на вход канала и далее пересчитывается в аппаратное и реальное значения. **Аппаратное** значение у каналов типа **INPUT** формируется **масштабированием (логической обработкой)** для дискретных каналов) входных значений. Используемые процедуры обеспечивают первичную обработку данных (исправление ошибок датчиков, масштабирование, коррекция температуры холодных спаев термопар и т.д.). Выходные значения в в каналах типа **INPUT** не используются.

Выходной канал (см. рис. 52) передает данные приемнику. Приемник может быть внешним (значение переменной в контроллере, в другом МРВ и пр.) или внутренним - одна из системных переменных (номер проигрываемого звукового файла, номер экрана, выводимого на монитор, и пр.). И внешние и внутренние приемники данных связываются с выходными значениями каналов. У каналов типа **OUTPUT** их входное значение формируется одним из следующих способов: процедурой **управление** данного канала; процедурами

управление или **трансляция** других каналов; метапрограммой на языке Техно II; каналом удаленного узла (например, по сети); оператором с помощью управляющих графических форм. У каналов типа **OUTPUT** аппаратное значение получается из реального процедурой **трансляция**. **Аппаратные** значения каналов имеют такое название, поскольку в них удобно получать величины унифицированных сигналов, с которыми работает аппаратура ввода/вывода (4-20 мА, 0-10 В и пр.). **Реальные** значения предназначены для хранения значений контролируемых параметров или сигналов управления в реальных единицах (например, кг/час, оС, % и пр.). **Выходное** значение определено только для каналов типа **OUTPUT**. Оно пересчитывается из аппаратного значения.

Данные из внешних устройств записываются в каналы, данные из каналов посылаются на внешние устройства. В каналы оператор заносит управляющие сигналы. Значения из каналов записываются в архивы, операторские отчеты и т.п. В каналах осуществляется преобразование данных. Меняя значения на системных каналах, можно управлять выводимой на экран информацией, звуковыми сигналами и т.д., т.е. всей системой.

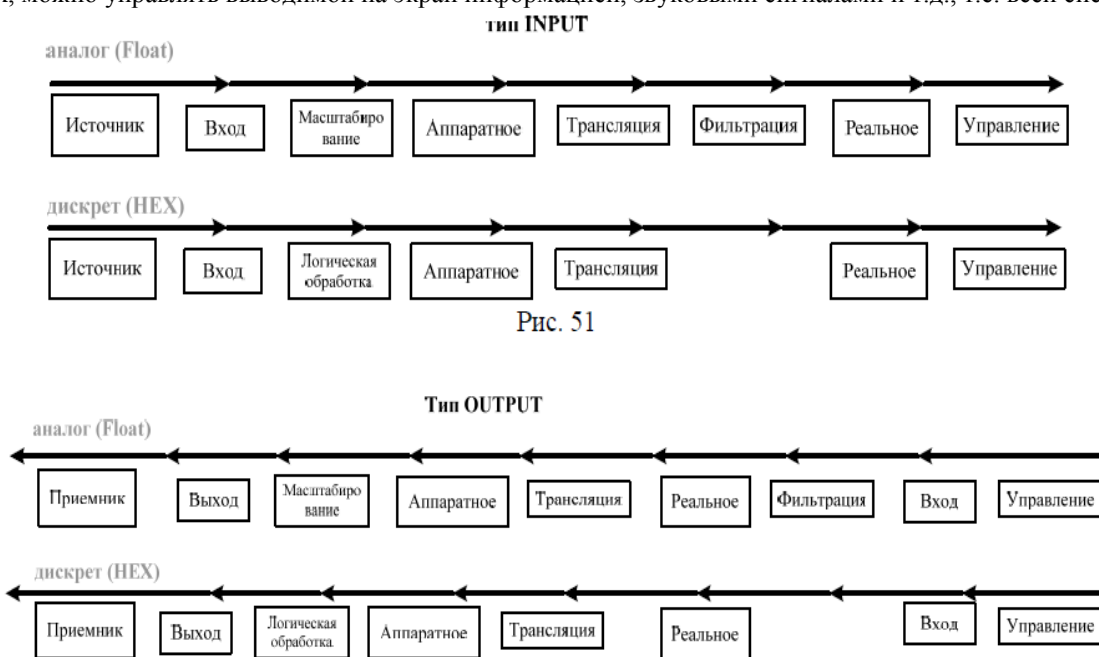


Рис. 51

Процедуры.

Входное значение канала преобразуется в аппаратное, реальное и выходное с помощью **процедур**.

Процедурами канала являются:

- **масштабирование** (умножение и смещение);
- **фильтрация** (подавление пиков, апертюра и сглаживание);
- **логическая обработка** (предустановка, инверсия, контроль сочетаемости);
- **трансляция** (вызов внешней программы);
- **управление** (вызов внешней программы).

Порядок следований и содержание процедур может меняться в зависимости от типа канала (входной или выходной, аналоговый или дискретный). Процедура **масштабирование** используется только в каналах, работающих с аналоговыми переменными. Она включает в себя две операции: **умножение** и **смещение**. Последовательность этих операций меняется в зависимости от типа канала:

- у каналов типа **INPUT** входное значение умножается на заданный множитель и к полученному результату добавляется величина смещения, результат присваивается аппаратному значению канала;

- у каналов типа **OUTPUT** к аппаратному значению добавляется величина смещения, затем эта сумма умножается на заданный множитель, а результат присваивается выходному значению канала.

Процедура **трансляция** определена для всех каналов независимо от их типа и вида представления. У входных каналов процедура трансляции преобразует **аппаратное** значение в **реальное**, а для выходных - наоборот. Для этого вызывается FBD-программа. Вызываемая программа выбирается при настройке процедуры. При настройке процедуры входные и выходные аргументы выбранной программы связываются с атрибутами текущего канала, а также любых других каналов из текущей базы. Поэтому процедура трансляции одного канала может также использоваться для формирования значений других каналов.

Рассмотрим пример использования процедуры **трансляция** [76]. Необходимо измерять расход вещества, транспортируемого по трубопроводу, и интегрировать его по времени для расчета технико-экономических показателей производства. На трубопроводе установлен датчик скорости потока.

Решение. Для решения этой задачи потребуется один канал типа

INPUT. Его аппаратное значение необходимо связать с данными, поступающими от датчика скорости потока (адресация каналов будетописана в следующем разделе), настроить коэффициенты масштабирования и дрейфа нуля исходя из геометрических характеристик трубопровода и физических свойств потока для перевода измеренной скорости в величину расхода. Далее следует создать FBD-программу, в которой будет выполняться интегрирование входной величины и результат записываться в выходную переменную. Затем эту программу надо установить для процедуры трансляция данного канала (написание программ для процедур канала будет рассмотрено ниже). При такой конфигурации во **входном** значении канала будет находиться информация о скорости потока, в **аппаратном** - величина расхода вещества, а в **реальном** – количество прошедшего по трубе вещества.

Набор процедур в канале зависит от формата данных. Каналы, работающие с аналоговыми переменными, используют процедуры **масштабирование, трансляция, фильтрация** и **управление**. В каналах обрабатывающих дискретные параметры используются **логическая обработка, трансляция** и **управление**.

Фильтрация - процедура, которая присутствует только у аналоговых каналов. Набор выполняемых ею операций отличается для входных и выходных каналов. У каналов типа **INPUT** фильтрация выполняется после процедуры трансляции до формирования реального значения. Фильтрация включает в себя следующие операции: подавление случайных всплесков в тракте измерения; подавление малых колебаний значения канала; экспоненциальное сглаживание; контроль шкалы – отслеживание выхода реального значения канала за установленные границы шкалы. У каналов типа **OUTPUT** данная процедура формирует реальное значение по входному значению. При этом выполняются следующие операции: ограничение скорости изменения реального значения; подавление малых колебаний значения канала; экспоненциальное сглаживание; контроль шкалы – обрезание величины управляющего воздействия до границ шкалы канала.

Управление – процедура, которая определена для всех каналов и реализует функцию управления. С ее помощью можно вызвать **FBD**- программу, в которой можно запрограммировать требуемые алгоритмы управления. В качестве аргументов программе могут передаваться значения и атрибуты любых каналов из текущей базы.

Эти аргументы могут быть как входными, так и формируемыми. Формально процедура управление связана с каналом только циклом пересчета. Она может вообще никак не участвовать в формировании его значений, а управлять другими каналами. Такая ситуация часто наблюдается при использовании процедуры **управление** на каналах типа **INPUT**.

Кроме основных значений канал имеет дополнительные переменные: шесть границ, гистерезис, настройки процедур обработки, начальные параметры, флаги архивирования и др. Переменные, настройки и идентификаторы канала образуют список его **атрибутов**.

Часть из них задается в редакторе базы каналов и не может быть изменена в реальном времени. Другие могут иметь начальные значения и доступны для изменения.

5.3.4. Подтип канала. Подтип канала указывает класс источников или приемников данных, с которыми будет связываться канал. Для каналов типа **INPUT** подтип характеризует получаемую ими информацию (**АНАЛОГ** - значение АЦП, считанное с платы УСО, **СИСТЕМНЫЙ** - состояние системы, **СВЯЗЬ** - данные с удаленных узлов проекта и пр.). Каналы **OUTPUT** имеют тот же набор подтипов, что и каналы **INPUT**. Однако для них подтип определяет класс приемников, а не источников данных (**АНАЛОГ** - значение ЦАП, **СИСТЕМНЫЙ** - состояние системы, **СВЯЗЬ** - значения управляемых каналов на удаленных узлах проекта и пр.). Всего существует шестнадцать подтипов каналов. Все они могут задаваться как для входных, так и для выходных каналов. Подтип канала задает класс источников или приемников данных. Кроме того, подтип канала определяет также количество его дополнительных настроек.

Уточнение источника или приемника в рамках заданного подтипом класса осуществляется с помощью дополнения к подтипу. Последний уровень адресации источника или приемника данных осуществляется с помощью настроек канала.

Пример 8. Пусть надо настроить канал для запроса данных от удаленного МРВ по протоколу **M_LINK**. Тип канала в этом случае следует установить **INPUT**, поскольку данные запрашиваются. Для обмена данными с удаленными мониторами **ТРЕЙС** **МОУД** по любой линии связи используется подтип каналов **СВЯЗЬ**. Дополнение к подтипу должно быть задано **InM_Link**. Такой канал будет иметь пять настроек. В них будет указываться номер последовательного порта, имя удаленного монитора, название объекта базы каналов, имя канала и его атрибут.

Лекция 17-18. Обмен данными в SCADA системе TRACE MODE

План:

1. Обмен Trace Mode с базами данных через механизмы ODBC.
2. Обмен данными в Trace Mode через механизмы OPC.
3. СДКУ системами вентиляции и теплоснабжением

1. Обмен Trace Mode с базами данных через механизмы ODBC.

Если речь идет о связи между компонентами одного узла, то вопрос об аппаратно-программном интерфейсе, который должен быть задействован для обеспечения связи, не возникает. В этом случае достаточно выполнить конфигурирование свойств связи/вызов компонентов. Если взаимодействующие компоненты относятся к разным узлам, интерфейс связи, как правило, должен быть указан и сконфигурирован.

Мониторы реального времени ТРЕЙС МОУД могут обмениваться данными по следующим линиям: локальная сеть; последовательный интерфейс RS-232, **RS-485**, RS-422; радиоканал; выделенная телефонная линия; коммутируемые телефонные линии; сети GSM. По этим носителям необходимо организовать информационные потоки всех уровней системы управления. При этом могут реализоваться как вертикальные связи (между уровнями), так и горизонтальные (между узлами одного уровня). Например, при задании связи двух каналов разных узлов по RS необходимо создать в узлах компоненты COM-порт, задать для них необходимые параметры и указать для канала-приемника используемый интерфейс связи.

Последовательный интерфейс.

Обмен по всем линиям, **кроме локальной сети**, реализуется через последовательный порт по протоколу **M-Link**. Узлы в сети **M-Link** неравноправны: один имеет статус **Master**, а остальные – **Slave**. Такие сети следует использовать для связи между операторскими станциями и контроллерами. Монитор со статусом **Master** является активным. Он посылает команды управления и запросы на передачу информации.

Монитор со статусом **Slave** принимает посланные ему команды и передает запрошенные данные. Команды управления содержат указания на изменение значений атрибутов каналов удаленного узла.

Таким образом, запросы, посылаемые монитором со статусом **Master**, могут быть двух типов:

1) запрос данных (используется для получения значений каналов или другой информации от монитора со статусом **Slave**);

2) запрос на изменение (используется для изменения значений атрибутов каналов на удаленном мониторе).

В запросах на изменение передаются новые значения корректируемых атрибутов удаленной базы.

Следует отметить, что в одной сети M-Link не может быть двух мониторов, для которых установлен статус Master. Чтобы один монитор выступал как Master, и как Slave, надо создать **параллельные** сети, используя при этом по два последовательных порта на каждом узле. Тогда два монитора смогут работать в режиме Master.

Обмен по протоколу M-Link.

Для обмена данными между мониторами ТРЕЙС МОУД по **последовательному интерфейсу** используется протокол M-Link. Он применяется для обмена по интерфейсам RS-232, RS-485, RS-422, радиоканалу, коммутируемым телефонным линиям и GSM сети.

Используя протокол M-Link, в рамках ТРЕЙС МОУД можно создавать **сетевые комплексы** на базе последовательного интерфейса **RS-485**. Такие комплексы могут включать в себя до 128 узлов (контроллеров и операторских станций). При этом связь может осуществляться по нескольким последовательным портам. **Для связи двух мониторов** можно использовать **интерфейс RS-232**. Чтобы связаться с несколькими удаленными узлами поэтому интерфейсу, нужно иметь соответствующее количество последовательных портов. Это позволяет организовать связь типа «звезда». Такая конфигурация может потребовать дополнительных затрат на многоканальные платы. Однако она позволяет быстрее передавать данные за счет распараллеливания обмена с разными удаленными узлами. ТРЕЙС МОУД поддерживает обмен одновременно по 32 последовательным портам.

Для **связи сильно разнесенных в пространстве мониторов** можно использовать радиоканал, выделенные или коммутируемые телефонные линии. В этих случаях нужны дополнительные устройства – модемы. Они согласуют электрические характеристики последовательных портов и используемой среды передачи.

2. Обмен данными в Trace Mode через механизмы OPC.

Одним из самых перспективных стандартов обмена данными между приложениями WINDOWS при создании систем управления является механизм OPC. OPC (OLE for Process Control) – стандартизованные интерфейсы для Microsoft технологии COM, предназначенные для применения в области автоматизации управления технологическими процессами. Стандарт OPC разработан международным фондом OPC Foundation, который был создан фирмами Fisher Rosemount, Intellution, Intuitive Technology, Opto22, Rockwell и [Siemens](#) в 1995 г. В 1996 г. появилась первая версия спецификации OPC.

OPC в настоящее время является стандартом, который признан разработчиками, системными интеграторами и пользователями АСУ ТП. Сегодня практически все производители программного и аппаратного обеспечения АСУ ТП разрабатывают продукты, соответствующие этому стандарту.

За последние несколько лет OPC серверы полностью вытеснили DDE (Dynamic Data Exchange) серверы и специализированные драйверы для аппаратных средств автоматизации. DDE – самый старый (время появления — 1989-1991 гг.) и очень медленный способ динамического обмена данными между Windows приложениями, был со временем заменен (преобразован) в OLE (Object Linking and Embedding). OLE первоначально и до середины 90-х годов использовался исключительно Microsoft для обмена данными между ее офисными приложениями. Во время разработки Windows NT появилась технология [DCOM](#) (Distributed Component Object Model) как продолжение технологии COM. DCOM была разработана для распределенных клиент-серверных приложений. Один клиент мог одновременно использовать несколько серверов, установленных на разных компьютерах в сети и каждый сервер одновременно мог обслуживать несколько клиентов. В настоящее время OPC базируется практически исключительно на DCOM технологии фирмы Microsoft для распределенных систем. Главным понятием DCOM является понятие интерфейса, посредством которого DCOM-объекты обслуживают клиентов.

Для обмена данными по OPC между мониторами ТРЕЙС МОУД используются каналы подтипа СВЯЗЬ с дополнениями In OPC – прием данных от MPB по OPC, Out OPC – передача данных MPB по OPC.

При настройке связи по OPC для каждого узла необходимо указать имя компьютера, на котором он будет запущен. Для этого в диалоге Параметры узла на бланке Основные предусмотрено поле Имя компьютера. Для доступа к удаленному компьютеру может потребоваться запуск утилиты DCOMCNFG.EXE и установка соответствующих разрешений пользователям.

Каналы для связи с OPC-сервером создаются процедурой автопостроения. Чтобы запустить её, следует, находясь в окне объектов настраиваемого узла, выполнить команду “Связать с OPC-сервером” из меню “Узел” или нажать сочетание клавиш “Alt”+”L”. При этом появится экран “Выбор сервера OPC”, на котором имеется тип кнопки: ”Добавить”, ”Удалить”, ”Изменить”. Нажатие кнопки ”Добавить” выводит на экран “Выбор сервера OPC” перечень серверов, зарегистрированных на локальной машине или на любом компьютере, присутствующем в сети.

Указанный сервер добавляется в список предыдущего диалога.

По нажатию кнопки ”Удалить” выделенный в списке сервер удаляется из окна. Кнопка ”Изменить” используется для замены выделенного сервера. Она выводит на экран тот же диалог, что и кнопка ”Добавить”. Выбранный в нем сервер заменяет текущий.

Чтобы создать каналы ТРЕЙС МОУД для обмена с выделенным в списке сервером, надо нажать ЛК на кнопке “Выбрать”. В левом окне появившегося экрана следует выбрать каналы OPC-сервера, которые надо контролировать в MPB, и перенести их в правое окно нажатием ЛК на кнопке “>>”. После выхода из этого диалога в базе каналов появится новый объект, имя которого образовано из идентификатора OPC-сервера. В нем создаются каналы для обмена с указанными каналами сервера.

3. СДКУ системами вентиляции и теплоснабжением(конец)

Кондиционирование.

Кондиционирование серверной. реализовано на прецизионных кондиционерах.

Назначение систем кондиционирования — это поддержание заданного микроклимата (во всем здании, отдельном блоке или отдельном помещении). Системы кондиционирования воздуха предназначены для охлаждения/нагрева и частичного осушения/увлажнения воздуха при создании комфортных условий для людей, находящихся в посещениях и стабильной работы серверов.

Система работает по сигналам с датчиков температуры, устанавливаемых в помещениях.

Система комплексной автоматизации и диспетчеризации кондиционирования обеспечивает управление установкой по заданному алгоритму:

- с АРМ оператора инженерных систем;
- с локальных панелей управления;
- по заданной временной программе установки.

Система комплексной автоматизации и диспетчеризации обеспечивает:

- индикацию параметров отдельных узлов подсистемы с возможностью их настройки;
- извещение диспетчера в случае отказа отдельных устройств и агрегатов, а также при возникновении внештатных ситуаций;
- оперативное изменение режимов работы установок в предопределенных ситуациях;
- регулирование температуры воздуха, проникающего в помещения;
- перевод системы в режим энергосбережения по сигналам датчиков;
- обработка заданных алгоритмов группового включения/выключения кондиционирующих установок.

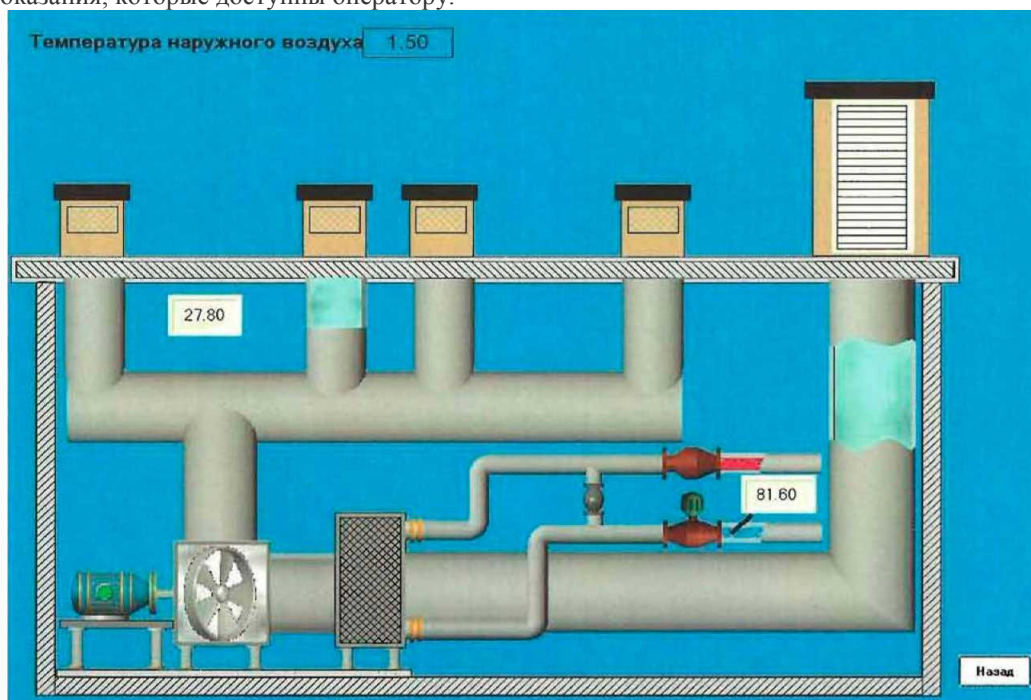
Контроллерный уровень управления выполнен на контроллере Carel mAC (производства Италия) и адаптере snmp/http Web-Gate™. Встроенная функция Web-сервера с использованием стандартного HTTP-протокола дает возможность получать информацию через Web-интерфейс посредством Web-браузера с любого компьютера локальной (или глобальной) вычислительной сети, а связь с [SCADA](#) системой [TRACE MODE](#) не осуществляется. На рисунке, который приведён ниже, отображена страница оператора.



Страница оператора системы кондиционирования

Адаптер Web-Gate является малогабаритным микропроцессорным устройством, предназначенным для интеграции климатического оборудования, управляемого встроенными контроллерами Carel, в стандартные вычислительные сети Ethernet, использующие протокол TCP/IP.

Так же отображён экран оператора с экраном-мнемосхемой, на котором отображена воздушная завеса, её работа и показания, которые доступны оператору.



Экран оператора «Воздушная завеса»

Тема 19-20	Тренажерные системы. Общие понятия о тренажерных системах.	
Технология проведения лекционного занятия		
Время 4 часа	Количество студентов 40-60 человек	
Вид занятия	Визуальная лекция	
План лекционного занятия	<ol style="list-style-type: none"> 1. Основные определения и понятия тренажеро-строения; 2. Концепция тренажера уровня установки 3. История создания тренажеров в промышленности; 4. Обзор тренажерных комплексов 5. Архитектура компьютерного тренажера 6. Основные характеристики тренажеров 7. Требования к тренажеру 8. Этапы создания тренажера 	
Цель учебного занятия: Формирование у студентов глубокого представления о целях и задачах химической науки и путях её дальнейшего развития		
Педагогические задачи:		Результаты обучения:
<ul style="list-style-type: none"> - Ознакомление с основными понятиями тренажеро-строения - Ознакомление с историей создания тренажеров в промышленности. - Внедрение в сознание студентов основных характеристики тренажеров, требования к тренажеру, этапы создания тренажера. 		Студент: <ol style="list-style-type: none"> 1. Имеет представление основные определения и понятия тренажеро-строения 2. Излагает историю создания тренажеров в промышленности 3. Характеризует требования к тренажеру и этапы создания тренажера
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (20 мин)	1.1. Приводит список используемой литературы по тренажерным системам. Знакомит студентов с рейтинговой оценкой знаний студентов. Объявляет тему и план занятия, знакомит с достижениями химической науки	1.1. Слушает и записывает
2-этап Основной	<ol style="list-style-type: none"> 2.1. Обращает внимание студентов на вопросы плана занятия и обращает внимание на основные понятия. Проводит опрос 2.2. Продолжает излагать лекционный материал. На примерах приводит и объясняет основные определения и понятия тренажеро-строения. 2.3. Обеспечивает студентом внимание и запись основных определения и понятия тренажеро-строения. 	<ol style="list-style-type: none"> 2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают
3-этап Итоговый	<ol style="list-style-type: none"> 3.1. Делает заключительные выводы лекционной темы и заостряет внимание студентов на основные вопросы 3.2. Отмечает активных студентов 3.3. Даст задание на самостоятельную подготовку: Дать определение каждого этапа создания тренажера, привести примеры 	<ol style="list-style-type: none"> 3.1. Слушают и усваивают 3.2. Записывают задание

Лекция 19-20. Тренажерные системы. Общие понятия о тренажерных системах.

План:

1. Основные определения и понятия тренажеро-строения
2. Структура и назначение тренажерных комплексов
3. Концепция тренажера уровня установки
4. История создания тренажеров в промышленности
5. Тенденции развития компьютерных тренажеров
6. Обзор тренажерных комплексов
7. Архитектура компьютерного тренажера
8. Основные характеристики тренажеров
9. Требования к тренажеру
10. Этапы создания тренажера

1. Основные определения и понятия тренажеро-строения

Тренажер определяется как техническое средство профессиональной подготовки человека-оператора, предназначенное для формирования и совершенствования у обучаемых профессиональных навыков и умений, необходимых им для управления материальным объектом, путем многократного выполнения обучаемыми действий, свойственных управлению реальным объектом.

Тренажер должен состоять из ряда обязательных элементов:

- ▶ рабочих мест операторов;
- ▶ моделирующего устройства;
- ▶ рабочего места инструктора;
- ▶ устройства контроля и оценки действий оператора.

В специальной литературе можно встретить употребление терминов «тренажеры», «компьютерный тренинг», «компьютерное обучение» в достаточно широком контексте в зависимости от специализации производителей того или иного элемента тренажерных систем.

Разработчикам тренажерных моделей ближе английское слово *simulator*, что более точно означает имитатор, т.е. воспроизводящий, копирующий. Ясно, однако, что не все имитаторы предназначены для тренинга. Не случайно в этой связи, что в англоязычной литературе принято употреблять одновременно два термина (*computer-based training simulation*) для обозначения компьютерного тренинга, основанного на имитационном моделировании процессов в реальном времени, т.е. с учетом возможного производственного вмешательства участников тренинга (обучаемого и инструктора) в ход моделируемого процесса. Разработчики тренажерных платформ оперируют понятием компьютерных тренажерных комплексов, понимая под ними программно-аппаратные средства, обеспечивающие моделирование в реальном времени непрерывного хода ТП, причем совсем необязательно - в целях обучения. Присутствие слова «комплекс» в данном определении подчеркивает наличие всех необходимых компонент для решения конкретных задач - программно-аппаратной платформы для моделирования процесса, необходимых рабочих мест и пользовательских интерфейсов (обучаемого, инструктора, инженера, технолога и др.). Такой подход подчеркивает необходимые требования к тренажерной модели и среде общения с ней пользователя, включает в рассмотрение задачи технологического и технического инжиниринга (часто тесно связанные с задачей обучения), но практически не учитывает инженерно- психологических аспектов тренинга.

С другой стороны, ГОСТ 21036-75 определяет тренажеры как техническое средство профессиональной подготовки человека-оператора для формирования и совершенствования профессиональных навыков и умений управления материальными объектами путем многократного выполнения обучаемым действий, свойственных управлению реальным объектом. И хотя условие много кратности повторения действий нельзя признать абсолютно необходимым, такое эргономическое определение вполне точно описывает цели тренажерных систем, если отвлечься от способов их достижения и вспомогательных инжиниринговых задач. Имеет смысл также учесть рассмотрение тренажеров как инструментального средства, реализующего модель иерархической системы и обеспечивающего контроль деятельности обучаемого.

Представляется, что феноменологический аспект компьютерных тренажерно-обучающих систем (КТОС) может быть сведен к последовательному определению четырех следующих «вложенных» элементов конструкции: «тренажер», «тренажер (для обучения) операторов», «тренажер (для обучения) операторов ТП» и «компьютерный тренажер (для обучения) операторов ТП». Причем уточнение вышеприведенных понятий позволяет, с одной стороны, сузить базовое понятие тренажера по типу деятельности обучаемого (оператора), специфике предметной области (ТП) и форме реализации (компьютерной), а с другой - описать и морфологическую структуру КТОС.

Начнем анализ с базового понятия «тренажер», рассматриваемого как средство формирования и совершенствования навыков субъекта обучения. Под субъектом тренинга здесь и далее будем понимать обучаемого (в конкретном случае КТОС - оператора ТП), а под предметом тренинга-объект реальной деятельности обучаемого (в случае КТОС - реальный ТП). В свою очередь, объект тренинга - те навыки и умения, на формирование и закрепление которых направлено обучение. В этом плане представляется важным

определить необходимые условия, при наступлении которых процесс обучения можно рассматривать как тренинг.

Во-первых, в тренинге должен присутствовать предмет, т.е. тренажерное обучение явно предполагает наличие некоторой операторской деятельности, отличной от самого процесса обучения. Так, развитие с помощью специальных средств (в обыденных случаях именно и называемых тренажерами) тех или иных физических качеств обучаемого, не связанных напрямую с какой-либо целенаправленной деятельностью, не является тренингом. В данном случае отсутствует предмет тренинга, а сам процесс использования такого «тренажера» представляет собой специальную деятельность субъекта с целью достижения желаемых свойств организма.

Во-вторых, даже наличие вышеуказанного предмета тренинга не является достаточным для существования тренажера в том случае, когда сам предмет и является инструментом выработки навыка. Так, если жонглер использует для отработки своего номера те же снаряды, что и во время представления, его упражнения нельзя рассматривать как работу на тренажере. Но как только упражнения на реальном объекте становятся невозможными по соображениям экономичности или безопасности, появляется «тренажерный заместитель» объекта (тренажерная модель) и возникает собственно тренажер. При этом существо дела не зависит от типа тренажерной модели - физической, математической и пр.

Описанное положение по-видимому, рассматриваемое разработчиками тренажеров как само собой разумеющееся, имеет принципиальный характер, поскольку позволяет обозначить тот нижний уровень, с которого может быть начат анализ феномена тренинга. Отметим, что это отчетливо осознается вне узких рамок собственно конструирования тренажеров. Согласно толкованию, тренажер представляет собой «моделирующее устройство для приобретения рабочих навыков или тренировки». Аналогично в зарубежной литературе общего назначения тренажер понимается как «машина или другое с устройством, способное воспроизводить состояния окружающей обстановки для контролируемого обучения. В психологической литературе можно встретить определение тренажера как «аппарата, сконструированного для тренинга с помощью имитации реальных ситуаций». Во всех вышеприведенных подходах существование автономного предмета тренинга и необходимость его замещения в тренажере видны очень четко.

На этой же первой стадии анализа возникает и другой элемент морфологической структуры тренажера, осуществляющий связь между субъектом тренинга и тренажерной моделью - информационная модель или, как принято называть ее в современной литературе, интерфейс обучаемого. Посредством этого элемента обучаемый наблюдает состояния тренажерной модели и осуществляет воздействия на нее, причем тип интерфейса может варьировать от простейших приборных панелей до продвинутых компьютерных виртуальных сред.

Наконец, третий образующий элемент структуры тренажера - модель обучения, содержащая инструментальную (контроль и организация тренинга), методическую (дидактика, упражнения) и оценочную (результаты и качество обучения) составляющие тренинга. Даже самые примитивные тренажерные системы обычно содержат набор тренировочных упражнений, инструкции и механизмы организации тренировок. Так, если рассматривать детский конструктор как тренажер навыков сборки, то предлагаемые модели сборки - элемент модели обучения.

Дальнейшее сужение исследуемого понятия до уровня «тренажер (для обучения) операторов» определяет специальный тип деятельности субъекта тренинга и ограничивает круг предметов тренинга разнообразными объектами управления. В большей степени переход к операторским тренажерам отражается на типе информационной модели (которая становится *интерфейсом оператора с технической системой*) и на модели обучения, отражающей специфику трудовых операторских навыков. Зато понятие «тренажер (для обучения) операторов ТП» в первую очередь связано с изменением характеристик тренажерной модели технической системы. Предметные особенности моделирования ТП, в частности, тренажерного моделирования - один из образующих вопросов построения тренажеров.

Наконец, конечное понятие «компьютерного тренажера (для обучения) операторов ТП» предполагает даже не столько принцип технической реализации составных элементов систем, а те поистине революционные возможности, которые открываются в тренажеро-строении на современном этапе развития информационных технологий. В общих чертах эти возможности определяются следующими:

Во-первых, используемые в современных компьютерных тренажерах математические модели точно воссоздают поведение процессов на основе фундаментальных принципов их функционирования и обеспечивают «в реальном времени» адекватные динамические реакции моделируемых объектов на произвольный набор операторских воздействий. Во-вторых, компьютерная среда управления на рабочем месте обучаемого оператора практически полностью имитирует (или даже копирует) реальную управляющую среду на объекте, так что в процессе тренинга оператор обладает тем же набором измерений и управляющих воздействий, что и в процессе реального труда. Наконец, современные тренажеры обязательно включают в свой состав рабочее место инструктора обучения, обеспечивающее широкий набор инструментов организации и анализа результатов тренинга, а также вмешательств в ход моделируемого процесса и работу системы управления.

Описанная выше конструкция позволяет проследить, как по мере последовательной конкретизации понятия КТОС изменяется содержание составных элементов тренажера. Так, тренажерная модель варьирует по цепочке: имитатор предмета тренинга - имитатор объекта управления (*технической системы*) - имитатор ТП -

компьютерная имитационная модель ТП. В свою очередь, информационная модель конкретизируется по схеме: система отображения информации о предмете тренинга - человеко- машинный интерфейс оператора и технической системы - человеко- машинный интерфейс оператора ТП - компьютерный интерфейс оператора ТП. Наконец, модель обучения предполагает развитие инструментальных средств (от общих механизмов организации тренинга до автоматического запоминания/воспроизведения динамических состояний моделируемого процесса и организации вмешательств инструктора в его ход), методической базы (от обычного набора рекомендуемых упражнений до специальных компьютерных методик выработки отдельных операторских навыков) и оценочных функций (от простейших оценок результативности тренинга до автоматических протоколов тренинга и расчета экономичности управления процессом).

Резюмируя изложение настоящего раздела, можно предложить следующее определение КТОТП, позволяющее приступить к анализу задачи построения тренажеров. *Компьютерный тренажер операторов ТП представляет собой программно-аппаратный комплекс для формирования и закрепления профессиональных навыков и умений оператора ТП, реализуемое путем имитационного моделирования процесса с учетом возможных вмешательств в его ход, воссоздания реального операторского интерфейса и обеспечения развитой инструментальной и методической поддержки обучения.*

2. Структура и назначение тренажерных комплексов

Структура тренажерного комплекса, как правило, имеет три уровня: уровень предприятий, средний уровень и уровень установки.

➤ Уровень предприятия – прогнозы и планирование производства; общий контроль стратегии системы; супервизорный контроль и мониторинг всей системы; обработка данных для административной информационной системы.

➤ Средний уровень – планирование, процесс оптимизации с учетом ряда требований с верхнего уровня; определение функций и набора узлов для отдельных контроллеров; супервизорный контроль и мониторинг процесса. Типичным примером здесь может служить управление работой компрессорных станций, станций подземного хранения газа и т.д.

➤ Уровень установки – посредственное цифровое управление, контроль экстремальных значений (выбросов) параметров; простейшие расчеты.

Тренажеры уровня установки являются наиболее распространенными и в своей работе интенсивно взаимодействуют (опираются) на аппаратно-программную платформу различных PCY (часто в том же смысле употребляется понятие SCADA-системы): Honeywell Plant Scape, Honeywell TPS, ABBMOD-300, Yokogawa Centum CS3000, Invensys I\A Series, SCADA Citect, SCADA InTouch и др.

Многие производители в зависимости от целей обучения подразделяют тренажеры уровня установки на:

- специализированные тренажеры;
- типовые тренажеры (тренажеры типовых технологических установок);
- базовые тренажеры (тренажеры для базовых технологических узлов и аппаратов).

Специализированные тренажеры с высокой точностью отражают динамику реальных технологических установок и, как правило, выполнены в среде применяемой на установке PCY. Это позволяет использовать их для глубокого тренинга операторов разной квалификации (в том числе перед пуском установки или PCY), проверять настройки системы базового регулирования и системы ПАЗ, отрабатывать сложные предаварийные и аварийные ситуации.

Типовые тренажеры (тренажеры типовых технологических установок) описывают динамику типовых технологических установок. Эти уже готовые средства обучения обычно выполняются в типовой среде PCY и направлены, в первую очередь, на ознакомление операторов низких разрядов и вновь поступающих операторов с устройством технологических процессов принципами управления ими.

Базовые тренажеры основаны на типовых моделях отдельных технологических узлов и аппаратов. Они полезны для подготовки вновь набираемых молодых операторов и мастеров КИПиА, поскольку дают представления об устройстве таких технологических объектов и особенностях управления ими.

По назначению тренажерные комплексы подразделяются на:

- тренажеры геолого-гео-физического поиска и разведки месторождений;
- тренажеры бурения скважин;
- тренажеры эксплуатации и ремонта скважин;
- тренажеры транспорта нефти и газа (трубопроводного и танкерного);
- тренажеры технологических процессов переработки.

3. Концепция тренажера уровня установки

В целом, в курсе работы на тренажере можно выделить четыре этапа:

- ознакомление с объектом;
- прогнозирование последствий по методу «что, если?»;
- поиск причин неисправностей;
- стандартные, типовые и аварийные процедуры действий.

Навыки по ознакомлению с объектом, т.е. использованию функций программного обеспечения станции оператора (навигация по системе меню, включение/выключение устройств, перевод устройств в ручной/автоматический режим и т.п.) позволяют получить простые тренажеры.

Формирование необходимого комплексного навыка принятия решений (включая заблаговременное выявление нестандартных ситуаций) возможно только в рамках более сложных тренажеров.

Эффективность обучения будет тем выше, чем выше сходство тренажерного комплекса и реальной АСУ ТП. Для повышения сходства целесообразно использовать программное обеспечение и функциональные модули той АСУ ТП, для которой разрабатывается тренажер, а также достаточно адекватную математическую модель моделируемого ТП.

Математические модели тренажера должны быть полномасштабны (охватывать всю технологическую установку, а не ее отдельные фрагменты), обладать реалистичными динамическими свойствами (т.е. оператор видит развитие ситуации в тренажере таким же, как и на реальной установке) и содержать все элементы технологического оборудования и систем управления, важные для обучения.

Операторский интерфейс тренажера должен соответствовать принятым стандартам отображения технологической информации. В зависимости от требования заказчика операторский интерфейс тренажера может эмулировать типовой операторский интерфейс микро процессорной системы управления или с высокой точностью воспроизводить применяемую на установке РСУ. Помимо практической полной функциональности РСУ, такой подход позволяет поддерживать актуальность тренажера в течение долгих лет после внедрения, поскольку пользователь может адаптировать систему управления в тренажере точно так же, как это делается для реального проекта РСУ на установке.

4. История создания тренажеров в промышленности

Слово происходит от английского **train** - **тренировать, обучать**.

Русский энциклопедический словарь 1890 года "Брокгауз и Ефрон": **Тренировка**: так называется воспитание организмов (главным образом животных, особенно лошадей) направленное к достижению известных целей (например к развитию у лошадей быстроты, у атлетов - мускула, у велосипедистов выносливого и быстрого хода и т.п.) и соответственно этому, соединенное с употреблением известных приемов, могущих оказать на организм желательное действие.

Ровно 100 лет спустя 1990 год Русский энциклопедический словарь: **Тренажер**: учебно-тренировочное устройство для выработки **навыков** и совершенствования техники управления машиной (механизмом). Там же: **"Навык"** - действие, сформированное путем повторения, характеризующееся высокой степенью освоения и **отсутствием поэлементной сознательной регуляции и контроля**.

Первые тренажеры были связаны с авиацией. Причина: высокая цена ошибки и требование действий по управлению машиной, доведенных до автоматизма, желательно на устройстве, находящемся на земле. Авиационные тренажеры (англ. Flight simulator) - электронная или механическая система для тренажа летчиков или космонавтов в условиях полной имитации полета. (Энциклопедия "Britanica", 1998 год.)

Первый авиационный тренажер был построен в Англии в 1910 году. Однако достаточно эффективный тренажер был построен Эдвином А. Линком лишь в 1927 году для обучения летчиков в США под Нью-Йорком.

После открытия коммерческих авиалиний в США в 1934 году усовершенствованный тренажер Линка получает широкое распространение.

Мощный толчок в тренажеростроении дала Вторая мировая война. И в Англии и в США создаются новые более приближенные к реальности тренажеры.

Статистические исследования английских ученых, проведенные во время войны, показали, что 85 % как английских, так и немецких летчиков погибают не от попадания в самолет зенитных снарядов, а от неумелых и неправильных действий по управлению самолетом в боевой обстановке.

Бесспорным доказательством высокой эффективности тренажеров является тот факт, что использование их в ВВС США во время Второй мировой войны позволило ежегодно сберечь не менее 524 жизней, около 133 млн. долл. США, 31 млн. чел./часов и высвободить 15 тыс. человек для выполнения других военных задач. Причем, наибольшая эффективность достигалась в тех случаях, когда тренажеры использовались для подготовки пилотов до принятия самолетов на вооружение.

В 50-е годы на основе аналоговой техники создаются тренажеры для Военно-воздушных сил США, полностью имитирующие кабину самолета и управление полетом в реальных условиях.

В 60-е - 70-е годы прошлого столетия после изобретения цифровой электроники на базе гибридных цифро-аналоговых устройств создаются еще более совершенные моделируемые системы. И, наконец, в 80-х - 90-х годах XX века на основе компьютерной программной технологии создаются виртуальные тренажеры, еще более приближенные к реальной жизни. Однако во всех реализациях авиационной тренажерной техники требование к адекватной конструкции кабины пилота является обязательным.

Такова вкратце история авиационного тренажеростроения в Англии и США.

В России авиационные тренажеры появились в середине 30-х годов. И дальнейшее их развитие примерно совпало с зарубежным.

В энергетике тренажеры начали входить в практику подготовки оперативного персонала в 60-е и 70-е годы, сначала в атомной энергетике (Нововоронежская АЭС), а затем и в традиционной энергетике (Конаковская ГРЭС, Трипольская ГРЭС, ТЭЦ-25 Мосэнерго).

В Мосэнерго работы по тренажеростроению начались в начале 1979 года. Причем, был издан приказ по энергосистеме со сроком окончания работ и вводу тренажера в эксплуатацию в конце 1979 года.

Однако первый полномасштабный тренажер в Мосэнерго был введен в только в 1984 году. Необходимо было создать методологию и методику моделирования энерго-объектов, причем, такую, чтобы моделирование могли проводить сами технологи. Дело в том, что в то время (как, впрочем, и сейчас) разработкой моделей технологического оборудования для тренажеров занимаются математики, специалисты по электронике, автоматике, программисты, системотехники, психологи и т.п.

5. Тенденции развития компьютерных тренажеров

Следует отметить следующий ряд ключевых характеристик компьютерных тренажеров, которые позволяют их однозначно идентифицировать среди всего множества тренировочных средств и оборудования.

1. *Искусственность.* Тренажеры являются искусственными объектами в том смысле, что они предназначены исключительно для выработки информации, необходимой для формирования навыков выполнения определенного задания, а не для осуществления задач, стоящих перед реальной системой.

2. *Хранение и обработка данных.* В тренажерных системах хранятся данные, описывающие динамику имитируемого объекта, а также сведения о влияющих внешних факторах на выполнение обрабатываемого задания. При этом, значительная часть информации хранится в памяти ЭВМ в форме математических выражений. В случае приложения управляющего воздействия его влияние отображается только после того, как компьютер определит те параметры, на которые оказывается воздействие, и оценит влияние этого управляющего воздействия.

3. *Динамика системы.* Отклик системы на входное управляющее воздействие является функцией многих факторов, знание о влиянии которых очень важно для тех, кто изучает принципы построения данной системы, учится управлению этой системой. Хранимая в тренажере информация о системе и внешних факторах организована и обрабатывается таким образом, чтобы воспроизводился динамический отклик системы так же, как он воспринимается и используется при работе с реальной системой.

4. *Органы управления и отображения.* Одно из основных назначений тренажера состоит в отображении информации, характеризующей реакцию системы на управляющее воздействие. Для обеспечения процесса обучения тренажер должен включать в себя органы управления и отображения, выполненные близко к реальным. Следует принимать во внимание условия визуального восприятия. Если какой-либо прибор имеется в тренажере, то он должен быть источником именно того количества информации, которое доступно в характерных для этого прибора условиях в реальной системе.

5. *Комплексное обучение.* Тренажеры, как правило, служат для достижения конкретных, четко определенных целей обучения. При этом тренажеры могут использоваться как для индивидуального обучения, так и для обучения взаимодействующих групп. Тренажер дает возможность обучаемому тренироваться в выполнении определенного задания при различных рабочих нагрузках.

6. *Управление процессом обучения.* Тренажер отличается от имитаторов других типов наличием возможности управления потоком информации, обеспечивающей приобретение профессиональных навыков, что способствует упрощению и углублению процесса обучения. Тренажер содержит математическую модель имитируемой системы и обучающую модель, которая обеспечивает управление, необходимое для того, чтобы обучение происходило максимально эффективно.

Как правило, структура типового тренажерного продукта, создаваемого для предметной области «Автоматизация технологических процессов и производств в промышленности» содержит: имитатор технологического объекта, воспроизводящего реакции объекта во всех режимах его эксплуатации; функциональный блок, обеспечивающий управление ходом обучения; модули декларативного и процедурного представления знаний; блок описания объекта и предмета изучения; интерфейсы обучаемого и инструктора обучения.

Имитатор технологического объекта представляет собой тренажерный заменитель реального объекта, без которого тренинг вообще невозможен. Для реализации имитатора проводится составление модели технической системы, состоящей из модели системы управления и модели технологического процесса. В свою очередь, в модели технологического процесса присутствует модель исполнительных механизмов, собственно модель процесса и модель измерения и преобразования информации.

Необходимо упомянуть, что имитаторы могут быть двух видов: технологические и программные. Первые разрабатываются на основе логических электронных элементов. Разработка вторых осуществляется на базе математических моделей процессов и аппаратов. Программные имитаторы, разработанные по модульному принципу, являются гибкими системами с возможностью построения и реализации желаемой структуры объекта, причём структура имитатора для конкретных тренажеров зависит от курса и выбранного метода программного обучения, сущность которого заключается в постоянном контроле за результатом обучения, анализе действий обучаемого, организации соответствующих рекомендаций и подсказок в процессе обучения, проведения экзамена по предмету и протоколирования хода процесса обучения.

В силу наблюдаемого в последние годы сотрудничества между крупными разработчиками динамических моделей технологических процессов и обладателями специфических тренажерных технологий, можно сделать заключение, что большая доля компьютерных тренажеров будет выполняться на основе недорогих моделей, разрабатываемых для инжиниринговых целей. Примером такого подхода может служить продукт PROTISS, созданный на базе моделирующего пакета PRO/11 фирмы Simulation Sciences Inc. и платформы моделирования реального времени OTISS разработки фирмы SAST (Special Analysis and Simulation).

Прогрессивной линией, прослеживаемой на современном этапе развития тренажеро-строения, является модификация стереотипа автоматизированного обучения производственного персонала. Суть её определяется в том, что компьютерный тренинг может мыслиться уже не как самостоятельная задача, требующая специфической технической и программной реализации, а как одно из звеньев как называемой «оптимизации реального времени» (on-line optimization). Следовательно, возникает прочный фундамент для связи тренинга с задачами инжиниринга: общие модели используются для обучения производственного персонала, прогнозирования реакции реального объекта, подстройки параметров прогнозирующих моделей, проверки и настройки системы управления, исследования технологических режимов, верификации измеряемых данных.

Следует отметить также другое очень важное положение, заключающееся в устремлённости в сторону выработки общего стандарта тренажерных продуктов и к обеспечению возможности объединения в тренажерных системах уже существующих многообразных продуктов разных производителей - библиотек моделей, интерфейсов оператора, наработанных упражнений и пр. Это положение во многом обязано своим существованием такой известной технологии, как технология OLE (Object Linking and Embedding).

В связи с тем, что в настоящее время мультимедиа и гипермедиа- технологии переживают вершину своего развития и широко применяются при создании электронных учебников, информационных компьютерных справочников и автоматизированных обучающих систем, появляются предпосылки для активного внедрения данных технологий на этапе разработки гипермедийных тренажерных комплексов, при которой необходимо будет учитывать научный подход к поставленной проблеме, методику и технологию их построения, которых пока не существует, а также для использования созданных гипермедийных тренажерных комплексов в обучении производственного персонала промышленных установок, в силу эффективности данных обучающих систем с точки зрения повышенной наглядности и понятности в представлении, интерпретации и передаче знаний.

Представляется, что при проектировании любого компьютерного тренажера, невозможно уже будет не принимать во внимание концепцию разработки мультимедийных образовательных программных комплексов. Актуальность данной концепции объясняется тенденцией к переходу от разработки "локальных" обучающих программ к комплексным программным средствам, предпосылками которого служат –общедидактические и частноидактические исследования, современные и программные технические средства, потребность учебных центров. Несмотря на это многие учебные электронные издания до сих пор представляют собой электронные версии текстовых файлов с использованием простейших гиперссылок и инструментов навигации. Не на должном уровне учитываются при разработке таких электронных учебников принципы когнитивного проектирования (повышение когерентности, снижение познавательной перегрузки и т.д.).

Суть обозначенной концепции заключается в следующих положениях:

1. Итерационный процесс создания мультимедийных образовательных программных комплексов для образовательной среды включает в себя взаимодействие авторов учебных материалов, специалистов по методике структурированности учебного материала и разработчиков компьютерных средств;

2. Наибольший обучающий эффект с методической точки зрения имеют сценарии компьютерного воплощения, сочетающие линейные фрагменты сценария с разветвленными, в том числе имеющими обратную связь;

3. Создание открытой программной структуры и мультимедиа- компонентов мультимедийных образовательных программных комплексов на основе базового средства разработки целесообразно после программной реализации каждого из функциональных блоков, имеющих программные модули, и организации взаимодействия между ними;

4. Компьютерная подготовка содержательной части мультимедийных образовательных программных комплексов есть процесс визуализации информации, включающий различные способы ее представления (текст, звук, графика). Для повышения интеллектуальности системы значительный эффект дает использование автоматических связей между различными видами визуализации информации внутри блоков, что особенно злободневно для большого объекта информации.

6. Обзор тренажерных комплексов

На сегодняшний день зарубежом существует несколько современных тренажерных платформ, разработанных и поддерживаемых основными мировыми производителями компьютерных тренажерных комплексов (КТК) (ABB Simeon, Inc., Honeywell, Inc., CAE Link, Inc. и ряд других).

Созданием тренажерных систем диспетчерского управления занимаются и российские специалисты. В частности, свои научные работы посвятили данной теме Григорьев Л.И. и Митичкин С.К. Ими разработана теория для диспетчерских тренажеров по транспорту газа, методика построения диспетчерских автоматизированных тренажеров, а также разработан комплекс математического, информационного и программного обеспечения для тренажера диспетчера по решению задач оперативного управления газотранспортной системой.

Тренажерные программы также разрабатываются специалистами ООО «Энерго автоматика». Ими разработан тренажер оператора магистрального продуктопровода, который имитирует функционирование всего продуктопровода, системы измерения, обработки и передачи информации. Математический модуль тренажера моделирует функционирование продуктопровода, системы измерения и датчиков, алгоритмы обработки информации в контроллерах нижнего уровня, процесс передачи данных по каналам связи и функционирование базы данных верхнего уровня. Этот тренажер функционирует совместно с системой управления на базе Micro

SCADA германской фирмы ABB. Собственно, у данной компании есть также разработки совместно с другими SCADA: MOTOROLA SCADA, MMG SCADA, Infinity SCADA.

В Калининграде существует «Отраслевой научно-тренажерный центр». Центром разработаны АСУ и тренажеры-имитаторы по бурению, добыче газа, транспорту, переработке газа и газового конденсата, нефтегазо-переработке, распределению газа. По транспорту газа создано более тридцати АСУ и тренажеров-имитаторов, одной из которых является АСУ «Эксплуатация линейной части магистрального трубопровода».

Среди российских разработчиков тренажерных комплексов следует отметить также российскую группу компаний ЗАО «Транзас» и ЗАО «Автоматизация Мониторинга Технологий».

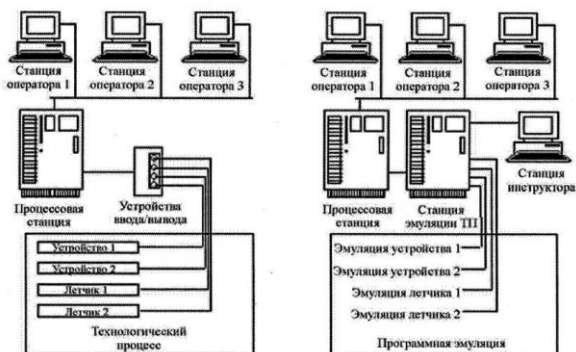
Существуют также тренажерные комплексы, предназначенные для определения остаточного ресурса магистральных нефтегазо-проводов, для определения долговечности трубопровода в различных режимах эксплуатации. Они помогут специалистам по управлению транспортировкой нефти и газа научиться выбирать оптимальный режим, который позволит максимально продлить срок службы трубопроводов, а, следовательно, будет создан столь ценный запас времени для проведения работ по профилактическому ремонту и замене «старых» участков.

В частности, в ОАО «ГИПР Отрубопровод» используется программная продукция по прогнозу ресурса трубопроводов, в основе которой лежат аналитические разработки по оценке надежности транспорта нефти и газа, созданные специалистами ОАО «ГИПР Отрубопровод», РАО «Газпром», ВНИИСПТ нефть. В качестве исходных данных используются результаты экспериментальных исследований (в ходе дефектоскопии и при осмотрах) образцов металла, реально работавших на трубопроводе. На основе моделей развития дефектов под воздействием нагрузок и прогноза будущих объемов транспорта рассчитываются зависимости ресурсов до отказа всех сортов и типоразмеров сталей труб от амплитуды изменений рабочего давления.

В институте проблем транспорта энергоресурсов (ИПТЭР - бывший ВНИИСПТ нефть) разработаны программы, позволяющие рассчитывать прочность, ресурс и надежность элементов трубопроводов. По словам авторов, программы дают возможность промоделировать рациональные параметры, испытательное давление, проанализировать разрушения.

7. Архитектура компьютерного тренажера

Опыт применения тренажеров свидетельствует о том, что эффект обучения достигается не просто идентичностью воспроизводимой на тренажере реальной деятельности оператора, а воспроизводимой на тренажере формы его деятельности, соответствующей решению практических задач в действительности.



Их эксплуатация показала, что при высокой стоимости эффективность применения для успешной подготовки операторов является недостаточной. Исходя из характера обучения операторов технологических процессов, основной задачей обучения является развитие интеллектуальных навыков, что достигается разработкой и внедрением компьютерных тренажеров. Отличительными особенностями компьютерного тренажера является возможность эффективного его использования как для повышения квалификации операторов разных уровней и специальностей и их периодических

тренировок, так и для обучения оперативного персонала в связи с вновь возникающими задачами, обусловленными совершенствованием технологического процесса.

Специфика конкретного производства, анализ деятельности операторов-технологов и способы управления технологическими процессами в АСУ ТП обуславливают ряд особенностей, которые необходимо учитывать при построении тренажеров для обучения операторов. Тренажерный комплекс для обучения операторов АСУ ТП должен обладать распределенной архитектурой, позволяющей легко изменять существующие компоненты системы, дополнять их новыми и т. д. Для достижения требуемого сходства тренажерного комплекса и реальной АСУ ТП целесообразно максимально использовать существующее программное обеспечение вышеупомянутой АСУ ТП.

В качестве примера АСУ ТП, имеющей удачно спланированную архитектуру, можно привести систему управления metso DNA фирмы Metso Automation, упрощенная схема которой приведена на рис. 1.

Ключевым звеном АСУ ТП является станция, в которую загружены алгоритмы обработки сигналов от устройств и датчиков технологического процесса, управляющих сигналов, поступающих от операторов, а также генераторы управляющих сигналов для устройств. Логика работы станции определяется конфигурацией программных модулей, которые получают информацию от устройств ТП через устройства ввода/вывода и от станций оператора. Таким образом, технологический процесс с точки зрения станции представлен набором интерфейсных единиц, каждая из которых позволяет получать данные от какого-либо устройства технологического процесса или передавать туда управляющие сигналы.

Такая архитектура системы metso DNA дает возможность разработать интерфейс, позволяющий заменить собой реальные устройства ввода/вывода, что, в свою очередь, позволит подключать различные модели технологических процессов, реализованные вне системы управления, используя существующие

программные средства математического моделирования, а также собственные средства реализации математических моделей, в том числе различные языки программирования.

8. Основные характеристики тренажеров

К основным характеристикам тренажеров относятся следующие:

- **полномасштабность:** полное подобие рабочего места оператора (информационного и моторного полей) как в случае пультного щитового, так и в случае компьютерного управления оборудованием;
- **всережимность:** воспроизводятся все без исключения режимы работы оборудования (пусковые, остановочные, регулировочные и аварийные);
- **адекватность:** применяется запатентованная методика количественной оценки точности идентификации моделируемых объектов по энтропийному показателю вариации переменных;
- **оптимизация:** контролируются технико-экономические показатели объекта с возможностью работы с максимальным КПД;
- **экологичность:** контролируются вредные выбросы (в сравнении с ПДК), предусмотрена возможность работы с их минимизацией;
- **научная обоснованность и юридическая защищенность:** разработаны и запатентованы всережимные модели технологического оборудования реального времени на базе разделения функций статического и динамического моделирования с направленной асимметрией точности разделенных функций, учитывающие как свойства объекта, так и свойства человека-оператора;
- **обновляемость:** в связи с постоянной международной ротацией программно-технических средств происходит периодическая замена программно-аппаратной платформы разрабатываемых тренажерных устройств. Так, в 2001 году выпущена **пятая версия** тренажеров и обучающих систем (**серия Е**), оснащенная новым программным техническим обеспечением, а также имеющая возможность наладки и коррекции программного продукта фирмой "ТЭСТ" через Интернет;
- **внутренняя и внешняя память, графопостроение:** производится запоминание всех действий оператора, режимов объекта, положения регулирующей и запорной арматуры, механизмов и параметров и их графическое отображение, возврат ситуаций;
- **масштаб времени:** имеется возможность ускорения и замедления масштаба времени (до 100 раз), т.е. ускорение медленных процессов (пуски, остановки) и замедление быстропротекающих процессов (аварийные ситуации);
- **объем дисплейной информации:** на один монитор выводится до 20 мнемосхем, расположенных в виде "окон" с увеличением или уменьшением масштаба;
- **возможности сети:** в случае сетевого варианта имеется возможность подключать основной монитор (сервер) к сети (до 10-ти компьютеров- станций или рабочих мест) и, таким образом, формировать или компьютерный класс или электростанцию полностью со всеми рабочими местами;
- **автоматический контроль работы оператора:** программа, позволяющая оценивать выполненные задания человеком-оператором: оценка ведется по отклонениям от допустимых значений текущих параметров и правильном (или неправильном) выполнении определенных операций за отведенный интервал времени;
- **система поддержки оператора:** в процессе выполнения задания оператор может обратиться к указанной системе за подсказкой;
- **количество задач для тренировки и аварийных ситуаций:** тренажеры снабжаются стандартным набором задач и аварийных ситуаций при серийной комплектации, увеличение количества задач аварийных ситуаций достигается по согласованию с Заказчиком.

9. Требования к тренажеру

Преимущественноиспользуемаятехнологиясозданиятренажерныхкомплексов предусматривает выполнение следующих требований к тренажеру:

- система должна моделировать реальные физические процессы, используя методы математического моделирования;
- тренажер должен заключать все модели физических процессов ТП, имитирующие свойства реального процесса с заданной точностью;
- для тренировки действий операторов в нестандартных ситуациях система должна генерировать различные возмущения в ходе моделируемых процессов как в автоматическом режиме в соответствии с заранее описанными правилами, так и с помощью обучающего инструктора;
- для сокращения времени работы инструктора система должна поддерживать автоматические режимы тренировки с использованием заранее заданного набора упражнений, причем в автоматическом режиме система должна не только ставить задачи перед оператором, но и выдавать рекомендации по их решению.
- тренажер должен реализовывать функцию сбора информации о действиях оператора с возможностью в последствии составления отчетов и анализа эффективности принятых решений;
- система должна быть распределена на несколько тренировочных станций операторов, с обеспечением возможности одновременного обучения нескольких операторов одним инструктором (при этом важным моментом с точки зрения экономии человеческих и машинных ресурсов является возможность тренировки нескольких операторов в один момент времени на разных моделях ТП);

- система должна предоставлять возможность конфигурации наборов упражнений в соответствии с моделью объекта обучения и специфическими требованиями заказчика.

10. Этапы создания тренажера

Основными этапами создания тренажера являются:

- изучение предоставленной инженерно-технологической документации и выработка проекта детальной функциональной спецификации (ДФС) будущего тренажера;
- сбор данных с производства, опрос персонала и систематизация данных;
- разработка модели в соответствии с ДФС на основе библиотеки модели и разработки новых моделей;
- проверка и наладка модели;
- организация обмена данными с РСУ;
- разработка методической базы тренажерного проекта, включающей в себя систему специальных тренировочных упражнений и методик обучения;
- проверка работоспособности всех систем модели настройка динамических свойств в присутствии специалистов заказчика (заводские приемные испытания);
- обучение будущих инструкторов, передача рабочей и методической документации (приемные испытания на месте).

Процесс тренировки оператора

Тренажерный комплекс обеспечивает следующие этапы тренировки оператора АСУТП:

- изучение набора средств оператора АСУ ТП, необходимых для работы со станцией оператора;
- изучение технологического процесса и устройств в нем задействованных;
- отработка наиболее распространенных ежедневных групп операций (пуск, останов, изменение режима);
- оценка состояния ТП по имеющейся информации, обнаружение симптомов неисправности, генерация гипотез о причинах возникновения неисправности и принятие мер по приведению системы в оптимальное состояние, исходя из возникшей неисправности;
- отработка действий в условия аварийной ситуации.

Рабочее место оператора обычно состоит из терминала, с помощью которого оператор получает информацию о ходе процесса, а также клавиатуры и мыши, посредством которых оператор передает системе информацию о своих решениях. Также могут использоваться специализированные (промышленные) варианты клавиатуры и манипуляторов, отличающиеся по форме и свойствам от стандартных, используемых для персональных компьютеров. Очевидно, что при обучении целесообразно использовать те устройства, которые будут использоваться в своей дальнейшей профессиональной деятельности.

Технологический процесс и устройства, в нем задействованные, обычно представляются на терминале оператора в виде символов, которые можно разделить на основные классы (клапан, мотор, насос, ПИД-контроллер и т. п.). Каждый символ несет в себе определенную смысловую нагрузку и обладает набором функций, связанных с ним.

Например, символы устройств, как правило, имеют функции включения или выключения устройства, перевод устройства в автоматический режим управления и т. п. Обучаемый должен усвоить эту символику, атак же получить и закрепить навыки использования всех основных функций символа каждого типа.

В работе оператора значительную часть занимают стандартные типовые последовательности, такие как пуск или останов различных участков ТП или перевод их из одного режима работы в другой. Для изучения, запоминания и повторения подобных последовательностей тренажерный комплекс должен включать набор соответствующих упражнений.

С точки зрения оператора технологический процесс представляется множеством компонентов (узлов, аппаратов, агрегатов, емкостей и т. п.), каждый из которых представлен на дисплее

в виде измерений и показателей состояния. Различные наборы значений измерений и показателей состояния соответствуют различным состояниям ТП.

Оператор должен научиться и адекватно оценивать по представленной информации состояние отдельных компонентов ТП и всего ТП в целом. Таким образом, группа упражнений должна содержать задачи по оценке состояния системы, обнаружения симптомов неправильного поведения, генерации гипотез о причинах неправильного поведения, предсказания развития поведения системы в случае невмешательства, предсказания последствий воздействий на систему, упражнения по генерации воздействий, оптимальных с точки зрения приведения системы в желаемое состояние.

Кроме того, необходимы навыки работы в аварийных ситуациях, либо, если возникновение аварийной ситуации приводит к автоматическому срабатыванию некоторого алгоритма, необходима отработка навыков поведения после завершения работы данного алгоритма.

На различных этапах обучения целесообразно использовать математические модели различной сложности.

Контрольные вопросы.

1. Какую функцию выполняет тренажер.
2. Классификация тренажеров.
3. Архитектура тренажеров.
4. Процесс тренировки в тренажере.

Тема 21-22	Программный пакет Experion PKS	
Технология проведения лекционного занятия		
Время 4 часа	Количество студентов 40-60 человек	
Вид занятия	Визуальная лекция	
План лекционного занятия	<ol style="list-style-type: none"> 1. Архитектура системы управления Experion PKS. 2. Структура системы. 3. Программное обеспечение. 4. Программирование задач контроллера и управление. 5. Программирование и реализация SCADA-программы. 6. ПО цифровой телевизионной системы наблюдения. 7. Описание уровней безопасности в программные пакеты Experion PKS 	
Цель учебного занятия: Формирование у студентов глубокого представления о целях и задачах химической науки и путях её дальнейшего развития		
Педагогические задачи:		Результаты обучения:
<ul style="list-style-type: none"> - Ознакомление с архитектуры систем управления Experion PKS. - Ознакомление с Структура системы, программное обеспечение. - Внедрение в сознание студентов основных характеристики тренажеров, требования к тренажеру, этапы создания тренажера. 		Студент: Имеет представление структура системы, программное обеспечение Излагает программирование задач контроллера и управление Характеризует требования к тренажеру и этапы создания тренажера
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.5. Приводит список используемой литературы по тренажерным системам. Знакомит студентов с рейтинговой оценкой знаний студентов. Объявляет тему и план занятия, знакомит с достижениями химической науки	1.1. Слушает и записывает
2-этап Основной	2.1. Обращает внимание студентов на вопросы плана занятия и обращает внимание на основные понятия. Проводит опрос 2.2. Продолжает излагать лекционный материал. На примерах приводит и объясняет структура системы, программное обеспечение 2.3. Обеспечивает студентом внимание и структура системы, программное обеспечение.	2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают
3-этап Итоговый	3.1. Делает заключительные выводы лекционной темы и заостряет внимание студентов на основные вопросы 3.2. Отмечает активных студентов 3.3. Даст задание на самостоятельную подготовку: Дать определение описание уровней безопасности в программные пакеты Experion PKS	3.1. Слушают и усваивают 3.2. Записывают задание

Лекция 21-22. Программный пакет Experion PKS

План:

1. Архитектура системы управления Experion PKS
2. Структура системы
3. Программирование задач контроллера и управление
4. Программирование и реализация SCADA-программы
5. Уровни безопасности
6. Безопасность по Операторам

1. Архитектура системы управления Experion PKS

Фирма Honeywell с 1974 г. продвигает технологию создания PCY. Пройдя через последовательные этапы эволюции (TDC 2000, TDC 3000, TPS, Plant Scape), в январе 2003 г. развитие технологии вышло на новый уровень. Была разработана платформа Experion, от крившая новое поколение систем сбора, хранения и обработки данных о ходе производственного процесса. И первой системой, созданной на базе новой платформы, стала PCY Experion PKS, которая уже внедрена на ряде предприятий нефтегазовой, химической, целлюлозно бумажной отраслей в разных странах мира. На рис. 1 представлена архитектура PCY Experion PKS.

Архитектура системы управления Experion PKS

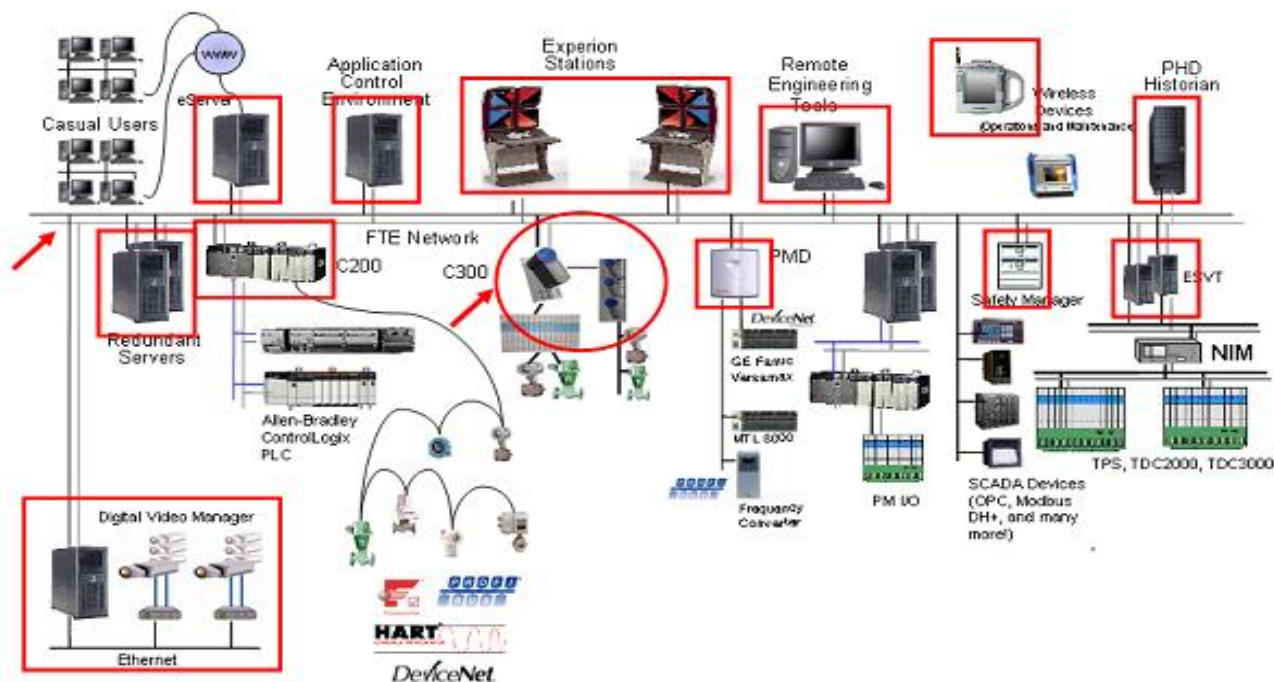


Рис. 1 Архитектура системы управления Experion PKS

Широкая масштабируемость, возможная (благодаря) специальной архитектуре PCY, позволяющей отдельным системам Experion PKS работать как единому комплексу, обеспечивая прямой доступ от каждой рабочей станции к любому контроллеру и к историческим данным любой системы. Благодаря этому Experion PKS можно использовать для управления отдельным ТП и для автоматизации всего производства в комплексе.

Объединение данных, полученных от различных средств системы, в единой БД, что позволяет иметь одноразовый ввод информации и многократное ее использование разными пользователями на различных уровнях управления.

Непосредственная доступность всей информации, вырабатываемой системой, через средства Internet/Intranet, поскольку все программные средства верхнего уровня системы базируются на запатентованной фирмой Honeywell технологии HMIWeb, реализующей ЧМИ с форматированием данных (их гипертекстовой разметкой) на языке XML.

Экономное функционирование сетей благодаря принятому способу передачи сигналов от любых средств нижнего уровня (датчиков и контроллеров): сигналы передаются только при изменении их значений. Это существенно снижает нагрузку сетей и повышает быстродействие системы.

Всесторонняя открытость системы, обеспечиваемая использованием промышленного интерфейса OPC (OPC-клиента/сервера); типовой связью с реляционными СУБД (язык SQL и драйвер ODBC); типовыми программными интерфейсами и стандартами сборки программ фирмы Microsoft, включая COM/DCOM, OLE, ActiveX; использованием в составе системы наиболее распространенных, стандартных (и) формационных,

промышленных и полевых сетей; наличием драйверов к техническим средствам ведущих мировых производителей продукции автоматизации.

Наследование линии предыдущих систем автоматизации фирмы Honeywell и реализация непосредственной связи и интеграции с существующими на предприятиях системами, комплексами, контроллерами производства Honeywell.

2. Структура системы

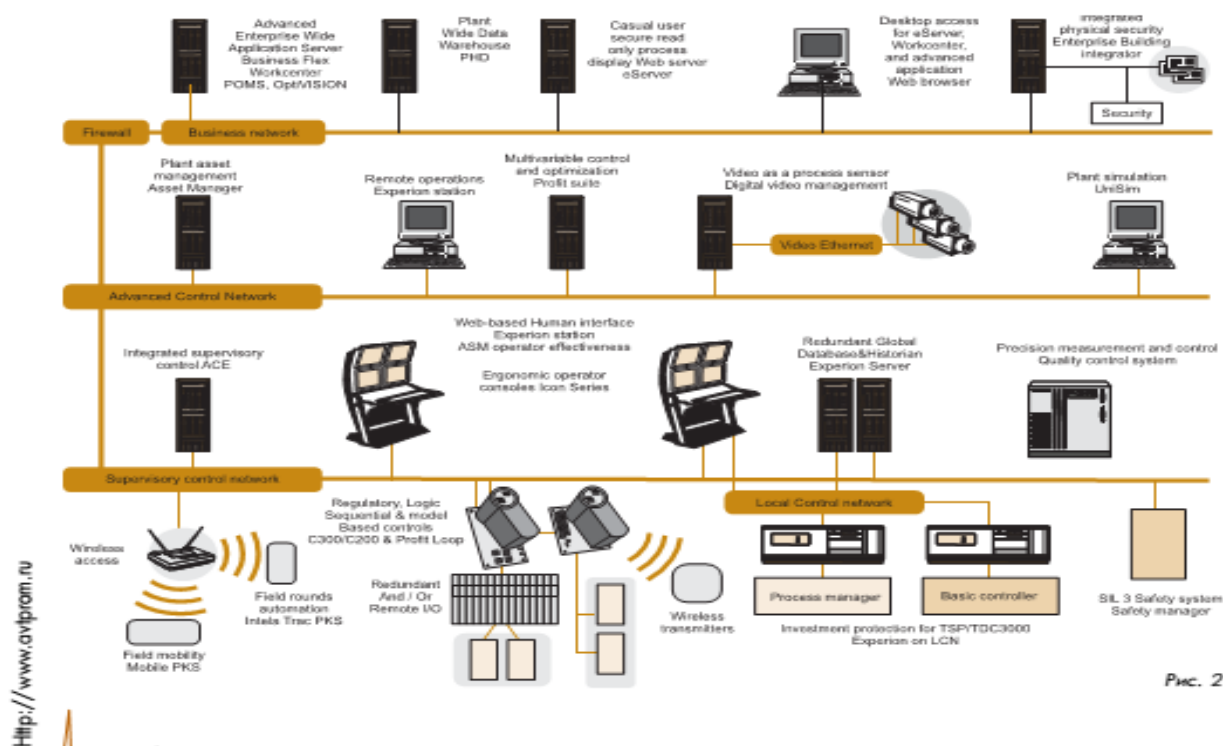


Рис. 2

Платформа Experion имеет три сетевых уровня, каждый из которых может быть реализован разными вариантами типовых сетей (рис. 2). На верхнем информационном уровне, формирующем связи рабочих станций друг с другом, с серверами, с корпоративной сетью предприятия, используется типовая сеть Ethernet в двух вариантах:

- нерезервированная сеть Ethernet/Fast Ethernet: скорость передачи данных – 10/100 Мбит/с, максимальная длина сегмента сети – 1 км, число сегментов – 6 ед. (5 повторителей), число узлов на сети – 12 ед.;
- Fault Tolerant Ethernet (отказоустойчивый Ethernet) – лицензированная разработка фирмы Honeywell, объединяющая достоинства широко распространенной, недорогой информационной сети Ethernet и живучесть промышленных сетей. Она построена на резервированных кабелях и обеспечивает множественные связи между узлами сети (до четырех портов сети на каждом узле), что приводит к сохранению связей при единичных и даже множественных сбоях в сети. В качестве отдельных узлов сети могут применяться и узлы обычной сети Ethernet. Любые локальные неисправности сети обнаруживаются за время ~1 сек и в подавляющем большинстве случаев компенсируются передачей информации через другой путь по имеющимся множественным связям. Скорость сети – 100 и 1000 Мбит/с, а другие ее характеристики идентичны типовой сети Ethernet.

В качестве промышленных сетей, соединяющих контроллеры друг с другом, с серверами и/или рабочими станциями, могут использоваться Ethernet, Fault Tolerant Ethernet, ControlNet.

Сеть ControlNet имеет следующие характеристики: скорость передачи данных – 5 Мбит/с, максимальная длина сегмента сети – 1 км, число сегментов – 6 ед. (5 повторителей), поддерживает одиночную и резервированную среды, число узлов на сети – 32 ед.

К одному серверу через любую промышленную сеть может быть подключено до 10 нерезервированных или резервированных контроллеров.

В качестве полевых сетей, соединяющих контроллер с удаленными каркасами модулей ввода/вывода и с интеллектуальными приборами, предусмотрено использование любой из широко распространенных цифровых протоколов данного уровня: Foundation Fieldbus, Profibus DP, HART)протокол, а также сети ControlNet. Основные характеристики полевых сетей представлены в табл. 1.

Таблица 1.

Характеристики	Foundation Fieldbus	Profibus DP	HART-протокол
Скорость, бит/с	31,25	9,6...1500	1,2
Длина сети, км	1,9	4,8	1,5
Число узлов на сети, ед.	< 32	< 126	<15

Основным контроллером системы Experion PKS является C200. Он поддерживает работу в резервируемой и не резервируемой конфигурации, по быстродействию может быть сконфигурирован для работы в одной из двух модификаций, характеризуемой базовым временем обработки одного функционального модуля (5 или 50 мс), имеет 4 Мбайта памяти для программ пользователя (табл. 2).

Модули ввода/вывода контроллера являются интеллектуальными: установленный в них процессор управляет опросом датчиков, выполняет преобразование из измеряемой величины в цифровую форму, производит за данные математические и логические действия. Модули ввода/вывода располагаются в том же каркасе, что и контроллер или выносятся в отдельных каркасах непосредственно в цеха и соединяются с ним через полевую сеть ControlNet с использованием оптоволоконных технологий. Возможно резервирование модулей ввода/вывода. Максимальное их число на один контроллер –64 ед. Существуют различные модификации модулей ввода/вывода: исполнение в каркасе, взрывобезопасное исполнение, экономичное исполнение блоков.

В РСУ предусмотрено использование следующих серий модулей ввода/вывода:

-модули аналоговых входов на 8, 16 каналов с параметрами 0...5В, 4...20 мА, разрядностью 14 бит, точностью 0,075%;

-модули входов милливольтных сигналов, термопар и термосопротивлений на 16, 32 (мультиплексор) канала, разрядностью 15 бит;

-модули аналоговых выходов на 8, 16 каналов с параметрами 4...20 мА;

-модули дискретных входов на 32 канала;

-модули дискретных выходов на 16, 32 канала; имеются релейные выходы на 265 В и 2,5 А.

Большинство модулей ввода/вывода имеют гальваническую развязку.

Кроме основного контроллера в системе могут использоваться полностью интегрированные в нее контроллеры PLC-5 и LC 5550 семейства Allen-Bradley, выпускаемые фирмой Rockwell Automation.

Для создания систем противоаварийной защиты используется специальный контроллер противоаварийной защиты фирмы Honeywell – Fail Safe Controller (FSC), имеющий центральный процессор с четырехкратным резервированием. Контроллер сертифицирован как в России, так и в Германии немецкой комиссией TUV по классу АК 6, SIL3, стандарта DIN V 19250; имеет два независимых интерфейса к сети Ethernet; непосредственно подсоединяется к техническим средствам верхнего уровня системы Experion PKS.

Удаленные модули ввода/вывода могут работать при влажности, загрязненности, агрессивности окружающей среды, соответствующей норме американского стандарта NEMA 4X; иметь модификации с защитой от коррозии (норма G1 или G3 по стандарту ANSI/ISA)S71/04) и без нее.

В системе Experion PKS могут функционировать несколько типов рабочих станций (табл. 3). Рабочая станция типа Flex обладает наибольшими возможностями.

В качестве клиента она может быть подсоединена к серверу Experion PKS через сеть Ethernet, Internet или посредством радиосвязи. Станция типа Flex либо непрерывно соединена с сервером (когда используется оператором автоматизируемого ТП), либо подключается к нему в отдельные периоды по желанию пользователя (инженерного персонала предприятия). Все механизмы безопасности и авторизации подключения сохраняются вне зависимости от способа ее соединения.

Рабочая станция типа Console имеет одно важное отличие: доступ к контроллерам осуществляется непосредственно (а не через сервер) с использованием сети Fault Tolerant Ethernet. Использование станций этого типа целесообразно при автоматизации быстрых непрерывных процессов, требующих постоянного внимания оператора.

Рабочая станция типа Console Extension является расширением рабочей станции Console и подключается непосредственно к ней, как к серверу через сеть Fault Tolerant Ethernet.

Кроме того, в системе Experion PKS могут использоваться мобильные рабочие станции беспроводной связи Mobile PKS.

Все типы рабочих станций имеют следующие рекомендуемые характеристики: процессор Pentium IV и выше с частотой 2,5 ГГц; 512 Мбайт ОЗУ; 512 Кбайт кэш памяти; 10 Гбайт дискового пространства; ОС Windows 2000 Professional/XP Professional.

Сервер системы Experion PKS содержит БД РВ и прикладные программы. В БД могут записываться одномоментные мгновенные значения и усредненные значения измеряемых величин за 6 мин и за 1, 8, 24 часа; а также аварийные сообщения с метками времени, подразделяемые на 15 приоритетных уровней. Сервер Experion PKS имеет следующую типовую конфигурацию: два процессора Pentium III или Xeon и выше с частотой 2,0 ГГц; 2 Гбайта ОЗУ; 512 Кбайт кэш памяти; 36 Гбайт дискового пространства; ОС Windows 2000 Server.

При управлении производством в комплексе могут использоваться несколько систем Experion PKS со своими серверами, каждый из которых обслуживает свой объект автоматизации. В этом случае устанавливается еще один центральный сервер, обслуживающий диспетчерский пункт предприятия и все указанные серверы обмениваются информацией для создания единой БД. В этом случае перечень данных, сообщений аварийной сигнализации, трендов на любой станции оператора (диспетчера) производства может формироваться из измеряемых значений разных объектов автоматизации, охватываемых системами Experion PKS. Серверы и рабочие станции могут оперативно резервироваться.

Программное обеспечение

ПО системы Experion PKS охватывает различные стороны контроля, управления, диагностики, анализа, документирования и усовершенствованного управления и содержит ряд оригинальных алгоритмов, базирующихся на изучении конкретных свойств автоматизируемого объекта.

Общая программная среда CEE (Control Execution Environment) выполнения алгоритмов контроля и управления охватывает:

- контроллер C200, реализующий типовые функции контроля и управления нижнего уровня;

Таблица 2. Контроллеры систем управления компании Honeywell

Название	Контроллер C300	Контроллер C200	Контроллер HPM
Год выпуска	Новейшая разработка компании Honeywell, начало поставок намечено на конец 2005 г. Позиционируется как основной контроллер системы Experion PKS.	Совместная разработка компании Honeywell и Allen-Bradley, впервые представлена в 1999 г. Является основным контроллером системы PlantScape, а в настоящее время – системы Experion PKS. Широко распространен в процессах, имеющих небольшое число входных/выходных каналов, не требующих резервирования.	Высокопроизводительный менеджер процесса (HPM), впервые представлен компанией Honeywell в 1997 г. Является основным контроллером системы TPS. Широко распространен в процессах, имеющих большое число входных/выходных каналов и имеющих повышенные требования по надежности и резервированию.
Производительность резервированного контроллера	3 600 единиц (PU) Один контур регулирования PID – 2,8 ед.	1 800 единиц (PU) Один контур регулирования PID – 2,8 ед.	800 ед. (PU) Один контур регулирования PID – 1 ед.
Частота сканирования	наиболее быстрая – 50 мс; наиболее медленная – 2 с		наиболее быстрая – 250 мс наиболее медленная – 1 с
Интерфейс подключения	Fault Tolerant Ethernet	ControlNet; Fault Tolerant Ethernet	Universal Control Network
Число модулей ввода/вывода	64		40
Система ввода/вывода	Series C; Series A (не все); PMIO (не все)	PMIO (не все); Series A	модули ввода/вывода PMIO
Резервирование	модули ввода/вывода Series C и PMIO (не все)	модули ввода/вывода PMIO (не все)	
Удаленный ввод/вывод	модули ввода/вывода PMIO (не все)	Rail I/O Series A; Rail I/O Series H Модули ввода/вывода PMIO (не все)	
Искрозащита	Терминальные панели GI/IS FTA Планируется использование искробезопасных модулей Series C (2006 г.)	Rail I/O Series H Терминальные панели GI/IS FTA	Терминальные панели GI/IS FTA
Поддержка протоколов	HART (Series C, PMIO); DE (PMIO); FieldBus (Series C); Profibus (Series A, Series C – планируется); DeviceNet (Series A, Series C – планируется); Modbus (Series A, Series C – планируется)	HART (Series A, PMIO); DE (PMIO); FieldBus (Series A); Profibus (Series A); DeviceNet (Series A); Modbus (Series A)	DE; FieldBus; Modbus

- сервер Experion PKS, реализующий более сложные и объемные (супервизорные) прикладные задачи управления, выходом которых являются воздействия на уставки контуров управления нижнего уровня или непосредственные команды исполнительным механизмам;

- ПК, реализующий моделирование работы контроллера и сервера (без подключения к компьютеру последнего), необходимое для проверки и отладки разработанного прикладного ПО контроля и управления, а также для обучения и тренажа операторов.

Для работы одних и тех же прикладных программ на разных технических средствах к общей программной среде добавляются определенные программные функции, поэтому программные среды подразделяются на управляющую решающую среду для контроллера, среду управления прикладными задачами для сервера, среду управления моделированием для реализующего модель ПК.

Управляющая решающая среда контроллера работает в двух вариантах быстрого действия: при базовом времени обработки одного функционального модуля 5 и 50 мс. При 50 мс период выполнения отдельных программных модулей может быть 50, 100, 200, 1000, 2000 мс, а при быстром цикле 5 мс период выполнения отдельных программных модулей может быть 5, 10, 20, 100, 200 мс.

3. Программирование задач контроллера и управление

Инструментальным средством программирования задач контроля и управления во всех трех программных средах является Control Builder – графический конфигуратор, базирующийся на обширной библиотеке типовых алгоритмов математических, логических, управленческих функций, которые группируются в модули контроля, регулирования, последовательного управления. В библиотеку типовых программных модулей входят:

- общие математические операции типа линеаризации, опережения/запаздывания, временной зоны нечувствительности;
- логические операции и функции последовательного управления механизмами. В отдельных программных модулях управления группой механизмов реализованы различные варианты воздействий при возникновении нештатных ситуаций в процессе последовательных действий, которые предотвращают возможность аварийных событий. Имеются также модули, обеспечивающие интерактивное взаимодействие автоматических и ручных воздействий в процессе проведения последовательного управления механизмами;
- программы регулирования, включающие типовые алгоритмы ПИД, ПИД с упреждением, каскадное управление, регулирование соотношения.

Кроме этого, в библиотеку включаются специальные модули, в частности, интерфейсы с приводами семейства Allen Bradley, расчеты расходов различных газов с учетом температуры окружающей среды, давления и состава газа.

С помощью конфигуратора Control Builder производится проектирование, документирование и мониторинг работы алгоритмов контроля и управления. Он поддерживает иерархическое вложение одних программных модулей в другие; позволяет многократно копировать и использовать спроектированные модули; реализует многопользовательское проектирование, когда несколько пользователей конфигурируют и загружают программы с разных АРМ.

Прикладные программы пользователя могут быть написаны на языках C/C++, Visual Basic, Visual C/C++ и реализовываться на сервере, рабочих станциях и контроллерах.

4. Программирование и реализация SCADA-программы

Связь любого пользователя с любой рабочей станцией на производстве (доступ к мнемосхемам, трендам, сообщениям, протоколам) производится (в отличие от других систем) по Web) технологии с использованием языка форматирования документов XML, что облегчает интегрирование технологических и экономических данных и сокращает время доступа пользователей к оперативной производственной информации.

Инженерным средством проектирования ПО рабочих станций по Web) технологии является пакет HMI Web Display Builder, обеспечивающий построение, модификацию и хранение дисплейных кадров всех типов. Формы работы с ним и типы проектируемых кадров подобны имеющимся в большинстве SCADA) систем.

Инструментальные средства генерируют все традиционные виды представления информации на рабочих станциях: различные типы мнемосхем с возможностью анимации, вывод на дисплей видеок кадров с разных видеокамер, окна в кадре с графическими панелями текущего состояния отдельных контуров управления (всего в кадре могут быть размещены четыре окна), барграфы, графики и тренды текущих и исторических данных, одиночные и групповые сообщения о тревогах и аварийных ситуациях.

Аварийные сообщения могут автоматически передаваться на пейджеры персонала (до 100 пейджеров могут быть сконфигурированы на прием сообщений в определенном временном графике) и на заданные адреса электронной почты.

Типовые рапорта, протоколы, отчеты включают, в частности, следующие формы:

- протокол всех тревог и событий за определенный интервал времени;
- протокол, фиксирующий определенные события и их взаимосвязанные группы с моментами времени их появления;
- отчет свободного формата, включающий значения измеряемых величин, вычисляемых показателей, статистических характеристик за заданный интервал времени.

Безопасность работы комплекса обеспечивается двумя типами допусков пользователей: допуск к работе на определенной рабочей станции, допуск к фиксированному виду работы с определенной группой данных. Перед работой каждый пользователь, обращающийся к комплексу, идентифицируется и авторизуется (выделяется разрешенный для него набор функций). Всего имеется шесть уровней допуска к функциям контроля и управления на определенной рабочей станции от показа только тревожных сообщений до общего мониторинга и реализации любых управляющих функций. Возможна поддержка электронной подписи под регламентируемыми действиями персонала.

ПО цифровой телевизионной системы наблюдения

В систему Experion PKS полностью интегрирована промышленная телесистема, которая состоит из четырех видеокамер (возможно ее расширение). Работа видеокамер может активироваться отдельными программами Experion PKS: наступлением заданных моментов времени, обнаружением системой определенных событий, командой оператора. Изображения записываются в СУБД SQL Server. Специальные программы через рабочую станцию позволяют операторам дистанционно управлять работой видеокамер: масштабировать, панорамировать, включать звуковое сопровождение в месте расположения камер. Операторы могут просматривать как текущие изображения, так и ранее записанные.

Программы анализа нештатных ситуаций

Углубленный контроль и диагностика автоматизируемого процесса, базирующиеся на знаниях операторов и технологов, реализованы в программах:

- анализа зафиксированных системой событий, где определяются число отдельных событий и их связанных групп за час и смену и максимальное их число за 10 мин; рассчитываются интервалы времени от момента возникновения события до его подтверждения оператором и его компенсации управляющими воздействиями; вычисляются скользящие во времени статистические характеристики появления отдельных событий и их связанных групп, что выявляет изменение хода автоматизируемого процесса во времени;
- анализа протекания периодических процессов, где сравниваются текущие измеряемые данные с заданными для данного момента эталонными значениями; сопоставляются текущие тренды измеряемых величин с соответствующими трендами прошлых периодов; выделяются отклонения хода процесса от эталона за счет воздействий оператора, изменений качества входных ресурсов, ухудшения работы оборудования.

Программы анализа работы оборудования и документирование его обслуживания

Контроль, анализ и документирование работы оборудования автоматизируемого объекта выполняется программами:

- раннего обнаружения изменений состояния оборудования, где проводится статистическое моделирование условий эксплуатации оборудования. На основе заданных логических моделей отказов (опыт операторов), а также наблюдений за изменениями колебательности автоматизируемого процесса и возникающими нелинейностями в зависимостях отдельных измеряемых величин программа прогнозирует возникновение отказов в работе оборудования;
- документирования, которая хранит данные по истории всех изменений, происходивших с оборудованием (включая технические средства автоматизации); отслеживает процедуры технического обслуживания оборудования и автоматизированной калибровки приборов.

Программы взаимосвязи системы с внешними средствами

Ряд программных пакетов расширяют возможности документирования системы и существенно упрощают ее связи с организационно) экономическими подразделениями предприятия:

- пакет eServer обеспечивает на основе Web) технологии нерегулярный доступ сотрудников предприятия и любых сторонних средств к информации системы Experion PKS, одновременно сохраняя целостность среды контроля и управления для операторов производства;
- центральная историческая СУБД (Uniformance PHD) концентрирует данные из всех систем Experion PKS на предприятии и из других средств и систем через серверы OPC. Здесь происходит сжатие информации и ее обработка по заданным алгоритмам.

Выводы

- Архитектура и характеристики системы Experion PKS характеризуются полной открытостью и возможностью интеграции со средствами и системами автоматизации производства других производителей, ERP)системами, отдельными программными пакетами управления бизнес-процессами; в системе отсутствуют какие)либо свойства замкнутости.
- Значительно расширена функциональность платформы Experion: полнее охвачены функции автоматизации технологического объекта и добавлены функции автоматизации обслуживания оборудования.
- Доступ к любой информации, вырабатываемой системой, благодаря использованию Web) технологии разрешен не только работающему с ней оператору, но и каждому сотруднику предприятия, авторизованному на получение этой информации независимо от места его нахождения.
- Функции контроля и определения нештатных ситуаций значительно расширились и углубились за счет использования в них знаний операторов и технологов, работающих с автоматизированным объектом.

Описание уровней безопасности в программном пакеты Experion PKS

Существует несколько уровней безопасности и каждый уровень определяет задачи, которые могут быть выполнены на этом уровне.

Станция поддерживает два типа безопасности, из которых мы будете использовать только один.

Operator-based security – Безопасность по Операторам – В этом случае нам назначается имя пользователя и пароль, или Ваше имя пользователя и пароль для входа на Станцию идентичны нашему имени пользователя и паролю для входа в систему Windows.

Station-based security – Безопасность по Станциям – В этом случае Вам не назначается имя пользователя. При запуске Станции, Вам автоматически назначается уровень безопасности Оператора(OPER). Впоследствии мы

можете изменить уровень доступа на более высокий, в том случае если мы знаете пароль для соответствующего уровня на конкретную Станцию.

5. Уровни Безопасности

Уровень Безопасности определяет, какие задачи Вам разрешается выполнять. Однако даже если нам разрешено осуществить специфическую задачу, Вам может быть не позволено выполнить эту задачу при определенных обстоятельствах.

Наш уровень безопасности появляется в правой части Строки Состояния. В нижеприведенном примере, уровень безопасности данной Станции- MNGR(Менеджер).



Уровни Безопасности: (от самого низкого до самого высокого)

VIEW Only (Только Просмотр)

ACK Only (Только Подтверждение)

OPER (Оператор)

SUPV (Супервизор)

ENGR (Инженер)

MNGR (Менеджер)

Уровень Безопасности	Описание
View Only	Разрешает просматривать мнемосхемы, тренды и репорты, но не разрешает производить изменения в технологическом процессе.
Ack Only	Разрешает подтверждать сигнализацию и просматривать мнемосхемы, тренды и репорты, но не позволяет производить изменения в технологическом процессе.
OPER	Разрешает Вам подтверждать сигнализацию и производить изменения в технологическом процессе только в тех точках, которые привязаны к зонам, назначенным или на Вашу Станцию или Вашему Идентификатору Оператора.
SUPV	Предоставляет Вам все права уровня Оператора и способность изменять или конфигурировать настройки Трендов и Групповые Дисплеи, Расписание Праздников и Смены, конфигурировать Рецепты, и Рапорты Продолжительности Сигнализации.
ENGR	Предоставляет Вам все права уровней Оператора и Супервизора, и способность производить изменения в конфигурации точек и других элементов системы.
MNGR	Предоставляет пользователю права изменять все, что может быть изменено или сконфигурировано в системе.

Различия между безопасностью по Станциям и по Операторам

Два типа безопасности для Станции Experion PKS®:

Безопасность по Станциям. Уровень по умолчанию для доступа на Станцию.

Безопасность по Операторам. Обеспечивает более высокий уровень безопасности, чем безопасность по Станциям.

6. Безопасность по Операторам

Безопасность по Операторам обеспечивает более высокий уровень безопасности, чем безопасность по Станциям. Существует два типа безопасности по Операторам:

Традиционная учетная запись Оператора – учетная запись Оператора Experion PKS не имеет ассоциированной учетной записи Windows.

Интегрированные учетные записи – учетные записи Windows используются для идентификации пользователей в системе Experion PKS с целью их регистрации.

В общих чертах, безопасность по Операторам с традиционными учетными записями, работает следующим образом:

Каждому Оператору назначается определенный уровень безопасности.

Пользователи не будут иметь доступа ко всем функциям Станции, пока пользователь не введет правильные идентификатор и пароль.

Для доступа с более высоким уровнем безопасности, чем тот, который используется на данный момент, пользователям необходимо выйти из системы и вновь войти в систему под другим пользователем, кто имеет более высокий уровень безопасности.

Зоны назначаются операторам независимо от того, на какую Станцию они вошли в данный момент. Зоны, которыми Вы можете управлять определяются понятием Кто Вы, а не Где Вы.

Безопасность по Станциям

Безопасность по Станциям используется по умолчанию для доступа на Станцию. Безопасность по Станциям работает следующим образом:

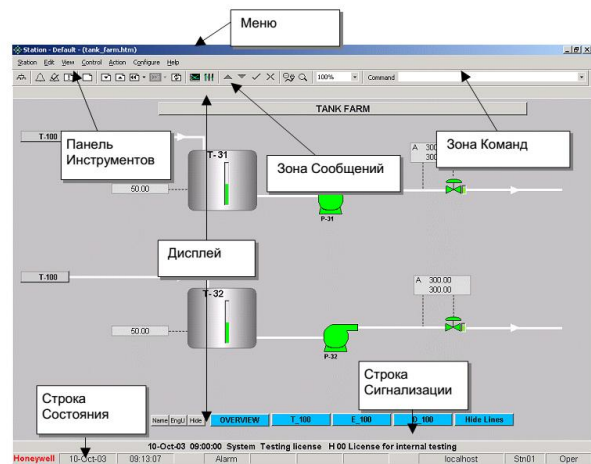
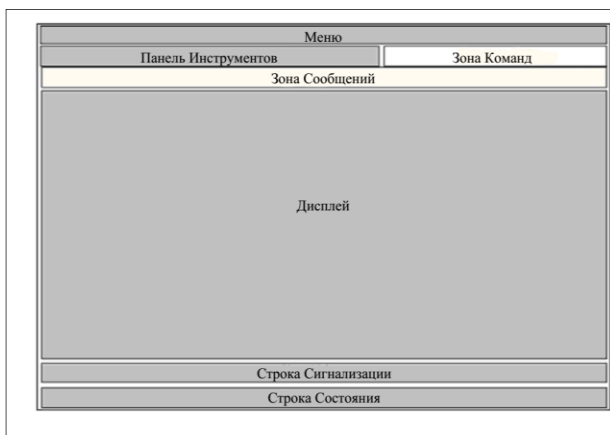
При запуске Станции пользователю не предлагается ввести в какой-либо форме идентификатор оператора или пароль. Начальные настройки уровня безопасности позволяют пользователям выполнять основные технологические функции, ассоциированные с уровнем пользователя OPER (например, подтверждение сигнализации и управление точками).

Для изменения уровня безопасности на более высокий, необходимо использовать только пароль.


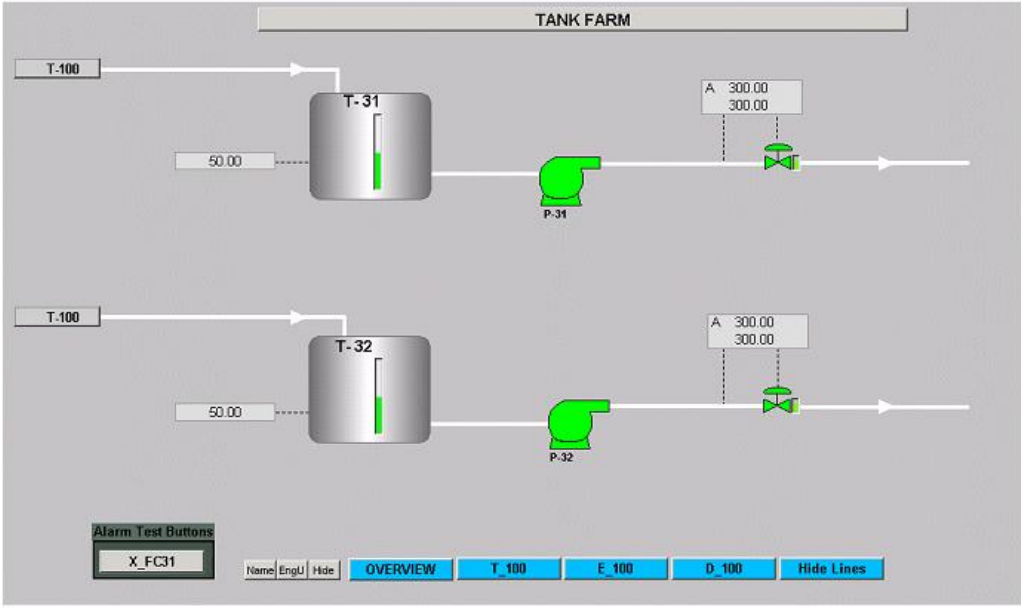
Назначение Зоны применяется к Станции, не к Оператору.

Распознавание разделов окна Станции и их функции

Станция Experion PKS имеет структурированную схему расположения экрана, которая делает наблюдение процесса более эффективным



Раздел	Описание
Меню	Выбор команд из меню Станции аналогичен выбору команд в других приложениях. Например, для вызова Сводки Событий, выберите View > Events > Event Summary .
Панель Инструментов	Нажатие кнопок на панели инструментов предоставляет быстрый доступ к часто запрашиваемым командам. Например, для вызова дисплея Сигнализации, нажмите кнопку Alarm (Сигнализация).

Зона Команд	Вводите команды в Зоне Команд .
	
Зона Сообщений	<p>Станция отображает объясняющие сообщения в зоне Сообщений.</p> <p>Например, если Вы попытаетесь изменить выход в автоматическом режиме (Auto), в зоне Сообщений появится сообщение <i>Invalid Mode</i> (Неправильный Режим).</p>
<p>Invalid Mode</p>	
Дисплей	<p>Каждый дисплей представляет собой отдельную <i>панель управления</i>, которую Вы используете для мониторинга и управления конкретной частью вашей системы.</p>
	
Строка Сигнализации	<p>В основном, эта строка отображает самую последнюю неподтвержденную сигнализацию.</p> <p>(Строка Сигнализации может быть скрыта в вашей системе, или может быть сконфигурирована для работы определенным образом.)</p>
<p>01-Dec-03 14:08:49 T100 X_FC18 PУНИИ U 00 T-100 Bottom To Storage Tank 385 m3/hr</p>	
Строка Состояния	<p>Предоставляет обзор состояния вашей системы.</p> <p>Например, мигание красного поля показывает, что имеется как минимум одна неподтвержденная сигнализация.</p>
<p>Honeywell 10-May-04 12:59:58 Alarm System devmserv1 Str04 Oper</p>	

Тема 23-24	Программный пакет UniSIM	
Технология проведения лекционного занятия		
Время 4 часа	Количество студентов 40-60 человек	
Вид занятия	Визуальная лекция	
План лекционного занятия	<ol style="list-style-type: none"> 1. Структура системы UniSim 2. Программное обеспечение; 3. UniSim Operations 4. UniSim Design 5. Ключевые особенности UniSim Design 6. Тренажёрные комплекс КТК-М 	
Цель учебного занятия: Формирование у студентов глубокого представления о целях и задачах химической науки и путях её дальнейшего развития		
Педагогические задачи:		Результаты обучения:
<ul style="list-style-type: none"> - Ознакомление с архитектуры систем управления UniSim. - Ознакомление с Структура системы, программное обеспечение. - Внедрение в сознание студентов основных характеристики тренажеров, требования к тренажеру, этапы создания тренажера. 		Студент: Имеет представление структура системы, программное обеспечение Излагает программирование задач контроллера и управление Характеризует требования к тренажеру и этапы создания тренажера
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.6. Приводит список используемой литературы по тренажерным системам. Знакомит студентов с рейтинговой оценкой знаний студентов. Объявляет тему и план занятия, знакомит с достижениями химической науки	1.1. Слушает и записывает
2-этап Основной	2.1. Обращает внимание студентов на вопросы плана занятия и обращает внимание на основные понятия. Проводит опрос 2.2. Продолжает излагать лекционный материал. На примерах приводит и объясняет структура системы, программное обеспечение 2.3. Обеспечивает студентом внимание и структура системы, программное обеспечение.	2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают
3-этап Итоговый	3.1. Делает заключительные выводы лекционной темы и заостряет внимание студентов на основные вопросы 3.2. Отмечает активных студентов 3.3. Даст задание на самостоятельную подготовку: Дать определение описание уровней безопасности в программные пакеты UniSim	3.1. Слушают и усваивают 3.2. Записывают задание

Лекция 23-24. Программный пакет UniSIM

План:

1. Структура системы UniSim
2. Программное обеспечение;
3. UniSim Operations
4. UniSim Design
5. Ключевые особенности UniSim Design
6. Тренажёрные комплекс КТК-М

Система UniSim является разработкой Honeywell и представляет собой набор унифицированных средств моделирования, позволяющих улучшить функционирование предприятия в течение всего жизненного цикла – от проектирования технологических процессов с использованием статических и динамических моделей, проверки системы управления и обучения операторов до использования в режиме реального времени для управления и оптимизации, мониторинга и планирования работы предприятия.

UniSim обеспечивает существенно расширенные возможности «off-line» и «on-line» моделирования для проектирования, инжиниринга, обучения операторов и оптимизации хода технологических процессов. Пользователи системы получают удобные возможности сбора и доступа к знаниям о технологическом процессе. Это повышает рентабельность предприятия и обеспечивает максимально выгодные условия возврата инвестиций в технологии моделирования.

В рамках линейки UniSim предлагаются следующие продукты, применяемые при обучении и тренировке: система динамического моделирования для повышения эффективности управления технологическими процессами и обучения персонала UniSim Operations и система имитационного моделирования UniSim Design.

Программный пакет UniSim Operations вобрал в себя самые лучшие стороны предыдущего решения Honeywell, Shadow Plant, а также программного обеспечения для имитационного моделирования, приобретенного у компании AspenTech. Программный пакет использует распределенную архитектуру, что сняло ограничения, связанные с нехваткой вычислительных ресурсов.

Модели могут полностью строиться на базе реальных данных, полноценных физических характеристик, и в то же время вычисления могут реализовываться с минимальной временной скажностью. При моделировании в реальном времени это позволяет обеспечить: более высокую точность, как в задачах обучения операторов, так и в инжиниринге; возможность проводить моделирование сложных технологических объектов, содержащих несколько установок; возможность воспроизводить сложные приложения (в т.ч. системы управления).

Основными компонентами Тренажера UniSim являются:

- платформа UniSim;
- рабочее место инструктора;
- рабочие места операторов.

Платформа Unisim – это базовый механизм, обеспечивающий управление и синхронизацию различных задач и прикладных приложений для проведения высокоточного имитационного моделирования крупномасштабных процессов в режиме реального времени. Задачи и приложения обычно распределяются по нескольким ПК.

Рабочее место инструктора – это главный интерфейс для управления процессом обучения.

Инструктор работает с настраиваемым графическим интерфейсом (рисунок 1), который обеспечивает широкий набор возможностей, в частности:

- управление сессией тренинга
- выбор модели, создание/загрузка «моментальных снимков» (начальных состояний), автоматическое и ручное запоминание моментальных снимков, возврат к предыдущим начальным состояниям, инициализация данных с технологической установки;
- управление временем – приостановка, возобновление, ускорение и замедление моделирования;
- инструменты инструктора – экраны для мониторинга динамики процессов, переменные инструктора, функции полевого оператора (удаленного управления), экран много переменных трендов, видео с привязкой к моделируемым событиям, доступ к моделируемым переменным, экраны параметров технологических потоков, процедуры управления;
- оценка результатов – автоматические тренировочные упражнения, прогревание сценариев, внесение возмущений в ход моделируемых процессов, мониторинг тревожных сообщений и событий, контроль и оценка знаний, отчеты и протоколы;
- среда и механизм моделирования процессов (см. ниже);
- воспроизведение среды управления (см. ниже).

Стандартный пользовательский интерфейс UniSim Design, показанный на рисунке 2, обеспечивает среду моделирования процессов. Это наглядный интерактивный инструмент для представления технологических процессов, который позволяет настраивать конфигурацию моделей в графическом виде путем добавления модулей с палитры процессов, а также управлять моделируемыми элементами оборудования, соединяя

потоками их входы и выходы и определяя значения параметров. Кроме того, UniSim Design – это механизм имитационного моделирования процессов для разработки и исполнения как статических, так и динамических моделей. Расширенная система физических свойств и термодинамики поддерживает свойства веществ и материалов, типичных для разных отраслей.

Также важно, чтобы обучение проходило в условиях, приближенным к реальным. Лучше всего, когда оператор работает за настоящей консолью PCY. Поэтому системы UniSim обычно комплектуются двойным набором консолей, чтобы оператор имел доступ к тому же набору функций, который есть в диспетчерской. Однако возможно реализовать операторский интерфейс PCY и на обычных ПК.

Ключевое отличие между моделью собственно технологического процесса и моделью технологической установки реального предприятия состоит в том, что последняя предполагает высокоточное воспроизведение поведения распределенной системы управления (PCY) и системы противоаварийной защиты. Для некоторых PCY эти системы можно эмулировать в среде UniSim, в других случаях моделирующая среда связывается с эмулируемыми контроллерами, предоставляемыми поставщиками PCY.

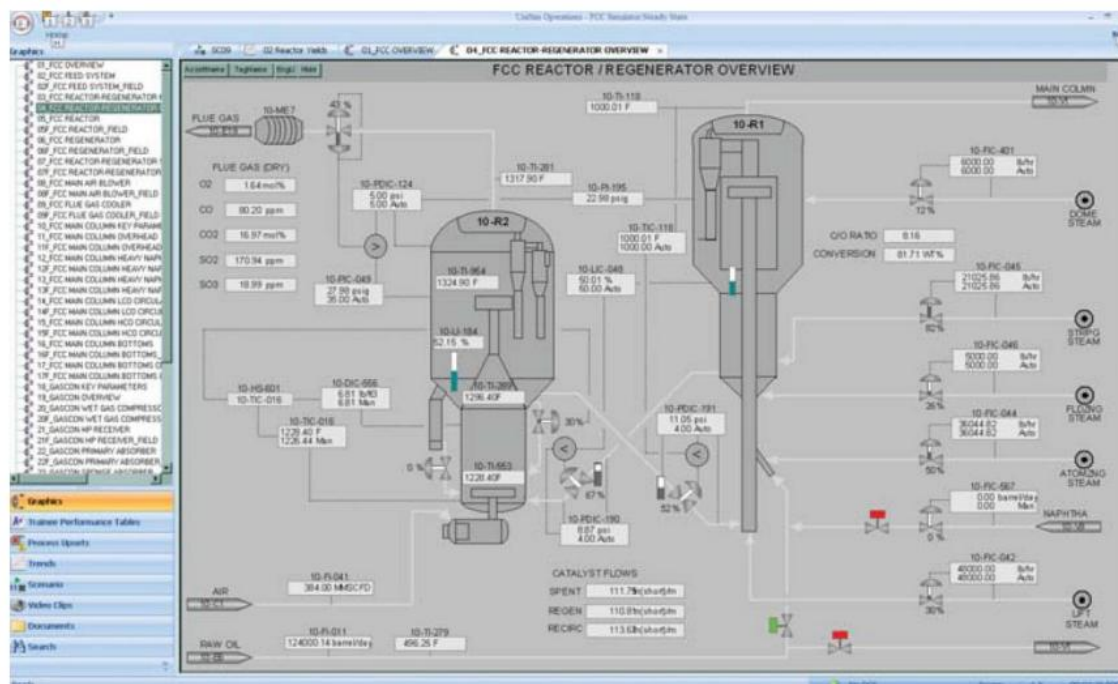


Рисунок 1. Интерфейс UniSim Operations

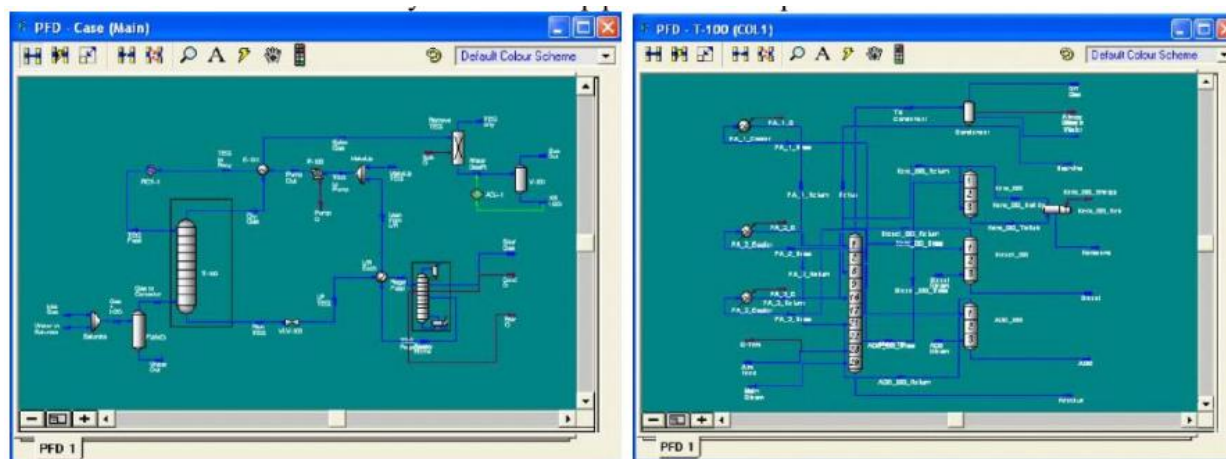


Рисунок 2. Интерфейс пользователя UniSim Operations

Программное обеспечение UniSim Design (последняя версия продукта R390 вышла весной 2009 г.) содержит последнюю версию HYSYS и, используемая не как стандартный интерфейс оператора в составе Unisim Operations, предлагает статистическое и динамичное моделирование, дизайн, контроль качества работы, оптимизацию и бизнес-планирование для предприятий по добыче и переработке нефти и газа, а также для предприятий химической промышленности.

Ключевыми характеристиками UniSim Design являются:

- удобная в использовании многооконная Windows среда (рисунок 2): технологические схемы обеспечивают наглядные графические представления ТП;

– мощный термодинамический пакет: обеспечивается аккуратный расчет физических свойств, свойств тепло- и массопереноса, фазового поведения для нефтегазовых, нефтеперерабатывающих и нефтехимических процессов; Поддерживается широкая база данных по компонентам, имеется возможность добавлять пользовательские компоненты;

– протокол ActiveX (OLE Automation): возможна интеграция с технологическими модулями и описаниями кинетики реакций, созданными пользователями, а также со специализированными пакетами физико-химических свойств. Обеспечивается легкая связь с программами MS Office через VBA;

– богатые библиотеки технологических модулей, реализующие в статических и динамических режимах различные процессы разделения, химические реакции, операции переноса тепла, ротационное оборудование, логические операции; указанные модули проверены на получение реалистических результатов и возможность моделирования разнообразных нештатных ситуаций, таких как опустошение и переполнение емкостей, обратные потоки, захват насосом инородной среды и пр. В тренажерной библиотеке – множество специализированных моделей технологических установок нефтепереработки, нефтехимии и химии, включая сложные установки первичного разделения нефти, гидроочистки, каталитического риформинга, каталитического крекинга и другие процессы различных лицензиаров. Кроме того, поскольку известный технологический лицензиар

– компания UOP – является дочерней компанией Honeywell, в библиотеке UniSim Design находятся эксклюзивные динамические модели ТП разработки UOP (Unionfining, Platforming, Tolaray, Isomar, Parex, Molex, CycleMAX, Unioncracking, Penex, Alkylation и др.).

Также в UniSim Design поддерживается полный набор моделей технологических аппаратов, включая дистилляционные колонны, реакторы, теплообменное оборудование, ротационное оборудование. Поддерживается работа с твердыми веществами и в смешанных средах (вибрационное сито, рукавный фильтр-пылеуловитель, центрифуга, конвейер, дробилка и ротационный измельчитель). Имеется большой выбор логических элементов, элементов конфигурирования системы управления. Реализуется моделирование разнообразных технологических ситуаций.

UniSim Design располагает специальными пакетами расчета термодинамики, обслуживающими моделирование всего спектра ТП. Вместе с новой моделью фтористо-водородных соединений в UniSim Design добавлен и пакет ее термодинамических свойств. Программный пакет для моделирования электролитов работает с гипотетическими компонентами, определенными пользователем. В пакет входят модель смешанных растворителей электролита и анализатор коррозии. Продвинутый инструментарий обеспечивает высокую функциональность UniSim Design на основе таких эксклюзивных опций, как матрица причинно-следственных зависимостей; планировщик событий (автоматизация операций, позволяющая создать последовательности действий при пуске и останове); новое средство верификации балансов (Simulation Balance Tool), обеспечивающее механизм проверки материального и энергетического балансов. Обеспечивается преемственность UniSim Design относительно системы HYSYS. В UniSim Design поддерживает чтение любых файлов моделирования, сохраненных в формате Aspen HYSYS 2006 или более старых версий. Также поддерживается сохранение файлов моделирования UniSim Design в формате HYSYS 2006. Таким образом, они могут быть использованы в обеих системах, что дает преимущество пользователю при переходе на UniSim Design.

Система UniSim Design представляет собой набор унифицированных средств моделирования, позволяющих улучшить функционирование предприятия в течение всего жизненного цикла от проектирования с использованием статических моделей, проверки системы управления и обучения операторов до использования в режиме реального времени для управления и оптимизации, а также поиска причин неисправностей и повышения качества эксплуатации.

Высокоточные динамические модели позволяют максимально приблизить процесс обучения к реальным задачам, что обеспечивает сокращение времени пуска, наилучшую организацию обучения, а также возможности раннего обнаружения проблем до возникновения неисправностей на реальном объекте.

Использование UniSim предоставляет следующие преимущества:

- Улучшение стратегий базового регулирования и усовершенствованного управления для повышения управляемости и эффективности производства
- Обучение персонала предприятия для обеспечения адекватной реакции в аварийных ситуациях и поддержания функционирования производства на безопасном и оптимальном уровне
- Использование сценариев What if (Что произойдет, если?) и точного анализа для идентификации оптимального проекта
- Гарантия правильного подбора технологического оборудования для обеспечения требуемой производительности и соблюдения технических условий
- Оценка влияния изменений сырья, нарушений режима и отказов оборудования на безопасность и надежность производства
- Мониторинг функционирования оборудования и выявление участков, требующих повышенного внимания, до появления реальных проблем на объекте

UniSim предоставляет существенно расширенные возможности «off line» и «on line» моделирования для проектирования и оптимизации хода технологических процессов.

UniSim можно легко соединить с системой Experion PKS, а также с системами управления сторонних производителей, таких как Invensys, ABB, Emerson и другие.

Ключевые особенности UniSim

Удобная в использовании многооконная Windows-среда:

- технологические схемы обеспечивают удобные графические представления ТП, включая опции «вырезать», «копировать», «вставить», «автоматически связать», «создать подсхему» и др.
- Мощный термодинамический пакет: обеспечивается точный расчет физических свойств, свойств переноса, фазового поведения для нефтегазовых, нефтеперерабатывающих и нефтехимических процессов.
- Поддерживается широкая БД по компонентам, имеется возможность добавлять пользовательские компоненты
- Поддержка OLE Automation: возможна интеграция с технологическими модулями и описаниями кинетики реакций, созданными пользователями, а также со специализированными пакетами физико-химических свойств. Обеспечивается легкая связь с программами MS Office через VBA.

Программа UniSim Design имеет в своем составе следующие модели технологических аппаратов

- Колонны дистилляции
- Абсорберы, адсорберы, десорберы
- Реакторы
- Теплообменники, холодильники, печи
- Ротационное оборудование (насосы, компрессоры)
- Оборудование для работы с твердыми веществами и в смешанных средах (вибрационное сито, рукавный фильтр-пылеуловитель, центрифуга, конвейер, дробилка и ротационный измельчитель) и др.

Кроме того, имеются следующие СПЕЦИАЛИЗИРОВАННЫЕ модели технологических установок нефтепереработки, нефтехимии и химии:

- Изомеризация
- Первичное разделение нефти
- Гидроочистка
- Каталитический риформинг
- Каталитический крекинг
- Производство серной кислоты
- Производство аммиака и др.

Также имеется богатый выбор логических элементов и элементов конфигурирования системы управления и ПАЗ.

КЛЮЧЕВЫЕ ОСОБЕННОСТИ UniSim Design

• **Удобная в использовании многооконная Windows-среда:** технологические схемы обеспечивают удобные графические представления ТП, включая опции «вырезать», «копировать», «вставить», «автоматически связать», «создать подсхему» и др.

• **Мощный термодинамический пакет:** обеспечивается точный расчет физических свойств, свойств переноса, фазового поведения для нефтегазовых, нефтеперерабатывающих и нефтехимических процессов. Поддерживается широкая БД по компонентам, имеется возможность добавлять пользовательские компоненты

• **Поддержка OLE Automation:** возможна интеграция с технологическими модулями и описаниями кинетики реакций, созданными пользователями, а также со специализированными пакетами физико-химических свойств. Обеспечивается легкая связь с программами MS Office через VBA

UniSim Operations: ProcessSync

• Инициализирует динамическую тренажерную модель реальными данными с установки из архивов PHD: задания на регуляторы, положения клапанов, показания датчиков, состояния оборудования и ручной арматуры

• Включает действия операторов в учебную программу, обеспечивая обучение на случившихся инцидентах

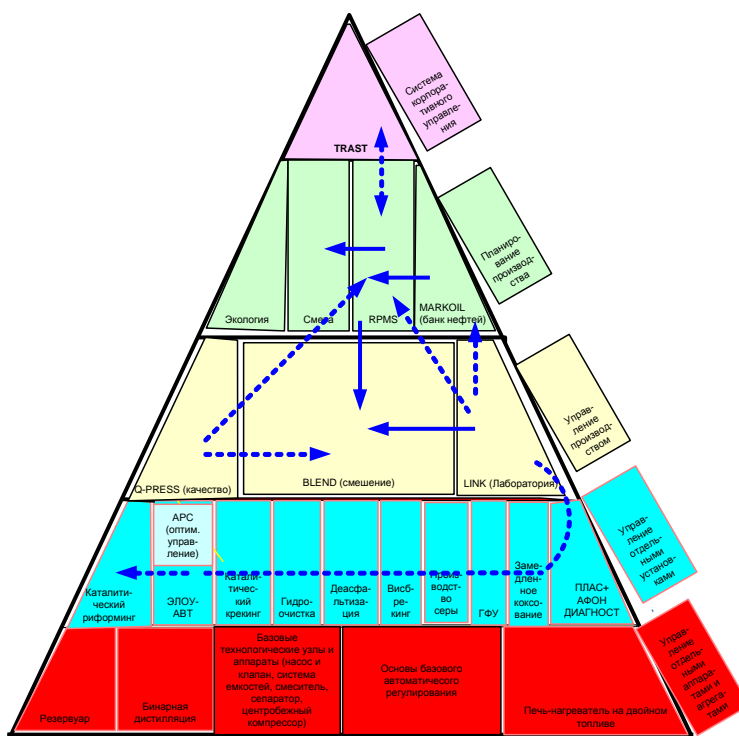
• Архивирует инциденты для анализа, обучения и обмена опытом между операторами

• Предоставляет операторам шанс на модели «забежать вперед» и эффективно проверить различные варианты управлений и их последствий

• Если рассматриваемый случай заслуживает внимания, его можно сохранить и использовать в дальнейшем обучении. Это обеспечивает:

- Повышение качества обучения
- Улучшение управления процессом
- Более уверенные действия операторов в нештатных ситуациях
- Избежание повторных ошибок
- Расширение учебной программы
- Максимальное использование инвестиций в тренажерный проект

- Тренажеры типовых установок
 - Семинары и курсовые работы для студентов-технологов, магистрантов, аспирантов
 - ВУЗы и колледжи в Москве, Салавате, Ачинске
- Система UniSim Design
 - Лабораторные и курсовые работы для студентов-технологов и магистрантов
 - ВУЗы в Москве, Санкт-Петербурге, Уфе, Омске, Иркутске, Нижнекамске и пр.
- Проект «Виртуальный НПЗ»
 - Моделирование НПЗ в целом, включая технологические установки, системы оптимального управления, системы планирования, системы компаундирования, лабораторию, экологическую систему, банки качества нефти, корпоративное управление и пр.
 - Российский государственный университет нефти и газа им. И.М. Губкина



Тренажёрные комплекс КТК-М

Много функциональный компьютерный тренажерный комплекс КТК-М (сертификат качества РОСС0001.04ЯА 747 от 14 февраля 2003 г.) Универсальный компьютерный тренажерный комплекс КТК-М имеет сетевую архитектуру и состоит из моделирующего компьютера (Рабочей станции инструктора), работающий в операционной среде Windows NT/2000/XP и одной или нескольких Рабочих станций обучаемых операторов (PCO). В качестве PCO может использоваться как реальная операторская станция распределенной системы управления, такие эмуляция (точное воссоздание)на персональном компьютере. Основные функции Рабочей станции инструктора:

- Выбор модели и операторского интерфейса;
- Выбор начальных условий моделирования;
- Имитация отказов оборудования;
- Активизация функций удаленного управления;
- Имитация изменения внешних условий;
- Просмотр значений, трендов и сигнализаций переменных ТП;
- Просмотр и изменение положений и параметров аналоговых и дискретных управляющих переменных;
- Протоколирование сеанса обучения;
- Фильтрация по типам событий, сохранение, редактирование и печать протоколов;
- Создание и активизация сценариев обучения.

Тренажерные системы МЧС

В качестве руководителя службы спасения вам предстоит столкнуться с множеством чрезвычайных ситуаций и спасти не одну человеческую жизнь. В вашем распоряжении спасательные службы, полиция, спец. подразделения и скорая помощь. Вам противостоят могущественные силы природы и разрушительные техногенные катаклизмы. "Служба спасения 911" - это не просто игра, это настоящая борьба за жизнь.

Особенности игры: смена дня ночи, реалистичные погодные эффекты; - 25 сценариев: от мелких происшествий до крупномасштабных катастроф; более 20 типов пожарных юнитов; более 10 полицейских юнитов; более 10 спасательных юнитов, включая вертолеты и спец. подразделения с поисковыми собаками.

Системные требования: Pentium 800 MHz, оперативная память: 256 Mb, видео: с 32 Mb RAM, CD-ROM:

8x

Тема 25, 26	ПРОГРАММИРУЕМЫЕ КОНТРОЛЛЕРЫ SIMATIC STEP 7. ОБЩИЕ СВЕДЕНИЯ.	
Технология проведения лекционного занятия		
Время 4 часа	Количество студентов 40-60 человек	
Вид занятия	Введение. Визуальная лекция	
План лекционного занятия	<ol style="list-style-type: none"> 1. Обзор STEP 7 2. Стандартный пакет STEP 7 3. Расширенное использование стандартного пакета STEP 7 4. Человеко-машинный интерфейс 5. Проектирование решения задачи автоматизации 6. Описание отдельных функциональных областей 7. Определение требований безопасности 	
Цель учебного занятия: Формирование у студентов глубокого представления о стандартном пакете STEP 7		
Педагогические задачи:		Результаты обучения:
<ul style="list-style-type: none"> - Ознакомление с основными понятиями стандартного пакета STEP 7; - Ознакомление с расширенным использованием стандартного пакета STEP 7; - Внедрение в сознание студентов инструментальных средств для связи с Windows 		<p>Студент:</p> <ul style="list-style-type: none"> . Имеет представление о примерах промышленного процесса смешивания и разделение процесса на связанные группы областей. . Описывает отдельные функциональные области в стандартном пакете STEP 7; . Характеризует основные определение требований безопасности;
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.2. Сообщает тему занятия и план, ожидаемые результаты	1.1. Слушает и записывает
2-этап Основной	<ol style="list-style-type: none"> 2.1. Обращает внимание студентов на вопросы темы лекционного плана, основные понятия, правила, законы, опрос 2.2. Продолжает излагать лекционный материал. Комментирует задачи, решаемые для большинства проектов и соответствующие им базовые процедуры комплекты в STEP 7. 2.3. Объясняет состав пользовательского интерфейса программного пакета STEP 7. 2.4. Объясняет языки программирования в STEP 7. 2.5. Подчеркивает студентам основные понятия и обращает внимание на основные моменты лекции для записи студентами и подчеркивает самостоятельное повторения 	<ol style="list-style-type: none"> 2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают 2.3. Записывают и запоминают 2.4. Записывают и запоминают 2.5. Записывают основные понятия и приводят примеры
3-этап Итоговый	<ol style="list-style-type: none"> 3.1. Делает выводы по лекции и обращает внимание студентов на основные вопросы 3.2. Отмечает активных студентов и поощряет их 	<ol style="list-style-type: none"> 3.1. Слушают и усваивают 3.2. Записывают задание на самоподготовку

Лекция 25, 26.
Программируемые контроллеры Simatic Step 7. Общие сведения.

План:

1. Обзор STEP 7
2. Стандартный пакет STEP 7
3. Расширенное использование стандартного пакета STEP 7
4. Человеко-машинный интерфейс
5. Проектирование решения задачи автоматизации
6. Описание отдельных функциональных областей
7. Определение требований безопасности

1. Обзор STEP 7

STEP 7 – это пакет стандартного программного обеспечения, используемый для конфигурирования и программирования программируемых логических контроллеров SIMATIC. Он является частью промышленного программного обеспечения SIMATIC. Имеются следующие версии стандартного пакета STEP 7:

- STEP 7 Micro/DOS и STEP 7 Micro/Win для относительно простых автономных приложений на SIMATIC S7-200.

- STEP 7 Mini для относительно простых автономных приложений на SIMATIC S7-300 и SIMATIC C7-620.

- STEP 7 для приложений на SIMATIC S7-300/S7-400, SIMATIC M7-300/M7-400 и SIMATIC C7 с более широким набором функций:

- Может быть расширен по выбору программными продуктами, имеющимися в промышленном программном обеспечении SIMATIC

- Возможность назначения параметров функциональным модулям и коммуникационным процессорам

- Принудительный и многопроцессорный режим

- Связь через глобальные данные

- Управляемая событиями передача данных с использованием коммуникационных функциональных блоков

При решении задачи автоматизации с помощью STEP 7 появляется ряд основных задач. Следующий рисунок 1. показывает задачи, которые должны быть решены для большинства проектов, и ставит им в соответствие базовые процедуры.

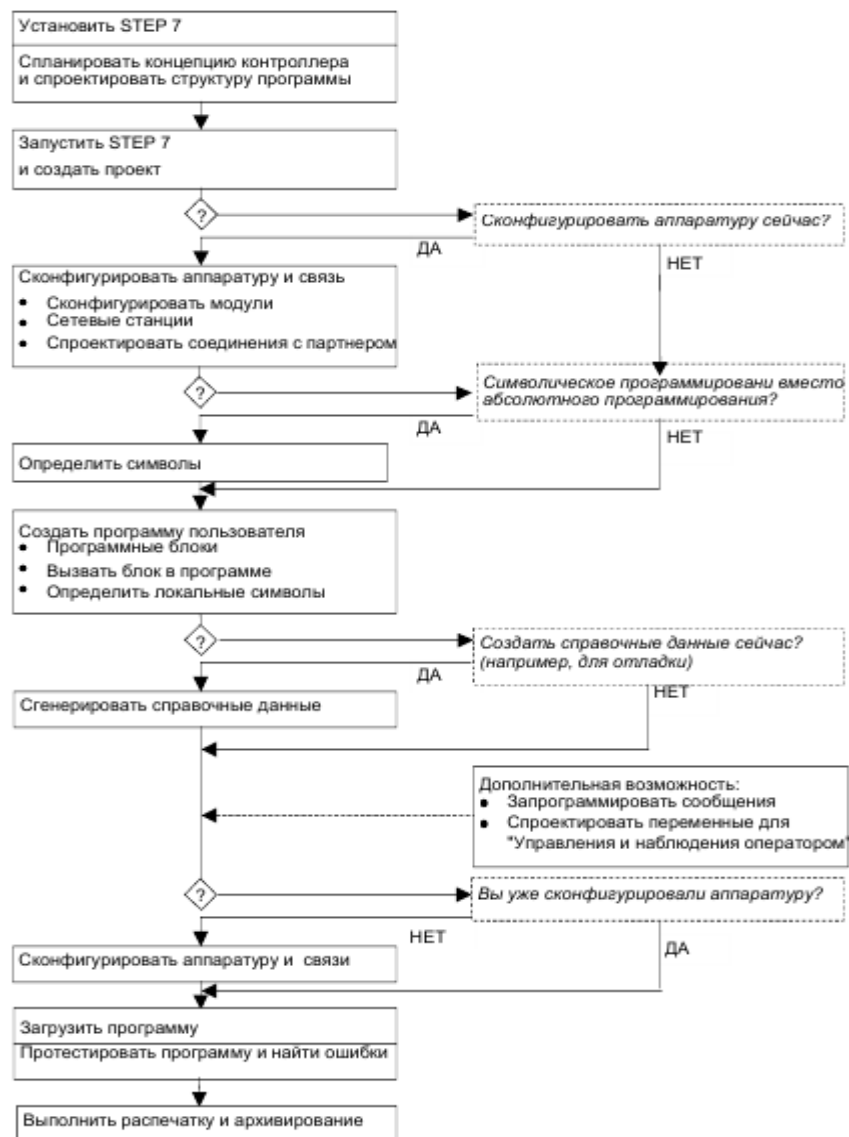


Рис. 1. Задачи, решаемые для большинства проектов и соответствующие им базовые процедуры.

Как показано на рисунке 1, в вашем распоряжении есть две альтернативы:

- Вы можете сначала сконфигурировать аппаратуру, а затем запрограммировать блоки.
- Вы можете, однако, сначала запрограммировать блоки, не конфигурируя аппаратуру. Это рекомендуется для работ, связанных с эксплуатацией и обслуживанием, например, для встраивания программных блоков в существующий проект.

Краткое описание отдельных шагов

- Установка и авторизация

При первом использовании STEP 7, установите его и перенесите авторизацию с дискеты на жесткий диск.

- Спланируйте концепцию использования вашего контроллера

Перед началом работы со STEP 7 спланируйте решение задачи автоматизации от деления процесса на отдельные задачи до создания диаграммы конфигурации.

- Спроектируйте структуру программы

Преобразуйте задачи, описанные в эскизном проекте вашего контроллера, в структуру программы, используя блоки, имеющиеся в STEP 7.

- Запустите STEP 7

STEP 7 запускается из пользовательского интерфейса Windows 95/98/NT.

- Создайте структуру проекта

Проект похож на папку, в которой все данные хранятся в виде иерархической структуры и доступны вам в любое время. После создания проекта все остальные задачи выполняются в этом проекте.

- Сконфигурируйте станцию

При конфигурировании станции вы указываете, какой программируемый контроллер вы хотите использовать; например, SIMATIC 300, SIMATIC 400, SIMATIC S5.

- Сконфигурируйте аппаратуру

При конфигурировании аппаратуры вы указываете в конфигурационной таблице, какие модули вы хотите использовать для решения своей задачи автоматизации и какие адреса должны быть использованы для доступа к модулям из программы пользователя. Модулям также могут быть назначены свойства с помощью параметров.

- Спроектируйте сети и коммуникационные связи

Основой для коммуникаций является предварительно спроектированная сеть. Для этого вам нужно будет создать подсети, необходимые для ваших задач автоматизации, установить свойства подсетей и установить свойства сетевых подключений и всех коммуникационных связей, требуемых для сетевых станций.

- Определите символы

Вы можете определить в таблице символов локальные или совместно используемые символы, имеющие более наглядные имена, для использования вместо абсолютных адресов в своей пользовательской программе.

- Создайте программу

Используя один из доступных языков программирования, создайте программу, связанную с модулем или независимую от модуля, и сохраните ее в виде блоков, исходных файлов или схем.

Только для S7: сгенерируйте и проанализируйте справочные данные. Вы можете использовать эти справочные данные для облегчения отладки и модификации программы пользователя.

- Спроектируйте сообщения

Сообщения, относящиеся к блокам, создаются, например, с помощью их текстов и атрибутов. Используя передающую программу, вы переносите созданные данные о конфигурации сообщений в базу данных системы взаимодействия с оператором (например, SIMATIC WinCC, SIMATIC ProTool).

- Спроектируйте переменные для управления и наблюдения оператором

Вы создаете переменные для управления и наблюдения оператором один раз в STEP 7 и назначаете им требуемые атрибуты. Используя передающую программу, вы переносите созданные переменные для управления и наблюдения оператором в базу данных системы взаимодействия с оператором WinCC.

- Загрузите программы в программируемый контроллер.

Только для S7: после завершения конфигурирования, назначения параметров и программирования задач вы можете загрузить всю свою пользовательскую программу или отдельные блоки из нее в программируемый контроллер (программируемый модуль для вашего аппаратного решения). CPU уже содержит операционную систему.

- Протестируйте программу
- Наблюдайте за работой, диагностируйте аппаратуру

Причина неисправности модуля определяется отображением информации о модуле в режиме online. Причины ошибок в обработке программы пользователя определяются с помощью диагностического буфера и содержимого стеков. Вы можете также проверить, может ли программа пользователя исполняться на конкретном CPU.

- Задокументируйте установку

После создания проекта/установки имеет смысл выполнить четкое документирование данных проекта, чтобы облегчить дальнейшее редактирование проекта и любую деятельность по обслуживанию. DOCPRO, дополнительное инструментальное средство для создания и управления документацией на установку, позволяет структурировать данные проекта, представить их в форме руководства по монтажу и распечатать их в обычном формате.

2 Стандартный пакет STEP 7

Языки программирования SIMATIC и встроенные в STEP 7 представления языков соответствуют требованиям стандарта EN 61131-3 или IEC 1131-3. Стандартный пакет работает в операционной системе Windows 95/98/NT и соответствует графической и объектно-ориентированной философии работы Windows.

Стандартное программное обеспечение оказывает вам поддержку на всех стадиях процесса решения задачи автоматизации, таких как:

- Создание и управление проектами
- Конфигурирование и назначение параметров аппаратуре и связям
- Управление символами
- Создание программ, например, для программируемых контроллеров S7
- Загрузка программ в программируемые контроллеры
- Тестирование системы автоматизации
- Диагностика неисправностей установки

Пользовательский интерфейс программного пакета STEP 7 спроектирован так, чтобы удовлетворить самым последним достижениям эргономики, и облегчает вам начало работы.

Стандартный пакет STEP 7 предоставляет ряд приложений (инструментальных средств) внутри программного пакета (рис.2)



Рис. 2. Состав стандартного пакета STEP 7

SIMATIC Manager

SIMATIC Manager управляет всеми данными, относящимися к проекту автоматизации – независимо от того, для какой системы программного управления (S7/M7/C7) они спроектированы. Инструментальные средства, необходимые для редактирования выбранных данных, запускаются автоматически SIMATIC Manager'ом.

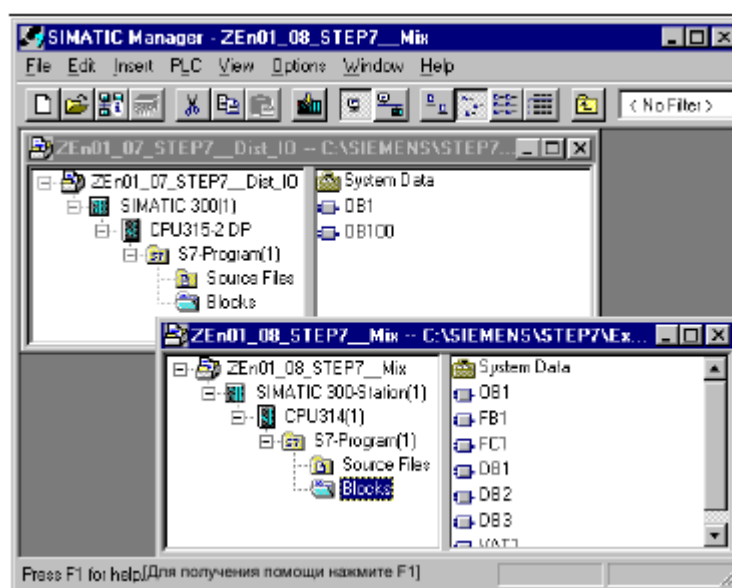


Рис. 3. Автоматически запускаемые инструментальные средства в SIMATIC Manager: File – файл, Edit – редактирование, Insert – вставка, PLC – ПЛК (программируемый логический контроллер), View – вид, Options – параметры, Window – окно, Help – помощь, No Filter – фильтр отсутствует, Source Files – исходные файлы, Blocks – блоки, System Data – системные данные

Редактор символов

С помощью редактора символов вы управляете всеми совместно используемыми символами. Доступны следующие функции:

- Установка символических имен и комментариев для сигналов процесса (входов/выходов), маркеров и блоков
- Функции сортировки
- Импорт/экспорт в другие программы/из других программ Windows

Таблица символов, созданная этим инструментальным средством, доступна и всем другим инструментам. Следовательно, любые изменения свойств символа автоматически распознаются всеми инструментальными средствами.

Диагностика аппаратуры. Эти функции предоставляют вам обзор состояния программируемого контроллера. Обзор может отображать символы, чтобы показать, неисправен какой-либо модуль или нет. Двойной щелчок на неисправном модуле отображает подробную информацию о неисправности. Объем этой информации зависит от конкретного модуля:

- Отображение общей информации о модуле (например, номер для заказа, версия, имя) и состояния модуля (например, неисправен)
- Отображение неисправностей модуля (например, неисправность канала) для центрального устройства и slave-устройств DP
- Отображение сообщений из диагностического буфера

Для CPU отображается следующая дополнительная информация:

- Причины неисправностей при обработке программы пользователя
- Отображение длительности цикла (самого длинного, самого короткого и последнего)
- Возможности и загрузка связей через MPI
- Отображение функциональных характеристик (числа возможных входов/выходов, меркеров, счетчиков, таймеров и блоков)

Языки программирования: контактный план, список операторов и функциональный план для S7-300 и S7-400 являются составной частью стандартного пакета.

- Контактный план (нем. KOP, англ. LAD) – это графическое представление языка программирования STEP 7. Его синтаксис для команд похож на релейно-контактные схемы: такая схема дает возможность проследить поток энергии между шинами при его прохождении через различные контакты, составные элементы и выходные катушки.

- Список команд (нем. AWL, англ. STL) – это текстовое представление языка программирования STEP 7, подобное машинному коду. Если программа написана в виде списка команд, то отдельные команды соответствуют шагам, с помощью которых CPU исполняет программу. Для облегчения программирования список команд расширен путем включения в него некоторых конструкций языков высокого уровня (таких как доступ к структурированным данным и параметры блоков).

- Функциональный план (нем. FUP, англ. FBD) – это графическое представление языка программирования STEP 7, использующее для представления логики логические блоки подобно булевой алгебре. Сложные функции (например, математические функции) могут быть представлены непосредственно в соединении с логическими блоками. Другие языки программирования доступны в виде дополнительных пакетов.

Конфигурирование аппаратуры – это инструментальное средство используется для конфигурирования и назначения параметров аппаратуре, используемой в проекте автоматизации. Имеются в распоряжении следующие функции:

- Для конфигурирования программируемого контроллера вы выбираете стойки из электронного каталога и размещаете выбранные модули в необходимых слотах на этих стойках.

- Конфигурирование децентрализованной периферии идентично конфигурированию центрального устройства. Поддерживаются также входы/выходы на уровне каналов.

- В процессе назначения параметров CPU вы можете установить такие свойства как поведение при запуске и контроль времени цикла под управлением меню. Поддерживается многопроцессорный режим.

Введенные данные хранятся в системных блоках данных.

- В процессе назначения параметров модулям все устанавливаемые параметры назначаются вами через диалоговые окна. Настройки, устанавливаемые с помощью двухпозиционных переключателей, отсутствуют. Присвоение параметров модулям производится автоматически при запуске CPU. Это значит, например, что модуль может быть заменен без назначения новых параметров.

- Назначение параметров функциональным модулям (FM) и коммуникационным процессорам (CP) производится с помощью инструментального средства Hardware Configuration [Конфигурирование аппаратуры] точно таким же образом, как и для других модулей. Для каждого FM и CP (включенного в сферу действия функционального пакета FM/CP) существуют специфические для модулей диалоговые окна и правила. Система препятствует неправильным вводам, предлагая только допустимые варианты в диалоговых окнах.

3. Расширенное использование стандартного пакета STEP 7

Стандартный пакет может быть расширен с помощью дополнительных программных пакетов, которые сгруппированы в следующие три класса программного обеспечения:

- Инструментальные средства для проектирования; это языки программирования высокого уровня и программное обеспечение, ориентированное на технологии.

- Рабочее (Run-Time) программное обеспечение; оно содержит готовые к использованию рабочие программы для производственного процесса.

- Human Machine Interfaces [человеко-машинные интерфейсы] (HMI); это программное обеспечение специально для управления и наблюдения оператором.

Инструментальные средства для проектирования – это инструментальные средства, ориентированные на задачи, которые могут быть использованы для расширения стандартного пакета. Инструментальные средства для проектирования включают в себя (рис. 4):

- Языки высокого уровня для программистов
- Графические языки для технического персонала
- Дополнительное программное обеспечение для диагностики, имитации, дистанционного обслуживания, документирования установки и т.д



Рис. 4. Состав инструментальных средств для проектирования

Языки высокого уровня. Следующие языки доступны как дополнительные пакеты для использования при программировании программируемых логических контроллеров SIMATIC S7-300/S7-400:

- S7 GRAPH – это язык программирования, используемый для программирования последовательного управления (состоящего из шагов и переходов). В этом языке ход процесса делится на шаги. Эти шаги содержат действия по управлению выходами. Переход от одного шага к другому управляется условиями переключения.

- S7 HiGraph – это язык программирования, используемый для описания асинхронных, непоследовательных процессов в виде графов состояний. Чтобы сделать это, установка разбивается на отдельные функциональные единицы, каждая из которых может принимать различные состояния. Эти функциональные единицы могут быть синхронизированы путем обмена сообщениями между графами.

- S7 SCL – это текстовый язык высокого уровня, удовлетворяющий требованиям стандарта EN 61131-3 (IEC 1131-3). Он содержит языковые конструкции, подобные имеющимся в языках программирования Pascal и C. Поэтому S7 SCL особенно пригоден для пользователей, привыкших работать с языками высокого уровня. Язык S7 SCL может быть использован, например, для программирования сложных и часто встречающихся функций.

Графический язык. CFC для S7 и M7 – это язык программирования для графического связывания существующих функций. Эти функции покрывают широкий диапазон от простых логических операций до сложных систем управления, работающих по замкнутому и разомкнутому циклу. Большое количество функций этого типа доступно в виде блоков в библиотеке. Вы программируете, копируя эти блоки в схему и соединяя их с помощью линий.

Дополнительное программное обеспечение:

- Borland C ++ (только для M7) содержит среду проектирования фирмы Borland.
- С помощью DOCPRO вы можете организовать все конфигурационные данные, которые вы создаете с помощью STEP 7, в руководства по монтажу. Это облегчает управление конфигурационными данными и позволяет подготовить информацию к распечатке в соответствии с указанными стандартами.

- HARDPRO – это система конфигурирования аппаратуры для S7-300, предназначенная для поддержки пользователя при крупномасштабном конфигурировании сложных задач автоматизации.

- M7-ProC/C++ (только для M7) позволяет встроить среду проектирования Borland для языков программирования C и C++ в среду проектирования STEP 7.

- Вы можете использовать S7 PLCSIM (только для S7) для имитации программируемых контроллеров S7, соединенных с устройством программирования или PC, в целях тестирования.

- S7 P DIAG (только для S7) предоставляет стандартизованную конфигурацию диагностики процесса для SIMATIC S7-300/S7-400. Используя диагностику процесса, вы можете обнаруживать дефекты и неисправные состояния вне программируемого контроллера (например, не достигнут конечный выключатель).

- TeleService позволяет вам программировать и обслуживать удаленные программируемые контроллеры S7 и M7 через телефонную сеть, используя ваше устройство программирования или PC.

Рабочее (Run-Time) программное обеспечение охватывает заранее запрограммированные решения, которые могут быть вызваны программой пользователя. Рабочее программное обеспечение непосредственно встроено в решение задачи автоматизации (рис. 5). Оно включает в себя:

- Регуляторы для SIMATIC S7, например, стандартный, модульный и нечеткий регулятор
- Инструментальные средства для связи программируемых контроллеров с приложениями Windows
- Операционную систему реального времени для SIMATIC M7

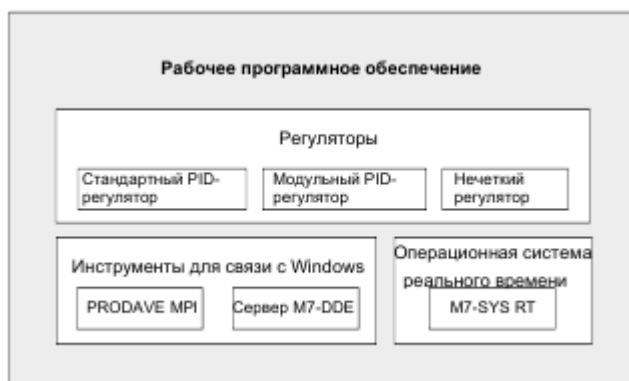


Рис. 5. Состав рабочего программного обеспечения

Регуляторы для SIMATIC S7:

- Стандартный PID-регулятор дает возможность встраивать в программу пользователя непрерывные регуляторы, импульсные регуляторы и ступенчатые регуляторы. Инструментальное средство назначения параметров со встроенной настройкой регулятора дает возможность настраивать регулятор на оптимальный режим за очень короткое время.
- Модульный PID-регулятор вступает в действие, если простой PID-регулятор недостаточен для решения вашей задачи автоматизации. Путем включения поставляемых стандартных функциональных блоков может быть спроектирована и настроена почти любая структура регулятора.
- С помощью нечеткого регулятора (Fuzzy Control) могут создаваться системы с нечеткой логикой. Эти системы вступают в действие, когда процессы очень сложны или не могут быть описаны математически, поведение процессов непредсказуемо или появляются нелинейности, но доступно экспериментальное исследование действующего процесса.

Инструментальные средства для связи с Windows:

- PRODAVE MPI – это набор инструментов для управления потоком данных между SIMATIC S7, SIMATIC M7 и SIMATIC C7. Он управляет потоком данных автономно через многоточечный интерфейс (MPI).
- С помощью сервера M7-DDE (Dynamic Data Exchange – Динамический обмен данными) приложения Windows могут быть связаны с переменными процесса в SIMATIC M7 без дополнительных затрат на программирование.

Операционная система реального времени:

- M7-SYS RT содержит операционную систему M7 RMOS 32 и системные программы. Это предпосылка для использования M7-ProC/C++ и CFC для пакетов SIMATIC M7.

4. Человеко-машинный интерфейс

Человеко-машинный интерфейс (Human Machine Interface, HMI) – это программное обеспечение специально для управления и наблюдения за процессом со стороны оператора в SIMATIC (Рис. 6). Он состоит из следующих частей:

- Открытая система визуализации процесса SIMATIC WinCC – это базовая система взаимодействия с оператором со всеми важными функциями управления и наблюдения оператора, которые могут быть использованы в любой отрасли промышленности и с любой технологией.
- SIMATIC ProTool и SIMATIC ProTool/Lite – это современные инструментальные средства для конфигурирования панелей оператора SIMATIC (OP) и компактных устройств SIMATIC C7.
- ProAgent дает возможность целенаправленной и быстрой диагностики в установках и машинах путем установления информации о месте и причине неисправности.

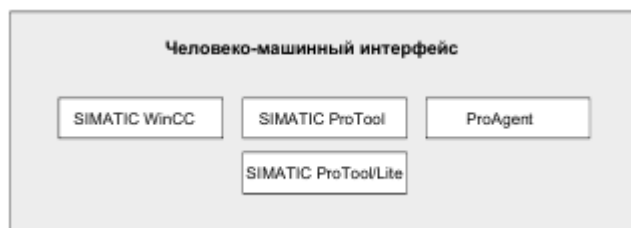


Рис. 6. Состав человеко-машинного интерфейса.

5. Проектирование решения задачи автоматизации

Существует много способов планирования проекта автоматизации. Основная последовательность действий, которую вы можете использовать для любого проекта, проиллюстрирована на следующем рисунке 7.

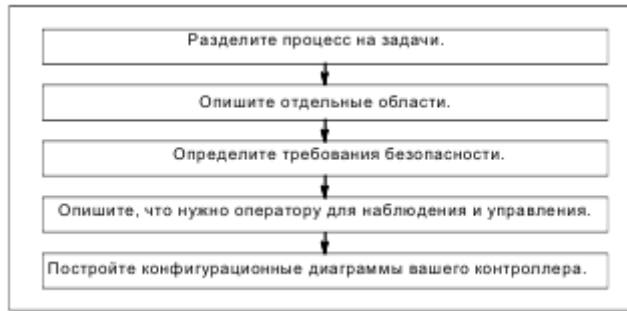


Рис. 7. Основная последовательность действий при планировании проекта автоматизации

Процесс автоматизации состоит из ряда отдельных задач. Путем выделения групп связанных задач внутри процесса и последующего разбиения этих групп на более мелкие задачи может быть определен даже самый сложный процесс.

Следующий пример промышленного процесса смешивания может быть использован для иллюстрации того, как представить процесс в виде функциональных областей и отдельных задач (рис. 8):



Рис. 8. Пример промышленного процесса смешивания

После определения процесса, подлежащего управлению, разделите процесс на связанные группы областей (рис.9):

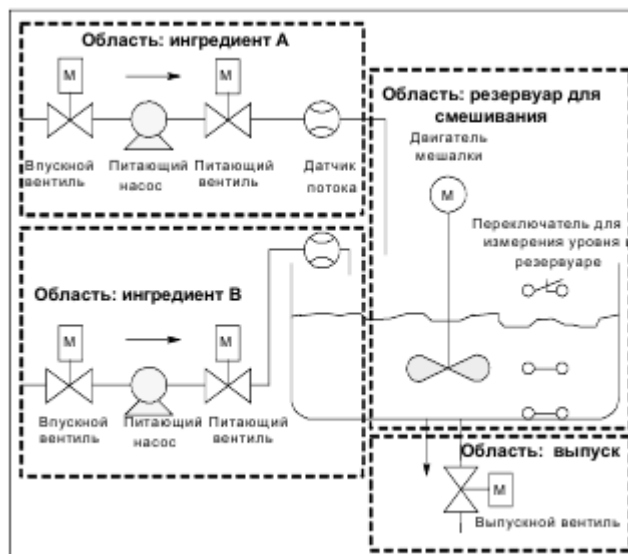


Рис. 9. Разделение процесса на связанные группы областей

Так как каждая группа разделена на более мелкие задачи, то задачи, необходимые для управления этой частью процесса, становятся менее сложными. В нашем примере промышленного процесса смешивания можно

выделить четыре отдельные области (см. следующую таблицу). В этом примере область для ингредиента А содержит такое же оборудование, как и область для ингредиента В.

Таблица 1.

Функциональная область	Используемое оборудование
Ингредиент А	Питающий насос для ингредиента А Впускной вентиль для ингредиента А Питающий вентиль для ингредиента А Датчик потока для ингредиента А
Ингредиент В	Питающий насос для ингредиента В Впускной вентиль для ингредиента В Питающий вентиль для ингредиента В Датчик потока для ингредиента В
Резервуар для смешивания	Двигатель мешалки Переключатель для измерения уровня в резервуаре
Выпуск	Выпускной вентиль

6. Описание отдельных функциональных областей

Описывая каждую область и задачу внутри вашего процесса, вы не только определяете функционирование каждой области, но и различные элементы, управляющие этой областью. Они включают в себя:

- электрические, механические и логические входы и выходы для каждой задачи
- блокировки и зависимости между отдельными задачами

Промышленный процесс смешивания в нашем примере использует насосы, двигатели и вентили. Они должны быть точно описаны для определения рабочих характеристик и типа блокировок, необходимых во время работы. В следующих таблицах приведены примеры описания оборудования, используемого в промышленном процессе смешивания. Завершив описание, вы можете его также использовать для заказа необходимого оборудования.

Ингредиенты А/В: двигатели питающих насосов

1. Двигатели питающих насосов подают ингредиенты А и В в резервуар для смешивания.
 - Скорость потока: 400 л в минуту
 - Номинальная мощность: 100 кВт при 1200 об/мин.
2. Насосы управляются (пуск/остановка) со станции оператора, расположенной рядом с резервуаром для смешивания. Количество пусков подсчитывается в целях обслуживания. Как счетчики, так и индикаторы могут быть сброшены одной кнопкой.

3. Для работы насосов должны быть выполнены следующие условия:

- Резервуар для смешивания не полон.
- Впускной вентиль резервуара для смешивания закрыт.
- Аварийное отключение не активизировано.

Ингредиенты А/В: двигатели питающих насосов

4. Двигатели выключаются, если выполнены следующие условия:

- Датчик потока извещает об отсутствии потока через 7 секунд после запуска двигателя насоса.
- Датчик потока извещает, что поток прекратился.

Ингредиенты А/В: впускной и питающий вентили

1. Впускной и питающий вентили для ингредиентов А и В разрешают или запрещают подачу ингредиентов в резервуар для смешивания. Вентили имеют электромагнит с пружинным возвратом.

- Когда электромагнит активизирован, вентиль открыт.
 - Когда электромагнит не активизирован, вентиль закрыт.
2. Впускной и питающий вентили управляются программой пользователя.
 3. Для активизации вентилей должны быть соблюдены следующие условия:
 - Двигатель насоса должен уже работать в течение не менее 1 секунды.
 4. Насосы выключаются, если выполнены следующие условия:
 - Датчик потока извещает об отсутствии потока.

Двигатель мешалки

1. Двигатель мешалки смешивает ингредиент А с ингредиентом В в резервуаре для смешивания.
 - Номинальная мощность: 100 кВт при 1200 об/мин.
2. Двигатель мешалки управляется (пуск/останов) со станции оператора, расположенной рядом с резервуаром для смешивания. Количество пусков подсчитывается в целях обслуживания. Как счетчики, так и индикаторы могут быть сброшены одной кнопкой.
3. Для работы двигателя мешалки должны быть выполнены следующие условия:
 - Датчик уровня жидкости в резервуаре не выдает сигнала "Уровень в резервуаре ниже минимального"

- Выпускной вентиль резервуара для смешивания закрыт.
 - Аварийный выключатель не активизирован.
4. Двигатель мешалки выключается, если выполнено следующее условие:
- Тахометр не показывает достижения номинальной скорости в течение 10 секунд после запуска двигателя.

Выпускной вентиль

1. Выпускной вентиль разрешает выпуск смеси (самотеком) на следующей стадии процесса. Вентиль имеет электромагнит с возвратной пружиной:

- Если электромагнит активизирован, выпускной вентиль открыт.
 - Если электромагнит не активизирован, выпускной вентиль закрыт.
2. Выпускной вентиль управляется (открытие/закрытие) со станции оператора.
3. Выпускной вентиль может быть открыт при следующих условиях:
- Двигатель мешалки выключен.
 - Датчик уровня жидкости в резервуаре не выдает сигнала "Резервуар пуст".
 - Аварийный выключатель не активизирован.
4. Выпускной вентиль закрывается, если выполнено следующее условие:
- Датчик уровня жидкости в резервуаре выдает сигнал "Резервуар пуст".

Переключатели для измерения уровня в резервуаре

1. Переключатели в резервуаре для смешивания измеряют уровень в резервуаре и используются для блокировки питающих насосов и двигателя мешалки.

Список входов, выходов и входов/выходов

Сделав физическое описание каждого устройства, подлежащего управлению, нарисуйте диаграммы входов и выходов для каждого устройства или группы задач (рис.10).



Рис. 10. Диаграммы входов и выходов

Эти диаграммы соответствуют логическим блокам, подлежащим программированию.

В нашем примере промышленного процесса смешивания используются два питающих насоса и одна мешалка. Каждый двигатель управляется своим собственным "блоком двигателя", одинаковым для всех трех устройств. Этот блок требует шести входов: два для запуска и остановки двигателя, один для сброса обслуживающего дисплея, один для ответного сигнала о работе двигателя (двигатель работает/не работает), один для времени, в течение которого должен быть получен ответный сигнал, и один для номера таймера, используемого для измерения времени.

Логический блок требует также четырех выходов: два для индикации рабочего состояния двигателя, один для индикации неисправностей и один для индикации того, что двигатель подлежит обслуживанию.

Для активизации двигателя необходим также вход/выход. Он используется для управления двигателем, но в то же время редактируется и изменяется в программе для "блока двигателя" (рис. 11).

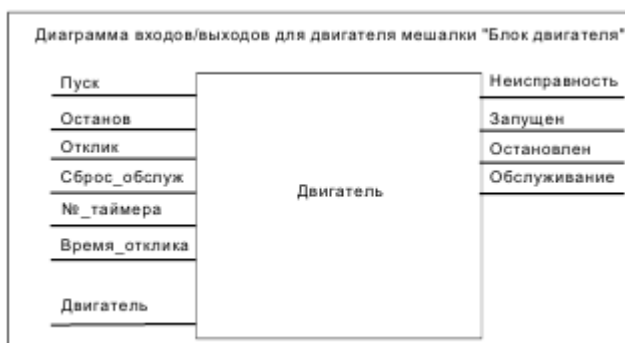


Рис. 11. Диаграмма входов и выходов для двигателя мешалки «Блок двигателя»

Каждый вентиль управляется собственным "блоком вентиль", одинаковым для всех используемых вентилях. Логический блок имеет два входа: один для открытия вентиля и один для его закрытия. У него

имеется также два выхода: один для индикации того, что клапан открыт, а другой для индикации того, что он закрыт.

Блок имеет вход/выход для активизации клапана. Он используется для управления клапаном, но в то же самое время редактируется и изменяется в программе для "блока клапана" (рис. 12).

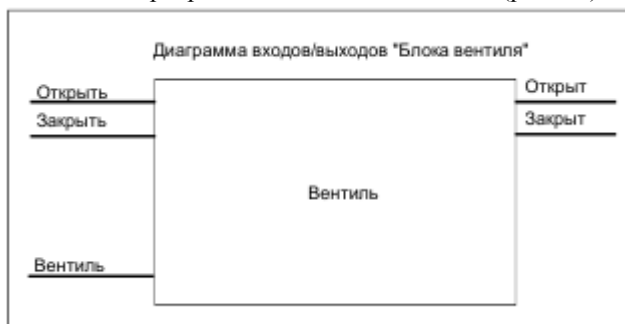


Рис. 12. Диаграмма входов и выходов «Блока клапана»

Определите, какие дополнительные элементы необходимы для обеспечения безопасности процесса – на основе юридических требований и корпоративной политики в области охраны здоровья и безопасности. В свое описание вам следует также включить все воздействия, которые элементы безопасности оказывают на области вашего процесса.

7. Определение требований безопасности

Выясните, какие устройства требуют аппаратно реализованных схем для удовлетворения требований безопасности. По определению, эти защитные схемы функционируют независимо от программируемого контроллера (хотя защитная схема в общем случае предоставляет интерфейс ввода/вывода для обеспечения координации с программой пользователя). Обычно проектируется матрица подключения каждого исполнительного устройства со своей собственной областью аварийного отключения. Эта матрица является основой для построения принципиальной схемы защитного устройства.

Для проектирования механизмов защиты действуйте следующим образом:

- Определите логические и механические/электрические блокировки между отдельными задачами автоматизации.
- Спроектируйте схемы, разрешающие ручное управление устройствами, относящимися к процессу, в случае аварии.
- Определите все остальные требования к защите для безопасного функционирования процесса.

Создание схемы защиты

Промышленный процесс смешивания в нашем примере использует следующую логику для своей схемы защиты:

- Один аварийный выключатель отключает следующие устройства независимо от программируемого контроллера (ПЛК):
 - питающий насос для ингредиента А
 - питающий насос для ингредиента В
 - двигатель мешалки
 - клапаны
- Аварийный выключатель находится на станции оператора.
- Один вход в контроллер индицирует состояние аварийного выключателя.

Описание требуемых для оператора устройств отображения и управления

Каждый процесс требует интерфейса с оператором, который обеспечивает вмешательство человека в процесс. Часть спецификации проекта включает в себя проект пульта оператора.

В промышленном процессе смешивания, описанном в нашем примере, каждое устройство может быть запущено или остановлено нажатием кнопки, расположенной на пульте оператора. Этот пульт оператора содержит индикаторы для отображения состояния функционирования (рис.13).

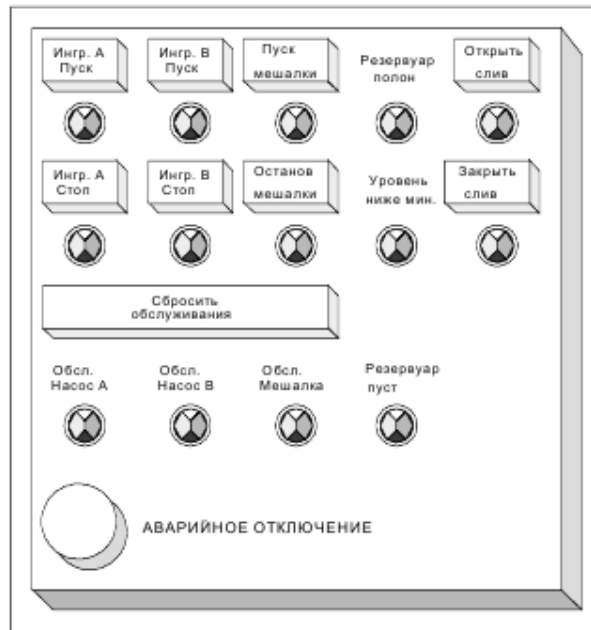


Рис. 13. Пульт оператора

Пульт содержит также индикаторные лампы для устройств, требующих обслуживания после определенного числа пусков, аварийный выключатель, с помощью которого процесс может быть остановлен немедленно. На пульте имеется также кнопка сброса для индикаторов обслуживания трех двигателей. С ее помощью вы можете отключить индикаторные лампы для двигателей, подлежащих обслуживанию, и сбросить соответствующие счетчики на 0.

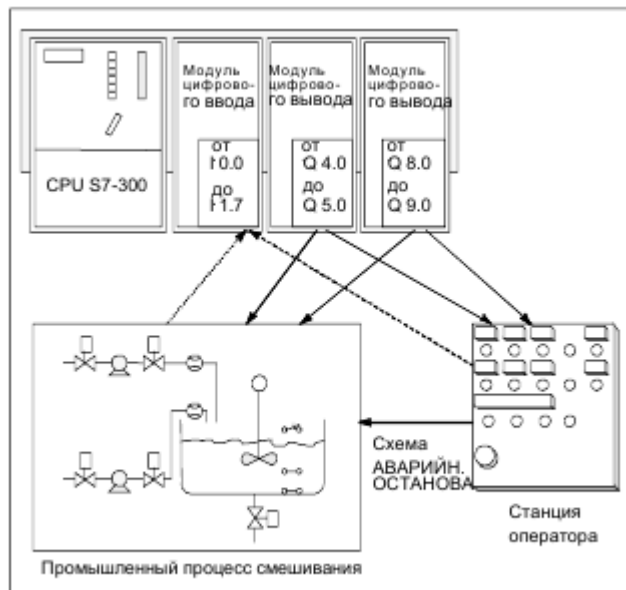


Рис. 14. Пример конфигурации S7 для промышленного процесса смешивания.

Составление конфигурационной диаграммы

После того как вы задокументировали требования к проекту, вы должны принять решение относительно типа управляющего оборудования, требующегося для проекта.

Принимая решение о том, какие модули вы хотите использовать, вы также определяете структуру программируемого контроллера. Составьте конфигурационную диаграмму, определяющую следующие аспекты:

- тип CPU
- количество и тип модулей ввода/вывода
- конфигурация физических входов и выходов

Следующий рисунок 14 иллюстрирует пример конфигурации S7 для промышленного процесса смешивания.

Контрольные вопросы:

1. Что такое STEP 7? Какие версии STEP 7 Вы знаете и в каких случаях они используются?
2. Каким образом запускается STEP 7?
3. Каким образом создается программа в STEP 7?
4. Что входит в состав программного пакета STEP 7?
5. Что такое SIMATIC Manager?
6. Как Вы понимаете «Контактный план языка программирования»?
7. Что Вы понимаете под «функциональным планом языка программирования»?
8. Что входит в инструментальные средства для проектирования?
9. Какие языки высокого уровня существуют в STEP 7?
10. Что входит в дополнительное программное обеспечение STEP 7?
11. Для чего служит ProAgent?

Тема 27	SIMATIC WINCC. ОБЩИЕ СВЕДЕНИЯ	
Технология проведения лекционного занятия		
Время 2 часа	Количество студентов 40-60 человек	
Вид занятия	Введение. Визуальная лекция	
План лекционного занятия	<ol style="list-style-type: none"> 1. Введение в SIMATIC WinCC 2. Расширения SCADA 3. Человеко-машинный интерфейс 4. Встроенное управление пользователями. SIMATIC Logon 5. Простое и эффективное проектирование. WinCC Graphics Designer 	
Цель учебного занятия: Формирование у студентов глубокого представления о SIMATIC WinCC		
Педагогические задачи:		Результаты обучения:
<ul style="list-style-type: none"> - Ознакомление с основными понятиями системы WinCC; - Ознакомление с основными опциями WinCC; - Внедрение в сознание студентов функций человеко-машинного интерфейса 		Студент: 13. Имеет представление о программном обеспечении современных ЭВМ 14. Излагает опции для масштабирования; 15. Характеризует основные особенности человеко-машинного интерфейса; 16. Характеризует основные возможности универсального использования WinCC.
Методика и техника обучения		-Информативная лекция, визуальной опрос, презентации
Средства обучения		- Текст лекции, проектор, плакаты, доска, мел
Форма обучения		-поток, группа, работа в парах
Условия обучения		-оснащенная аудитория

Технологическая карта лекционного занятия

Время этапов	Содержание деятельности	
	Преподаватель	Студент
1-этап Введение (10 мин)	1.3. Сообщает тему занятия и план, ожидаемые результаты	1.1. Слушает и записывает
2-этап Основной	2.1. Обращает внимание студентов на вопросы темы лекционного плана, основные понятия, правила, законы, опрос 2.2. Продолжает излагать лекционный материал. Комментирует понятия комплекты SIMATIC WinCC. 2.3. Объясняет контроль достоверности и изменений в WinCC. 2.4. Объясняет дополнения WinCC. 2.5. Приводит примеры графической системы WinCC 2.6. Подчеркивает студентам основные понятия и обращает внимание на основные моменты лекции для записи студентами и подчеркивает самостоятельное повторения	2.1. Студенты отвечают на вопросы 2.2. Слушают и записывают 2.3. Записывают и запоминают 2.4. Записывают и запоминают 2.5. Записывают и запоминают 2.6. Записывают основные понятия и приводят примеры
3-этап Итоговый	3.1. Делает выводы по лекции и обращает внимание студентов на основные вопросы 3.2. Отмечает активных студентов и поощряет их	3.1. Слушают и усваивают 3.2. Записывают задание на самоподготовку

Лекция 27.
Simatic WinCC. Общие сведения.

План:

1. Введение в SIMATIC WinCC
2. Расширения SCADA
3. Человеко-машинный интерфейс
4. Встроенное управление пользователями. SIMATIC Logon
5. Простое и эффективное проектирование. WinCC Graphics Designer

1. ВВЕДЕНИЕ В SIMATIC WINCC

SIMATIC WinCC представляет собой масштабируемую систему визуализации процессов, обладающую мощными функциями для контроля автоматизированных процессов. WinCC предоставляет все функциональные возможности SCADA под Windows для всех отраслей – от однопользовательских до распределенных многопользовательских систем с резервируемыми серверами и глобальных решений с web-клиентами. В частности, WinCC характеризуется абсолютной открытостью. Она без труда может быть использована в комбинации со стандартными и пользовательскими программами, что позволяет создавать человеко-машинные интерфейсы, точно соответствующие практическим требованиям. Благодаря открытым интерфейсам системные интеграторы, разрабатывающие программное обеспечение, могут создавать свои собственные приложения, используя WinCC как основу для системных расширений.

WinCC – это современная система с удобным пользовательским интерфейсом. Она подходит для применения, как в офисах, так и на производстве, функционально закончена и надежна в работе, эффективно проектируется и легко приспосабливается к решению как простых, так и сложных задач.

Вместе со встроенной базой данных WinCC образует информационный центр для вертикальной интеграции всего предприятия и с помощью набора средств Plant Intelligence увеличивает прозрачность производства. Для обеих систем справедливо утверждение о том, что комплексная автоматизация (Totally Integrated Automation) сокращает расходы как на разработку, так и на обеспечение жизненного цикла системы. Так, например, программное обеспечение человеко-машинного интерфейса имеет непосредственный доступ к базе переменных и сообщений контроллера SIMATIC, а также использует их коммуникационные параметры для настройки связи. Это позволяет с самого начала избежать больших затрат времени на многократный ввод данных и дополнительных источников ошибок. Кроме того, встроенные диагностические возможности обеспечивают поддержку обслуживающего персонала в течении всего жизненного цикла системы.

Системное программное обеспечение WinCC имеется в двух основных вариантах:

- Полный пакет WinCC (RC: лицензия на исполнение и проектирование)
- Пакет WinCC для исполнения (RT: лицензия на исполнение). Оба пакета имеются в вариантах, содержащих 128, 512, 2k, 8k, 64k, 100k, 150k или 256k так называемых внешних тегов.

Внешними тегами называют только переменные, которые связаны с контроллером или другими источниками данных через коммуникационные каналы WinCC.

При этом из одного тега может быть получено до 32 сообщений и до 256 определяемых пользователем сообщений по пределам аналогового сигнала. Кроме того, бесплатно предоставляются в распоряжение внутренние переменные, не связанные с процессом. Количество используемых внешних тегов можно увеличить с помощью пакетов Powerpack. Когда растет Ваше приложение, вместе с ним растет и WinCC. Так что Вы можете начинать работу с самым маленьким вариантом, а затем, по мере необходимости, наращивать теги с помощью пакетов Powerpack. С помощью пакетов PowerPack может быть также увеличено количество имеющихся в распоряжении архивных переменных с 512 (содержащихся в базовой поставке) до 1 500, 5 000, 10 000, 30 000 или 80 000 вплоть до 120 000.

WinCC – комплексная поддержка

WinCC предлагает услугу по обновлению программного обеспечения (Software Update Service, SUS) в виде пакета комплексной поддержки (Comprehensive Support), содержащего текущие обновления, а также много полезной информации и программного обеспечения для WinCC.

Автоматическая отправка текущих обновлений гарантирует, что в Вашем распоряжении всегда будет самая последняя версия WinCC.

Комплекты с исполняемым программным обеспечением WinCC

Комплекты SIMATIC Panel PC с WinCC – это оптимально согласованные комбинации аппаратных и программных средств человеко-машинного интерфейса. Эти комплекты дают вам следующие преимущества:

- Простота заказа
- Экономия затрат по сравнению с заказом по отдельности
- Оптимально согласованные аппаратные средства
- Системное протестированное решение

Опции WinCC

Отдельные функциональные или ориентированные на конкретные отрасли расширения программного обеспечения доступны в форме опций и дополнений WinCC (WinCC options и WinCC Addons).

Опции WinCC – это продукты от разработчиков SIMATIC. По этим продуктам Вы можете получить поддержку от наших консультативных служб и по центральной горячей линии. Они предлагают множество полезных расширений базовой системы WinCC и могут свободно комбинироваться в соответствии с Вашими потребностями. Здесь приведен обзор опций, имеющихся в распоряжении для SIMATIC WinCC V7:

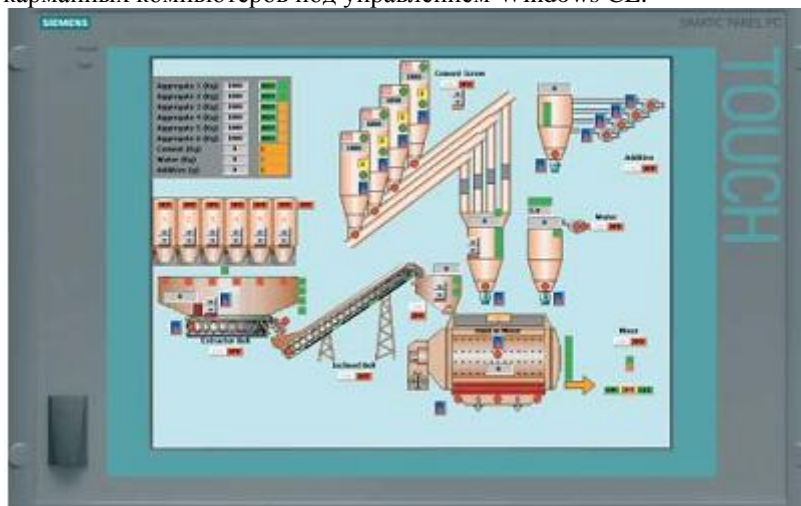
Опции для масштабирования

WinCC/Server – служит для расширения однопользовательской системы до мощной (распределенной) системы клиент-сервер, содержащей до 12 резервируемых серверов WinCC и 32 клиентов.

WinCC/Central Archive Server (центральный сервер архивов, CAS) – для создания расширяемого, централизованного, в случае необходимости резервируемого архива данных процесса, содержащего до 120 000 архивных переменных, на основе MS SQL-сервера.

WinCC/Web Navigator – предоставляет возможность управления и контроля установки через Интернет или Интранет с помощью MS Internet Explorer или поставляемой программы просмотра WinCC Web Viewer без необходимости внесения изменений в проект WinCC.

Решения для «тонких» клиентов допускают использование наряду с ПК также надежных локальных устройств и мобильных карманных компьютеров под управлением Windows CE.



Plant Intelligence и интеграция информационных технологий и бизнеса

Решающими факторами для комплексной интеграции информационных технологий и бизнеса и оптимизации производства посредством Plant Intelligence являются стандартные интерфейсы и мощные инструментальные средства для отображения, анализа и оценки. WinCC/Data Monitor – служит для отображения, анализа, оценки и представления текущих состояний процесса и исторических данных (измеренных значений, сообщений, данных пользователя) из базы данных процесса.

Необходимый для этого клиент Data Monitor может быть установлен на любом офисном ПК. Data Monitor предлагает несколько инструментальных средств для отображения и анализа.

WinCC/Downtime Monitor – может регистрировать и анализировать времена простоя в производственном оборудовании, ориентированном на использование отдельных станков или технологических линий, и вычислять относящиеся к этому оборудованию параметры, так называемые ключевые индикаторы производительности (Key Performance Indicators, KPI). Анализ неисправностей дает информацию о частоте и длительности простоев оборудования. Соответствующие управляющие элементы для представления результатов анализа (WinCC Controls) без труда могут быть встроены в мнемосхемы WinCC.

WinCC/ProcessMonitor – способствует оптимизации производства, действуя как информационная система управления (Management Information System) и инструментальное онлайн-средство анализа качества. Он рассчитывает пользовательские KPI, позволяет анализировать архивные данные и сообщения процесса и передает рассчитанные значения в WinCC. WinCC/Connectivity Pack – позволяет другим приложениям получить доступ к архивам WinCC через OPC HDA или WinCC OLE-DB, а также к текущим значениям через OPC XML и передает дальше текущие или прошлые сообщения через OPC (historical) A&E системам верхнего уровня. С помощью WinCC/ConnectivityStation любой компьютер с Windows без установленного на нем WinCC может быть сконфигурирован как станция для анализа.

WinCC/Industrial Data Bridge – поддерживает подключение внешних баз данных, офисных приложений и информационно-технологических систем через WinCC OLE-DB и OPC DA посредством параметризуемого стандартного программного обеспечения.

Увеличение коэффициента готовности

WinCC/Redundancy – увеличивает коэффициент готовности системы посредством резервирования станций WinCC или серверов, которые контролируют друг друга, гарантирует работоспособность системы и делает возможным непрерывный сбор данных.

WinCC/ProAgent– делает возможной целенаправленную и быструю диагностику процесса для машин и установок. Благодаря полной интеграции в диагностику процесса SIMATIC ProAgent предлагает комплексное решение на основе STEP 7, инжиниринговых средств разработки и контроллеров SIMATIC S7.

SIMATIC Maintenance Station– визуализирует данные, имеющие отношение к обслуживанию системы автоматизации. При этом нужная информация получается из проекта STEP 7. Необходимость в дополнительном проектировании отсутствует.

Контроль достоверности и изменений

С помощью опций WinCC/Audit(протоколирование действий оператора, контроль изменений в проекте и отслеживание производственного процесса посредством защищенных журналов Audit Trails), WinCC/ChangeControl (контроль версий проекта и отслеживание изменений в проекте) и SIMATIC Logon, встроенной в систему управления пользователями WinCC (централизованное в рамках всего предприятия управление пользователями) и соответствующих мероприятий при проектировании SIMATIC WinCC облегчает выполнение требований свода федеральных постановлений США 21 CFR, часть 11, в фармацевтической промышленности, в производстве биологически активных добавок и лекарств, а также требований директивы Европейского союза EU 178/2002 в пищевой промышленности, производстве напитков и табака.

2. РАСШИРЕНИЯ SCADA

WinCC/User Archives– предназначен для создания и использования архивов пользователя, в которых возможно хранение данных в виде наборов данных и обмен ими между WinCC и контроллером в виде рецептов или данных пакета.

Процессы рецептурного управления

SIMATIC Batch (для WinCC) – делает возможным рецептурное управление для технологических процессов. Расширения системы WinCC/Industrial X– делает возможным проектирование пользовательских объектов с помощью технологии ActiveX. Эти объекты могут быть стандартизованы, многократно использованы и централизованно изменены. WinCC/ODK– описывает открытые интерфейсы программирования (C-API), с помощью которых Вы получаете доступ к данным и функциям системы проектирования и исполнения WinCC и даже можете создавать собственные приложения.

Дополнения WinCC (WinCC Add-ons)

WinCC Add-ons разрабатываются и продаются другими подразделениями фирмы Siemens, а также внешними поставщиками. Поддержка WinCC Addons осуществляется поставщиком соответствующего продукта, который является также контактным лицом при интегрировании продукта в решение задачи автоматизации.

Premium Add-ons

WinCC Premium Add-ons – это качественные продукты, которые проверяются в тестовом центре для продуктов SIMATIC на их совместимость с базовой системой WinCC и поддерживаются на начальном этапе горячей линией SIMATIC. WinCC Premium Addons в настоящее время имеются в следующих категориях:

- Возможности связи и подключений
- Управление процессами
- Диагностика и обслуживание
- Отраслевые и технологические решения
- Инструментальные средства для проектирования

Основные особенности:

Универсальность

Рассчитанный с самого начала для международного использования, интерфейс разработки WinCC может переключаться между различными языками, среди которых также четыре азиатских языка проектирования. Свой проект тоже можно разрабатывать одновременно на нескольких целевых языках и переключаться между этими языками во время работы. Концепция базовой системы нейтральна относительно технологии и отрасли. Это доказывают применения во всех секторах и отраслях в машиностроении и производстве промышленного оборудования – вплоть до фармацевтической промышленности – где WinCC с соответствующими опциями удовлетворяет требованиям 21 CFR часть 11 и EU 178/2002 (пищевая и фармацевтическая промышленность).

В объеме поставки WinCC содержатся все основные коммуникационные каналы для подключения к контроллерам SIMATIC, AllenBradley Ethernet IP и Modbus TCP/IP, а также не зависящие от производителей каналы, такие как PROFIBUS/PROFINET и OPC.

Открытые интерфейсы, многочисленные опции и уже включенный в базовую систему MS SQL-сервер для архивирования данных поддерживают интеграцию информационных технологий и бизнеса на предприятии.

3. ЧЕЛОВЕКО-МАШИННЫЙ ИНТЕРФЕЙС

В базовый пакет поставки входят все стандартные для промышленности функции человеко-машинного интерфейса:

- Графическая визуализация состояний процесса
- Управление станком или установкой через индивидуально разрабатываемый интерфейс оператора с собственными меню и панелями инструментов
- Генерация и квитирование событий
- Архивирование измеренных значений и сообщений в базе данных процесса

- Протоколирование текущих данных процесса и исторических данных
- Управление пользователями и их правами доступа

Процессы и события, влияющие на качество продукции, непрерывно записываются и, таким образом, могут быть полностью, без пропусков, проанализированы.

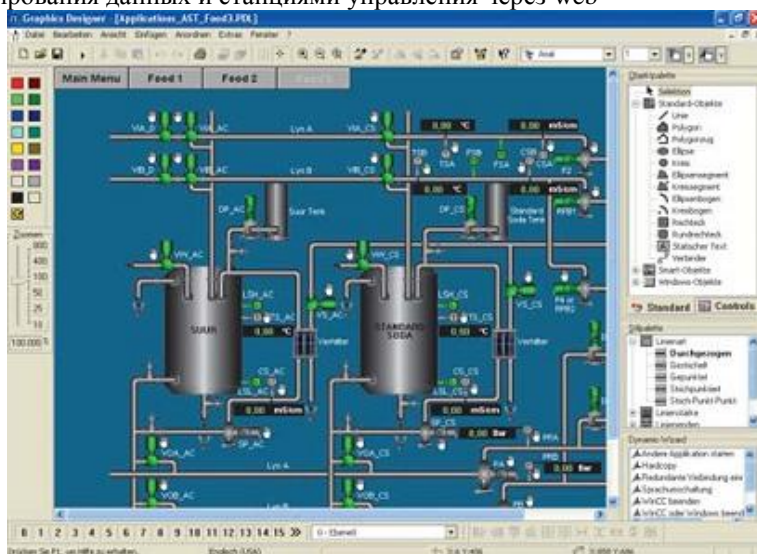
Простое и эффективное проектирование

Совершенные функции проектирования радикально сокращают затраты на разработку и обучение:

- Удобный, объектно-ориентированный графический редактор
- Обширные библиотеки
- Эффективная модульная технология
- Быстрота изменений благодаря онлайн-овому проектированию
- Инструментальное средство проектирования для обработки больших массивов данных
- Прозрачность благодаря списку перекрестных ссылок

Универсальная масштабируемость

Чтобы удовлетворять все возрастающим требованиям, визуализация должна иметь возможность расширения, не вызывая при этом нарушений в технологии и не требуя проектирования заново. Наивысшим приоритетом является защита инвестиций. WinCC предоставляет требуемую масштабируемость, от небольших однопользовательских систем до резервируемых систем клиент-сервер с центральным MS SQL-сервером для архивирования данных и станциями управления через web



Основные особенности человеко-машинного интерфейса

Универсальность

- Решения для всех отраслей промышленности - Удовлетворение стандарту 21 CFR
- Поддержка множества языков для использования по всему миру
- Возможность встраивания во все решения в области автоматизации и информационных технологий

Открытые стандарты для облегчения интеграции Встроенный MS SQL-сервер для архивирования данных может являться информационным центром в компании

Повышение прозрачности производства благодаря использованию Plant Intelligence

Расширяемость через опции и дополнения

Часть концепции Totally Integrated Automation

Открытые стандарты для облегчения интеграции

WinCC последовательно делает ставку на максимальную открытость и способность к интеграции: управляющие элементы ActiveX для технологических и отраслевых расширений, независимый от производителей обмен данными через OPC, стандартные интерфейсы для внешнего доступа к базе данных (WinCC OLE-DB и OPC HDA), встроенные стандартные языки сценариев (VBScript и ANSI-C), доступ к данным и системным функциям через прикладной программный интерфейс (API) с помощью открытого комплекта для проектирования (Open Development Kit, WinCC/ODK), пользовательские расширения графического редактора WinCC с помощью Visual Basic for Applications (VBA).

Встроенный MS SQL-сервер для архивирования данных в качестве информационного центра SIMATIC WinCC обладает мощными масштабируемыми функциональными возможностями для архивирования данных на основе Microsoft SQL-сервера, уже встроенного в базовую систему. Тем самым пользователю предоставляются неограниченные возможности: от высокопроизводительного архивирования текущих данных процесса и событий до долговременного архивирования с высокой степенью сжатия данных и функцией резервного копирования и далее, вплоть до образования информационного центра в виде центрального, действующего для всего предприятия, MS SQL-сервера для архивирования данных.

Повышение прозрачности производства благодаря использованию Plant Intelligence. Plant Intelligence означает стремление к снижению затрат, предотвращению брака, лучшему использованию производственного оборудования и, тем самым, к достижению большей эффективности и экономичности на производственных предприятиях с помощью разумного использования информации на предприятии. Высокие функциональные возможности (например, статистические функции для измеренных значений и сообщений, встроенные в базовую систему), неограниченная открытость, встроенное архивирование данных и ряд опций являются гарантией прозрачности в производственном процессе и обоснованности принятия решений.

Возможность универсального использования

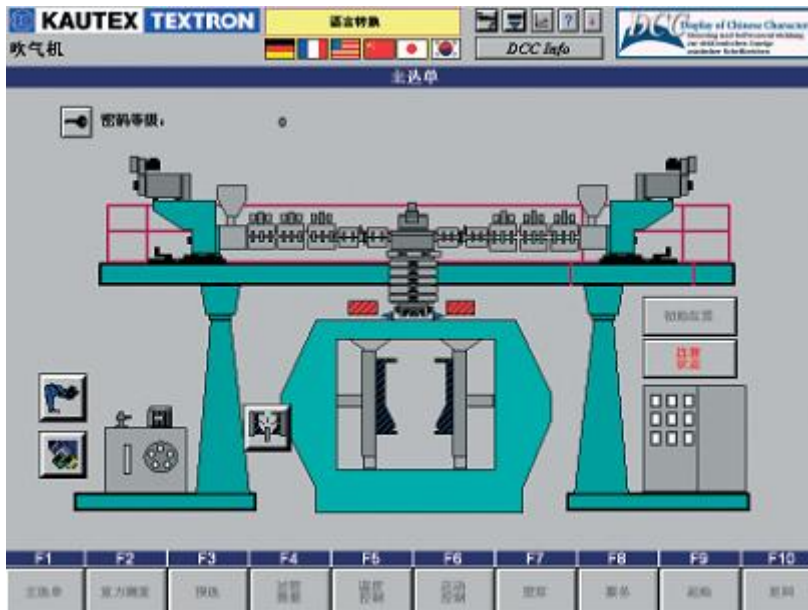
Решения для всех отраслей промышленности и технологий.

Базовая система WinCC разработана нейтральной по отношению к технологии и отрасли промышленности, она модульна и гибко расширяема. Она позволяет создавать как однопользовательские приложения в машиностроении, так и сложные многопользовательские решения или даже распределенные системы с несколькими резервируемыми серверами и клиентами.

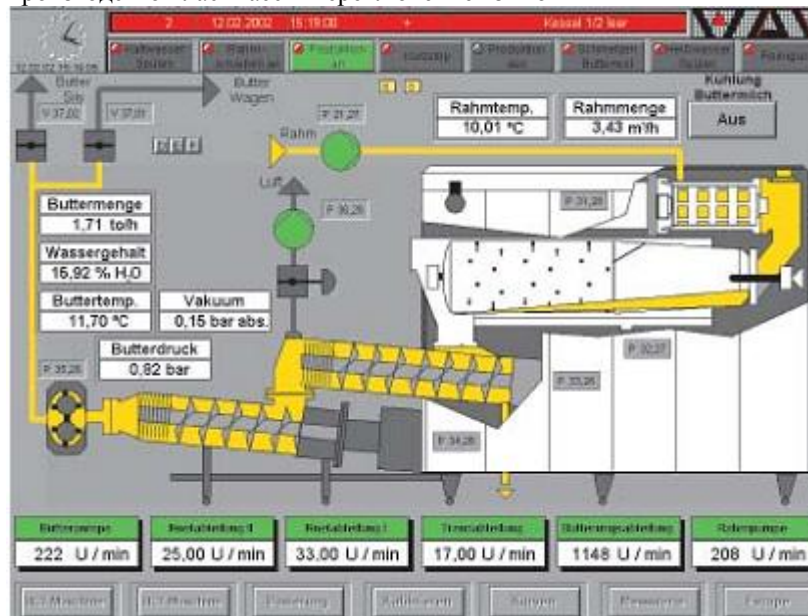
Многочисленные применения во всех приложениях и отраслях промышленности подтверждают универсальность и производительность системы:

- Производство автомобилей и комплектующих изделий для них
- Химическая и фармацевтическая промышленность
- Пищевая промышленность и производство напитков
- Машиностроение
- Производство и распределение энергии
- Торговля и сфера обслуживания
- Производство пластмасс и резины
- Металлургия и металлообработка
- Производство и переработка бумаги, полиграфическая промышленность
- Транспорт, перевозки и логистика
- Водоподготовка и водоочистка
- Управление зданиями и другой недвижимостью

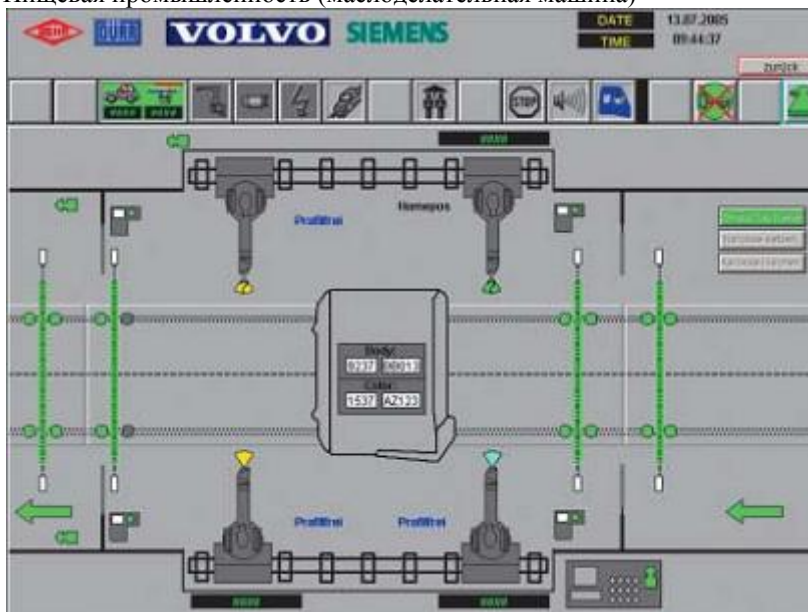




Производство пластмасс и переключение языков



Пищевая промышленность (маслоделательная машина)



Автомобилестроение (Volvo)



Упаковочная промышленность

Возможность встраивания во все решения задач автоматизации на любом предприятии

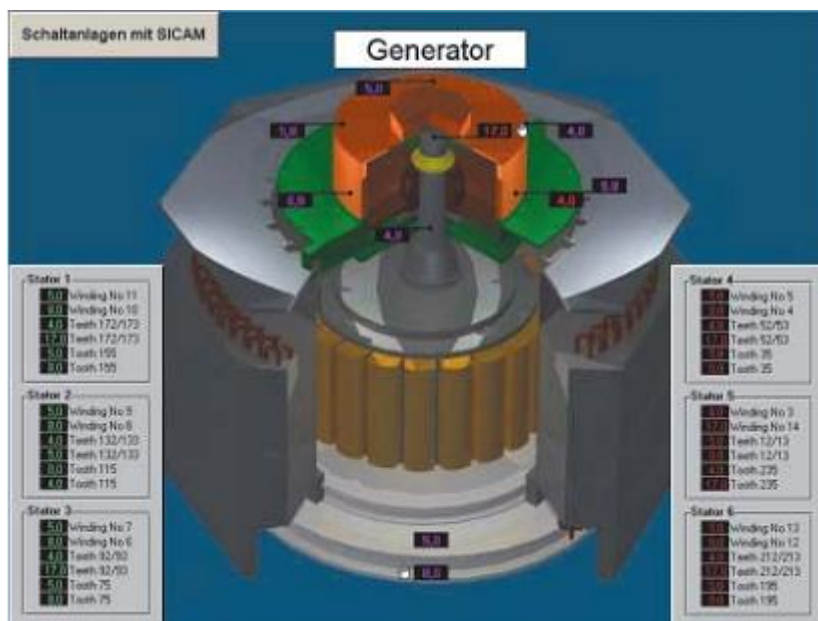
В объем поставки WinCC входят все основные коммуникационные каналы для подключения к контроллерам SIMATIC S5/ S7/ 505 (например, через S7 Protocol Suite), AllenBradley Ethernet IP и Modbus TCP/IP, а также независимые от производителей каналы, как PROFIBUS/PROFINET и OPC. Так как каждый производитель контроллеров предлагает к своей аппаратуре соответствующие OPC-серверы, то для возможностей подключения к WinCC нет почти никаких ограничений.

В базовую систему SIMATIC WinCC входит мощная масштабируемая система архивирования данных на основе Microsoft SQL-сервера, который можно использовать как единый информационный центр. Открытые интерфейсы и опции образуют основу для эффективной интеграции информационных технологий и бизнеса.

Все это обеспечивает также подключение к автоматизированной системе управления производственными процессами (MES) и к системе планирования и управления ресурсами предприятий (ERP).



Мнемосхема из области водоочистки.



Распределение энергии (распределительная система SICAM)

4. ВСТРОЕННОЕ УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ. SIMATIC Logon

С помощью администратора пользователей WinCC (WinCC User Administrator) производится назначение прав и управление правами доступа пользователей для режима проектирования и исполнения. Как администратор, Вы можете в любое время, в том числе и во время работы системы, создавать до 128 групп пользователей, каждая из которых содержит до 128 отдельных пользователей, назначая им соответствующие права доступа к функциям WinCC. В целом возможно до 999 различных авторизаций.

В управление пользователями включены все операторские станции, в том числе и клиенты Web Navigator и Data Monitor. С помощью SIMATIC Logon имеется возможность централизованного для всего предприятия управления пользователями, интегрированного в систему управления пользователями Windows.

SIMATIC Logon – централизованное для всего предприятия управление пользователями. SIMATIC Logon предоставляет многообразные механизмы безопасности как на стороне администратора, так и на стороне пользователя. Как обычно, пользователю задаются уникальный идентификатор, имя пользователя и пароль.

Такие функции, как устаревание пароля, автоматическое завершение сеанса по истечении заданного времени и блокирование после нескольких неправильных вводов

пароля, гарантируют максимальную безопасность управления. Кроме того, администратор имеет возможность в онлайн-режиме создавать новых и блокировать существующих пользователей в масштабе предприятия.

SIMATIC Logon может использоваться в самых разнообразных структурах, таких как однопользовательские станции или конфигурации клиент-сервер. В структурах с SIMATIC Logon высокий коэффициент готовности обеспечивается первичным и вторичным контроллерами доменов и локальным управлением пользователями Windows.

Графическая система

Графическая система WinCC обрабатывает во время работы все вводы и выводы на экране. Экраны, которые Вы используете для визуализации и управления своей установкой, создаются с помощью графического дизайнера WinCC (WinCC Graphics Designer).

Имеете ли Вы дело с небольшими, простыми применениями визуализации или сложными задачами управления: с помощью стандартов WinCC можно создать индивидуально спроектированные интерфейсы оператора для любого применения – для надежного управления и для оптимизации всего производства. Соответствующий для данного проекта вид может быть создан с помощью централизованных настроек дизайна и цветовой палитры, а также собственных меню и панелей инструментов. В Vista Design в качестве особых эффектов имеются тени, прозрачность, тематики (стили) и внешний вид объектов (скиннинг).

Преимущества

Централизованное для всего предприятия управление пользователями SIMATIC Logon встроено в управление пользователями Windows

Высокая степень безопасности благодаря мерам предосторожности на стороне администратора и стороне пользователя

Могут быть использованы различные процедуры регистрации: клавиатура, считыватель чип-карт

Возможно использование в различных конфигурациях (однопользовательская, системы клиент-сервер и вплоть до решений, обеспечивающих высокий коэффициент готовности)

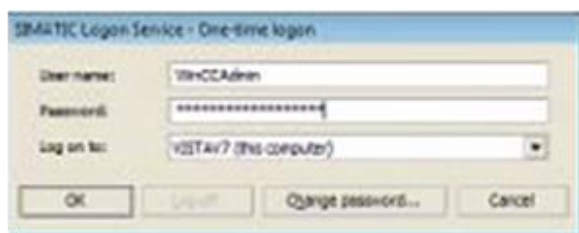
Любое управление установкой, архивами и WinCC может быть заблокировано для предотвращения несанкционированного доступа. WinCC может записывать действия с переменными – с датой, временем,

именем пользователя и сравнением между старым и новым значением. При этом hover-эффект обеспечивает дополнительное выделение управляемых объектов.

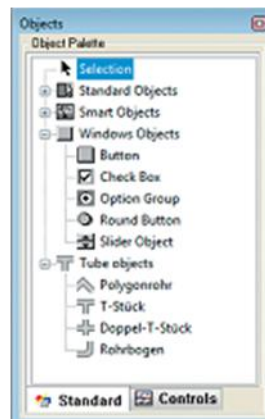
Для создания привлекательного, ориентированного на процесс интерфейса, система предлагает различные элементы, от графических объектов до собственных меню, панелей инструментов, кнопок и элементов управления и вплоть до собственных централизованно изменяемых лицевых панелей.

Проектировщик динамически управляет внешним видом частей графических изображений и может для этого обращаться к значениям переменных или принимать их непосредственно из программ. WinCC поддерживает экраны размером до 10 000 x 10 000 пикселей, причем разрешение в любое время может быть адаптировано до необходимого разрешения экрана. Это обеспечивает защиту инвестиций при последующих расширениях установки.

Панорамирование, изменение масштаба изображения и уменьшение детализации при уменьшении масштаба, т.е. представление содержимого экрана в зависимости от коэффициента увеличения изображения, придают завершенность графическим функциям системы визуализации и дают пользователю совершенно новые ощущения при управлении и контроле



Регистрация пользователя



**Палитры объектов графического
ливайнена WinCC**

Минимизация простоев – с помощью алармов и сообщений. SIMATIC WinCC регистрирует сигналы процесса и локальные события, сохраняет их в архивах и по мере необходимости предоставляет их в распоряжение в отфильтрованном или отсортированном виде. Сообщения могут возникать по изменению отдельных битов переменной процесса (макс. 32), как результат отправки кадра сообщений, инициированной непосредственно системой автоматизации, как результат нарушения граничных значений аналоговыми сигналами или при фиксации действий оператора (->сообщение об управлении). Можно спроектировать любое сообщение так, что оператор должен будет его квитировать.

Свободно определяемая структура сообщения. Так как структура сообщения свободно определяема, то она может быть составлена с учетом специальных требований Вашей установки. Деление структуры на различные текстовые блоки, общим числом до 10 (идентификатор установки, место неисправности, текст и т.д.), обеспечивает большую наглядность информации и позволяет инициировать целенаправленный анализ посредством функций фильтрации и сортировки.

Подразделение сообщений на классы (до 16 классов) позволяет сконфигурировать как сообщения о неисправностях и состояниях, так и отдельную обработку алармов, предупреждений, неисправностей и ошибок для различных частей установки. Для класса сообщений (например, алармов) может быть задано до 16 приоритетов.

Удобное отображение сообщений. Сообщения отображаются на экране с помощью свободно настраиваемого элемента управления WinCC Alarm Control. Здесь можно точно приспособить отображение информации в сообщении в соответствии с потребностями операторов. Созданные настройки могут быть сохранены в глобальных или пользовательских шаблонах.

На основе содержимого отдельных блоков сообщений может быть выполнена фильтрация, отбор и сортировка, например, в хронологическом порядке, в соответствии с приоритетами или местом неисправности. Затем это содержимое может быть экспортировано непосредственно в CSV-файл или распечатано как отчет. Кроме того, в панель инструментов можно добавить свой элемент со свободно определяемой функцией, что обеспечивает наивысшую степень гибкости. Так, например, можно встроить собственные, относящиеся к данному проекту функции.

Основные особенности

Система сообщений на основе MS SQL –сервера с:

– 10 текстовыми блоками

– 10 блоками переменных процесса и множеством системных блоков (например, дата и время)

Свободно выбираемые и произвольно комбинируемые критерии отбора

Различные пользовательские фильтры могут сохраняться в виде матрицы фильтров

Может быть сгенерирован список наиболее часто приходящих сообщений

Одно двойное слово может запускать до 32 сообщений

256 пороговых значений для аналоговых величин

Для сохранения наглядности при большом количестве поступающих сообщений оператор может подавлять менее важные сообщения при их отображении на экране с помощью функции Alarm Hiding [скрытие сообщений].

При этом эти сообщения продолжают архивироваться в фоновом режиме.

Архивирование и протоколирование сообщений

Для архивирования сообщений используется Microsoft SQL-сервер. Он гарантирует непрерывную, без пропусков, регистрацию всех событий. Сообщения архивируются при возникновении связанных с ними событий, например, при появлении сообщения или изменении его состояния.

В протоколе последовательности сообщений события (выборочно) могут документироваться в хронологическом порядке. При этом все изменения состояния (прибытие, уход, квитирование) произошедших сообщений выводятся на принтер. В протоколе архива сообщений могут генерироваться определенные подписки заархивированных сообщений.



WinCC Alarm Control для отображения сообщений

Встроены все функции человеко-машинного интерфейса

Статистический анализ

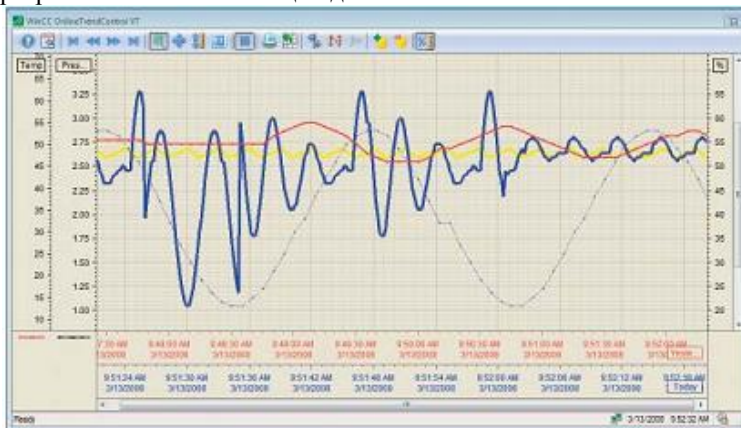
Большое количество встроенных статистических функций позволяет проанализировать состояния процесса. Список статистики сообщений показывает, как долго определенные сообщения были активными в среднем и суммарно (длительность сообщений) и, аналогично, среднее и общее время квитирования. Само собой разумеется, что при этом возможна фильтрация по соответствующим событиям, месту сообщений и интервалов времени. Благодаря этому быстро выясняется, где находятся критические точки и узкие места в производстве. Чтобы отсортировать сообщения для анализа, можно просто выбрать заголовки столбцов и указать желаемый критерий сортировки (например, «по убыванию частоты»).

Система архивирования

Исторические значения и последовательности значений сохраняются в архивах значений процесса. Наряду со значениями процесса WinCC архивирует также сообщения и данные пользователей. Архивирование производится в архивах базы данных MS SQL-сервера с высокой производительностью: до 10 000 измеренных значений и 100 сообщений в секунду в качестве постоянной нагрузки (при пиковой нагрузке даже 15 000 сообщений в течение 10 секунд) не является проблемой для центрального сервера архивов. Потребность в памяти благодаря мощным функциям сжатия без потерь очень невелика. Значения процесса могут сжиматься непрерывно (например, вычисление средних значений), а также по событию или управлению от процесса (например, при нарушении граничных значений), и архивироваться через функцию обобщения.

Открытость и возможность интеграции

Отчеты WinCC могут также содержать данные из базы данных и внешние данные в формате CSV в виде таблицы или кривой. Для отображения данных из других приложений в виде таблицы или графика можно также разработать свой поставщик данных.



Online Trend Control и клавиша для экспорта данных



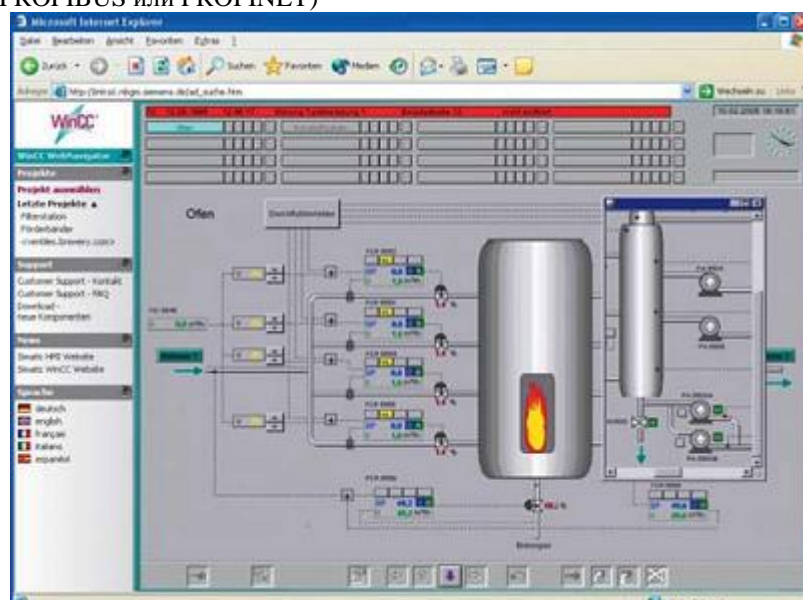
Статистические функции для архивов значений процесса (Trend Ruler Control)

Функции управления процессом.

Basic Process Control дополняет SIMATIC WinCC важным набором инструментов. Этот пакет предоставляет дополнительные объекты и инструментальные средства проектирования для упрощения реализации типовых требований систем контроля и управления процессами.

Благодаря этому при реализации систем управления процессами может быть использован целый ряд функций управления:

- Эффективное разделение экрана на обзорную область, рабочую область и область клавиатуры
- Прокрутка в иерархии изображений
- Сохранение и вызов пользовательских композиций экранов
- Выбор мнемосхем и точек измерения по имени
- Создание трендов в режиме онлайн
- Специальный элемент «групповой дисплей» для отображения соподчиненных неисправностей и непосредственного перехода к соответствующей мнемосхеме
 - Мониторинг работоспособности связей процесса с помощью экрана конфигурации установки и автоматической генерации сообщений системы управления
 - Активизация внешних устройств (например, сирены)
 - Синхронизация времени (установка часов ПК через DCF77 или GPS; дальнейшая передача времени через PROFIBUS или PROFINET)



Мнемосхема системы управления процессом (Basic Process Control) с отображением через WinCC/WebNavigator

5. ПРОСТОЕ И ЭФФЕКТИВНОЕ ПРОЕКТИРОВАНИЕ. WinCC Graphics Designer

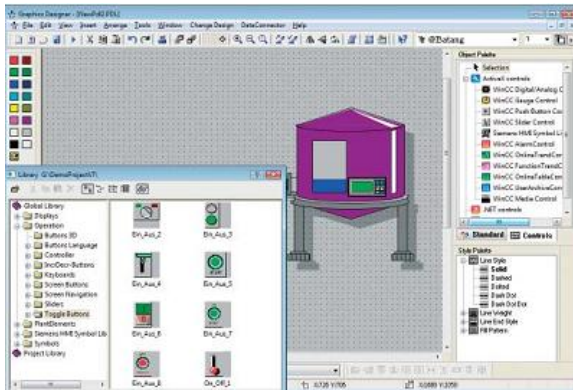
Простое использование ноу-хау из области ПК для промышленного процесса – эта идея определила развитие. Появилась объектно-ориентированная многоязычная среда разработки с настраиваемым интерфейсом для проектирования, контекстными подсказками, обширной онлайн-помощью и примерами проектирования.

Уточненность в деталях

Связь объекта с внутренней переменной или переменной процесса чрезвычайно проста и понятна: как только соответствующий объект помещается в изображение, появляются наглядные конфигурационные диалоги. Кроме того, WinCC Graphics Designer предоставляет возможность динамизации всех свойств объекта – для достижения максимальной гибкости возможна также обработка через сценарий.

WinCC Graphics Designer поддерживает проектирование на 32 уровнях. Для сложных мнемосхем, включающих множество накладывающихся друг на друга объектов, для большей наглядности можно выводить на экран отдельные уровни. Удобна также возможность изменять свойства отдельных объектов, входящих в состав группы, без разбора группы.

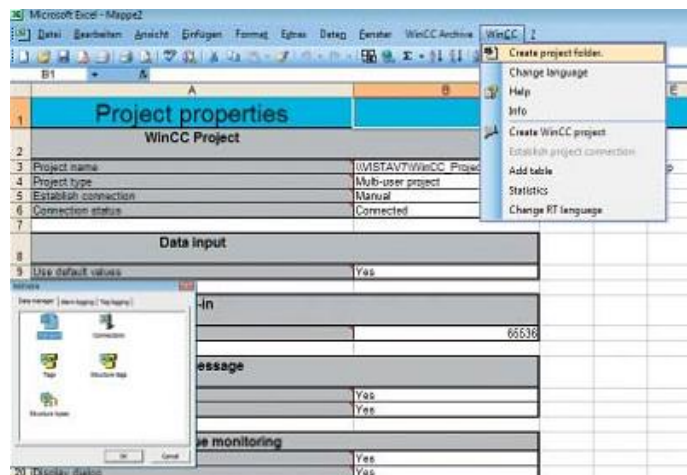
Объекты, многократно появляющиеся в изображениях, как правило, просто копируются. При копировании объектов их связи с переменными также копируются 1:1. Для облегчения привязки, т.е. подключения к другим переменным, WinCC предоставляет в распоряжение диалоговое окно привязки, в котором перечислены все переменные, связанные с выбранными объектами. Здесь же они могут быть централизованно «переподключены»



Глобальная библиотека объектов SIMATIC WinCC



Проектирование мнемосхем с помощью WinCC Graphics Designer



Инструментальное средство проектирования для работы с большими проектами

Контрольные вопросы

1. Для чего служит WinCC?
2. Опишите возможности WinCC Graphics Designer.
3. Функции SIMATIC Logon.
4. Возможности человеко-машинного интерфейса.

ЛАБОРАТОРНЫЕ РАБОТЫ

Лабораторная работа № 1.

Создание таблицы истинности и логической схемы на основе логических элементов

Цель работы: Получить навыки создания таблицы истинности и логической схемы на основе логических элементов.

Теоретическая часть

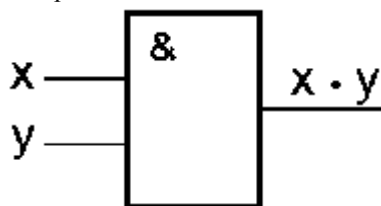
Современный этап промышленного развития характеризуется тем, что разработчики систем автоматики и вычислительной техники стремятся использовать функциональные модули, выполняющие определённые схемные задачи: логические преобразования, хранение информации и т.д. Конкретный вид электрической схемы, использованной для реализации заданной логической функции, как правило, не имеет существенного значения. Техническое устройство, реализующее логическую функцию, может рассматриваться просто как логический элемент, внутренняя структура которого не конкретизируется.

На рис. 1 представлена таблица истинности элемента "И" с двумя входами. Хорошо видно, что логическая единица появляется на выходе элемента только при наличии единицы на первом входе и на втором. В трёх остальных случаях на выходе будут нули.

Таблица истинности		
X1	X2	Y1
0	0	0
0	1	0
1	0	0
1	1	1

Рис. 1. Таблица истинности элемента "И" с двумя входами

На принципиальных схемах логический элемент "И" обозначают так

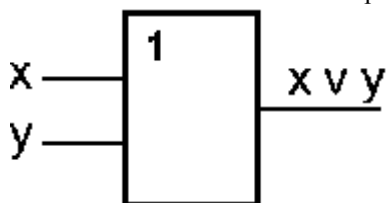


Элемент "ИЛИ" с двумя входами работает несколько по-другому. Достаточно логической единицы на первом входе или на втором как на выходе будет логическая единица. Две единицы так же дадут единицу на выходе (рис. 2).

Таблица истинности		
X1	X2	Y1
0	0	0
0	1	1
1	0	1
1	1	1

Рис. 2. Таблица истинности элемента "ИЛИ" с двумя входами

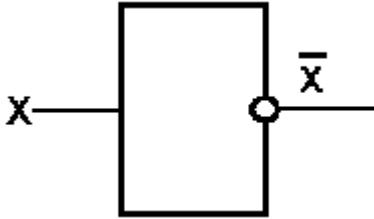
На схемах элемент "ИЛИ" изображают так



Элемент, выполняющий функцию инверсии «НЕ» имеет один вход и один выход. Он меняет уровень сигнала на противоположный. Низкий потенциал на входе даёт высокий потенциал на выходе и наоборот (рис. 3).

Таблица истинности	
X1	Y1
0	1
1	0

Рис. 3. Таблица истинности элемента "НЕ" с двумя входами
Вот таким образом его показывают на схемах.



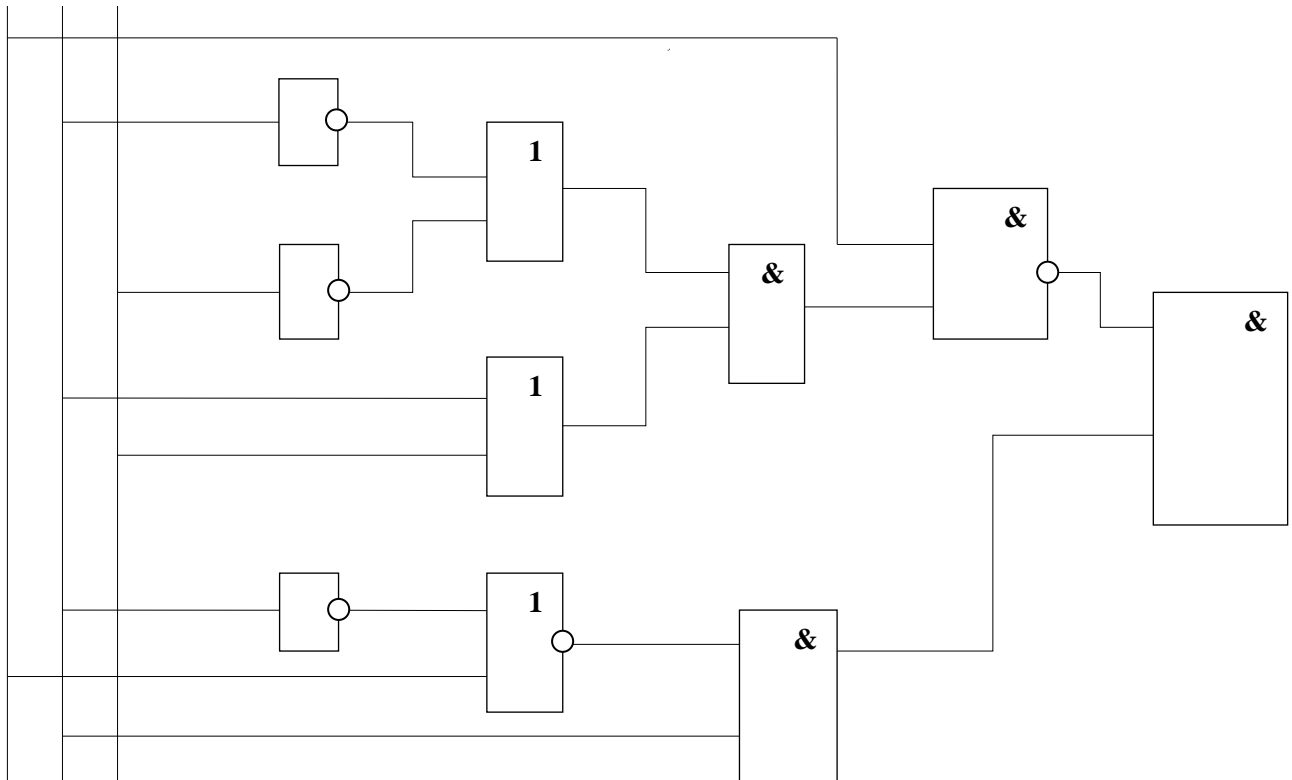
Порядок выполнения работы

Заданная формула $x \wedge (z \leftrightarrow \bar{y} \vee z) \rightarrow y \rightarrow x \wedge y$ с помощью равноправных формул, изученных на лекции, преобразовать в упрощенный вид. Далее для получившейся формулы составить таблицу истинности и схему.

$$x \wedge (z \leftrightarrow \bar{y} \vee z) \rightarrow y \rightarrow x \wedge y = x \wedge (z \rightarrow \bar{y}) \wedge (\bar{y} \rightarrow z) \rightarrow y \rightarrow x \wedge y = x \wedge (\bar{z} \vee \bar{y}) \wedge (y \vee z) \vee \bar{y} \vee x \wedge y$$

x	y	z	F
0	0	0	0
0	0	1	0
0	1	0	1
1	0	0	0
0	1	1	1
1	1	1	1

X Y Z



Задания:

1- Задание. Составьте таблицы истинности для следующих формул:

№	Логическая формула
1.	$z \wedge (x \wedge y \vee x) \rightarrow (z \leftrightarrow x) \vee y$
2.	$x \wedge z \leftrightarrow y \vee (z \rightarrow y \rightarrow x)$
3.	$y \wedge (z \wedge y) \vee x \leftrightarrow x \wedge z$
4.	$y \wedge x \leftrightarrow y \vee x \rightarrow z \wedge x$
5.	$z \wedge y \rightarrow x \leftrightarrow y \vee (x \rightarrow z \wedge x)$
6.	$z \leftrightarrow x \wedge (z \leftrightarrow y \vee z) \rightarrow y \vee x$
7.	$x \rightarrow (z \wedge z \wedge y) \vee x \leftrightarrow y \wedge x$
8.	$y \vee z \wedge (x \leftrightarrow y \vee x) \rightarrow z \wedge x$
9.	$z \vee x \wedge z \leftrightarrow y \vee (z \rightarrow y \rightarrow x)$

2-Задание. Составьте таблицы истинности и логические схемы для следующих формул:

№	Логическая формула
1.	$z \wedge (x \wedge y \vee x) \wedge (z \vee x) \vee y$
2.	$x \wedge z \vee y \vee (z \vee y \wedge x)$
3.	$y \wedge (z \wedge y) \vee x \vee x \wedge z$
4.	$y \wedge x \wedge z \vee x \vee z \wedge x$
5.	$z \wedge y \wedge x \vee y \vee (x \vee z \wedge x)$
6.	$z \vee x \wedge (z \wedge y \vee z) \wedge y \vee x$
7.	$x \wedge (z \wedge z \wedge y) \vee x \vee y \wedge x$
8.	$y \vee z \wedge (x \wedge y \vee x) \vee z \wedge x$
9.	$z \vee x \wedge z \vee y \vee (z \vee y \wedge x)$

Вопросы для самоконтроля.

1. Перечислите логические элементы?
2. Принцип работы логических элементов?
3. Сфера применения логических элементов?

Лабораторная работа №2.

Работа с массивами

Цель работы: научиться строить таблицы значений функции, строить графики функции одной переменной, а также научиться перемножать вектора (скалярное произведение, векторное произведение, внешнее произведение).

Порядок работы

1. Выведите таблицу значений функции

$$y(x) = \frac{\cos^2 x}{1 + \sin x} - e^{-x} \cdot \log x$$

в точках 0.1, 0.4, 0.6, 0.7, 1.1, 1.5, 2.3.

Вначале необходимо записать данные значения в вектор-строку x , а затем вычислить значения функции $y(x)$ от каждой этой точки, поэтому необходимо в записи самой функции использовать поэлементные операции:

```
>> x = [0.1 0.4 0.6 0.7 1.1 1.5 2.3];
>> y = cos(x).^2/(1+sin(x)) - exp(-x).*log(x)
y =
    2.3728    0.9035    0.5696    0.4664    0.2576    0.1988
    0.2058
```

2. Заполните вектор-столбец элементами, начинающимися с нуля и до единицы с шагом 0.3 (используя операцию транспонирование) и вектор-строку, начинающуюся с единицы до двух с шагом -0.2 :

```
>> x = [0:0.3:1]';
x =
    0
    0.3000
    0.6000
    0.9000
>> y = [2: -0.2:1]
y =
    2.0000    1.8000    1.6000    1.4000    1.2000    1.0000
```

3. Построить график функции $y(x) = e^x \cos 5x$ на отрезке $[0, 5]$.

Вывод отображения функции в виде графика состоит из следующих этапов:

1. Задание вектора значений аргумента x .
2. Вычисление вектора y значений функции $y(x)$.
3. Вызов команды `plot` для построения графика.

Команды для задания вектора x и вычисления функции лучше завершать точкой с запятой для подавления вывода в командное окно их значений (после команды `plot` точку с запятой ставить не обязательно, т.к. она ничего не выводит в командное окно). Не забудьте использовать поэлементное умножение `.*`:

```
>> x = [0:0.05:5];
>> y = exp(x).*cos(5*x);
>> plot(x, y)
```

$$g(x) = \cos \frac{1.5}{x^2};$$

```
>> x = [-1:0.005: -0.3];
>> f = cos(x.^ -2);
>> g = cos(1.5*x.^ -2));
>> plot(x, f, x, g)
```

$$f(x) = \cos \frac{1}{x^2},$$

4. Постройте на отрезке $[-1, -0.3]$ графики функций

$$a = \begin{bmatrix} 1.5 \\ -4.8 \\ 0.9 \end{bmatrix}, \quad b = \begin{bmatrix} 6.3 \\ 9.7 \\ -1.4 \end{bmatrix};$$

а) Найдите скалярное произведение векторов:

б) найдите длину (модуль) вектора a ; в) найдите векторное произведение векторов a и b .

Для вычисления скалярного произведения необходимо просуммировать компоненты вектора, полученного в результате поэлементного умножения a на b , т.е. надо использовать функцию `sum` и поэлементное умножение:

```

a) >> a = [1.5; -4.8; 0.9];   б) >> d = sqrt(sum(a .* a))   в) >> c = cross(a, b)
>> b = [6.3; 9.7; -1.4];     d =                               c =
>> s = sum(a.*b)             5.1088                               -2.0100
s =                               7.7700
-38.3700                          44.7900

```

$$a = \begin{bmatrix} 2.5 \\ 0 \\ 0 \end{bmatrix}, b = \begin{bmatrix} 0.3 \\ 3.2 \\ 0 \end{bmatrix}, c = \begin{bmatrix} -0.3 \\ -2.1 \\ 4.2 \end{bmatrix};$$

6. Найдите объем параллелепипеда, если

```

>> a = [2.5; 0; 0];
>> b = [0.3; 3.2; 0];
>> c = [-0.3; -2.1; 4.2];
>> v = abs(sum(a.*cross(b, c)))
v =
33.6000

```

Задания

Вариант 1

1. Выведите таблицу значений функции

$$y(x) = \frac{2 \cdot \cos^2 x}{1 + \tan x} - e^{2x} \cdot \log x$$

в точках 0.1, 0.3, 0.4, 0.6, 1.3, 1.4, 2.1.

2. Заполните вектор-столбец элементами, начинающимися с нуля и до единицы с шагом 0.1 (используя операцию транспонирование) и векторстроку, начинающуюся с единицы до двух с шагом -0.2.

3. Постройте график функции $y(x) = 2e^{-x} \sin 3x$ на отрезке $[0, 4]$.

$$f(x) = \sin \frac{1}{3 \cdot x^2}, \quad g(x) = 2 \cdot \cos \frac{1.8}{x^2}.$$

4. Постройте на отрезке $[-1, -0.5]$ графики функций

$$a = \begin{bmatrix} 2.1 \\ -3.7 \\ 0.6 \end{bmatrix}, \quad b = \begin{bmatrix} 4.1 \\ 7.5 \\ -0.9 \end{bmatrix};$$

5. а) Найдите скалярное произведение векторов:

б) найдите длину (модуль) вектора a ; в) найдите векторное произведение векторов a и b .

$$a = \begin{bmatrix} 1.5 \\ 0 \\ 0 \end{bmatrix}, b = \begin{bmatrix} 0.5 \\ 6.1 \\ 0 \end{bmatrix}, c = \begin{bmatrix} -0.7 \\ -3.6 \\ 8.5 \end{bmatrix}.$$

6. Найдите объем параллелепипеда, если

Вариант 2

$$y(x) = \frac{5 \cdot \cos^2 x}{3 + \sin 2x} + 4 \cdot e^{-x} \cdot \tan x$$

1. Выведите таблицу значений функции

в точках 0.1, 0.2, 0.5, 0.8, 1.1, 1.3, 2.2.

2. Заполните вектор-столбец элементами, начинающимися с нуля и до единицы с шагом 0.2 (используя операцию транспонирование) и векторстроку, начинающуюся с единицы до двух с шагом -0.2.

3. Постройте график функции $y(x) = e^{-x} \cdot (1 + \cos 7x)$ на отрезке $[0, 7]$.

$$f(x) = \cos \frac{1.4}{x^2}, \quad g(x) = \sin \frac{1}{2 \cdot x^2}.$$

4. Постройте на отрезке $[-1, -0.6]$ графики функций

$$a = \begin{bmatrix} 0.5 \\ -1.6 \\ 0.4 \end{bmatrix}, \quad b = \begin{bmatrix} 7.3 \\ 4.7 \\ -5.4 \end{bmatrix};$$

5. а) Найдите скалярное произведение векторов:

б) найдите длину (модуль) вектора a ; в) найдите векторное произведение векторов a и b .

$$a = \begin{bmatrix} 7.2 \\ 0 \\ 0 \end{bmatrix}, b = \begin{bmatrix} 0.4 \\ 8.6 \\ 0 \end{bmatrix}, c = \begin{bmatrix} -1.8 \\ -4.8 \\ 5.4 \end{bmatrix}.$$

6. Найдите объем параллелепипеда, если

**Лабораторная работа 3-4.
Двухмерные массивы и матрицы**

Цель работы: научиться вводить матрицы; производить математические операции с ними, поэлементные операции; вычислять функции от элементов матриц, читать и записывать с использованием текстового файла простейшее визуализации матричных данных.

Порядок работы

$$C = \begin{pmatrix} 4 & 5 & -2 \\ -6 & -3 & 7 \end{pmatrix}, A = \begin{pmatrix} 4 & -7 & 8 \\ 2 & 6 & 0 \end{pmatrix}; B = \begin{pmatrix} 3 & 5 & -2 \\ 7 & 2 & -8 \\ -4 & 6 & 0 \end{pmatrix};$$

1. а) Найдите сумму и разность матриц
б) умножьте матрицы С и В в) полученную матрицу умножьте на 5:

```
>> C = [4 5 -2; -6 -3 7];
```

```
>> A = [4 -7 8; 2 6 0];
```

```
>> S = C + A
```

```
S =
```

```
8 -2 6
```

```
-4 3 7
```

```
>> R = C - A
```

```
R =
```

```
0 12 -10
```

```
-8 -9 7
```

```
>> B = [3 5 -2; 7 2 -8; -4 6 0];
```

```
>> P = C*B
```

```
P =
```

```
55 18 -48
```

```
-67 6 36
```

```
>> T = P*5
```

```
T =
```

```
275 90 -240
```

```
-335 30 180
```

2. Найдите значение выражения $(A - C)B^2(A + C)^T$:

```
>> (A-C)*B^2*(A+C)'
```

```
ans =
```

```
-4328 1820
```

```
4166 -5505
```

$$\begin{bmatrix} 3 & 2 & -4 \end{bmatrix} \cdot \begin{pmatrix} 6 & 3 & 0 \\ -8 & 5 & -1 \\ 7 & 2 & -4 \end{pmatrix} \cdot \begin{bmatrix} -6 \\ 5 \\ 1 \end{bmatrix} :$$

3. Вычислите выражение

```
>> a = [3 2 -4];
```

```
>> B = [6 3 0; -8 5 -1; 7 2 -4];
```

```
>> c = [-6; 5; 1];
```

```
>> a*B*c
```

```
ans =
```

```
225
```

4. Решите систему из 3 уравнений
$$\begin{cases} 1.6x_1 + 6.3x_2 - 0.7x_3 = 3.4; \\ 0.8x_1 + 6.8x_2 + 9.1x_3 = 7.5; \\ -0.5x_1 + 5.5x_2 + 8.1x_3 = 5.1; \end{cases}$$

```
>> A = [1.6 6.3 -0.7; 0.8 6.8 9.1; -0.5 5.5 8.1];
```

```
>> b = [3.4; 7.5; 5.1];
```

```
>> x = A\b
```

```
x =
```

```
1.1635
```

```
0.2995
```

```
0.4981
```

5. Решите систему линейных уравнений № 4, матрица и вектор правой части которой хранятся в текстовых файлах matr.txt, rside.txt, и запишите результат в файл sol.txt.

Задания

Вариант 1

$$C = \begin{pmatrix} 1 & 3 & -5 \\ -2 & -6 & 9 \end{pmatrix}, A = \begin{pmatrix} 9 & -5 & 4 \\ 1 & 7 & 0 \end{pmatrix}; B = \begin{pmatrix} 6 & 3 & -7 \\ 5 & 4 & -9 \\ -6 & 2 & 1 \end{pmatrix} \quad |$$

1. а) Найдите сумму и разность матриц
- б) умножьте матрицы С и В
- в) полученную матрицу умножьте на 4.

2. Найдите значение выражения $(A + C)B^3(A + C)^T$.

$$[6 \ 1 \ -5] \cdot \begin{pmatrix} 5 & 3 & 1 \\ -7 & 4 & -1 \\ 8 & 6 & -9 \end{pmatrix} \cdot \begin{bmatrix} -2 \\ 9 \\ 1 \end{bmatrix}.$$

3. Вычислите выражение

$$\begin{cases} 0.6x_1 + 8.8x_2 - 4.2x_3 = 9.3; \\ 6.1x_1 + 5.9x_2 + 1.1x_3 = 5.5; \\ -2.5x_1 + 4.5x_2 + 0.1x_3 = 0.1. \end{cases}$$

4. Решите систему из 3 уравнений

5. Решите систему линейных уравнений № 4, матрица и вектор правой части которой хранятся в текстовых файлах matr.txt, rside.txt, и запишите результат в файл sol.txt.

Вариант 2

$$C = \begin{pmatrix} 7 & 1 & -3 \\ -4 & -2 & 6 \end{pmatrix}, A = \begin{pmatrix} 3 & -5 & 9 \\ 4 & 7 & 0 \end{pmatrix}; B = \begin{pmatrix} 6 & 2 & -3 \\ 4 & 5 & -7 \\ -9 & 1 & 0 \end{pmatrix} \quad б)$$

1. а) Найдите сумму и разность матриц
- б) умножьте матрицы С и В
- в) полученную матрицу умножьте на 6.

2. Найдите значение выражения $(A + C)B^2(A - C)^T$.

$$[4 \ 8 \ -1] \cdot \begin{pmatrix} 3 & 7 & 0 \\ -6 & 4 & -2 \\ 5 & 1 & -8 \end{pmatrix} \cdot \begin{bmatrix} -9 \\ 4 \\ 1 \end{bmatrix}.$$

3. Вычислите выражение

$$\begin{cases} 1.3x_1 + 0.6x_2 - 2.5x_3 = 8.1; \\ 5.4x_1 + 2.7x_2 + 0.1x_3 = 7.3; \\ -0.8x_1 + 9.2x_2 + 4.9x_3 = 0.6. \end{cases}$$

4. Решите систему из 3 уравнений

5. Решите систему линейных уравнений № 4, матрица и вектор правой части которой хранятся в текстовых файлах matr.txt, rside.txt, и запишите результат в файл sol.txt.

**Лабораторная работа №5-6.
Построение графиков функции в системе Matlab**

Цель работы: Работа с графическими элементами в среде MATLAB для графическими, 2 и 3-х мерных графиков функций по укреплению навыков и опыта, чтобы рисовать и работать.

Оборудовать: Персональный компьютер типа **Pentium**, программа MATLAB.

Визуализация данных в MATLAB для графических возможностей высокого уровня. Все графики, меню и панели инструментов описаны в отдельном графическом окне. В меню и панели элементов совершенствовать графическое окно ясно и может изменяться в соответствии с запросом пользователя

1. Двумерные графики.

В MATLAB для создания двумерную графика используются следующие команды:

- **loglog, polar, stairs, area, pcolor, line, pie, plot, semilogx, comet, bar, fill, colormap, ribbon, pie3, strips, semilogy, stem, barh, patch, rectangle, scatter, errorbar, imagesc** va h.k.;

Рисунки, графики и графические дизайны окон и графического управления окнами, меню элементы и инструменты, выполняемых следующих команд:

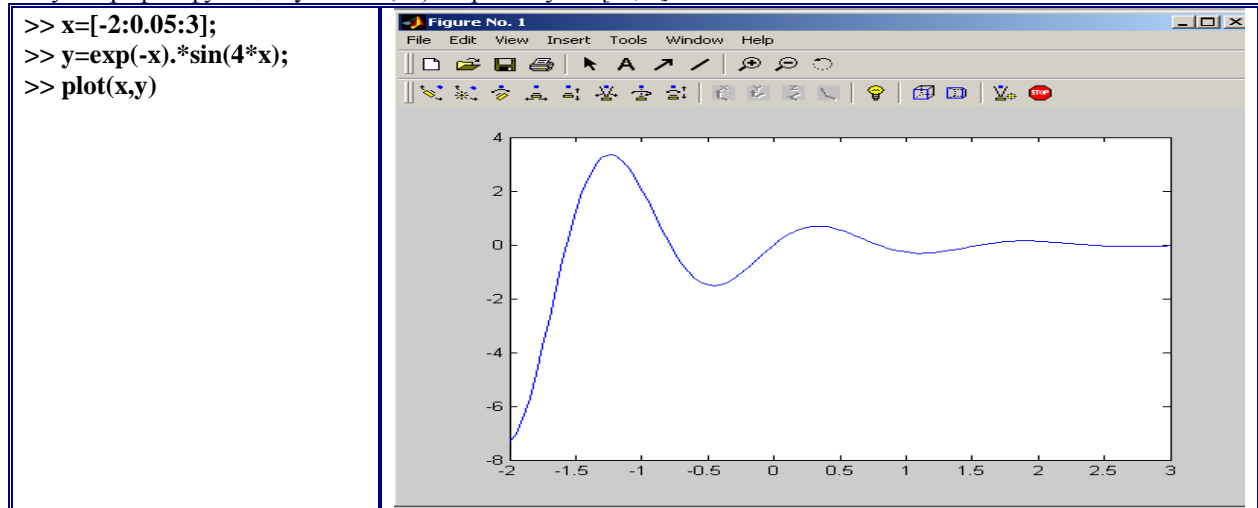
- **grid, axis, hold, figure, shg, clf, subplot** и т.д.;

Обычно имеют переменную порядок сюжета и его различные формы будут использоваться для создания графики.

Сокращенный вид	Выполнение
<pre>>> x=[a:h:b]; >> y=f(x); >> plot(x,y)</pre>	<p>Первый шаг з куски добавляются , то функция будет добавлена, а затем добавляется к команде участок.</p>

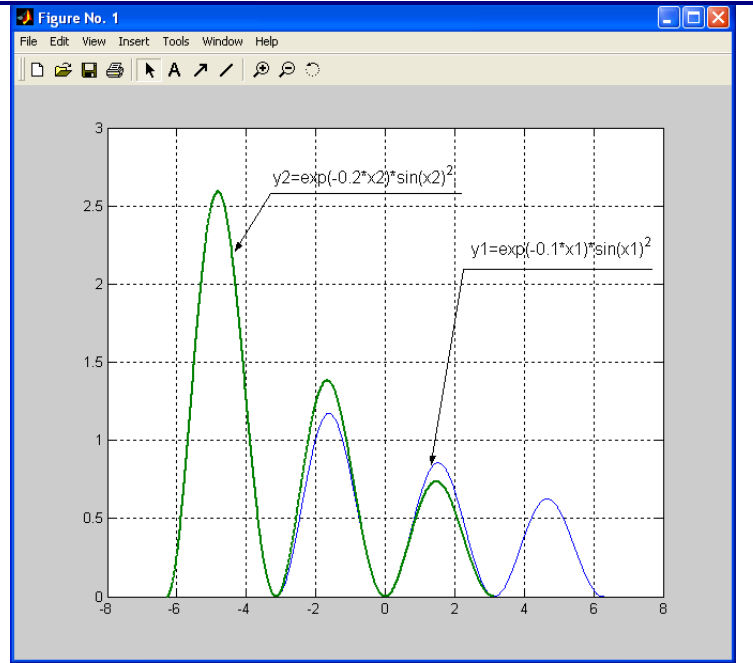
ЗАДАНИЕ - 1.

Рисуем график функции $y = e^{-x}\sin(4x)$ в промежутке [-2; 3]:



ЗАДАНИЕ -2.

```
>>x1=[-pi:0.01:2*pi];
>>y1=exp(-0.1*x1).*sin(x1).^2;
>>x2=[-2*pi:0.01:pi];
>>y2=exp(-0.2*x2).*sin(x2).^2;
>>plot(x1,y1,x2,y2)
```

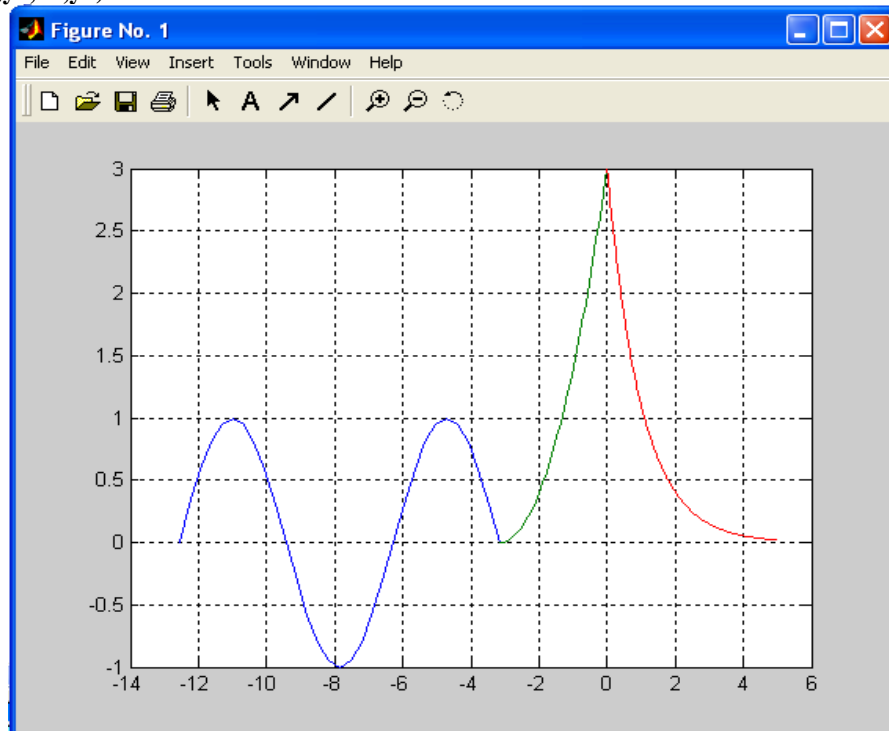


ЗАДАНИЕ -3.

$$y(x) = \begin{cases} \sin x & -4\pi \leq x \leq -\pi \\ 3(x/\pi + 1)^2 & -\pi < x \leq 0 \\ 3e^{-x} & 0 < x \leq 5 \end{cases}$$

Рисуем график функции:

```
>>x1=[-4*pi:pi/10:-pi];
>>y1=sin(x1);
>>x2=[-pi:pi/30:0];
>>y2=3*(x2/pi+1).^2;
>>x3=[0:0.02:5];
>>y3=3*exp(-x3);
>>plot(x1,y1,x2,y2,x3,y3)
```

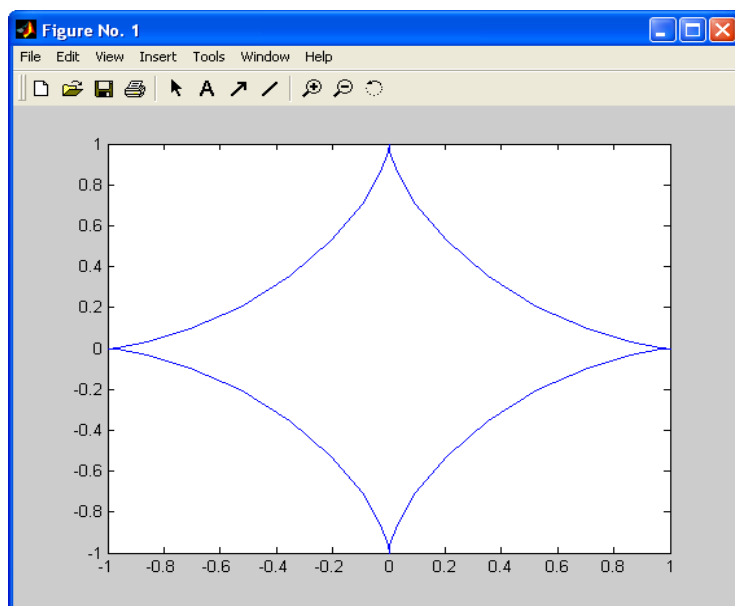


ЗАДАНИЕ – 4.

$$x(t) = \cos^3 t, \quad y(t) = \sin^3 t, \quad t \in [0, 2\pi].$$

Рисуем график функции с параметрами:

```
>> t=[0:pi/20:2*pi];  
>> x=cos(t).^3;  
>> y=sin(t).^3;  
>> plot(x,y)
```



2. Трехмерная график.

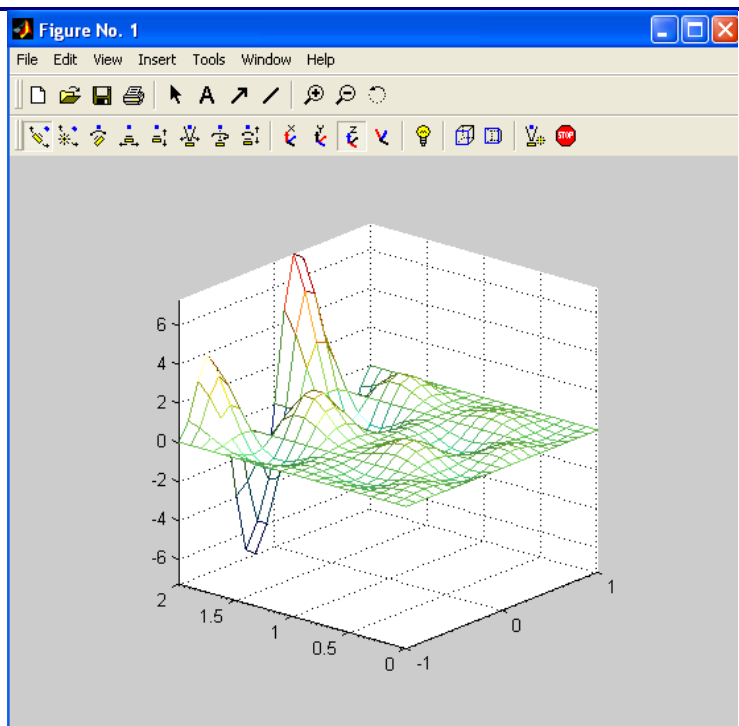
Используются следующие команды в рисовании трехмерные графики в MATLAB:

- **bar3, plot3, mesh, surf, sphere, cylinder, bar3h, contour, meshgrid, fill3, ellipsoid, logo** va h.k.;

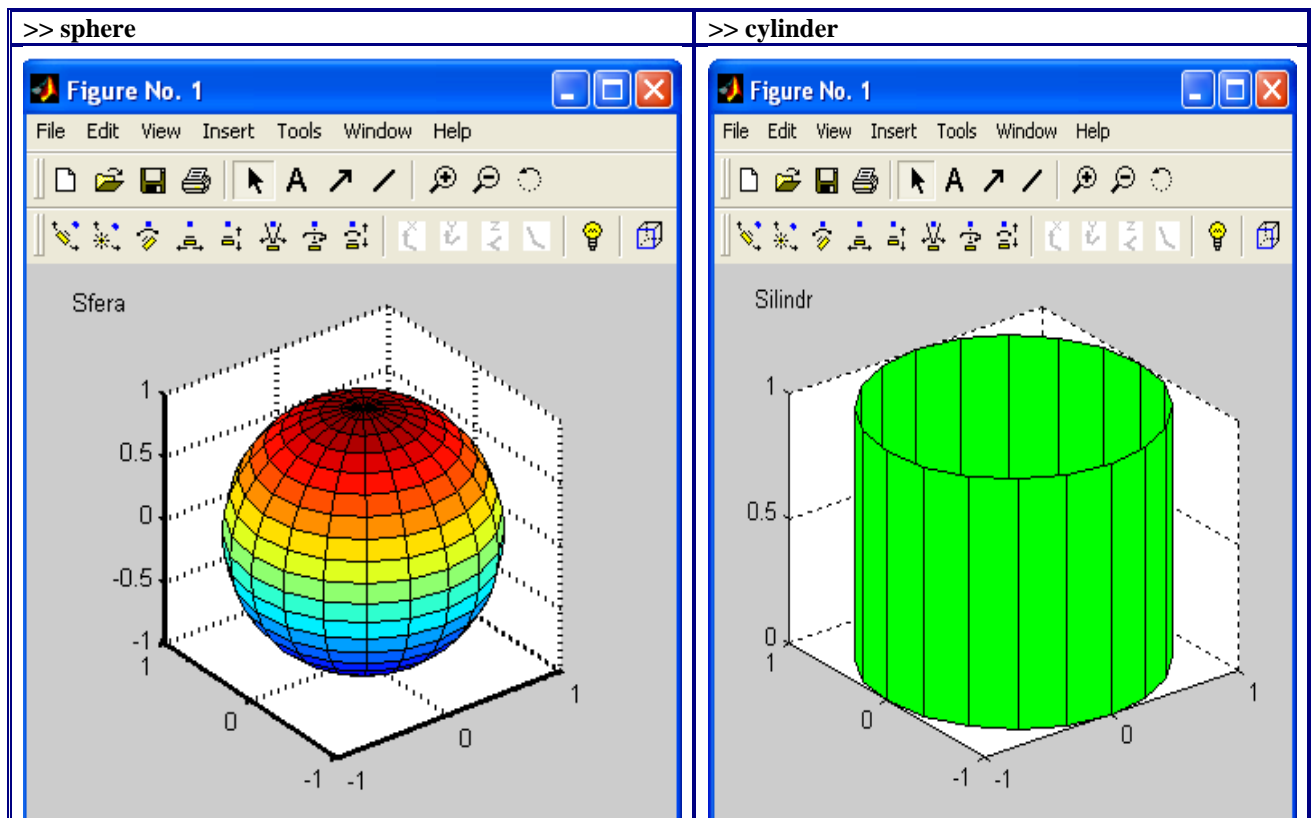
Обычно много переменных функционируют тест для того, чтобы рисовать графики и различные ее проявления.

Пример:

```
>>[X,Y]=meshgrid(-1:0.1:1, 0:0.1:2);  
>> Z = 4*sin(2*pi*X).*cos(1.5*  
pi*Y).*(1-X.^2).*Y.*(1-Y);  
>> mesh(X,Y,Z)
```



Описание геометрических объектов в космической сфере **sphere, cylinder, bar3h, ellipsoid** эллипсоида в виде серии команд.



КОНТРОЛЬНЫЕ ЗАДАНИЯ

1. Построить графики функции с помощью декартовой системы координат:

1. $f(x) = \sin x$; $g(x) = \sin^2 x$; $x \in [-2\pi, 3\pi]$.
2. $u(x) = 0.01x^2$; $v(x) = e^{-|x|}$; $x \in [-0.2, 9.4]$.
3. $f(x) = \sin x^2$; $g(x) = \cos x^2$; $x \in [-\pi, \pi]$.
4. $u(x) = x/20$; $v(x) = e^x$; $x \in [-2, 2]$.
5. $f(x) = x^3 + 2x^2 + 1$; $g(x) = (x - 1)^4$; $x \in [-1, 1]$.
6. $u(x) = \sqrt{x}$; $v(x) = e^{-x^2}$; $x \in [0, 1]$.
7. $f(x) = \ln x$; $g(x) = x \ln x$; $x \in [0.2, 10]$.
8. $u(x) = x^{1/\beta}$; $v(x) = \sqrt{x}$; $x \in [0, 8]$.
9. $f(x) = |2x|^3$; $g(x) = |2x|^5$; $x \in [-0.5, 0.5]$.
10. $u(x) = \sqrt{|x|}$; $v(x) = x^{1/\beta}$; $x \in [-0.6, 0.5]$.

2. Построить графики функции с помощью декартовой системы координат:

$$1. f(x) = \begin{cases} -1, & -3 \leq x \leq -1 \\ x, & -1 < x \leq 1 \\ e^{1-x}, & 1 < x \leq 3 \end{cases}$$

$$2. \quad f(x) = \begin{cases} \sqrt{x}, & 0 \leq x \leq 1 \\ 1, & 1 < x \leq 3 \\ (x-4)^2, & 3 < x \leq 5 \end{cases}$$

$$3. \quad f(x) = \begin{cases} \ln x, & 1 \leq x \leq e \\ x/e, & e < x \leq 9 \\ 9e^{8-x}, & 9 < x \leq 12 \end{cases}$$

$$4. \quad f(x) = \begin{cases} \sin x, & -2\pi \leq x \leq 0 \\ -x^3, & 0 < x \leq 1 \\ \cos \pi x, & 1 < x \leq 3\pi \end{cases}$$

$$5. \quad f(x) = \begin{cases} \arcsin x - 1, & 0 \leq x \leq 1 \\ \frac{\pi}{2} - x, & 1 < x \leq \frac{\pi}{2} \\ \cos x, & \frac{\pi}{2} < x \leq \pi \end{cases}$$

3. Построить графики в пространстве следующие функции:

$$1) \quad z(x, y) = \sin x \cdot e^{-3y} \quad x \in [0, 2\pi] \quad y \in [0, 1]$$

$$2) \quad z(x, y) = \sin^2 x \cdot \ln y \quad x \in [0, 2\pi] \quad y \in [1, 10]$$

$$3) \quad z(x, y) = \sin^2(x - 2y) \cdot e^{-|y|} \quad x \in [0, \pi] \quad y \in [-1, 1]$$

$$4) \quad z(x, y) = \frac{x^2 y^2 + 2xy - 3}{x^2 + y^2 + 1} \quad x \in [-2, 2] \quad y \in [-1, 1]$$

$$5) \quad z(x, y) = \frac{\sin xy}{x} \quad x \in [0.1, 5] \quad y \in [-\pi, \pi]$$

КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Как сделать пространство в равнинных и графических команд? Разговор о параметрах этой команды .
2. Выражают взгляды и графики функций , какие команды ушли в прошлое? Их настройки
4. Каков порядок сюжета, и цель ее?
5. В данной системе определяется через двумерной сфере неравенства , как командовать делом прошлого?
6. Космические поверхности и кривые графика команды ушли в прошлое?
7. bar3 о возможностях и заказов plot3

Лабораторная работа 7. Ознакомление с операционной средой Simulink и простейшие модели

Цель работы: изучение основ работы в среде SIMULINK и создание простейших моделей.

1.1 Общие сведения о программе MatlabSimulink

Simulink – это среда имитационного моделирования. Для начала работы в Simulink необходимо запустить пакет Matlab. При запуске этого пакета появится окно, показанное на рис. 1.1.

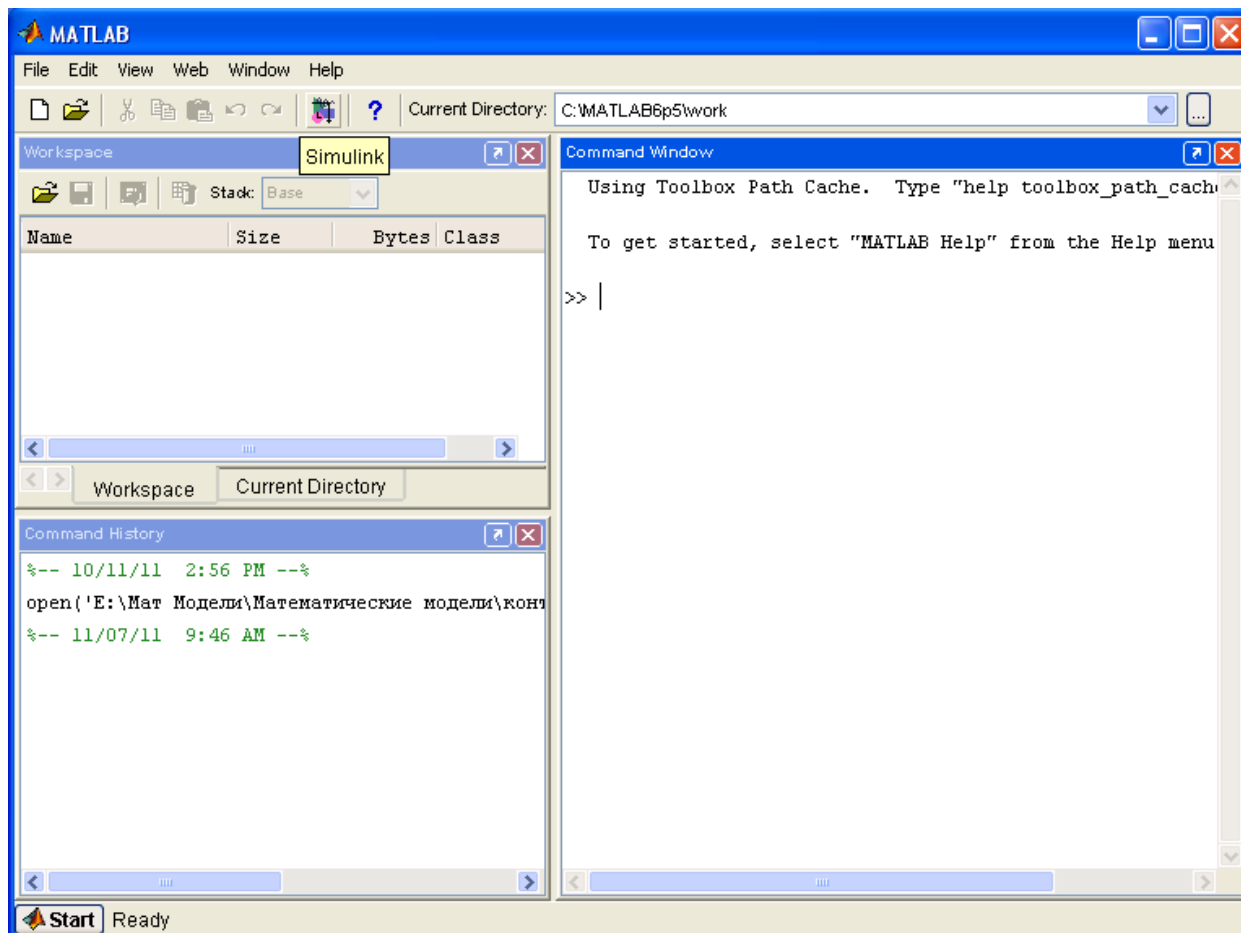



Рис.1.1. Окно программы Matlab

Программа Simulink запускается с помощью кнопки  (Simulink) на панели инструментов командного окна Matlab (на рис. 1.1 на эту кнопку наведён курсор мыши). При нажатии этой кнопки открывается библиотека компонентов. Библиотека компонентов содержит большое количество элементов (блоков), необходимых для создания модели. Из программы Simulink библиотека вызывается с помощью кнопки (LibraryBrowser), расположенной на панели инструментов. Внешний вид обозревателя окна библиотеки представлен на рис.1.2.

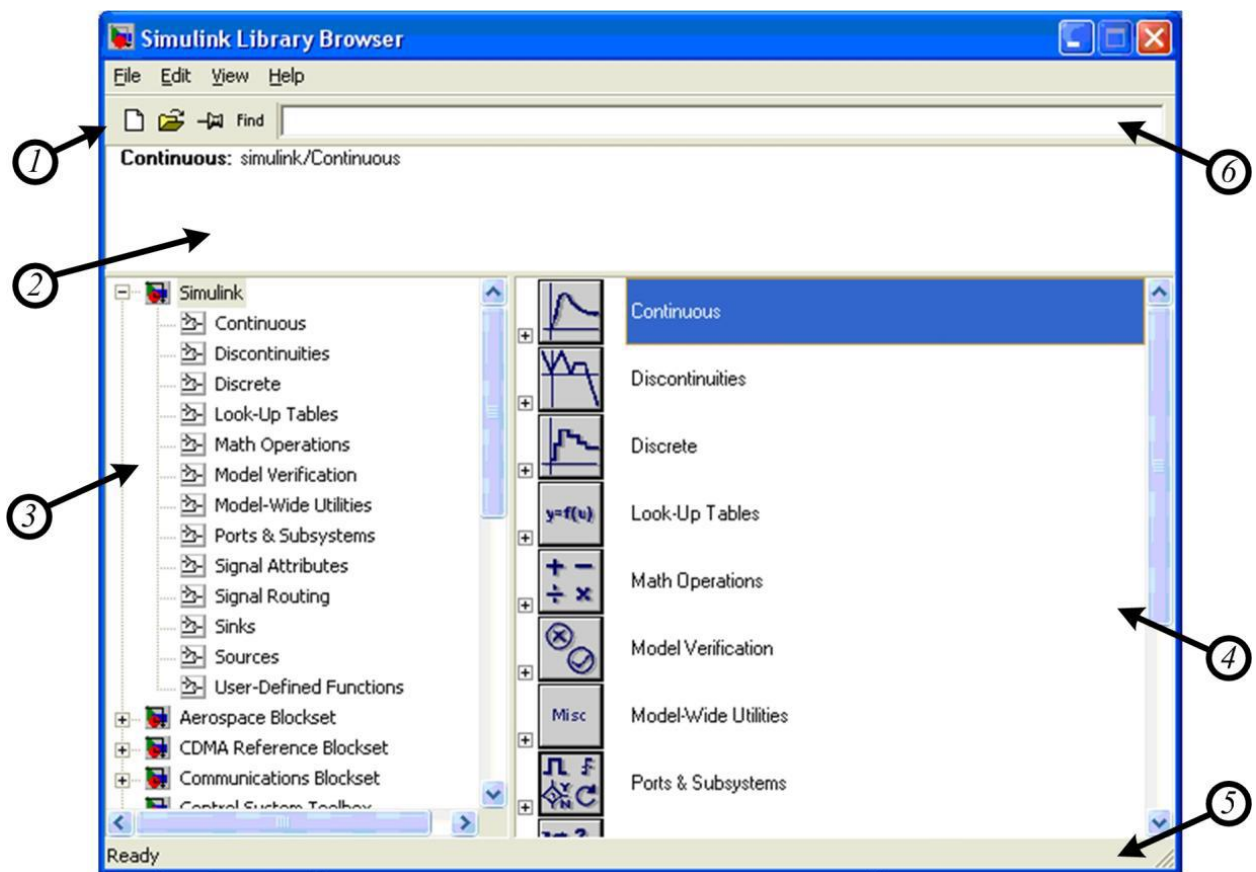


Рис. 1.2. Внешний вид обозревателя окна библиотеки:

1 – панель инструментов с кнопками наиболее часто используемых команд; 2 – окно комментария для вывода поясняющего сообщения о выбранном разделе библиотеки и блоке; 3 – список разделов библиотеки; 4 – список вложенных блоков или разделов библиотеки. Показывается содержимое раздела, указанного в списке разделов библиотеки; 5 – строка состояния, содержащая подсказку по выполняемому действию; 6 – строка поиска блоков

Для поиска необходимого блока проще всего воспользоваться автоматическим поиском по названию. Для этого в строку поиска записывается название (можно только несколько букв из названия) искомого блока и нажимается клавиша **Enter**. После этого начинается автоматический поиск компонента по введенному названию. Если требуемый блок найден, то он отображается в списке вложенных блоков как выбранный. Если при этом найденный блок не является искомым, а имеется лишь совпадение в названии, то следует повторно нажать клавишу **Enter**, после чего начнется дальнейший поиск блоков по заданному названию.

Для создания модели в среде Simulink необходимо создать новый файл с помощью команды **File**→**New**→**Model** или с помощью кнопки (Newmodel). После выполнения этой команды появится окно, в котором строится модель. Открытие созданной ранее модели происходит с помощью команды **File**→**Open**.

После того, как необходимый блок найден в библиотеке, его необходимо переместить в окно модели. Это осуществляется «перетаскиванием» необходимого блока из списка обозревателя библиотеки в рабочее поле окна модели с помощью курсора (левой клавишей мыши).

После перемещения блока необходимо расставить в нужном порядке. Для удаления блока необходимо его выделить и нажать клавишу **Delete**. Для операций над блоком необходимо нажать на нём правой кнопкой мыши и выбрать пункт **Format**. Этот пункт имеет следующие команды:

Font – формат текста названия блока;

HideName (ShowName) – убрать название (показать название);

FlipName – переместить название на противоположную сторону блока;

FlipBlock – зеркально отразить блок;

RotateBlock – повернуть блок по часовой стрелке (Clockwise) или против часовой стрелки (Counterclockwise);

ShowDropShadow (HideDropShadow) – показать тень (убрать тень).

Копирование блоков можно как с помощью стандартных сочетаний клавиш Windows, так и с помощью клавиши **Ctrl**. Во втором случае необходимо зажать клавишу **Ctrl**, а затем перетащить соответствующий блок в необходимое место.

Также, если требуется, изменяются параметры блоков. Параметры блока вызываются двойным щелчком левой кнопкой мыши по соответствующему блоку. При изменении численных параметров в качестве десятичного делителя используется точка, а не запятая. После редактирования параметров следует нажать клавишу **OK** или **Apply**.

На рис. 1.3 представлено окно модели с блоками *Scope*, *Constant* и *Display*.

После расстановки блоков в рабочем поле модели их необходимо соединить. Блоки соединяются через входы и выходы. Вход обозначается стрелкой, направленной в блок, а выход – стрелкой, направленной из блока (на рис. 1.3 элемент *Scope* имеет один вход и не имеет выхода, а элемент *Constant* имеет один выход и не имеет входа). Для соединения двух блоков необходимо нажать на выход первого блока, а затем, не отпуская клавиши мыши, провести линию связи до входа второго блока. Если соединение произведено правильно, то линия связи будет сплошной. В случае неправильного (незаконченного) соединения линия связи будет штриховой красного цвета. При этом следует помнить, что таким способом нельзя соединить два входа или два выхода. Для этого требуются специальные соединители, о которых будет рассказываться позже. Если требуется соединить три блока, то следует сначала провести линию связи между двумя из них, а вход (выход) третьего соединить с созданной линией связи, соединяющей два первых блока. Можно также соединение с третьим блоком осуществлять от линии связи. Для этого соединение необходимо производить правой кнопкой мыши.

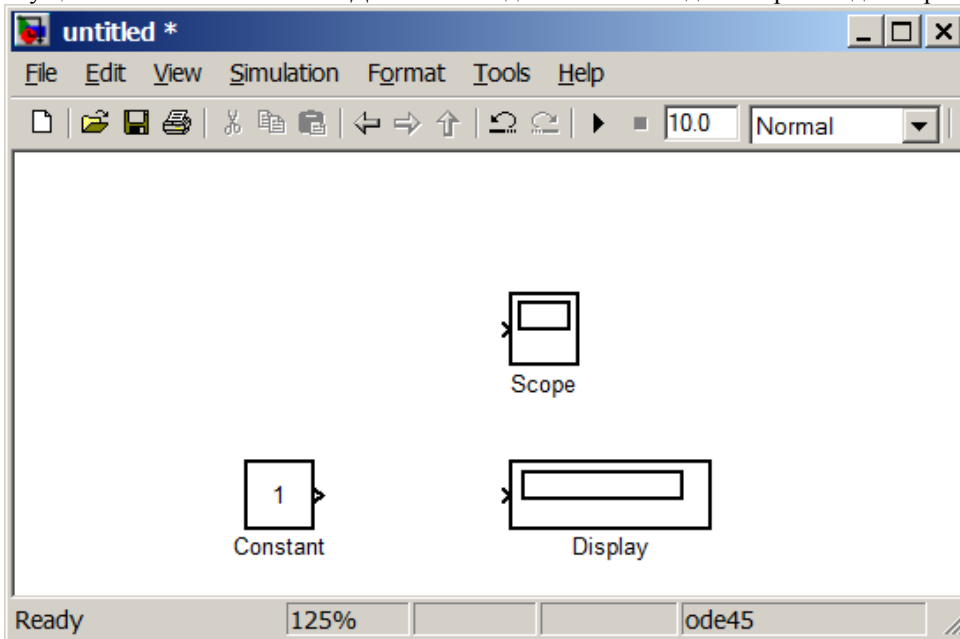


Рис. 1.3. Окно модели с блоками

Также существует другой способ соединения блоков. Для этого нужно выделить первый блок и нажать клавишу **Ctrl**. Не отпуская этой клавиши, следует нажать на второй блок. При этом будет проведена линия связи между выходом первого блока и входом второго блока. Если число выходов и входов больше одного, то будет проведено столько линий связи, сколько имеется выходов на первом элементе.

На рис. 1.4 показано окно модели с соединёнными блоками

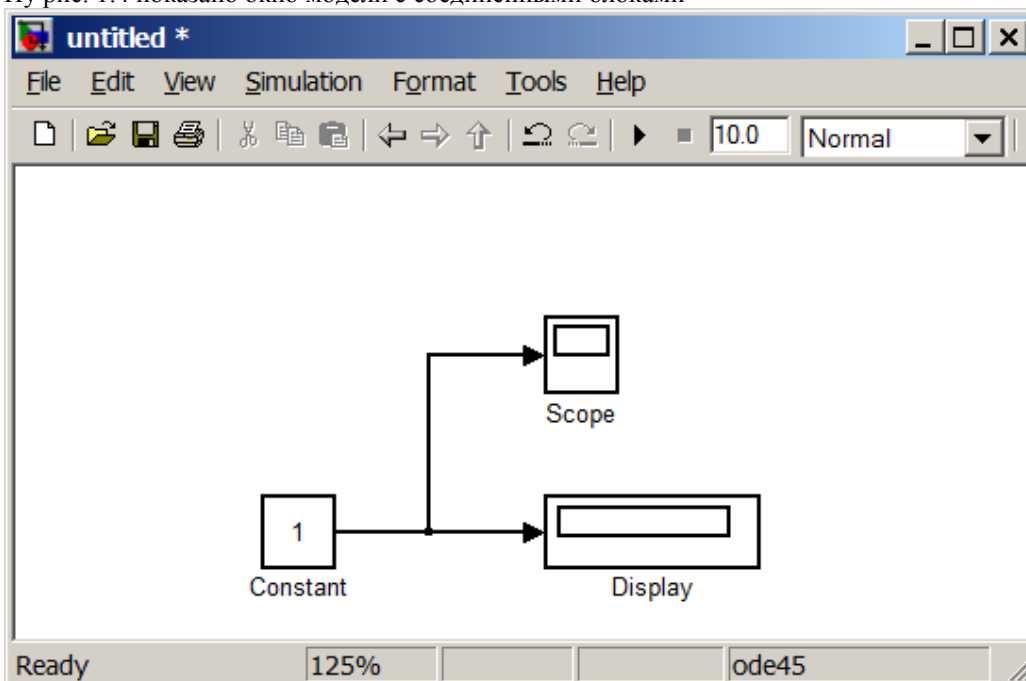
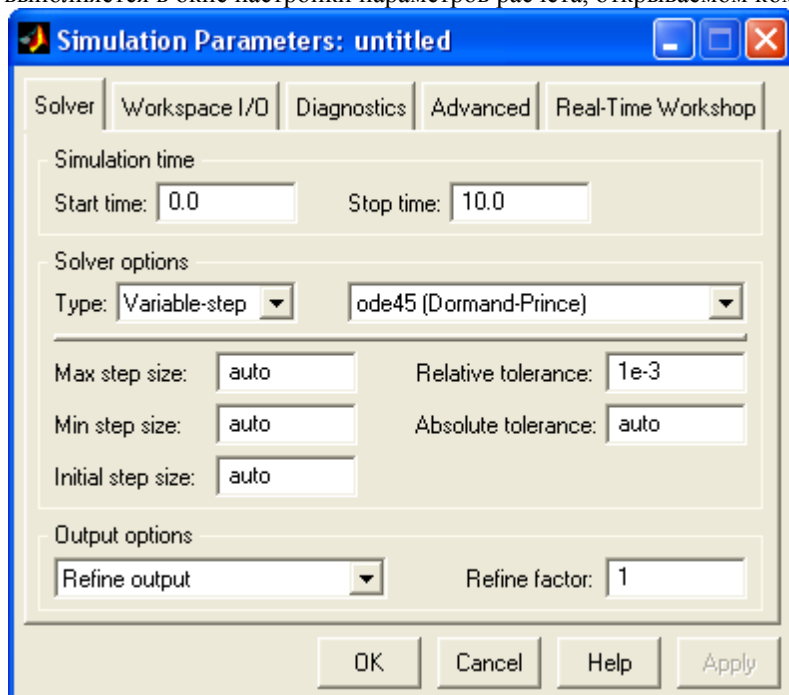


Рис.1.4. Окно модели с соединёнными блоками

Сохранение модели производится с помощью команды **File**→**SaveAs**. Имя файла не должно содержать более 64 символов, должно начинаться с буквы и содержать только алфавитно-цифровые символы латиницы. Такое же требование (за исключением количества символов) относится к пути файла.

Рис. 1.5. Вид окна настройки параметров расчёта

Перед запуском модели необходимо установить параметры моделирования. Задание этих параметров выполняется в окне настройки параметров расчёта, открываемом командой **Simulation**→**SimulationParameters**.



Вид окна настройки параметров расчёта показан на рис. 1.5.

В левой части окна находится список групп настраиваемых параметров. Установка параметров моделирования выполняется с помощью панели **Solver**. Элементы этой панели разделены на две группы: **Simulationtime** (интервал времени моделирования) **Solveroptions** (параметры расчёта).

Интервал времени моделирования задаётся путём указания начального (**Starttime**) и конечного (**Stoptime**) значений времени расчёта. Как правило, начальное время указывается равным нулю, а конечное время зависит от условий поставленной задачи.

В параметрах расчёта необходимо выбрать метод интегрирования (**Type**): с фиксированным шагом (**Fixed-step**) или с переменным шагом (**Variable-step**). Максимальный шаг интегрирования задаётся в графе **MaxStepSize**. Также в параметрах расчёта необходимо указать один из следующих методов:

ode45 – одношаговые явные методы Рунге-Кутты 4-го и 5-го порядка;

ode23 – одношаговые явные методы Рунге-Кутты 2-го и 3-го порядка;

ode113 – многошаговый метод Адамса-Башворта-Мултона переменного порядка;

ode15s – многошаговый метод переменного порядка (от 1 до 5), использующий формулы численного дифференцирования;

ode23s – одношаговый метод, использующий модифицированную формулу Розенброка 2-го порядка;

ode23t – метод трапеций с интерполяцией;

ode23tb – неявный метод Рунге-Кутты в начале решения и метод, использующий формулы обратного дифференцирования 2-го порядка в последующем.

После редактирования параметров следует нажать клавишу **OK** или **Apply**.

Запуск расчёта (моделирования) выполняется с помощью команды **Simulation**→**Start** или нажатием кнопки на панели инструментов.

1.2 Простейшие модели

Рассмотрим простейшие модели. На рис. 1.6 представлена модель со следующими элементами:

Constant – генератор постоянного сигнала. На выходе создаёт постоянный сигнал. Величина сигнала задаётся в параметрах (**Constantvalue**).

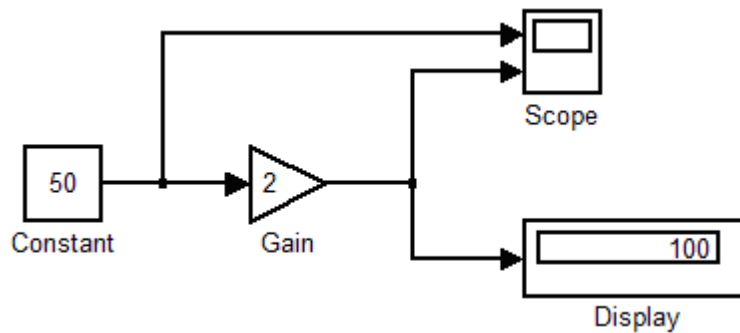







Рис. 1.6. Модель умножения числа (по-сле проведения моделирования)

Gain– линейный умножитель. На выходе создаёт сигнал, равный входному сигналу, умноженному на заданное значение. Величина множителя задаётся в параметрах (Gain). Для получения математической функции деления необходимо в строку Gain записывать «1/x», где x – величина, на которую требуется поделить входной сигнал.

Scope– осциллограф. Строит график зависимости входного сигнала от времени. Для просмотра графика необходимо дважды щелкнуть левой кнопкой мыши на блоке. Осциллограф может иметь несколько входов.

Для этого необходимо в параметрах осциллографа (кнопка ) указать количество входов (Numberofaxes). В случае, когда входной сигнал имеет значения больше или значительно меньше шкалы графика, следует нажать на кнопку  (Autoscale). Для увеличения масштаба используются кнопки:  – увеличение масштаба

по заданной области;  – увеличение масштаба по горизонтальной оси;  – увеличение масштаба по вертикальной оси. Также в параметрах можно задавать предел точек, из которых строится график. Во всех лабораторных работах необходимо исключить такой предел точек. Для этого нужно в настройках убрать галочку «LimitDataPointstoLast».

Display– дисплей. Показывает величину входного сигнала в виде числовой записи на самом блоке. В параметрах можно менять формат записи числа (целое число из пяти цифр, число с символами степени десяти и т. д.).

На рис. 1.6 элемент *Constant* имел значение 50, а элемент *Gain* – значение 2. Начальное время моделирования было 0, а конечное – 1. Метод моделирования – ode23s. После того как схема собрана и параметры установлены, необходимо запустить моделирование. Осциллограмма, построенная элементом *Scope* показана на рис. 1.7.

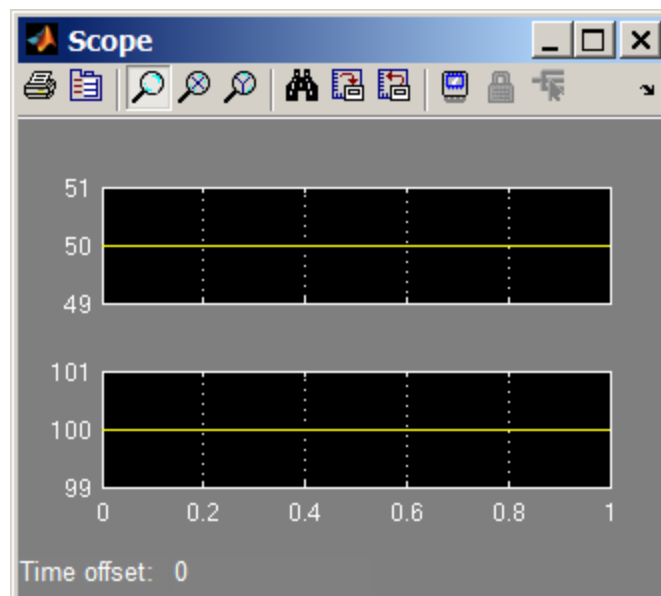


Рис. 1.7. График, построенный элементом *Scope* Анализируя представленную модель, можно наглядно убедиться в работоспособности элементов: с выхода элемента *Constant* на вход элемента *Gain* поступает сигнал, равный 50. Элемент *Gain* умножает этот сигнал на заданное значение 2 и на его выходе образуется сигнал, равный 100.

На следующих моделях можно рассмотреть моделирование электрических цепей. При этом будут использованы следующие блоки:

- *DC voltage source*– источник постоянного напряжения. В параметрах можно изменять величину генерируемого напряжения (Amplitude), В. Внешний вид блока показан на рис. 1.8, а.

- *AC voltage source*– источник переменного напряжения. Выходное напряжение изменяется по синусоидальному закону. В параметрах изменяется амплитуда напряжения (Amplitude), В; начальная фаза (Phase), эл. град.; частота (Frequency), Гц. Внешний вид блока показан на рис. 1.8, б.
- *Series RLC Branch*– последовательный RLC контур. Представляет собой последовательно соединённые активное, индуктивное и ёмкостное сопротивление. В параметрах можно изменять активное сопротивление (Resistance), Ом; индуктивность (Inductance), Гн; ёмкость (Capacitance), Ф. Внешний вид блока показан на рис. 1.8, в.
- Для получения только одного сопротивления необходимо в элементе *Series RLC Branch* исключить остальные два. Исключения сопротивлений осуществляются проставлением для них следующих параметров: для активного сопротивления – «0», для индуктивного сопротивления – «0», для ёмкостного сопротивления – «inf» (англ. *infinity*– бесконечность). На рис.1.8, г показан элемент, представляющий два последовательно соединённых сопротивления RL. Для получения такого элемента в *Series RLC Branch* заданы параметры R = 10; L = 10; C = inf.
- *Ground*– заземление. Внешний вид блока показан на рис. 1.8, д, е. Может быть с выходом (рис.1.8, д) или с входом (рис. 1.8, е).
- *Voltage measurement*– датчик напряжения. На выходе создаёт сигнал, численно равный разности потенциалов на входе блока (между входами «+» и «-»). Внешний вид блока показан на рис. 1.8, ж.
- *Current measurement*– датчик тока. На выходе создаёт сигнал, численно равный силе тока, проходящего через него (между входами «+» и «-»). Внешний вид блока показан на рис. 1.8, з.
- T-connector*– соединитель линий связи. Используется для соединения двух выходов с одним входом. Внешний вид блока показан на рис. 1.8, к.

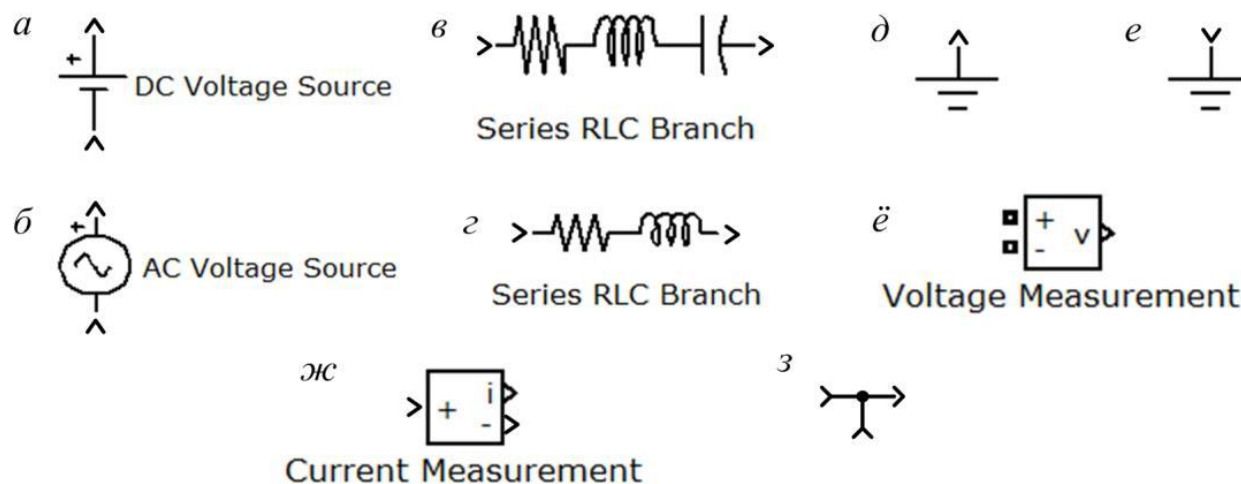


Рис. 1.8. Элементы, необходимые для создания моделей электрических цепей

На рис. 1.9 представлена модель простой электрической цепи постоянного тока, содержащей источник постоянного напряжения и активное сопротивление. Источник постоянного напряжения генерирует напряжение величиной $U = 10$ В, активное сопротивление составляет $R = 3$ Ом. Время моделирования 0–2 с; метод расчёта *ode23s*.

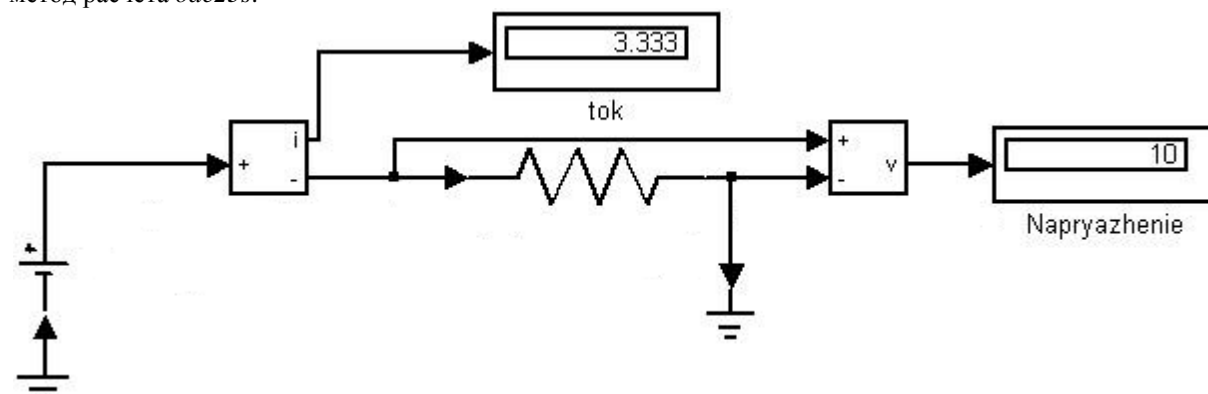


Рис. 1.9. Модель простой электрической цепи постоянного тока

В модели названия блоков Constant изменены, а названия остальных элементов убраны. Для изменения названия блока необходимо щелкнуть по его названию левой кнопкой мыши.

Правильность расчёта составленной модели (рис. 1.9) можно проверить с помощью закона Ома:

$$I = \frac{U}{R} = \frac{10}{3} = 3,333 \text{ A.}$$

Проведённый расчёт подтверждает правильность моделирования.

На рис. 1.10 представлена модель простой электрической цепи переменного тока, содержащая источник переменного напряжения и последовательно соединённые активное, индуктивное и ёмкостное сопротивление. Параметры источника напряжения: амплитуда – 10 В, частота 50 Гц, фаза – 0. Параметры RLC: R = 0,1 Ом, L = 0,001 Гн, C = 0,15 Ф. Время моделирования 0,9–1 с; метод расчёта *ode23s*; максимальный шаг интегрирования 0,0001.

Правильность моделирования составленной модели (рис. 1.10) можно также проверить с помощью закона Ома:

$$\dot{I}_m = \frac{U_m}{Z} = \frac{10}{0,1 + \frac{1}{j314,16 \cdot 0,15} + j314,16 \cdot 0,001} = 10,437 - j30,574 = 32,3e^{-71,2} \text{ A.}$$

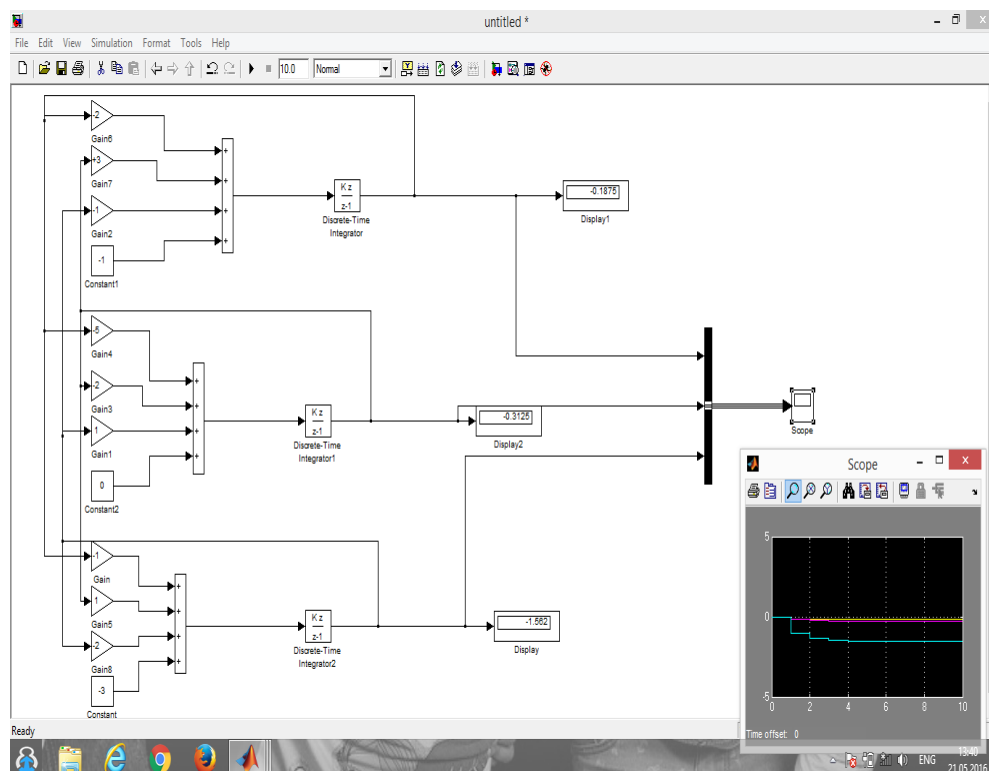
Проведённый расчёт подтверждает правильность моделирования

1.3. Порядок выполнения работы

1. Изучить интерфейс программы и выучить основные понятия.
2. Создать простейшие модели (рис. 1.6, 1.9, 1.10) и произвести их расчёт.
3. Создать в Simulink простейшую модель деления числа. Модель создавать по аналогии с рассмотренной на рис. 1.6. Числа выбираются индивидуально для каждого студента из табл. 1.1. Сохранить внешний вид схемы модели и полученные осциллограммы.
4. Создать и рассчитать в Simulink каждую из электрических схем постоянного тока, представленных на рис. 1.11. Модель создавать по аналогии с рассмотренной на рис. 1.9. Величина эдс источника напряжения и параметры сопротивлений выбираются индивидуально для каждого студента из табл. 1.2. Сохранить внешний вид схемы модели. Проверить правильность моделирования с помощью первого закона Ома.
5. Создать и рассчитать в Simulink одну из электрических схем переменного тока, представленных на рис. 1.12. Модель создавать по аналогии с рассмотренной на рис. 1.10. Величина эдс источника напряжения и параметры сопротивлений выбираются индивидуально для каждого студента из табл. 1.3. Частота источника напряжения – 50 Гц. Сохранить внешний вид схемы модели и полученные осциллограммы. Проверить правильность моделирования с помощью первого закона Ома.
6. По окончании работы сделать вывод.

Пример:

$$\begin{aligned} 2x - 3y + z &= -1 \\ 5x + 2y - z &= 0 \\ x - y + 2z &= 3 \end{aligned}$$



Лабораторная работа . Изучение основ работы на AutoCAD

AutoCAD 2008 может работать как в автономном режиме, так и в локальной сети. Для эффективной работы AutoCAD 2008 под управлением операционной системы Windows необходимы следующие программные и аппаратные средства:

- процессор Intel® Pentium® IV;
- операционные системы: Microsoft® Windows® XP Professional или Home Edition (пакет обновлений SP1 или SP2), Windows XP Tablet PC Edition (SP2) или Windows 2000 (SP3 или SP4). Рекомендуется устанавливать и эксплуатировать AutoCAD либо на операционной системе, локализованной на одном языке с программой, либо на англоязычной версии одной из перечисленных систем;
- Microsoft® Internet Explorer 6.0 (пакет обновлений SP1 или более поздний);
- 512 Мбайт оперативной памяти;
- монитор VGA с разрешением не менее 1024×768 и поддержкой режима true color;
- 750 Мбайт свободного места на жестком диске для установки;
- привод компакт-дисков: любой (только для установки программы).

Требования к системе при использовании AutoCAD 2008 для концептуального проектирования:
процессор Intel®

- с тактовой частотой 3,0 ГГц и выше;
- Windows XP Professional (пакет обновлений SP2);
- не менее 2 Гбайт оперативной памяти;
- 2 Гбайт свободного места на жестком диске, не считая места, необходимого для установки программы;
- монитор с минимальным разрешением 1280×1024 и поддержкой режима true color;
- графический адаптер класса рабочих станций, снабженный не менее 128 Мбайт памяти и поддерживающий технологию OpenGL.

Установка AutoCAD

Чтобы успешно установить AutoCAD, понадобится выполнить несколько подготовительных шагов.

Для подготовки к установке следует:

- выяснить серийный номер — он должен быть напечатан на упаковке AutoCAD 2008. В случае обновления одной из прежних версий AutoCAD требуется использовать серийный номер предыдущей версии. Его можно либо найти на соответствующей упаковке, либо узнать, выбрав в меню AutoCAD пункты Help>About;
- воспользоваться правами администратора локального компьютера, на который будет установлен AutoCAD. Иметь права администратора домена при этом не обязательно;
- закрыть все работающие приложения;
- отключить средства антивирусной защиты.

Установка AutoCAD выполняется с помощью специальной программы-инсталлятора, которая переписывает файлы с компакт-дисков в папку, созданную на жестком диске компьютера. В ходе установки AutoCAD автоматически настраивается на работу с системным устройством указания и WHIP-драйвером монитора.

Запуск системы AutoCAD

Запуск AutoCAD осуществляется следующими способами:

- на Панели задач выберите из меню Start/Programs/Autodesk AutoCAD 2008;
- на Рабочем столе Windows дважды щелкните на пиктограмме AutoCAD 2008.

При запуске AutoCAD создается новый неименованный рисунок.

Можно либо начать создавать в нем объекты, либо загрузить с диска один из уже имеющихся файлов

После запуска AutoCAD выводится диалоговое окно начала работы

Startup(рис. 1.1)

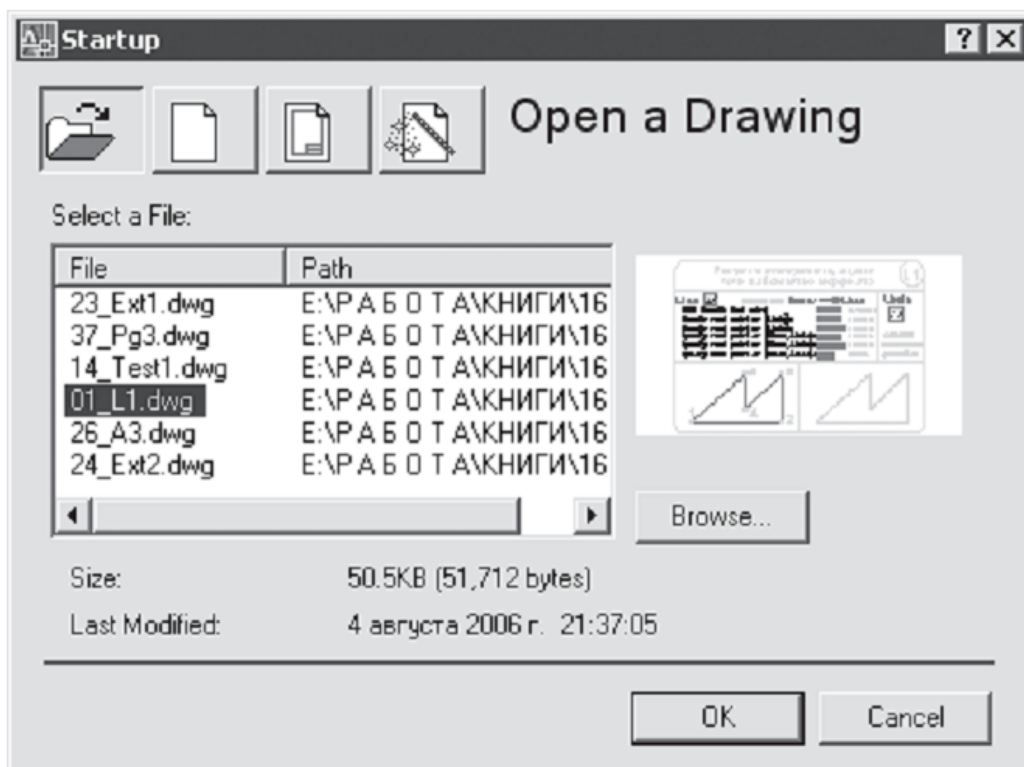


Рис. 1.1. Диалоговое окно начала работы

В диалоговом окне Startup пользователю предлагается четыре кнопки, в зависимости от выбора которых меняется содержимое диалогового окна:

Open a Drawing— для открытия ранее созданного чертежа; **Use a Wizard**— вызов Мастера для установки параметров нового чертежа. Вызов справочной системы В любой момент работы с AutoCAD вы можете получить доступ к электронной документации по программе. Для этого необходимо выбрать в падающем меню пункт Help. Альтернативный вариант — нажать клавишу F1 на функциональной клавиатуре, ввести символ ? в командной строке или щелкнуть на пиктограмме со значком вопроса на стандартной панели инструментов. Открытие рисунков Программа AutoCAD по умолчанию записывает внутреннее представление рисунка в файл с расширением DWG.

AutoCAD предлагает многооконную среду проектирования Multiple Design Environment (MDE), которая допускает одновременное открытие нескольких чертежей. Открыть существующий рисунок можно с помощью команды OPEN, которая вызывается из падающего меню FileOpen...или щелчком на пиктограмме Open...CTRL+Она стандартной панели инструментов. После обращения к команде OPENна экране AutoCAD появляется диалоговое окно выбора файла Select File(рис. 1.2). Здесь можно выбрать имя файла из списка или ввести это название в поле File name:. Для открытия рисунка можно дважды щелкнуть на имени соответствующего файла в Проводнике Windows, что приведет к автоматическому запуску AutoCAD. Если же в системе уже ведется сеанс работы в AutoCAD, то рисунок откроется в нем. Рисунки, созданные в AutoCAD любой предшествующей версии, открываются как любой документ версии 2008 и автоматически преобразуются в новый формат. Создание рисунков Создать новый рисунок позволяет команда NEW, вызываемая из падающего меню FileNew...или щелчком на пиктограмме QNew на стандартной панели инструментов. Start from Scratch— для создания чертежа, где устанавливаются только единицы измерения в области Default Settings: британские (футы и дюймы) или метрические (миллиметры);

Use a Template— для создания чертежа по шаблону — документу, установки которого используются как основа для нового рисунка;

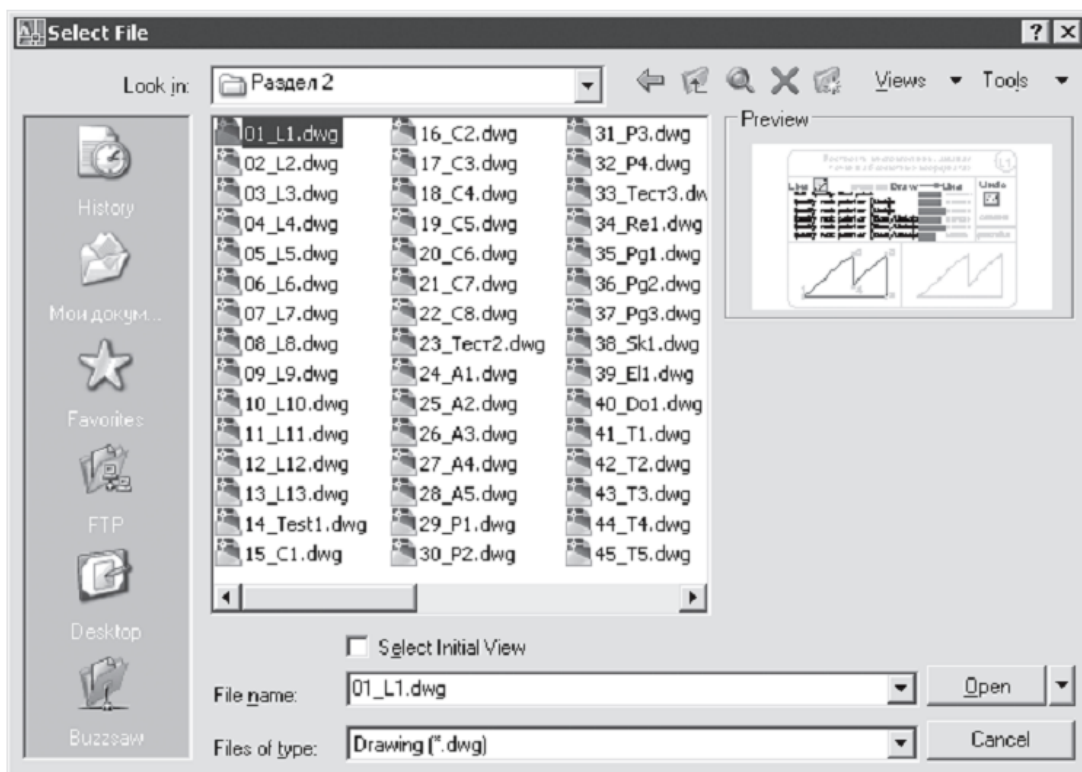


Рис. 1.2. Диалоговое окно выбора файла

После запуска команды необходимые настройки параметров рабочей среды AutoCAD производятся в диалоговом окне создания нового рисунка Create New Drawing — рис. 1.3.

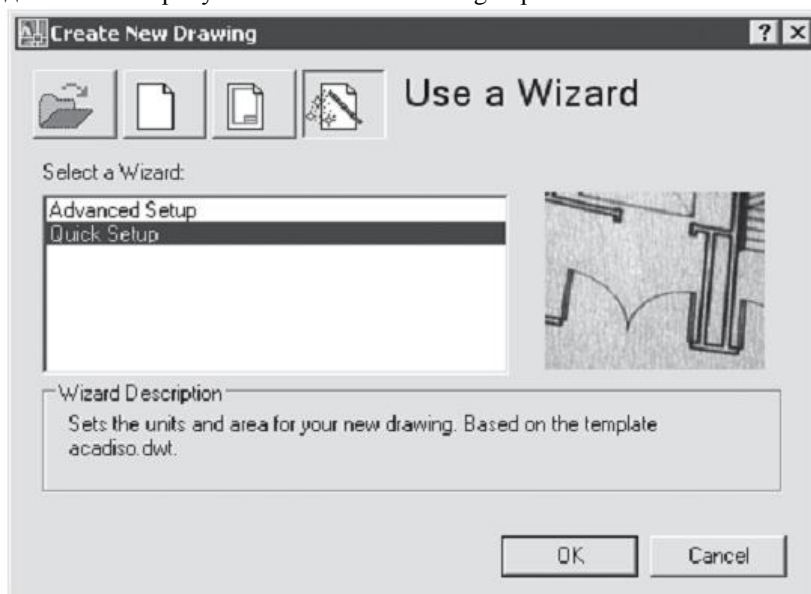


Рис. 1.3. Диалоговое окно создания нового рисунка

В AutoCAD имеется так называемый Мастер подготовки Wizard — служебное средство для создания нового рисунка. Чтобы вызвать Мастер подготовки, необходимо в диалоговом окне Create New Drawing щелкнуть на пиктограмме Use a Wizard. Мастер быстрой подготовки Quick Setup позволяет задать для нового рисунка единицы измерения Units и область рисования Area. Поддерживаются следующие типы единиц для рисования и вычерчивания: десятичные — Decimal, инженерные — Engineering, архи-тектурные — Architectural, дробные — Fractional, научные — Scientific. Указывая ширину Width и длину Length области рисования, пользователь тем самым задает граничные пределы рисунка, так называемые лимиты. Именно лимитами определяется размер чертежа, впоследствии выводимого на плоттер. Мастер детальной подготовки Advanced Setup позволяет задать для нового рисунка тип линейных единиц измерения Units и способ измерения углов Angle, указать начало отсчета угла Angle Measure и направление измерения угла Angle Direction, определить границы области рисунка Area. Любая из установок, произведенных в начале рисования, в дальнейшем может быть изменена.



Лабораторная работа. Создание проекта в TRACE MODE 6

Цель работы: познакомиться с SCADA– системой TRACE MODE 6, научиться создавать и настраивать канал, выводить информацию на экран в среде TRACE MODE.

Задание: создать канал, генератор заданного сигнала (синусоидального, пилообразного, треугольного), произвести привязку генератора к созданному каналу, масштабирование сигнала, обеспечив заданный диапазон; на экране вывести отмасштабированный сигнал с помощью стрелочного прибора, тренда, текста.

Ход работы

1. Создание проекта Trace Mode. Запустите программу

Trace Mode 6 (файл tmdevenv.exe). Для создания нового проекта щелкните левой клавишей мыши на иконке  или на строчке **Новый** в меню **Файл**. Созданный проект примет вид, изображенный на рис. 3.7. При отсутствии навигатора проекта щелкните левой клавишей мыши на строчке **Навигатор проекта** меню **Вид**. Сохраните созданный проект, щелкнув левой клавишей мыши на иконке  или на строчке **Сохранить** или **Сохранить как** в меню **Файл**.

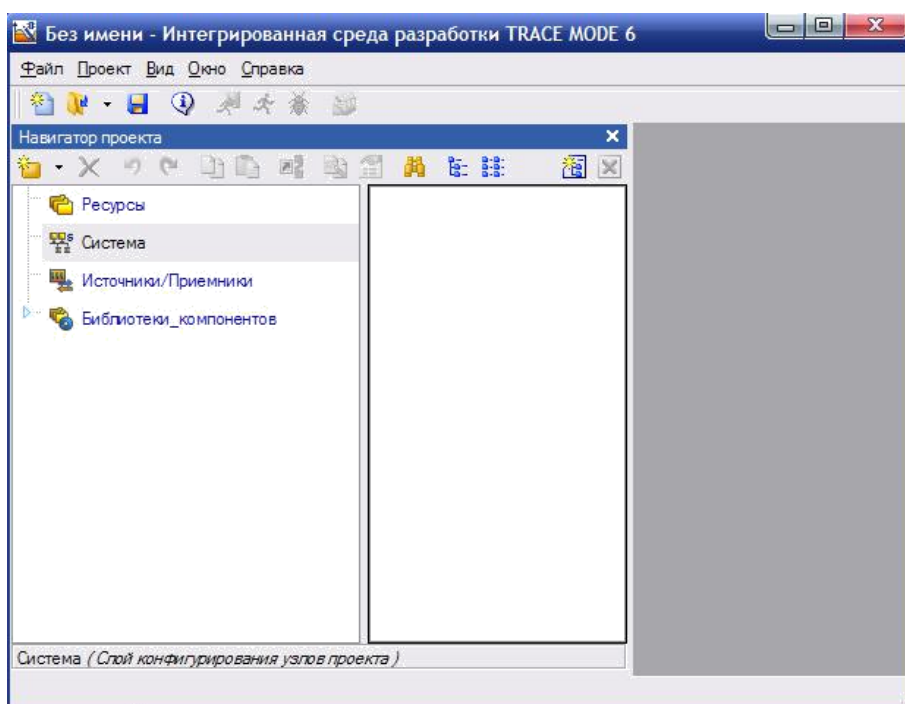


Рис. 3.7 Новы проект Trace Mode

2. **Создание узла.** Выделите строку *система* в навигаторе проекта (рис. 3.8). Вызовите контекстное меню щелчком правой клавиши мыши. В контекстном меню выберите строку *создать узел*. Среди предложенных типов узлов выберите **RTM**.

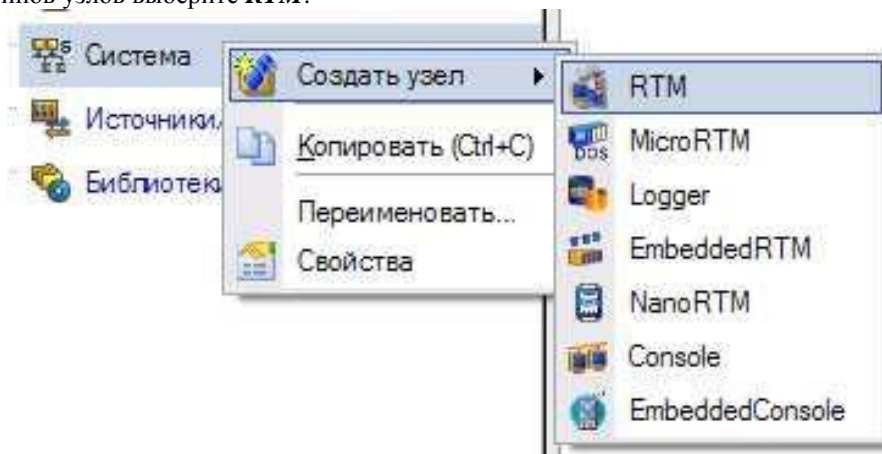


Рис. 3.8 Создание узла

Навигатор проекта в разделе *система* отобразит созданный узел. Выделите созданный узел *RTM* и щелкните левой клавишей мыши по нему. Появится возможность изменить имя узла. Пример созданного узла RTM изображен на рис. 3.9.

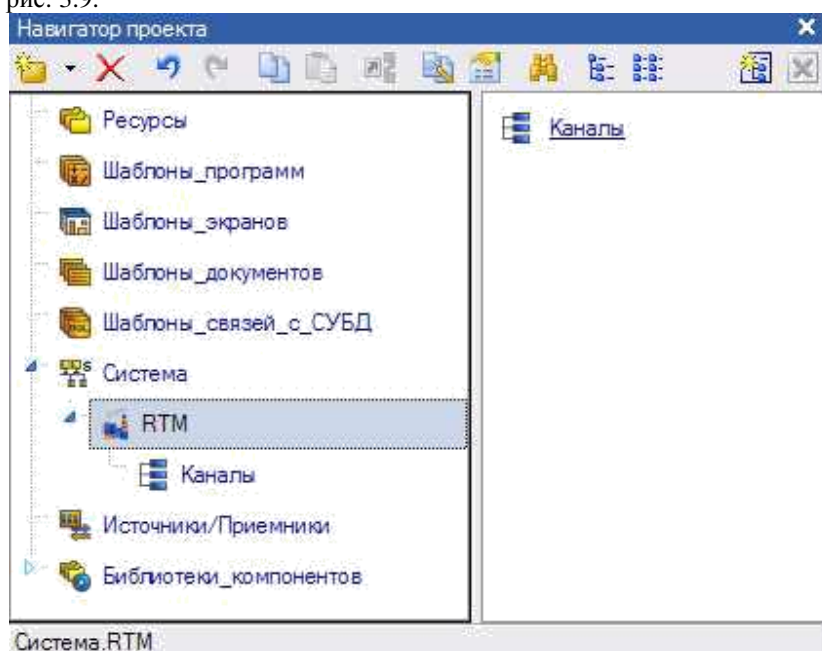


Рис. 3.9 Узел

3. **Создание канала.** Выделите группу *каналы* RTM узла. Вызовите контекстное меню. В появившемся контекстном меню выберите строку *создать компонент*. Среди предложенных компонентов выберите *канал_FLOAT* (рис. 3.10).

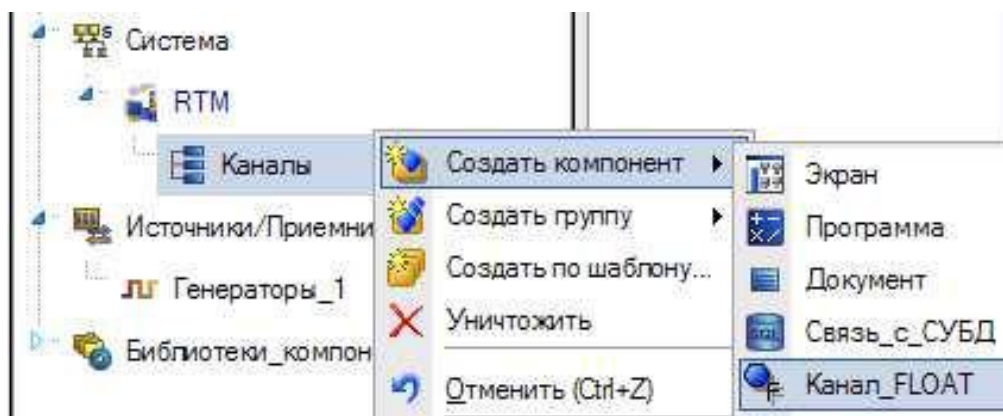


Рис. 3.10 Создание канала

4. **Произведите расчет множителя K и смещения Z .**

Преобразование сигнала будет производиться по формуле:

$$Y = K \cdot X + Z, \text{ где}$$

Y —результат преобразования;

X — входное значение канала;

K — множитель;

Z —смещение.

Входное значение канала изменяется в диапазоне $[0; 100]$. Найдите значение множителя K и смещения Z , которые позволят получить требуемый диапазон изменения результата преобразования (диапазон сигнала).

Пример: необходимо обеспечить диапазон сигнала $[-10; 10]$. Множитель K составит 0,2, смещение Z — -10.

5. **Редактирование созданного канала.** Дважды щелкните левой клавишей мыши на имени созданного канала (канал#1 на рис. 3.11). Откроется окно для редактирования канала (рис. 3.12)

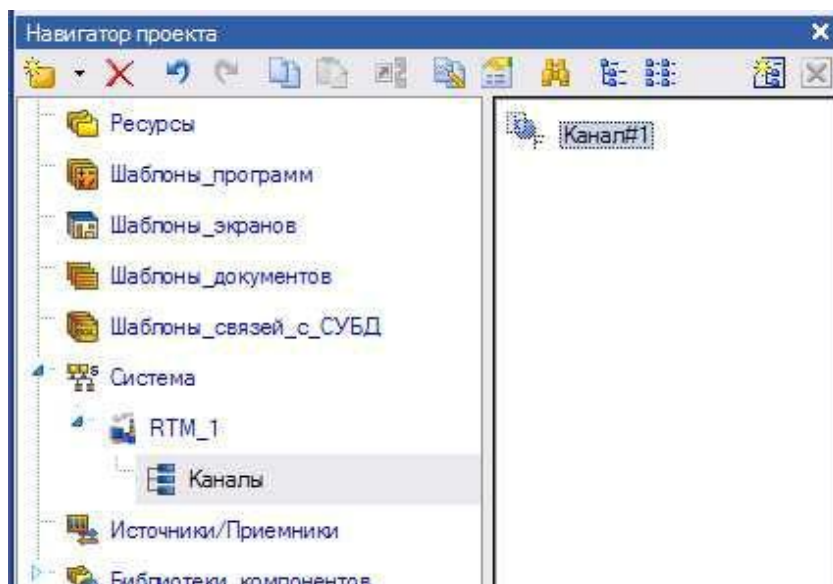


Рис. 3.11 Редактирование канала

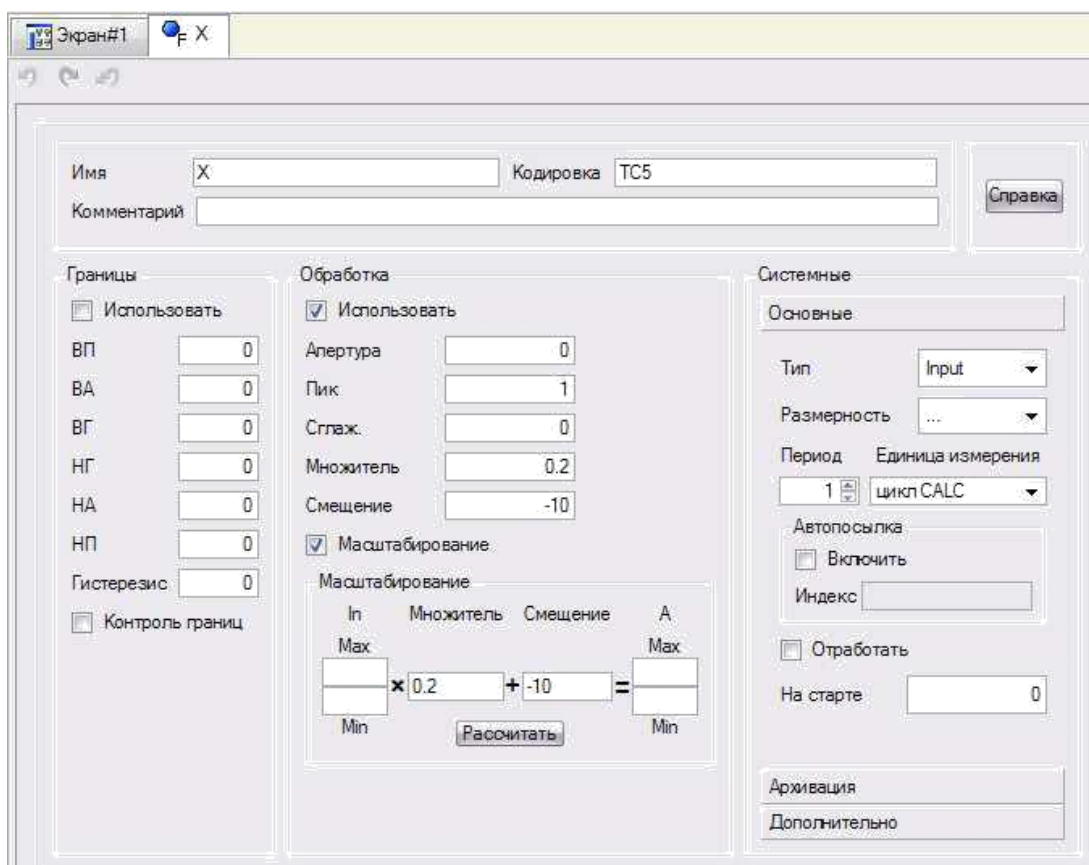


Рис. 3.12 Редактирование канала



В поле *имя* введите новое имя канала. Установите флажок *использовать* на панели *обработка*. Установите *апертуру* равной 0, *пик* равным 1, *сглаж.* равным 0, вычисленные значения множителя *K* и смещения *Z*. Убедитесь, что тип канала— *input*.

6. **Генерация сигнала.** Создайте группу генераторы. Для этого выделите строку *источники/приемники* навигатора проекта. Выделите контекстное меню. В появившемся меню выберите строку *создать группу*. Среди предложенных групп выберите *генераторы* (рис. 3.13). Выделив созданную группу (генераторы 1) и щелкнув левой клавишей мыши, измените имя, к примеру, на «генератор».

Выделите созданную группу. Вызовите контекстное меню. В контекстном меню выберите строку *создать компонент*. Среди предложенных генераторов выберите требуемый тип генератора (рис. 3.14).

Выделив созданный генератор и щелкнув левой клавишей мыши, измените имя генератора, к примеру, на «Пила».

7. **Привязка созданного генератора.** Щелкните левой клавишей

мыши по иконке . Откроется еще одно окно навигатора проекта. В левой части верхнего навигатора проекта выберите группу *каналы RTM* узла. В левой части нижнего навигатора проекта выберите группу *генераторы* группы *источники/приемники* (рис. 3.9). Нажмите левую клавишу мыши на созданном ранее генераторе. Не отпуская левой клавиши мыши, наведите курсор мыши на созданный канал и отпустите левую клавишу мыши. Иконка канала должна измениться на  (рис. 3.15).

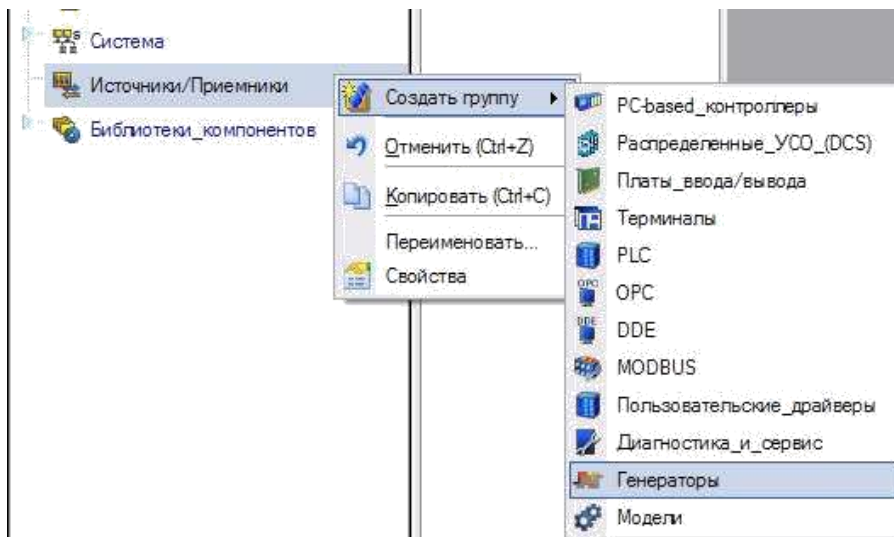


Рис. 3.13 Создание группы для генераторов

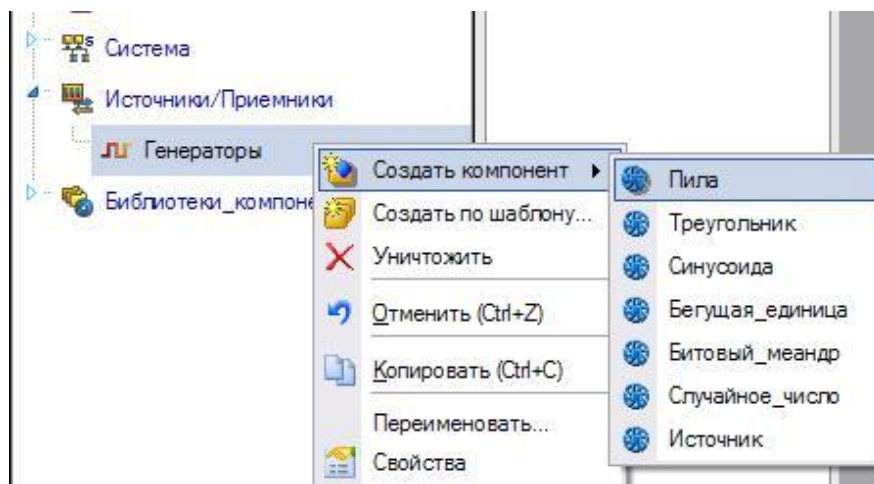


Рис. 3.14 Создание генератора

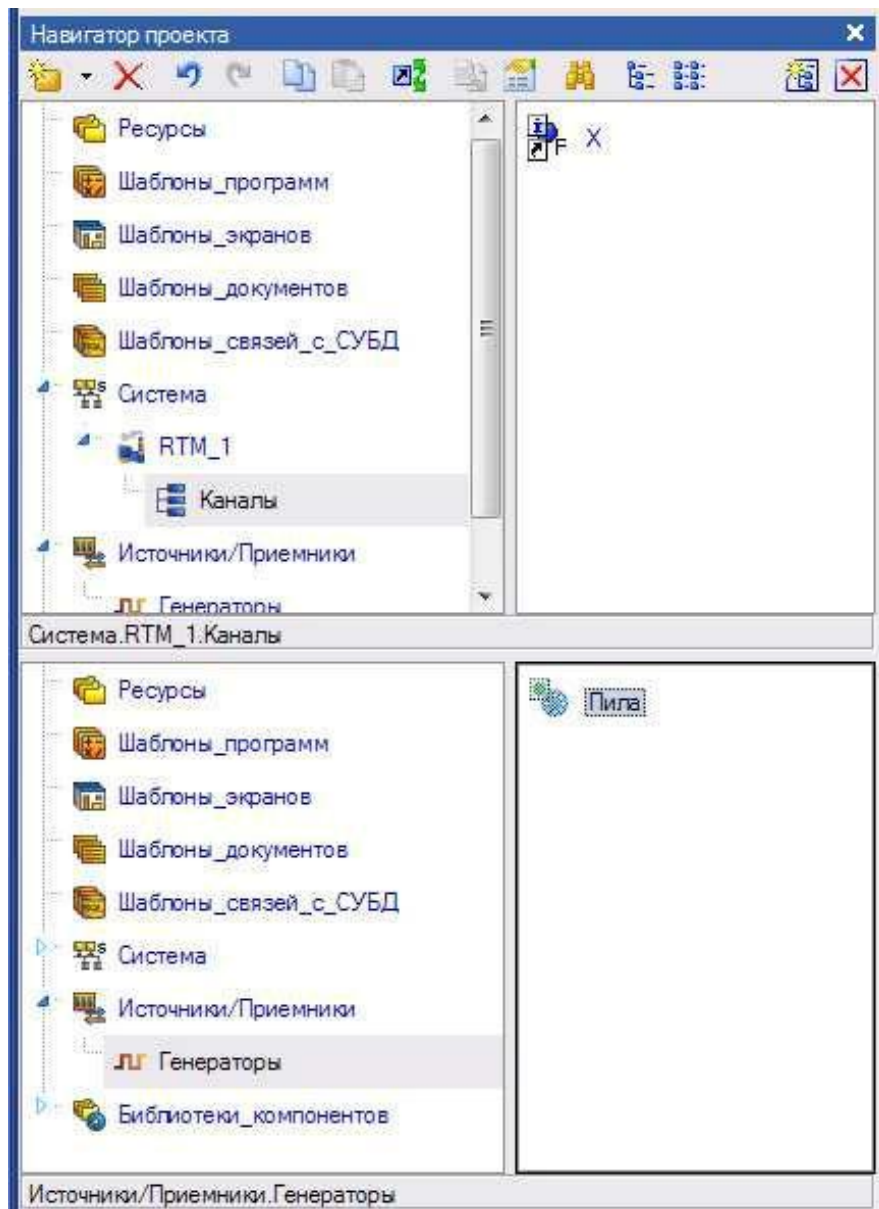



Рис. 3.15 Привязка генератора к каналу

Закрывать одно из созданных окон навигатора проекта можно щелкнув левой клавишей мыши по иконке .

8. **Создание экрана.** Выделите группу *каналы*, вызовите контекстное меню. В контекстном меню выберите строку *создать компонент*. Среди предложенных компонентов следует выбрать *экран* (рис. 3.16).

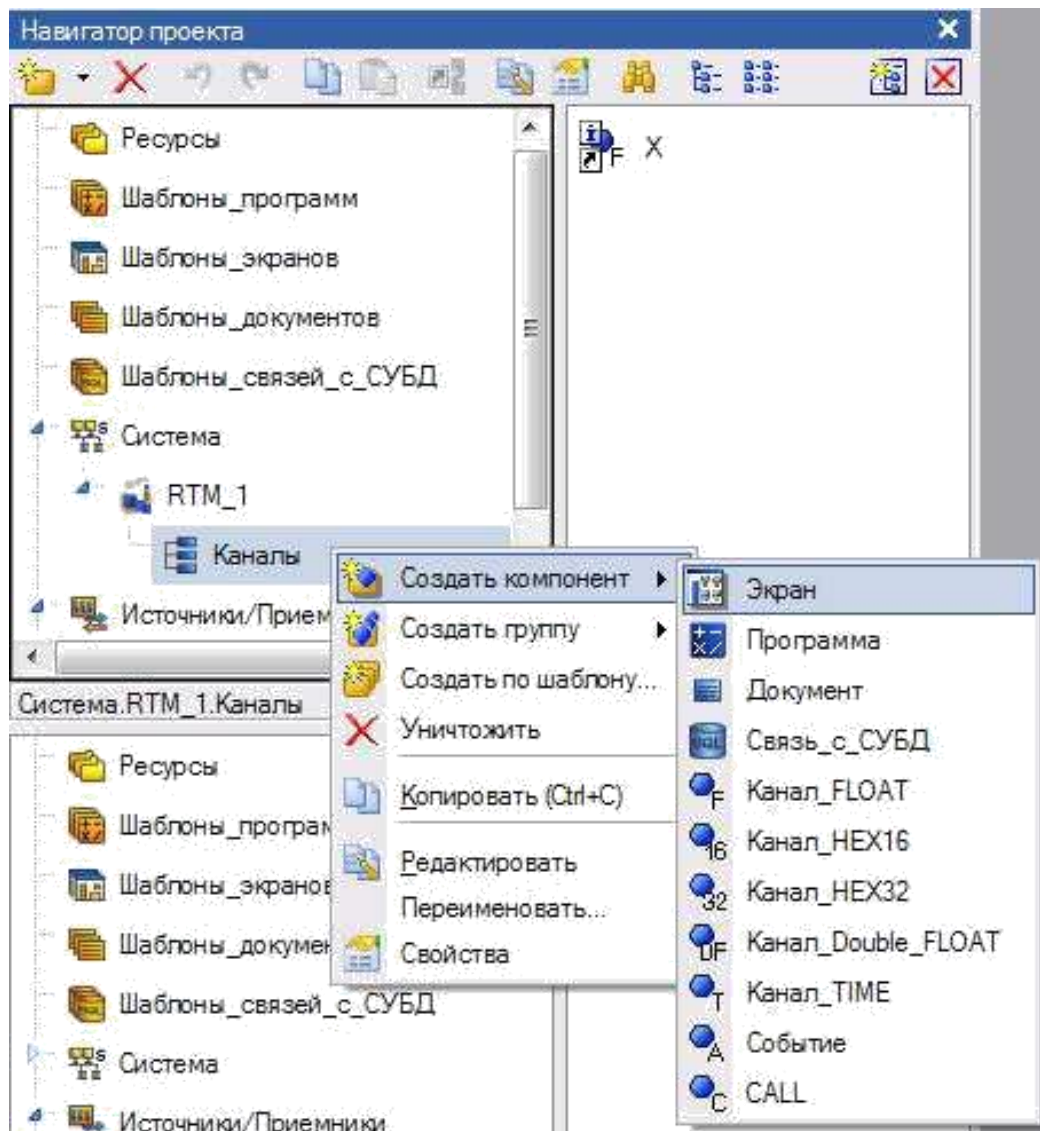


Рис. 3.16 Создание экрана

Выделите созданный экран и щелкните левой клавишей мыши, измените имя экрана. К примеру, можно изменить имя на «экран». Дважды щелкните левой клавишей мыши по созданному экрану. Откроется окно для редактирования шаблона экрана (рис. 3.17).

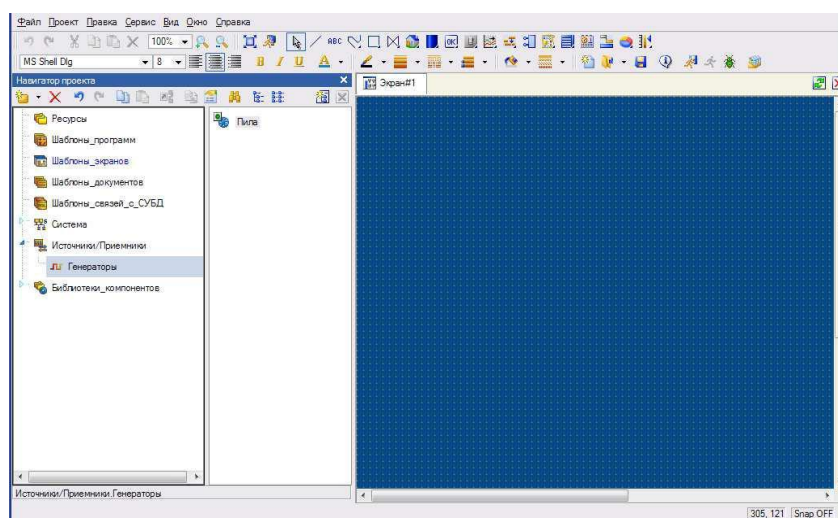







Рис. 3.17 Редактирование шаблона экрана

9. **Размещение объекта «стрелочный прибор».** Щелкните левой клавишей мыши по иконке . Если на панели инструментов вместо иконки  присутствует иконка , щелкните правой клавишей мыши по иконке . Среди предложенных объектов выберите стрелочный прибор .


После выбора  щелкните левой клавишей мыши в том месте экрана, где должен располагаться один из углов стрелочного прибора. Переместите курсор в положение, соответствующее положению противоположного угла стрелочного прибора и щелкните левой клавишей мыши. Будет создан стрелочный прибор на экране (рис. 3.18).



Рис. 3.18 Создание стрелочного прибора

Щелкните левой клавишей мыши по иконке .

10. **Настройка стрелочного прибора.** Если окно свойств объекта открыто выделите стрелочный прибор, в противном случае дважды щелкните левой клавишей мыши по созданному стрелочному прибору. Откроется окно свойств объекта (рис. 3.19). Раскройте раздел **заголовок** двойным щелчком левой клавиши мыши по подчеркнутой строчке **заголовок**. В появившемся поле **текст** введите текст заголовка, который будет выведен на стрелочном приборе. Раскройте раздел **полоса** двойным щелчком левой клавиши мыши по подчеркнутой строке **полоса**. В поле **верхний предел шкалы** введите верхнюю границу диапазона, в поле **нижний предел шкалы**— нижнюю границу диапазона. Заполните поля HL, HA, HW, LL, LA, LW любыми значениями, удовлетворяющими условию:

Нижний предел шкалы < LL < LA < LW < HW < HA < HL < Верхний предел шкалы

Пример заполнения полей свойств стрелочного прибора приведен на рис. 3.20.

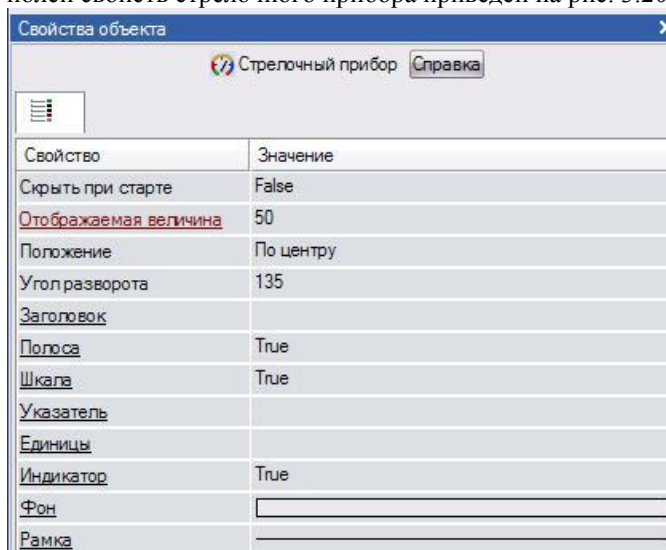


Рис. 3.19 Настройка стрелочного прибора

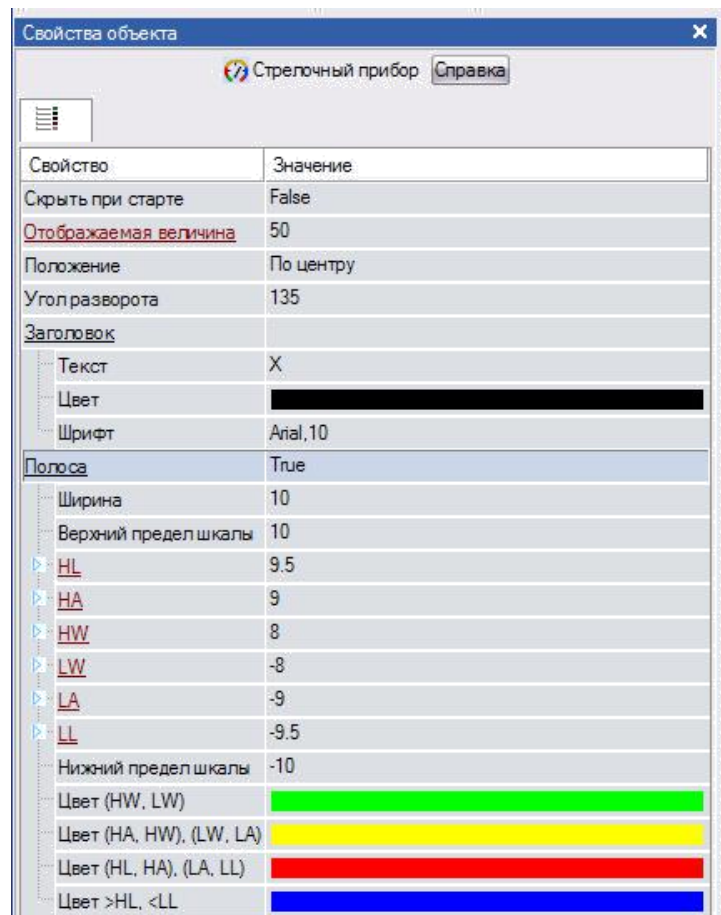


Рис. 3.20

11. **Привязка объекта к каналу.** Раскройте раздел *отображаемая величина*. Для этого дважды щелкните левой клавишей мыши на подчеркнутой строчке *отображаемая величина*. Щелкните левой клавишей мыши в поле *привязка*. Откроется окно свойств привязки с пустой таблицей.

Щелкните левой клавишей мыши на иконке . Установите тип IN у созданного аргумента ARG_000. Дважды щелкните левой клавишей мыши в столбце *привязка* таблицы. Откроется окно привязки. В левой части открытого окна выделите канал RTM узла, созданный ранее. В правой части окна выберите аргумент реальное значение (рис. 3.21). Щелкните левой клавишей мыши по кнопке *привязка*. Окно свойств привязки примет вид, изображенное на рис. 3.22. В поле *имя* можно ввести новое имя канала аргумента. Щелкните левой клавишей мыши по кнопке *готово*. В поле привязка будет выведено имя созданного аргумента.

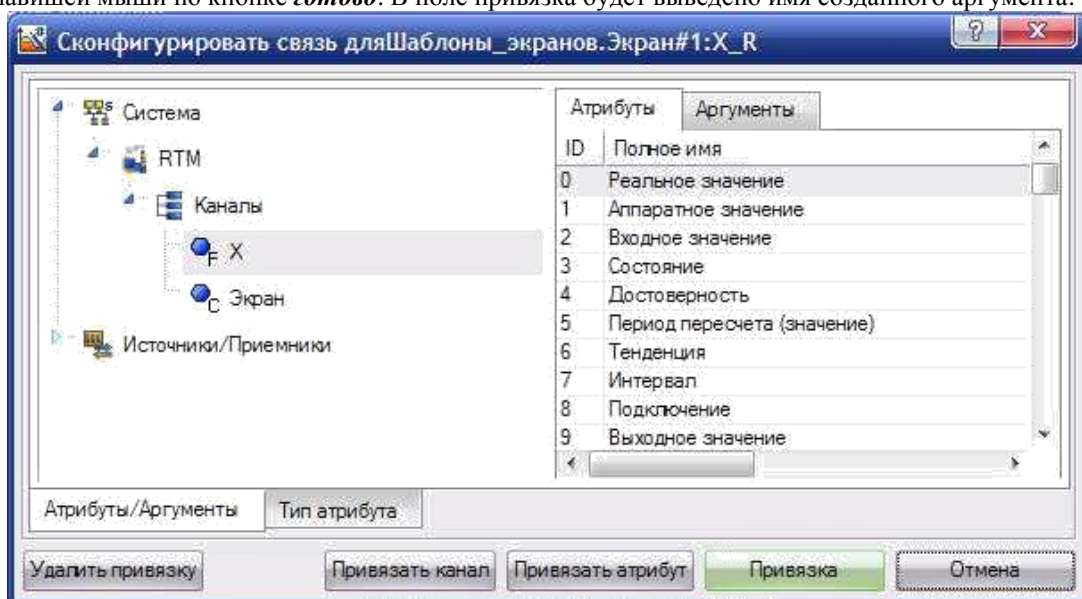


Рис. 3.21 Привязка

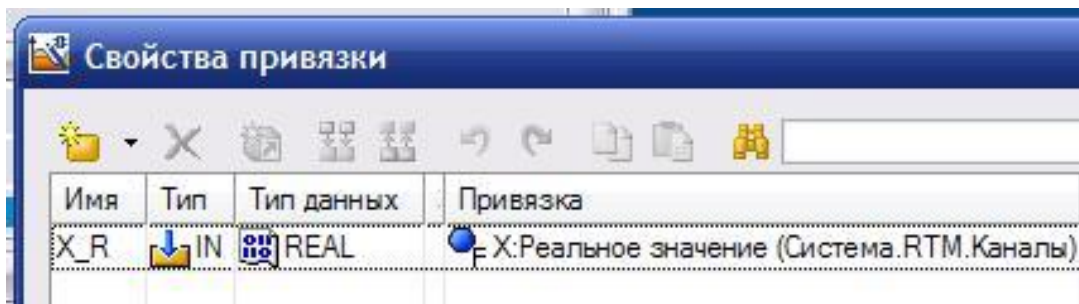


Рис. 3.22 Результат привязки

12. **Создание тренда.** Для создания тренда щелкните левой клавишей имени по иконке . Если вместо нее располагается на панели инструментов иконка архивный тренд , тренд XY или архивная гистограмма , то необходимо щелкнуть правой клавишей мыши на соответствующей иконке и выбрать тренд . щелкните левой клавишей мыши по иконке .

Для размещения тренда на экране щелкните левой клавишей мыши там, где должен располагаться один из углов тренда. Переведите курсор в положение, где должен располагаться противоположный угол тренда и щелкните левой клавишей мыши (рис. 3.23).

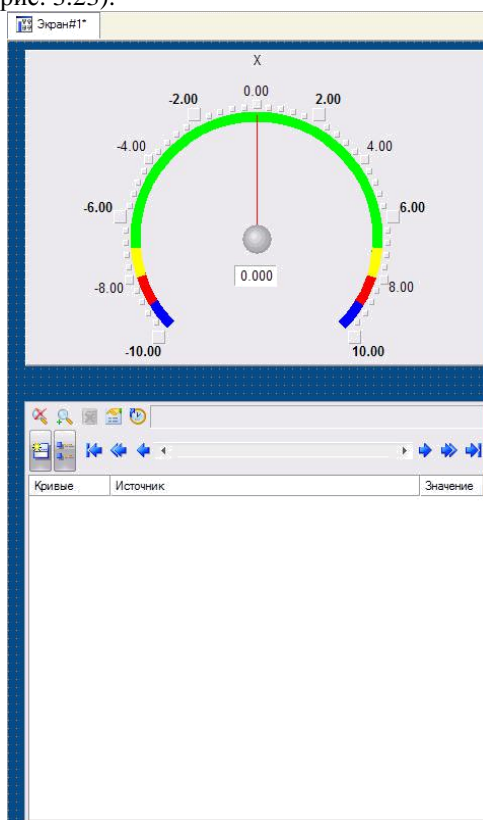


Рис. 3.23 Создание тренда

13. **Настройка тренда.** Выделите созданный тренд. Откроется окно свойств тренда. Изначально будет открыта закладка **основные свойства**. В поле **заголовок** введите текст, который вы хотите вывести в качестве заголовка. Перейдите на закладку **кривые**. Выделите строку **кривые** и вызовите контекстное меню. В появившемся контекстном меню выберите **кривая** (рис. 3.24).

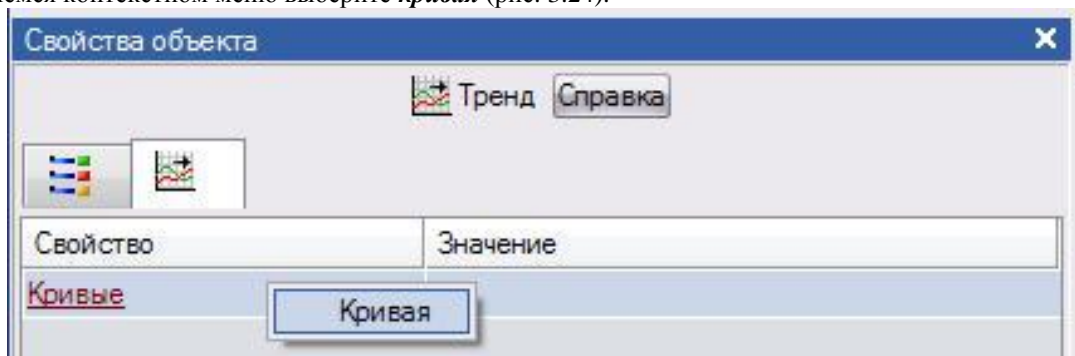


Рис. 3.24 Создание кривой

В появившемся поле **имя** введите имя созданной кривой. В поле **макс. значение** введите значение верхней границы диапазона, в поле **мин. значение**— нижней границы диапазона. Для привязки кривой щелкните левой клавишей мыши в поле **привязка**. Откроется окно, изображенное на рис. 3.22. Выберите созданный ранее аргумент и щелкните левой клавишей мыши по кнопке **готово**. Тренд примет внешний вид, изображенный на рис. 3.25.

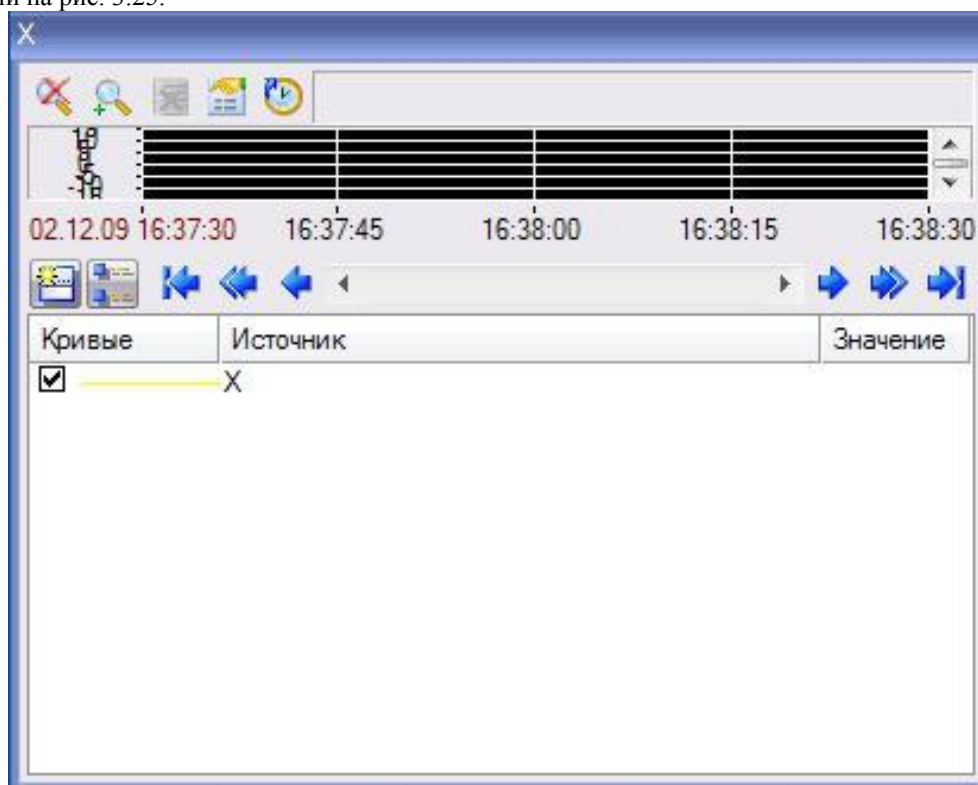


Рис. 3.25 Созданный тренд

Для увеличения размеров графика наведите курсор на границу легенды и графика. Курсор примет форму двух параллельных линий со стрелками. Нажмите левую клавишу мыши. Не отпуская левой клавиши мыши, опустите курсор вниз и отпустите левую клавишу мыши. Тренд примет вид, изображенный на рис. 3.26.

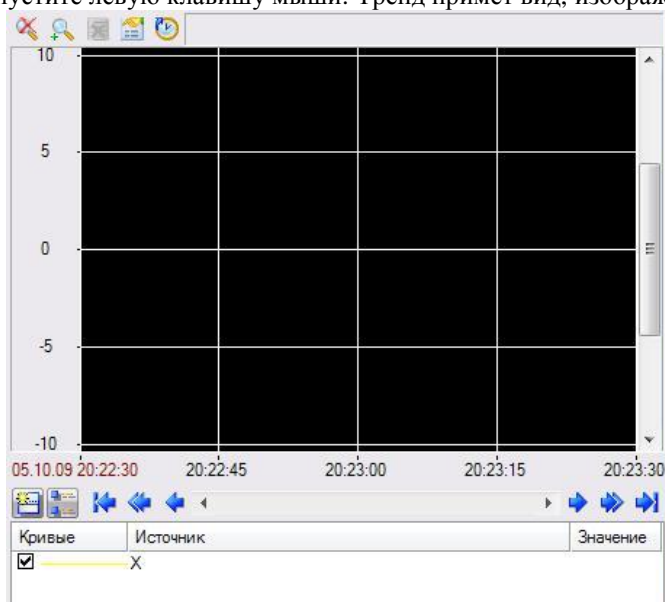



Рис. 3.26 Результат создания тренда

14. **Создание объекта текст.** Щелкните левой клавишей мыши на иконке **ABC**. Щелкните левой клавишей мыши там, где должен располагаться один из углов объекта текст. Переведите курсор мыши в положение, где должен располагаться другой край объекта текст и щелкните левой клавишей мыши. Рядом аналогично разместите еще один объект текст (рис. 3.27). Щелкните левой клавишей мыши по иконке .

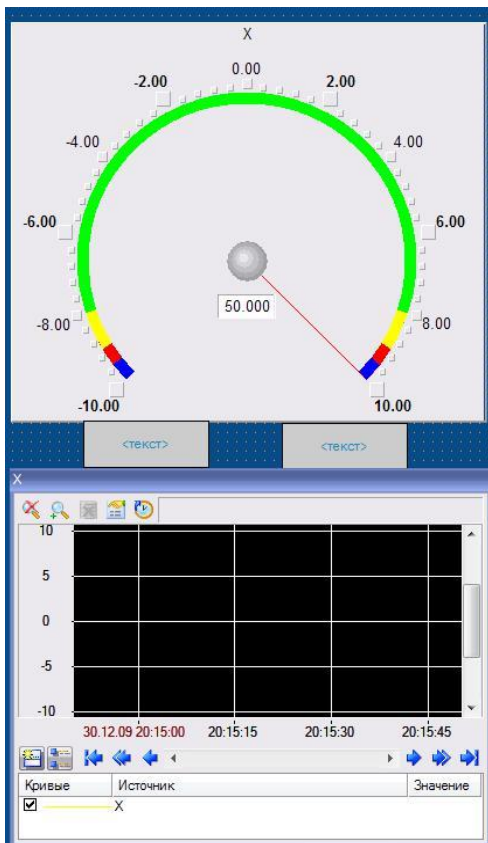


Рис. 3.27 Пример размещения объектов текст

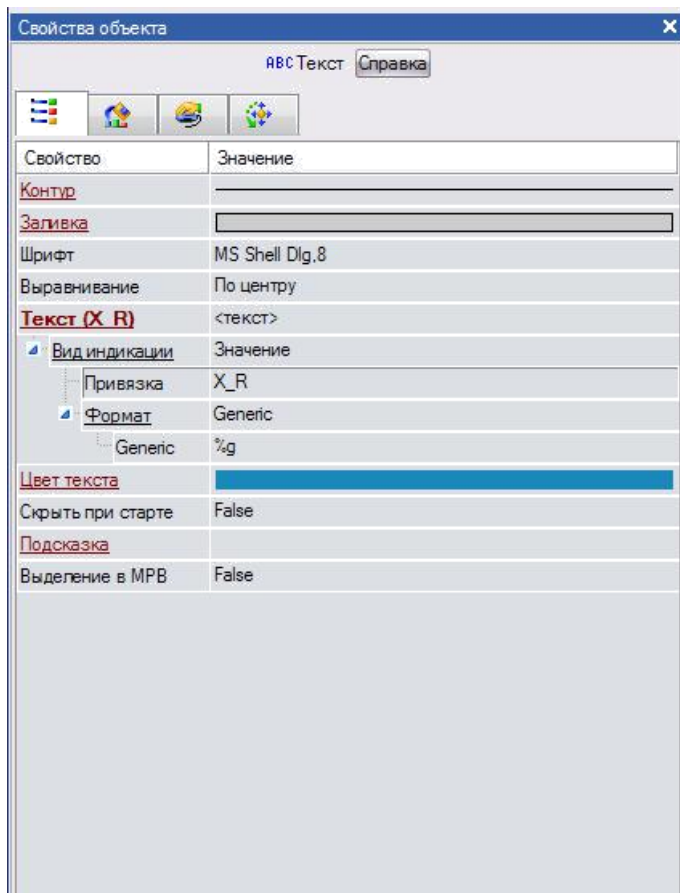


Рис. 3.28 Редактирование объекта текст

15. **Настройка объектов текст.** Выделите объект текст, расположенный слева. В поле *текст* введите имя канала RTM узла. Выделите объект текст, расположенный справа. Откройте раздел *текст* двойным щелчком по подчеркнутой строчке *текст*. В поле *вид динамизации* выберите *значение* (рис. 3.28). Щелкните левой клавишей мыши в появившемся поле *привязка*. Откроется окно, изображенное на рис. 3.22. Выберите созданный ранее аргумент и щелкните левой клавишей мыши по кнопке *готово*. Объекты текст примут вид, изображенный на рис. 3.29.



Рис. 3.29 Объекты текст

16. **Запуск проекта.** Щелкните левой клавишей мыши по иконке или по строчке *сохранить для MPB* в меню *файл*. В навигаторе проекта выделите созданный RTM узел. Щелчком левой клавиши мыши по иконке откройте профайлер. Если профайлер не запустился, запустите файл *rtc.exe*, щелкните левой клавишей мыши по иконке или строчке *открыть* в меню *файл*. Для запуска проекта щелкните левой клавишей мыши по иконке . В результате запуска профайлера стрелка стрелочного прибора должна перемещаться, тренд строить график, а объект текст выводить текущее значение созданного канала (рис. 3.30).

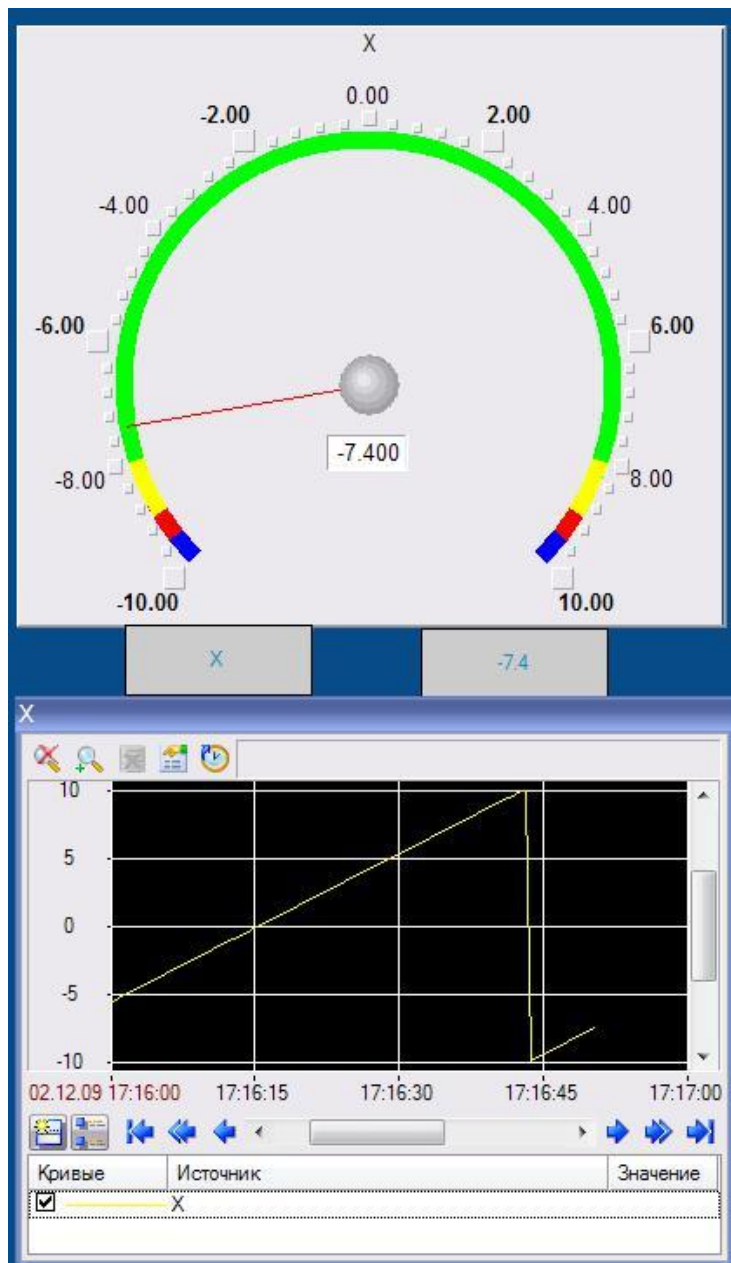


Рис. 3.30 Результат запуска проекта

Контрольные вопросы

1. Что такое SCADA-система.
2. Основные функции SCADA-систем.
3. Причины появления SCADA-систем.
4. Что такое АСУ ТП, их назначение?
5. Что такое узел?
6. Что такое канал?
7. Что такое база каналов?
8. Что такое объекты базы каналов?
9. Функции и назначение контроллеров нижнего уровня АСУ ТП.
10. Функции и назначение контроллеров верхнего уровня.
11. Что такое микроSCADA?
12. Опишите диспетчерский уровень АСУ ТП.

Лабораторная работа

Основная часть: Система Experion PKS R201 состоит из нескольких программ:

Станция Experion PKS.

Станция служит для организации рабочего места оператора-технолога, инженера и реализует Windows-ориентированную платформу пользователя с возможностями, позволяющими получить доступ к актуальным сведениям процесса и системы.

На станции отображается и выводится информация в виде стандартных и пользовательских дисплеев, таких как сводный дисплей сигнализации, дисплей трендов, мнемосхем процессов, различных рапортов.

В этой лабораторной работе Вы изучите:

Вход в систему Станции Experion, которая имеет Безопасность по Станциям Изменение уровня доступа на более высокий Выход из системы Станции или возвращение на уровень Оператора

Закрытие приложения Станции Experion

Основная Идея

Вход в систему с безопасностью по Станциям

При запуске Станции Вам автоматически назначается уровень безопасности Оператора OPER. Вы можете изменить на более высокий уровень доступа, в том случае если Вы знаете соответствующий пароль для конкретной Станции.

Объяснение

Вход в систему

Для входа в систему Станции Experion, которая имеет безопасность Входа в Систему по

Станциям, необходимо выполнить следующие шаги:

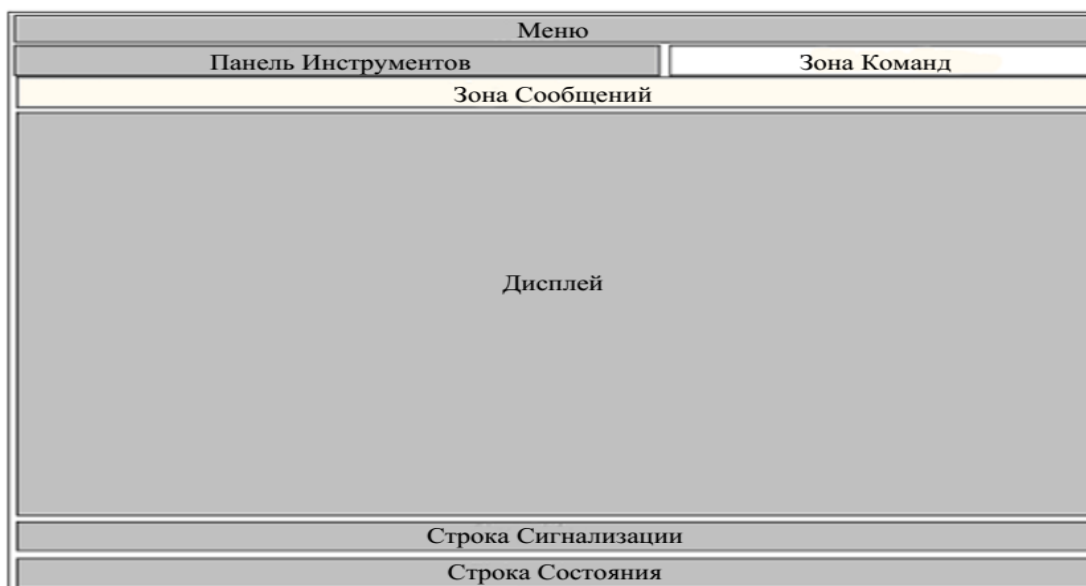
Выбрать в меню Station (Станция)

В меню выбрать Станцию, например, "StudentA"

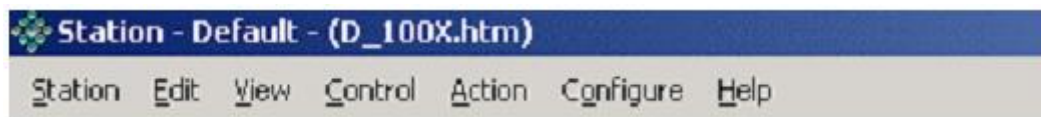
Нажать Connect (Установить Связь)

В этой лабораторной работе Вы познакомитесь со следующими инструментами навигации.

Окно Станции



Меню



Панель Инструментов



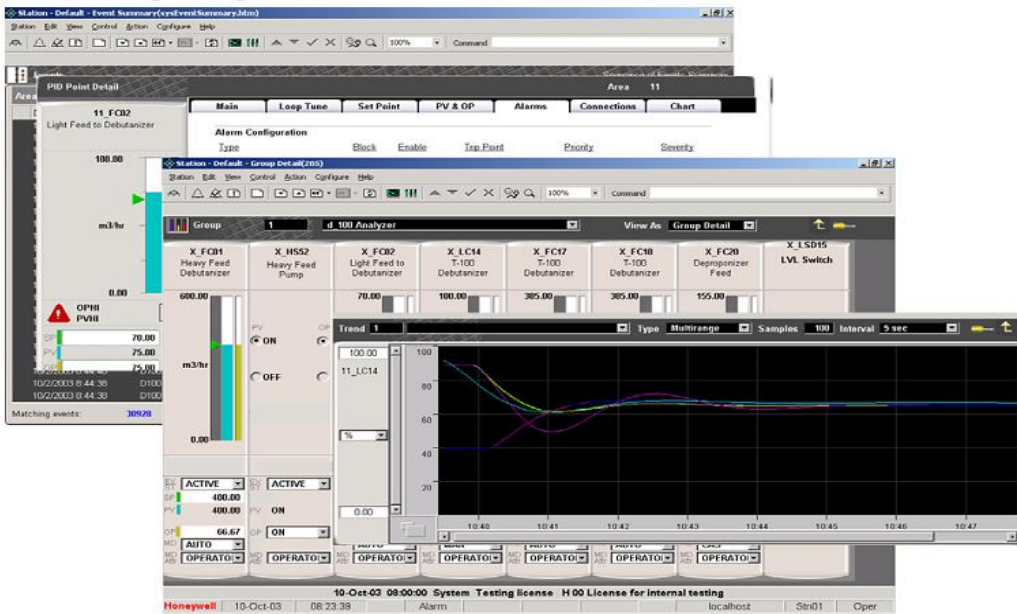
Функциональные клавиши

c<F1> по<F12>

ИКВ (Интегрированная клавиатура) – дополнительная опция



Просмотр Системных Дисплеев




Объекты Дисплея



Использование иконок(кнопок) Панели Инструментов для просмотра дисплеев и подтверждения сигнализации и или отключения звукового сопровождения.

ОБУЧЕНИЕ НА ПРАКТИКЕ		
✓	Шаг	Действие
	1	Нажмите кнопку System Menu для вызова дисплея Системного Меню.
	2	Нажмите кнопку Alarm Summary для вызова Обзорного дисплея Сигнализации.
	3	На Обзорном Дисплее Сигнализации, выделите точку, которая мигает. Используйте кнопку Acknowledge/Silence Alarm для того, чтобы подтвердить выбранную сигнализацию.
	4	Выберите другую точку на дисплее Сигнализации и вызовите ассоциированный дисплей этой точки, используя кнопку Associated Display .
	5	Выберите кнопку Request Page и введите имя дисплея: T_100 .



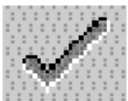

Использование иконок (кнопок) Панели Инструментов для перехода из одного дисплея в другой.

ОБУЧЕНИЕ НА ПРАКТИКЕ		
✓	Шаг	Действие
	1	Для вызова Группы #2, нажмите клавишу <F6> и введите число 2 и нажмите Enter, затем используйте  (кнопка Page Down) на панели инструментов для вызова дисплея следующей группы.
	2	Вернитесь на дисплей первой Группы, используя кнопку Page Up.
	3	Используйте кнопку Navigate Back для повторного вызова дисплея T_100.
	4	Чтобы обновить дисплей, выберите кнопку Reload Page.
	5	Кликните на стрелку вниз ▼ справа от кнопки Navigate Forward.
	6	В списке присутствует Alarm Summary? ДА НЕТ Выберите один из дисплеев в списке и повторно вызовете его на ваш экран.

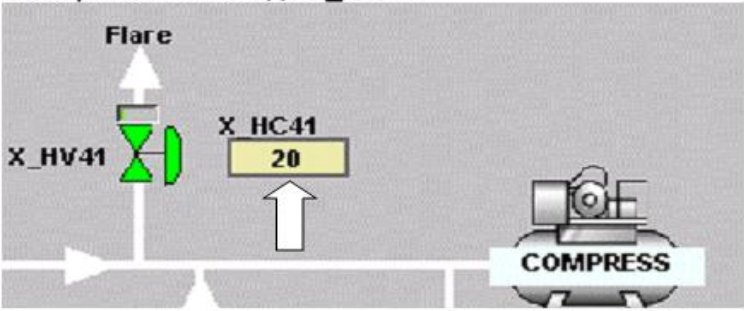
Группы и тренды конфигурируются по порядку и заносятся в нумерованный список, что позволяет использовать инструменты Page Up и Page Down для навигации последовательно в следующий дисплей.

ОБУЧЕНИЕ НА ПРАКТИКЕ		
✓	Шаг	Действие
	1	Нажмите кнопку Group, введите номер группы: 3 , и нажмите ENTER.
	2	Используйте кнопку Page Up для вызова Группы 2, затем вызовите Группы 3, 4 и 5, используя кнопку Page Down.
	3	Для вызова Тренда #3, нажмите кнопку Trend; введите номер тренда, и нажмите ENTER.


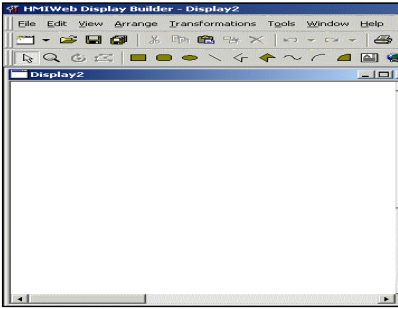
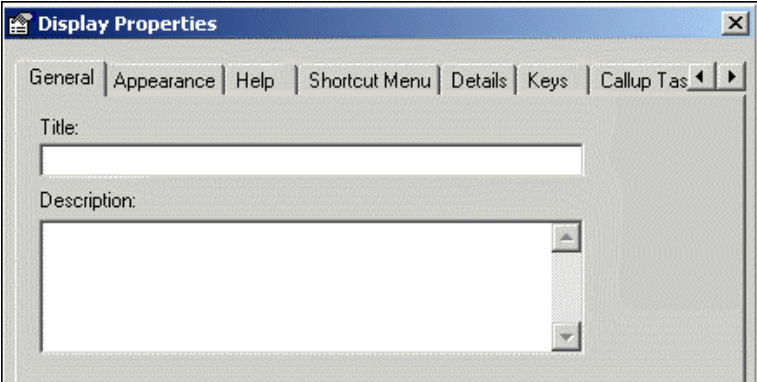
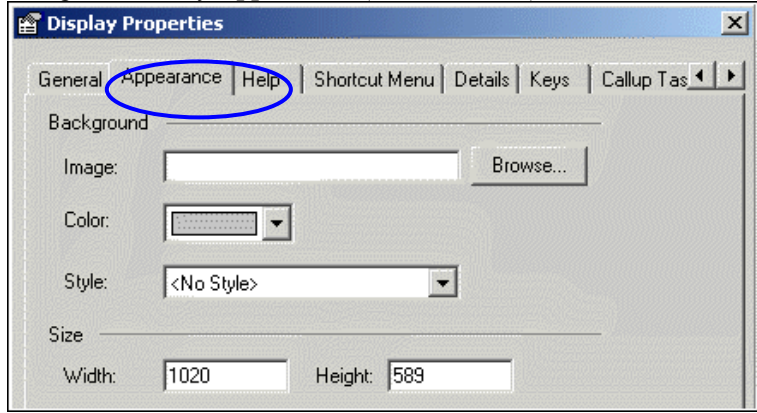
Следующие четыре кнопки Панели Инструментов используются для ввода данных при управлении точками.


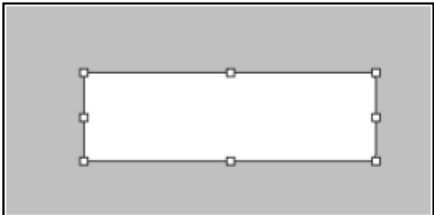
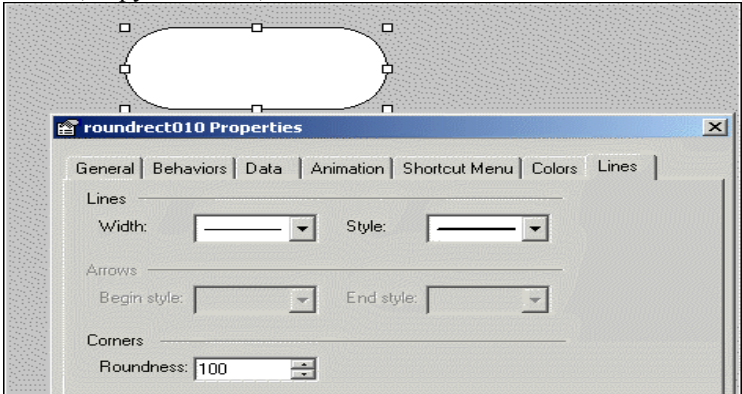
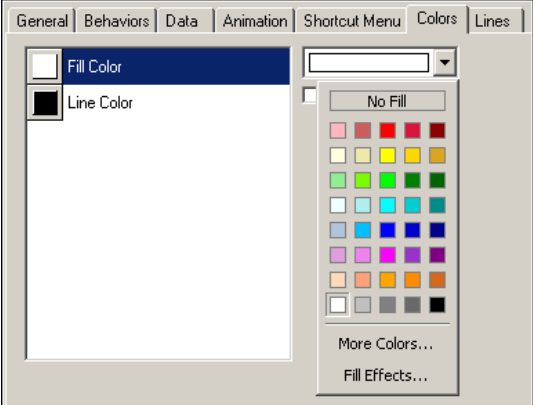

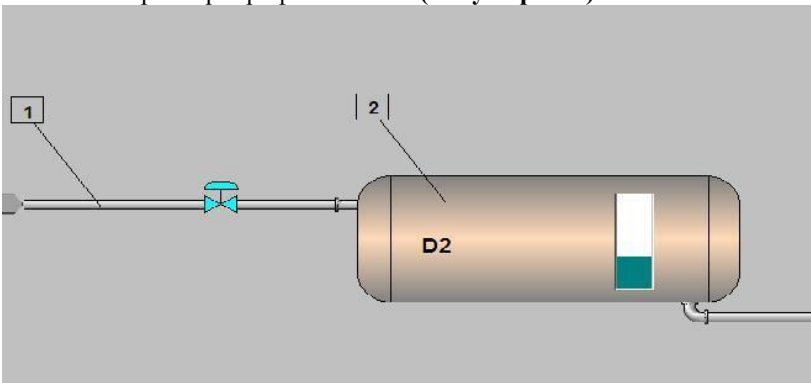
Кнопка	Описание
	Raise — Повышает значение параметра. Может использоваться для изменения Задания или Выхода.
	Lower — Понижает значение параметра. Может использоваться для изменения Задания или Выхода.
	OK — Принимает вновь введенные значения.
	Clear — Отменяет вновь введенные значения, и возвращает исходные.


Ввод некоторых данных и использование кнопок Панели Инструментов для ввода, подтверждения или очистки введенных данных.

ОБУЧЕНИЕ НА ПРАКТИКЕ		
✓	Шаг	Действие
	1	Нажмите на иконку Request Page для вызова дисплея D_100x (Не забудьте заменять x в соответствии с вашей буквой идентификации).
	2	<p>Выберите блок ввода x_HC41.</p>  <p>Используйте кнопку Raise для увеличения значения.</p>
	3	<p>Выберите блок ввода x_HC44 и понизьте значение, используя кнопку Lower.</p> <p>Эти кнопки могут также использоваться для изменения Задания (SP) и Выхода (OP).</p>
	4	Теперь выберите блок ввода x_HC41 , введите 50 , и нажмите Enter для завершения операции.
	5	Теперь выберите блок ввода x_HC44 , введите 80 , и нажмите кнопку Clear для отмены операции.

**Лабораторная работа .
Построение статических объектов дисплея**

✓	Шаг	Действие
	1	<p>Откройте новый дисплей в HMI Web Display Builder:</p> <p>Нажмите кнопку  New Display (Новый Дисплей).</p> <ul style="list-style-type: none"> Откроется пустой дисплей, как показано ниже.  <p>Примечание: Каждому новому дисплею присваивается, по умолчанию, имя (Display#).</p>
	2	<p>Установка цвета фона и размера дисплея:</p> <p>Дважды кликните на дисплей, для вызова Properties Window (Окно Свойств) как показано ниже.</p>  <p>Выберите вкладку Appearance (Внешний Вид).</p> 
	3	<p>Установите размер дисплея 1276 x 894 (данный размер выбран как основной в конфигурации. Измените цвет фона по умолчанию на цвет silver (серебряный). Оставьте поля image (отображение) и style (стиль) пустыми. Закройте Properties Window (Окно Свойств).</p>

4	<p>Выберите Rounded Rectangle (Скругленный Прямоугольник) на панели инструментов . Перетащите курсор диагонально по дисплею, до нужного размера объекта, который вы создаете.</p> 
5	<p>Откройте Properties window (Окно свойств) для объекта rounded rectangle (скругленного прямоугольника). Выберите вкладку Lines (Линии). Измените roundness (закругленность) на 100.</p>  <p>Выберите вкладку Colors (Цвета). Нажмите на стрелку выпадающего меню для изменения Fill Effects (Эффекты заливки).</p> 
6	<p>Создание трубопровода, клапана и емкост D2:</p>  <p>Выберите кнопку Textbox (Текст) на панели инструментов. Начертите текстовый объект в середине прямоугольника, как показано ниже. Замените слово "Text" на слово D2. Установите размер шрифта 8 и bold (полужирный).</p>  <p>Сохраните ваш дисплей под именем Lab1.htm в папке "abstract", (путь: C:\Program Files\Honeywell\Experion PKS\Client\Abstract)</p>

7	<p>Выберите Edit > Insert shape, и выберите шайпы LoopLabPid.sha, KlapLab.sha и IndiLab.sha и вставьте их на дисплей.</p> <p>Ваш дисплей должен выглядеть примерно так, как показано ниже.</p> 
8	

1. Введение в Control Builder.

- Control Builder – средство инжиниринга системы Experion PKS.
- Control Builder можно запускать на Сервере или на Инженерной Рабочей Станции. Разрешается подключать к Серверу Experion PKS одновременно максимум четыре клиента Control Builder.
- Устройства, так же как стратегии управления, создаются в Control Builder и загружаются в Контроллер C200 и Сервер Experion PKS.
- Для входа в Control Builder требуется пользователь:
 - Профиль входа в систему, который использует Control Builder, - один из профилей Оператора, который использует Сервер Experion PKS.
- Предъявление пароля пользователем содержит уровень доступа пользователя, например, LVL1, LVL2, OPER, SUPV, ENGR или MNGR.
 - LVL1 и LVL2 → только Просмотр
 - OPER и SUPV → Позволяет производить операции
 - ENGR и MNGR → Позволяет конфигурировать
- Профиль Оператора Experion PKS может использовать идентификатор и пароль учетной записи Windows (Интегрированные Учетные Записи).

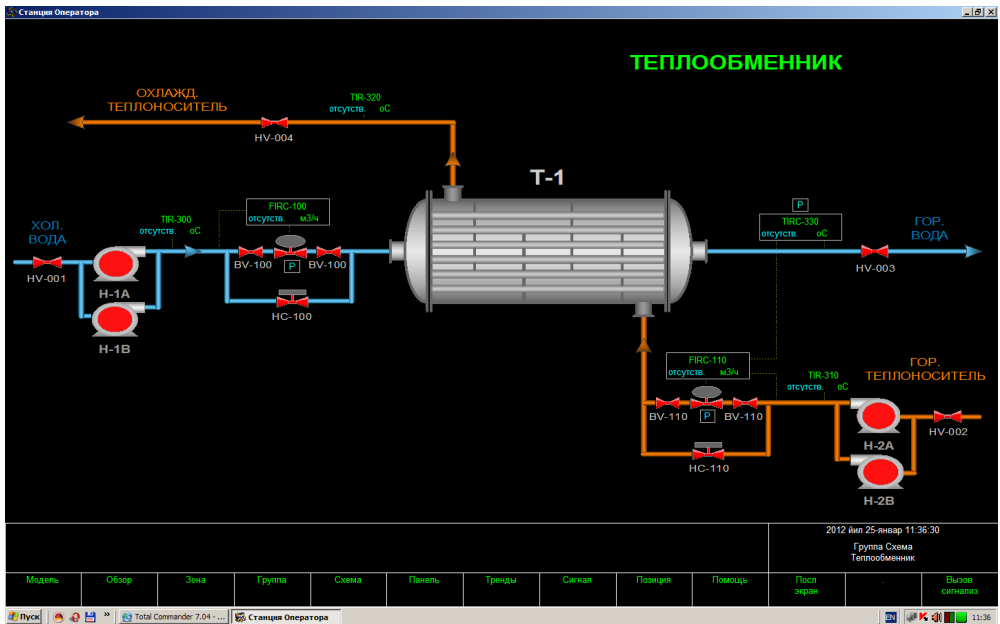
Цель задачи: Принцип работ тренажерного комплекса.

Основная часть: Внедрение современной АСУТП не может быть полноценным без организации тренажерной подготовки эксплуатационного персонала. При этом наибольший эффект по обучению и сокращению затрат на освоение персоналом новой техники может быть достигнут, если тренажер будет введен в действие ещё до начала внедрения реальной АСУТП на объекте.

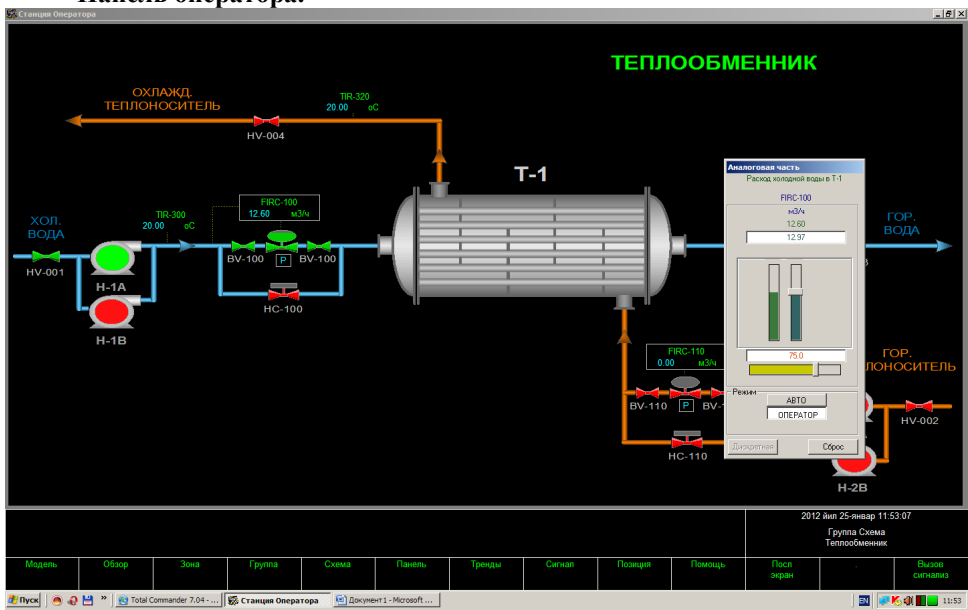
Компьютерный тренажер состоит из следующих составных частей: математической модели объекта управления (например, энергоблока), модели системы управления и специального программного обеспечения инструктора, обеспечивающего организацию тренировочного процесса.

Совершенно очевидно, что оптимальным решением для современных тренажеров является отказ от создания имитационной модели системы управления и замена её реальной АСУТП с использованием как технических средств, так и программного обеспечения, полностью соответствующих АСУТП энергоблока-прототипа.

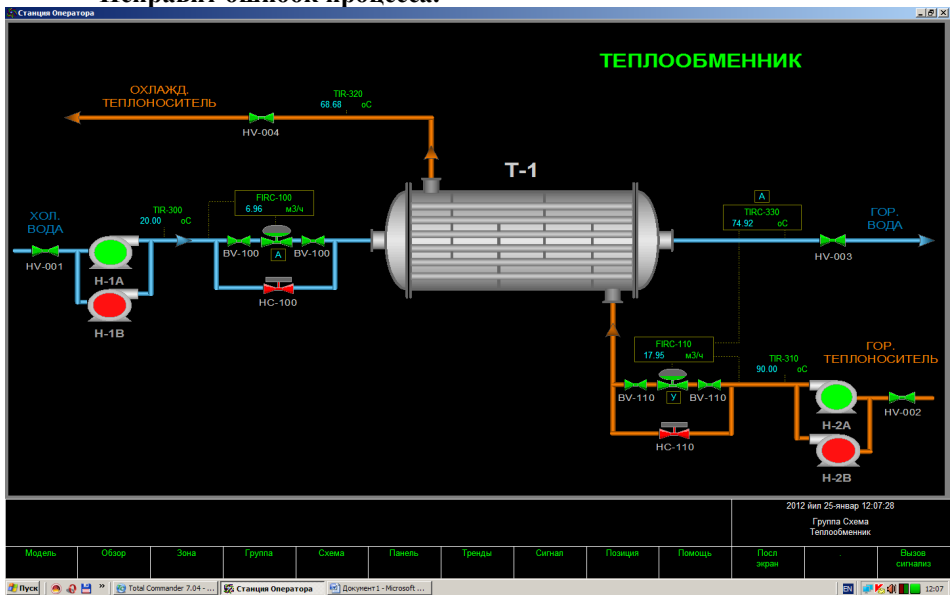
КТК-М тренажерного комплекса состоит из двух частей экрана 1 экран оператора 2 экран инструктору. Для проверки теплообменного процесса показано на рисунку.



Панель оператора.



Исправит ошибок процесса.



Процесс в норме

Лабораторная работа. Разработка программы управления

1. Цель работы

Изучить особенности разработки полнофункциональной программы для управления объектом автоматизации.

2. Краткие теоретические сведения

Программирование с использованием символов.

В таблице символов назначаются символические имена и типы данных всем абсолютным адресам и переменным (меркерам), к которым можно позднее обращаться в программе. Используя символическое представление, облегчается задача создания и обеспечивается читаемость написанной программы. Имена задаются в компоненте Symbols[Символы] в окне проекта SIMATIC Manager. Для программы S7 создается только одна таблица символов, независимо от того, какой язык программирования вы выбрали.

Структура программы.

Любое управление состоит из частных функций управления частями машины или процесса. Этого можно добиться путем соответствующего разделения программы управления на отдельные части (функции или блоки).

Функциональный блок (FB*) содержит часть программы, которая может многократно вызываться OB1. Все формальные параметры и статические данные функционального блока сохраняются в отдельном блоке данных (DB*), назначаемом функциональному блоку. Для создания функционального блока необходимо выбрать в контекстном меню в папке Blocks [Блоки] команду **Insert New Object>Function Blocks [Вставить новый объект> Функциональный блок]**. Затем открыть FB1 и в диалоговом окне "**Properties –Function Block [Свойства – Функциональный блок]**" выбрать язык, на котором надо создавать этот блок, активизировать триггерную кнопку "Multiple instance FB [Мультиэкземплярный FB]" и подтвердить остальные параметры настройки, щелкнув на ОК. Все сигналы, специфические для блока, передаются функциональному блоку из организационного блока как параметры блока и должны быть перечислены в таблице описания переменных как входные и выходные параметры (описание "in" и "out").

Чтобы в будущем получить возможность вызова функционального блока в OB1, необходимо сгенерировать соответствующий блок данных. Экземплярный блок данных (DB) всегда ставится в соответствие функциональному блоку. Возможно также давать различным блокам символические имена в таблице символов.

Централизованно программируя функциональный блок один раз, вы можете сократить объем программирования. Для этого надо создать блок данных DB в папке **Блоки** и принять все параметры, отображаемые в диалоговом окне **Properties**. Для каждого вызова функционального блока используется свой блок данных.

Создавая структуры программ с организационными блоками, функциональными блоками и блоками данных, необходимо программировать вызов для подчиненных блоков (таких, как FB1) в блоке, расположенном в иерархии более высоко (например, OB1).

Тестирование программы с помощью функции Статус.

Используя функцию "Статус программы", можно тестировать программу в блоке. Предпосылкой для этого является установление соединения online с CPU, CPU должен находиться в режиме RUN или RUN-P, а программа должна быть загружена. Необходимо открыть OB1 в окне проекта "Your program ONLINE." Откроется окно для программирования LAD/STL/FBD. Затем активизировать функцию **Debug >Monitor [Отладка > Наблюдение]**.

В контактном плане отображается последовательная цепь. Путь тока представляется сплошной линией, это значит, что в цепь уже подано питание.

В случае функционального плана состояние сигнала показывается с помощью "0" и "1." Пунктирная линия показывает, что результат логической операции отсутствует. В графических языках программирования Контактный план и Функциональный план можно проследить результаты тестирования, просматривая изменение цвета в сегментах программы. Это изменение цвета показывает, что результат логической операции выполнен до этой точки.

Для списка операторов в табличной форме отображается следующее:

- результат логической операции (RLO)
- бит состояния (STA)
- стандартное состояние (STANDARD)

При этом отображение в столбцах STA и RLO меняется, когда результат логической операции выполняется.

Наблюдение за переменными.

Если предыдущий способ отладки почему-либо неприемлем, то можно щелкнуть на кнопке **Monitor Variables [Наблюдать переменные]** на панели инструментов. В строке состояния отобразится режим работы CPU.

Анализ диагностического буфера.

Если CPU переходит в STOP при обработке программы S7, или невозможно переключить CPU в RUN после загрузки программы, то можно определить причину ошибки из событий, перечисленных в диагностическом буфере. Предпосылкой для этого является установление связи online с CPU и нахождение CPU в состоянии STOP.

Чтобы запросить информацию о модуле используется команда меню **PLC > Module Information [ПЛК > Информация о модуле]**. Окно "**Module Information**" предоставляет информацию о свойствах и параметрах CPU. Теперь можно выбрать вкладку "**Diagnostic Buffer [Диагностический буфер]**", чтобы определить причину перехода в состояние STOP. Самое последнее событие (номер 1) находится наверху списка. Отображается причина перехода в состояние STOP. Если переход CPU в состояние STOP вызвала ошибка программирования, необходимо выбрать событие и щелкнуть на кнопке "Open Block [Открыть блок]". В окне для программирования LAD/STL/FBD открывается блок, и сегмент, содержащий ошибку, выделяется подсветкой.

3. Методические указания

- 3.1. Для решения задачи необходим отдельный проект.
- 3.2. До начала процесса написания кода программы необходимо сопоставить символьные имена каждой переменной с ее абсолютным адресом. Это делается в окне Symbol Editor.
- 3.3. Программу необходимо разделить на функциональные блоки. Порядок их вызова определяется в организационном блоке.
- 3.4. Для отладки задач их необходимо загружать симулятор S7-PLCSIM. При отладке можно использовать пошаговый режим (если выполняется одиночный цикл управляющей программы), или приостановку выполнения программы в режиме множественного цикла.

4. Порядок выполнения работы

- 4.1. Заполнить таблицу символов именами объектов и переменных.
- 4.2. Разработать и ввести коды функциональных блоков.
- 4.3. Ввести код блока OB1, содержащего последовательный вызов всех функциональных блоков.
- 4.4. Отладить работу программы, показать преподавателю, сделать скриншот решения (скопировать листинг программы).

5. Варианты заданий

- 5.1. Светофор. Красный сигнал горит 30 с, последние 2 с одновременно горит желтый. Когда красный и желтый сигналы гаснут, то зажигается зеленый сигнал на 15 с, причем последние 2 с он мигает с частотой 2 Гц. Зеленый сигнал гаснет, загорается на 2 с желтый. Цикл повторяется.
- 5.2. Стиральная машина. При включении питания загорается индикатор питания. При нажатии кнопки «старт» запускается таймер стирки (15 с), загорается индикатор стирки, по истечении таймера - гаснет. Далее запускается таймер полоскания (10 с) и загорается индикатор, по истечении таймера - гаснет. Далее запускается таймер отжима (5 с) и загорается индикатор, по истечении таймера – гаснет, начинает мигать индикатор питания и отжимается кнопка «старт». При отключении питания гаснет индикатор питания.
- 5.3. Лифт. Если лифт занят, и этаж задан, он движется на заданный этаж, если этаж не задан, то стоит. Если лифт свободен и имеется вызовы с этажей, то лифт движется к ближайшему этажу. Если вызова с этажей нет, то лифт стоит. Положение лифта отображается лампами текущего этажа.
- 5.4. Конвейер (повышенной сложности). Подающий транспортер должен перемещать изготовленные на трех рабочих местах (РМ) «монтажные элементы» в пункт конечного монтажа. При изготовлении элементов на рабочих местах лампы Н2 ... Н4 должны гореть. Для сообщения об окончании их изготовления используются кнопки S2 ... S4, при этом гаснут соответствующие лампы (Н2 ... Н4). После укладки всех трех монтажных элементов на транспортер (нажатия всех кнопок S2 ... S4) включается мотор транспортера (К1) на время, необходимое для перемещения элемента с рабочего места 3 до позиции конечного монтажа (например, 5 с). Пока транспортер находится в движении, должна мигать лампа Н1 на месте конечного монтажа. При остановке транспортера, лампа Н1 должна гореть непрерывно. После освобождения приемного накопителя, что квитируется нажатием кнопки S1, лампа Н1 должна погаснуть, а лампы Н2 ... Н4 вновь включиться. Управление включается с помощью кнопки S5 (нормально разомкнут) и выключается кнопкой S6 (нормально разомкнут). В состоянии «ВКЛ» должна гореть лампа Н5.

6. Содержание отчета

- 6.1. Цель работы.
- 6.2. Коды всех блоков.
- 6.3. Скриншот решения задач.

7. Контрольные вопросы

- 7.1. Адресация переменных в STEP 7.
- 7.2. Типы данных в STEP 7.
- 7.3. Структура STL-выражений.
- 7.4. Ключевые слова для программирования кодовых блоков на STL.
- 7.5. Какие средства проверки значений переменных имеются в STEP 7?
- 7.6. Какие операторы STL и для чего применены в Ваших программах?
- 7.7. Чем функциональные блоки программы отличаются от остальных?
- 7.8. Как определить причину остановки CPU?

Лабораторная работа №18. Разработка мнемосхемы технологического процесса

1. Цель работы

Изучить общие принципы построения систем визуализации, смены изображений в главном окне, организацию взаимодействия элементов мнемосхем с тегами процесса и оператором.

2. Краткие теоретические сведения

Традиционно в мнемосхемах окно разбивается на три части:

- обзорная область (10%, видно всегда), в которой расположены окно сообщений, кнопки навигации по мнемосхеме, обобщенные индикаторы состояния, системное время, текущий пользователь, состояние связи с контроллерами,
- рабочая область (80%, может меняться), в которой расположены мнемосхемы, экраны диагностики, графики и таблицы,
- область дополнительной клавиатуры (10%, видимость зависит от рабочей области), в которой содержатся кнопки вызова диагностики, сообщений, графиков, выхода из среды исполнения и др.

Для реализации этого подхода создадим кадр (main.pdl) размером 1024x768, в него вставим 3 объекта Picture Window размерами 1024x100, 1024x600, 1024x68. Для подгонки отображаемых кадров под размер объекта в свойствах Picture Window установим Adopt Picture в Yes. В свойство Picture Name верхнего объекта введем "menu.pdl", среднего - имя кадра с основной мнемосхемой.

Создадим новый кадр menu.pdl размером 1024x100. В нем вставим кнопки "Мнемосхема", "Графики". По их нажатию будут выполняться сценарии, изменяющие свойство Picture Name объекта Picture Window2 кадра main.pdl на имя кадра с мнемосхемой (1 кнопка) и кадр graphics.pdl (2 кнопка).

Для организации взаимодействия элементов мнемосхем в тегами процесса и оператором в свойствах этих элементов есть опции Dynamic Dialog, Direct Connection, C-Action, а также возможность отслеживать события. В свойствах всех объектов есть закладка "Events", в которой возможно отслеживание пяти событий мыши.

Возможно использовать C-макросы для программирования реакции на событие, произошедшее с графическим объектом. C-макрос следует использовать, если возможности, предоставляемые прямым соединением с тегом, не достаточны для решения вашей задачи.

Использование макросов для определения реакции на изменение свойства объекта влияет на производительность системы исполнения.

Событие происходит, если изменяется значение свойства объекта. В этом случае, макрос, связанный с этим событием, начинает выполняться. Когда кадр закрывается, выполнение всех макросов по очереди завершается. Это может привести к большой нагрузке на систему.

Кроме использования прямого соединения с тегом, C-макросов и тегов, в WinCC можно задавать динамику графических объектов в системе исполнения с помощью VBS-макросов. VBS-макросы следует использовать в следующих случаях

- если вы хотите в макросе обработать несколько входных параметров,
- если вы хотите использовать условную конструкцию (if ... then ...),
- если вы хотите изменить в макросе несколько свойств объекта,
- если вы хотите обратиться к диалоговым окнам операционной системы, как, например, диалоговому окну выбора файла или цвета

VBS-макросы создаются в редакторе VBS-макросов в Graphics Designer [Графическом дизайнера]. Редактор макросов предлагает такой же набор функций, как и редактор VBS в "Global Script [Глобальный сценарий]". Макросы, созданные в Graphics Designer [Графическом дизайнера] всегда сохраняются вместе в кадре, в котором они были созданы. Документация по созданным VBS-макросам добавляется в проектную документацию Graphics Designer [Графического дизайнера] вместе со свойствами всех сконфигурированных объектов. Все VBS-макросы, созданные в кадре, можно посмотреть в WinCC Explorer [Проводнике WinCC] в диалоговом окне Properties [Свойства]. Это диалоговое окно вызывается из контекстного меню для этого кадра.

VBS-макросы могут использоваться для динамизации свойств объекта. Вы можете определить динамику для свойства объекта в системе исполнения в зависимости от триггера, тега или состояния другого объекта. VBS-макрос следует использовать, если возможности динамизации, предоставляемые соединением с тегом или динамическим диалогом, не достаточны для решения вашей задачи.

Если к WinCC не подключен ПЛК, для тестирования проекта можно использовать имитатор. Чтобы запустить имитатор, перейдите на панель задач Windows и щелкните на пункте меню "Start [Пуск]" "SIMATIC" "WinCC" "Tools [Инструменты]" "WinCC Tag Simulator [Имитатор тегов WinCC]". Для того чтобы имитатор функционировал правильно, проект должен быть активизирован (в режиме исполнения). В диалоговом окне имитатора выберите тег, который вы хотите смоделировать. Для этого щелкните на "Edit [Правка]" "New Tag [Новый тег]". В диалоговом окне "Tags – Project [Теги - Проект]" выберите внутренний тег "Position" и щелкните на кнопке "OK". На закладке "Properties [Свойства]" выберите режим имитации.

Введите начальное и конечное значения. Установите флажок "active [активный]". На закладке "Tags [Теги]" нажмите на кнопку "Start Simulation [Начать моделирование]". В таблице "Tags [Теги]" будут

отображаться изменяющиеся значения выбранного тега. Перейдя к окну режима исполнения, вы сможете увидеть, как имитатор поставляет "реальные" значения кадру.

3. Методические указания

- 3.1. Для динамизации свойств объекта выбирайте наиболее адекватный инструмент: Dynamic dialog, Direct Connection, C-Action.
- 3.2. При использовании Dynamic dialog выбирайте верный формат выходных данных "Analog", "Boolean", и "Direct".
- 3.3. При использовании Dynamic dialog не забывайте настраивать триггеры.
- 3.4. При написании сценариев используйте встроенные функции, список которых отображается в левой части редактора сценариев.
- 3.5. Для отладки и диагностики ошибок сценариев можно использовать элемент Application Window, задав свойства Global Script/GSC Diagnostics, в который можно печатать из макроса: printf("Something wrong!\n").
- 3.6. Для вывода сообщений о работе сценариев можно использовать стандартную функцию MessageBox.

4. Порядок выполнения работы

- 4.1. Определить входные, внутренние и выходные параметры объекта автоматизации.
- 4.2. Создать внутренние (не связанные с контроллером) теги соответствующих типов.
- 4.3. Разработать структуру не менее чем из 5х кадров мнемосхем, 2 из которых (для вывода аварийных сообщений и значений технологических параметров) будут заполнены позднее. Организовать переходы между кадрами.
- 4.4. Используя библиотеку элементов НМІ заполнить основной кадр мнемосхемы элементами, отображающими как сам объект автоматизации, так и элементы управления им и отображения его параметров.
- 4.5. Связать элементы мнемосхемы с соответствующими тегами, используя возможности для придания элементам мнемосхемы динамических свойств.
- 4.6. Для внесения возмущений в технологический процесс использовать имитатор тегов.
- 4.7. Отладить сценарии, продемонстрировать работоспособность мнемосхем преподавателю.
- 4.8. Сохранить скриншоты мнемосхем и листинги сценариев.

5. Варианты заданий

- 5.1. Автоматизация процесса управления водоснабжением. Включает насос, резервуар с отображением уровня (с предупреждением переполнения), выпускной вентиль.
- 5.2. Автоматизация процесса управления электроснабжением. Включает шкаф с выключателем и отображением напряжения, тока, температуры (с предупреждением превышения), автомат отключения при коротком замыкании.
- 5.3. Автоматизация процесса управления бетоносмесителем. Включает дозаторы, смеситель с отображением веса (с предупреждением превышения), таймер смешения, кнопки запуска, опорожнения.
- 5.4. Автоматизация процесса управления паровой турбиной. Включает вентиль подачи пара, турбину с отображением скорости вращения (с предупреждением превышения), автомат отключения при превышении скорости.
- 5.5. Автоматизация процесса управления отопительным котлом. Включает вентиль подачи топлива, котел с отображением температуры и давления пара (с предупреждением превышения), автомат отключения при превышении давления.
- 5.6. Автоматизация процесса управления дизельной электростанцией. Включает кнопки пуска/останова, дизель с отображением температуры и скорости вращения (с предупреждением превышения), автомат отключения при превышении температуры.
- 5.7. Автоматизация процесса сортировки деталей. Включает кнопки пуска/останова, датчик определения вида деталей, два контейнера (для разных видов деталей), датчики переполнения контейнеров (с предупреждением переполнения), кнопки смены режима работы (автоматический, ручной), кнопки смены контейнеров (для ручного режима).

6. Содержание отчета

- 6.1. Цель работы.
- 6.2. Вариант задания.
- 6.3. Скриншоты мнемосхем.
- 6.4. Листинги сценариев.

7. Контрольные вопросы

- 7.1. Общие принципы построения систем визуализации.
- 7.2. Свойства объекта Picture Window, используемые для навигации по мнемосхемам.
- 7.3. Как обеспечить адаптацию размеров кадра и объекта Picture Window?
- 7.4. В каких случаях целесообразно создавать пользовательские объекты?
- 7.5. Способы переноса пользовательских элементов в другие проекты и на другие рабочие станции.
- 7.6. Способы управления внешним видом элементов мнемосхемы.

ПРАКТИЧЕСКИЕ РАБОТЫ

Практическое занятие № 1.

Составьте таблицы истинности для следующих формул:

№	Логическая формула
10.	$x \wedge (z \leftrightarrow y \vee z) \rightarrow \square \rightarrow \square \wedge y$
11.	$(y \wedge \square \wedge \square) \vee x \leftrightarrow \square \vee \square \rightarrow \square$
12.	$z \wedge (x \leftrightarrow y \vee x) \rightarrow \square \wedge \square \vee y$
13.	$x \wedge z \leftrightarrow y \vee (z \rightarrow \square \rightarrow \square)$
14.	$y \wedge (\square \wedge \square) \vee x \leftrightarrow \square \wedge \square$
15.	$y \wedge x \leftrightarrow y \vee x \rightarrow \square \wedge \square \wedge y$
16.	$z \wedge y \rightarrow x \leftrightarrow y \vee (x \rightarrow \square \wedge \square)$
17.	$y \wedge (z \leftrightarrow y \vee x) \rightarrow \square \rightarrow \square$
18.	$(x \wedge x \wedge \square) \vee z \leftrightarrow \square \vee \square \rightarrow \square$

Составьте таблицы истинности и логические схемы для следующих формул:

№	Логическая формула
10.	$x \wedge (z \wedge y \vee z) \vee \square \vee \square \wedge y$
11.	$(y \wedge \square \wedge \square) \vee x \wedge \square \vee \square \wedge \square$
12.	$z \wedge (x \wedge y \vee x) \wedge \square \wedge \square \vee y$
13.	$x \wedge z \wedge y \vee (z \vee \square \wedge \square)$
14.	$y \wedge (\square \wedge \square) \vee x \wedge \square \wedge \square$
15.	$y \wedge x \vee y \vee x \wedge \square \wedge \square \wedge y$
16.	$z \wedge y \vee x \wedge y \vee (x \vee \square \wedge \square)$
17.	$y \wedge (z \vee y \vee x) \wedge \square \wedge \square$
18.	$(x \wedge x \wedge \square) \vee z \wedge \square \vee \square \vee \square$

Практическое занятие №2

Формирование векторов и матриц.

Если надо задать вектор из n элементов, то их значения следует перечислить в квадратных скобках через пробел или через запятую, например:

```
>> V=[2 -3 4 6];
```

При задании матрицы используется символ ; (точка с запятой) для разграничения строк:

```
>> A=[1 3; 0 -3];
```

Можно вводить элементы векторов и матриц в виде арифметических выражений, содержащих доступные системе функции. Элементы матриц могут быть комплексными. Для указания отдельного элемента используется выражение вида $B(k)$ или $A(m,n)$, например:

```
>> A(1,2)
```

```
ans=3
```

Выражение $A(k)$ дает доступ к элементам матрицы, развернутым в один столбец:

```
>> A(4)
```

```
ans=-0.3
```

Индекс k может быть логическим выражением:

```
>> A(A>0)
```

```
ans= 1 3
```

В качестве индекса может быть вектор. Например, запись $A([1,2],2)$ обозначает первый и второй элементы второго столбца. Для указания на m -ю строку или n -й столбец матрицы используются выражения вида $A(m,:)$ или $A(:,n)$. Запись $A(k:m;n)$ обозначает вектор-столбец, сформированный из нескольких элементов n -го столбца матрицы A (с k по m -й).

Имеется ряд функций для задания особых векторов и матриц, например $\text{magic}(n)$ (матрица размером $n \times n$, у которой сумма строк, сумма столбцов и сумма диагоналей одинаковы), $\text{hadamard}(n)$ (матрица Адамара), $\text{hilb}(n)$ (матрица Гильберта), $\text{zeros}(m,\dots,q)$ (многомерный массив из нулей), $\text{ones}(m,\dots,q)$ (массив из единиц), $\text{rand}(m,\dots,q)$ (массив из случайных чисел), $\text{eye}(n)$ (единичная матрица), $\text{repmat}(A,m,n)$ ($m \times n$ дублей матрицы A , m вниз, n вправо), $\text{linspace}(a,b,n)$ (вектор-строка из n чисел, равномерно расположенных на отрезке $[a,b]$, по умолчанию $n=100$), $\text{logspace}(a,b,n)$ (узлы логарифмической сетки по десятичному логарифму, по умолчанию $n=50$) и др.

Элементами матрицы могут быть вектора и матрицы, что позволяет формировать большие матрицы, объединяя малые. Для удаления строк или столбцов удобно использовать пустые квадратные скобки []:

```
>> A(1,:)=[]
```

```
A= 0 -0.3
```

Основную информацию о векторах и матрицах можно получить с помощью функций $\text{size}(A)$ (вектор-строка с размерами матрицы A), $\text{length}(B)$ (число компонентов вектора B), $\text{ndims}(X)$ (размерность массива X), $\text{disp}(X)$ (массив X без его имени), $\text{isempty}(X)$ ($=1$, если $X=[]$), $\text{isequal}(X,Y)$ ($=1$, если $X=Y$), $\text{isnumeric}(X)$ ($=1$, если X – числовая переменная), $\text{islogical}(X)$ ($=1$, если X – логическая переменная).

При формировании матриц можно использовать следующие функции:

$\text{reshape}(X,m,n)$ – формирует матрицу $m \times n$ из элементов массива X ;

$\text{diag}(A,k)$ – формирует вектор-столбец из k -й диагонали A , по умолчанию – из главной;

$\text{tril}(A)$ – формирует нижнюю треугольную матрицу для A ;

$\text{triu}(A)$ – формирует верхнюю треугольную матрицу для A ;

$\text{fliplr}(A)$ – меняет на обратный порядок столбцов;

$\text{flipud}(A)$ – меняет на обратный порядок строк;

$\text{rot90}(A,k)$ – поворачивает A на 90 градусов k раз;

find – определяет индексы ненулевых элементов;

end – на месте индекса задает его последнее значение.

Определитель квадратной матрицы вычисляется с помощью функции $\text{det}(A)$. Для вычисления обратной матрицы методом Гаусса-Жордана используется функция $\text{inv}(A)$.

матриц и векторов поэлементные операции

```
>> v1=[3 6 9 12];
```

```
>> v2=[1,2,3,4];
```

```
>> v1/v2
```

```
ans=3
```

```
>> v1.*v2
```

```
ans=3 12 27 48
```

```
>> v1./v2
```

```
ans=3 3 3 3
```

Пример (анализ устойчивости и ограниченности). Дана неоднородная система дифференциальных уравнений

$$\dot{x} = 2y - 4x + a,$$

$$\dot{y} = 2x - y + b.$$

Требуется проанализировать ее устойчивость и выяснить, при каких постоянных a и b все решения системы ограничены.

$$\dot{X} = AX + B, \text{ где } A = \begin{bmatrix} -4 & 2 \\ 2 & -1 \end{bmatrix}, B = \begin{bmatrix} a \\ b \end{bmatrix} \text{ и}$$

Решение. Запишем систему в матричной форме

найдем корни ее характеристического уравнения, т.е. собственные числа матрицы A :

```
>> A=[-4 2;2 -1]; eig(A)
```

```
ans = -5 0
```

Одно из собственных чисел отрицательно, а другое лежит на мнимой оси, следовательно, однородная система находится на границе устойчивости.

Для ответа на вопрос об ограниченности неоднородной системы найдем ее аналитическое решение с помощью команды `dsolve`:

```
>> s yms a b;s=dsolve('Dx =2*y-4*x+a,Dy=2*x-y+b')
```

```
s = x: [1x1 sym]
```

```
      y: [1x1 sym]
```

```
>> x =s.x
```

```
x =2/5*exp(-5*t)*C1-1/10*b+1/5*a+2/5*t*b+1/5*t*a+1/2*C2
```

```
>> y = s.y
```

```
y =-1/5*exp(-5*t)*C1+4/5*t*b+2/5*t*a+C2
```

$$x = (a + 2b)t + 2c_1 e^{-5t} + a + c_2$$

$$y = 2(a + 2b)t + c_1 e^{-5t} + 2c_2$$

Приведем подобные члены и перейдем к обычной нотации:

Решение будет ограниченным, если $a=-2b$, тогда $x = 2c_1 e^{-5t} + a + c_2, y = c_1 e^{-5t} + 2c_2$.

Нули и полюсы системы, заданной передаточной функцией – это просто корни

z_i и p_i полиномов, стоящих в числителе и знаменателе. Поэтому для вычисления вектора нулей z и вектора полюсов p передаточной функции $Q(p) = \text{num}/\text{den}$, могут использоваться команды $z = \text{roots}(\text{num}); p = \text{roots}(\text{den})$. Если система `sys` задана как `tf`-модель или `ss`-модель, то используются команды $p=\text{pole}(\text{sys}), z=\text{zero}(\text{sys})$. Для нахождения полюсов допустимо также использование команды $p=\text{eig}(\text{sys}), z=\text{zero}(\text{sys})$, что эквивалентно команде $p=\text{eig}(\text{sys}.a)$, т.е. вычислению собственных чисел матрицы A . Функция `tzero` (от `transfer zeros` – передаточные нули) позволяет находить нули системы по матрицам описания в пространстве состояний $z=\text{tzero}(A,B,C,D)$.

Функция `pzmap` предназначена для одновременного вычисления нулей и полюсов. Если набрать $[p, z]=\text{pzmap}(\text{sys})$, то будут выведены столбцы p и z полюсов и нулей, а просто `pzmap(sys)` показывает расположение нулей и полюсов на комплексной плоскости (на графике нули изображаются ноликами, а полюсы – крестиками). При анализе управляемости и наблюдаемости линейных систем используются матрицы управляемости R и наблюдаемости D , построенные на основе матриц A, B, C описания в пространстве состояний:

$$R = [B, AB, \dots, A^{n-1}B], D = \begin{bmatrix} C \\ CA \\ \dots \\ CA^{n-1} \end{bmatrix}$$

Пример (анализ минимальности). Рассмотрим систему с передаточной функцией

$$Q(p) = \frac{p+1}{p^2+3p+2}$$

Она имеет один нуль $z_1 = -1$ и два полюса $p_1 = -1, p_2 = -2$. Система неминимальна, поскольку $p_1 = z_1$. Сформируем `tf`-модель этой системы и найдем ее нули и полюсы:

```
>>s=tf([1 1],[1 3 2])      >>zero(s)      >>pole(s)      >>eig(s)      >> [p, z]=pzmap(s)
      s + 1              -1              -2              -2              p = -2  z = -1
      -----              -1              -1              -1              -1
      s^2 + 3 s + 2
```

Расположение нулей и полюсов на комплексной плоскости можно получить, набрав команду `pzmap(s)` без выходного аргумента (на рис. 2.7, полюсы помечены крестиками, нуль – ноликом).

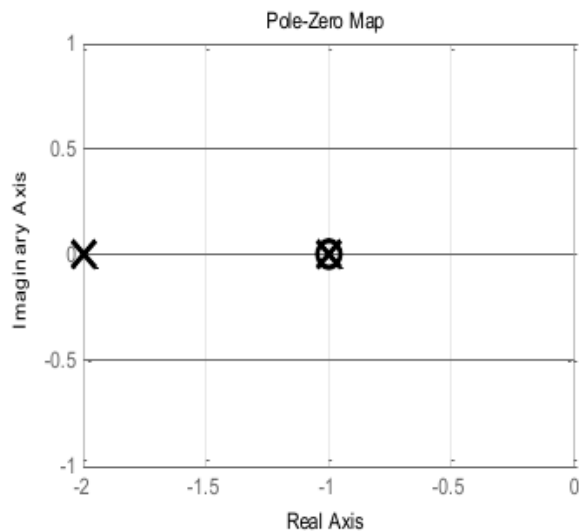


Рис. 2.7

Оба полюса отрицательны, следовательно, система устойчива. Наличие диполя (совпадающего нуля и полюса) в точке $(-1, 0)$ говорит о неминимальности системы. Перейдем к ss-модели и проанализируем ее управляемость и наблюдаемость, используя команды `ctrb`, `obsv` и `rank`:

```
>>s1 = ss(s)           >>R=ctrb(s1)       >>D=obsv(s1)   >>rank(R) >>rankD
a = -3 -2             b = 2                 R = 2 -6      D = 0,5 0,5
   1  0              c = 0.5 0.5           0  2        -1 -1      2      1
```

Анализируя ранги матриц R и D , заключаем, что система $s1$ управляема, но ненаблюдаема, следовательно, ее порядок может быть понижен.

Найдем передаточную функцию минимальной реализации: $q = \min_{\text{real}}(s)$, получаем ответ

$$q = \frac{1}{s + 2}.$$

К тому же результату приходим, сокращая числитель и знаменатель исходной передаточной функции $Q(p)$ на общий множитель $p+1$.

Заметим, что статический коэффициент усиления при переходе к минимальной реализации не изменился:

```
>>K=dcgain(s)         >>K=dcgain(q)
K=0.5                 k=0.5
```

Весовая и переходная функции также остаются прежними. В этом можно убедиться с помощью команд `impulse(s, q)`, `step(s, q)`, по которым будут построены графики указанных функций для обеих систем. Команды `ctrb` и `obsv` можно использовать при работе с символьными выражениями, например, когда часть элементов матриц A , b , c заданы в буквенном виде.

1.1.4 Определение и обработка числовых матриц

Файл-функция для заполнения матрицы:

```
function A=exam(n)
A=zeros(n);
for i=1:n
    for j=1:n
        A(i,j)=1/(i^2+j^2);
    end
end
end
```

Процедура перемножения матриц:

```
subroutine multmtr(n1, n2, n3, X, Y, Z)
С Вычисление  $Z=X*Y$ , где  $X - n1 \times n2$ ,  $Y - n2 \times n3$ ,  $Z - n1 \times n3$ 
integer n1, n2, n3, i, j, k
real*8 X(n1,n2), Y(n2,n3), Z(n1,n3)
do 30 i=1,n1
```

```

do 20 j=1,n3
  Z(i,j)=0.0d0
  do 10 k=1,n2
    Z(i,j)=Z(i,j)+X(i,k)*Y(k,j)
  10 continue
20 continue
30 continue
End

```

Вычисление элементов матрицы с предварительным выделением памяти:

```

A=zeros(400);
for i=1:400
  for j=1:400
    A(i,j)=1/(i^2+j^2);
  end
end
end

```

Сравнение матричных операндов:

```

a=[1 2 3; 4 5 6]
b=[2 2 2; 5 5 5]
c=a >= b
pause

```

Объединение малых матриц в большую:

```

A=magic(3)
B=[A A+16; A+32 A+16]
sum(B)
sum(B.')
D=magic(6)
sum(D)
sum(D.')
pause

```

Удаление столбцов и строк матриц:

```

M=[1 2 3; 4 5 6; 7 8 9]
M(:,2)=[]
M(2,:)=[]
pause

```

Практическое занятие

1. Построить графики функции с помощью декартовой системы координат:

1. $f(x) = x^2$; $g(x) = x^3$; $x \in [-1, 1]$.
2. $u(x) = x^4$; $v(x) = x^5$; $x \in [-1, 1]$.
3. $f(x) = \arcsin x$; $g(x) = \arccos x$; $x \in [-1, 1]$.
4. $u(x) = \operatorname{arctg} x$; $v(x) = \operatorname{arctg} 3x$; $x \in [-1, 1]$.
5. $f(x) = \operatorname{sh} x$; $g(x) = \operatorname{ch} x$; $x \in [-1, 1]$.
6. $u(x) = e^x$; $v(x) = e^{-x}$; $x \in [-0.6, 0.6]$.
7. $f(x) = \frac{\sin x}{x}$; $g(x) = e^{-x} \cos x$; $x \in [0, 0.1, 2\pi]$.
8. $u(x) = \sin(\ln(x+1))$; $v(x) = \cos(\ln(x+1))$; $x \in [0, 2\pi]$.
9. $f(x) = x^x$; $g(x) = x^{x^x}$; $x \in [0.1, 1]$.
10. $u(x) = \frac{1}{1+x}$; $v(x) = \frac{1}{1+\frac{1}{1+x}}$; $x \in [0, 1]$.

2. Построить графики функции с помощью декартовой системы координат:

$$f(x) = \begin{cases} |x|, & -2 \leq x \leq 1 \\ \sin \frac{\pi}{2} x, & 1 < x \leq 2 \\ (2-x)^3, & 2 < x \leq 3 \end{cases} \quad 2. \quad f(x) = \begin{cases} (x-1)^2, & -2 \leq x \leq 1 \\ \cos \frac{\pi}{2} x, & 1 < x \leq 3 \\ 1 - e^{3-x}, & 3 < x \leq 8 \end{cases}$$

1.

$$f(x) = \begin{cases} e^x, & -2 \leq x \leq -1 \\ \frac{|x|}{e}, & -1 < x \leq 1 \\ e^{-x}, & 1 < x \leq 2 \end{cases} \quad 4. \quad f(x) = \begin{cases} e^{x+1}, & -2 \leq x \leq -1 \\ x^2, & -1 < x \leq 1 \\ (2-x)^3, & 1 < x \leq 2 \end{cases}$$

3.

$$f(x) = \begin{cases} x^2 \log_2 x, & 1 \leq x \leq 2 \\ x^3/2, & 2 < x \leq 3 \\ x^x/2, & 3 < x \leq 3.5 \end{cases}$$

5.

3. Построить графики в пространстве следующие функции:

- 1) $z(x, y) = (\sin x^2 + \cos y^2)^{xy}$ $x \in [-1, 1]$ $y \in [-1, 1]$
- 2) $z(x, y) = \arctan(x+y)(\arccos x + \arcsin y)$ $x \in [-1, 1]$ $y \in [-1, 1]$
- 3) $z(x, y) = (1+xy)(3-x)(4-y)$ $x \in [0, 3]$ $y \in [0, 4]$
- 4) $z(x, y) = e^{-|x|} (x^5 + y^4) \sin(xy)$ $x \in [-2, 2]$ $y \in [-3, 3]$
- 5) $z(x, y) = (y^2 - 3) \sin \frac{x}{|y|+1}$ $x \in [-2\pi, 2\pi]$ $y \in [-3, 3]$

Практическое занятие

Задачи:

$$\begin{aligned}x + y - z &= 36 \\x + z - y &= 13 \\y + z - x &= 7 \\x + 2y + z &= 4 \\3x - 5y + 3z &= 1 \\2x + 7y - z &= 8 \\2x - 4y + 9z &= 28 \\7x + 3y - 6z &= -1 \\7x + 9y - 9z &= 5 \\2x + y &= 5 \\x + 3z &= 16 \\5y - z &= 10 \\x + y + z &= 36 \\2x - 3z &= -17 \\6x - 5z &= 7 \\7x + 2y + 3z &= 15 \\5x - 3y + 2z &= 15 \\10x - 11y + 5z &= 36 \\x + 2y - 4z &= 1 \\2x + y - 5z &= -1 \\x - y - z &= -2 \\2x - y + z &= -2 \\x + y + 3z &= -1 \\x - 3y - 2z &= 3 \\2x - 3y + z &= -1 \\5x + 2y - z &= 0 \\x - y + 2z &= 3 \\3x + 2y + z &= 20 \\4x + 0 * y - 10z &= -10 \\-x - 2y + 2z &= -1\end{aligned}$$

Практическая работа

Цель работы: Преобрести навыки о программе AutoCAD. Начертить простые чертежи в программе AutoCAD.

Необходимое оборудование: Компьютер в типе Pentium. Программа AutoCAD.

Теоретическая часть

AutoCAD – универсальный графический пакет, предназначенный для любого специалиста, работающего с технической графикой.

В основе структуры AutoCADa используется принцип открытой архитектуры, который позволяет адаптировать и развивать многие функции применительно к своим требованиям. Поэтому AutoCAD представляет помимо основных возможностей по созданию сложной технической графики и новые:

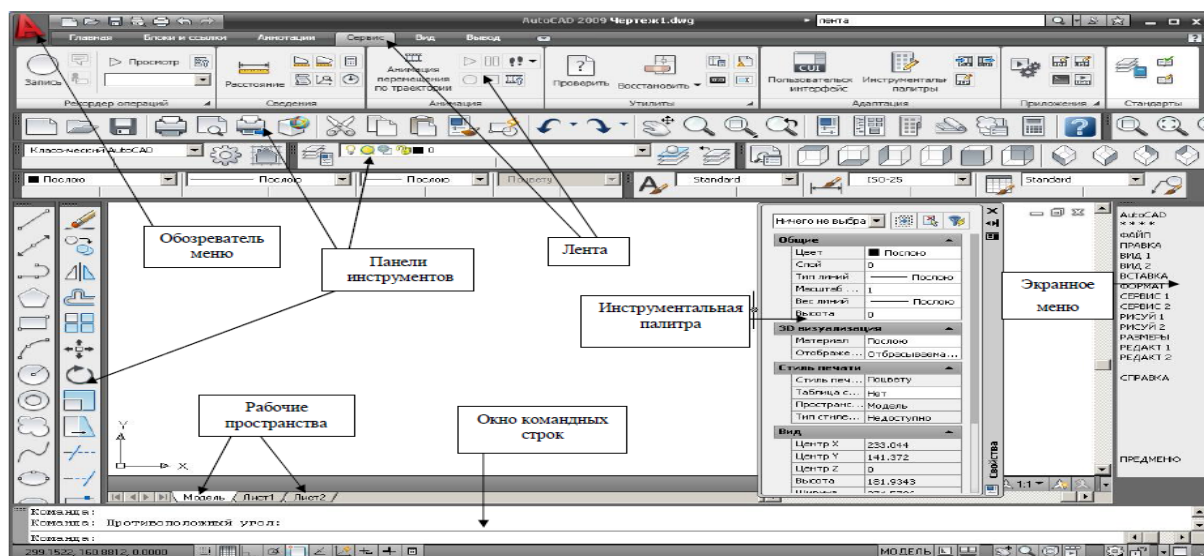
- *графическое моделирование* (использование компьютера как мощного вычислительного средства, а также для работы с пространственными моделями);
- *создание и ведение информационной базы данных (архива) чертежей;*
- *создание библиотеки стандартных элементов чертежей;*
- *параметризация чертежей* (построение чертежа с новыми размерами на основе один раз построенного чертежа-модели);

Лента состоит из последовательности панелей, которые организованы в виде вкладок, помеченных названием задачи. Панели ленты содержат многие из тех же инструментов и элементов управления, которые доступны на панелях инструментов и в диалоговых окнах.

Дополнительным элементом управления являются инструментальные палитры, они представлены отдельными вкладками и являются эффективным средством управления, упорядочения, распределения и размещения различных примитивов: блоков, штриховок и других инструментов.

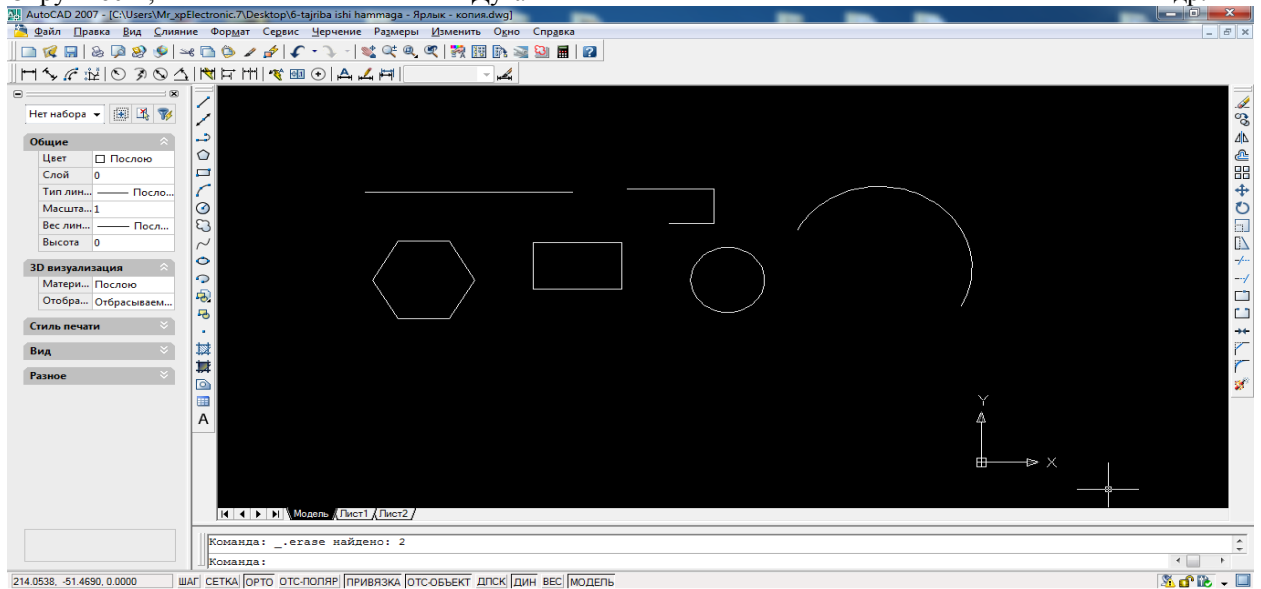
Структура выполнения работ.

1. Запустите программу AutoCAD с помощью пиктограммы на рабочем столе.
2. Изучим структуру AutoCAD.

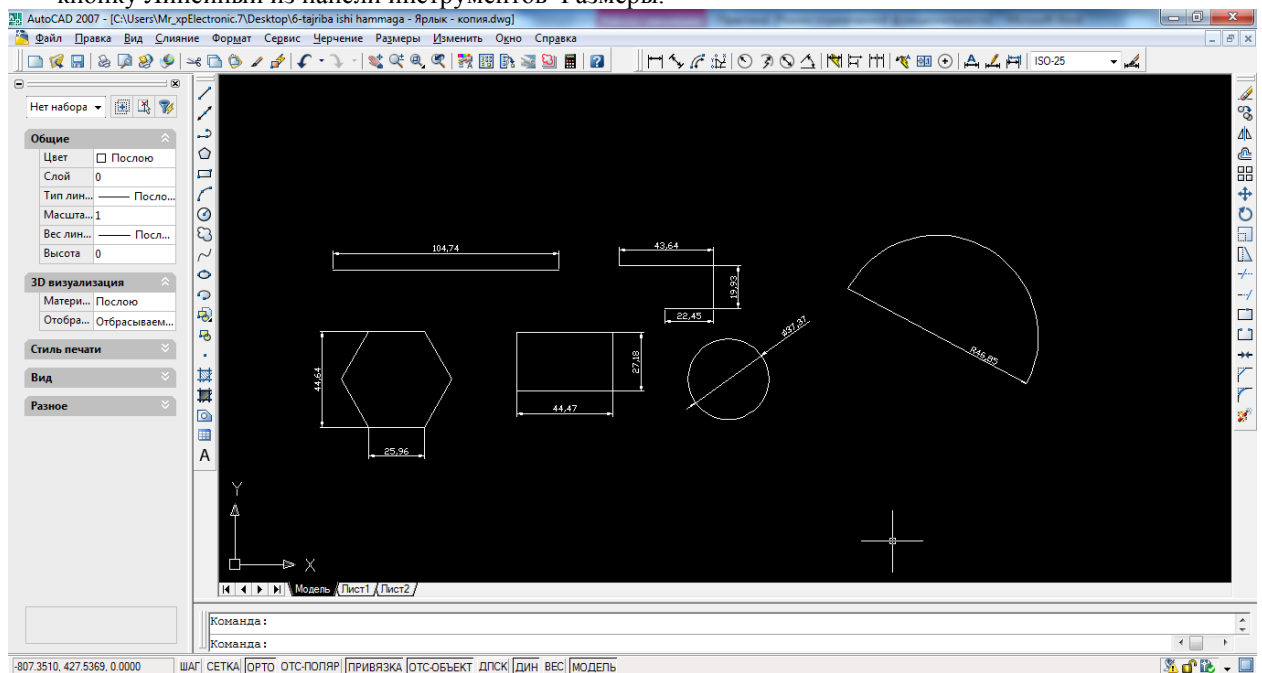


3. Начертим простые геометрические фигуры: Линия, прямоугольник, многоугольник, окружность, дуга и др.

4. Используйте кнопки из панели инструментов Черчение: Линия, Прямоугольник, Многоугольник, Окружность, Дуга и др.



5. Чтобы изменить параметры чертежа используйте кнопки панели меню Изменить: Стереть, Копировать, Зеркальное отражение, Переместить, Повернуть и др.
6. Начертив чертежи проверьте единицы отрезка используя кнопку Линейный из панели инструментов Размеры.



Задания по теме:

1. Начертите автомобиль.
2. Начертите двухэтажный дом.
3. Начертите угловой штамп.
4. Начертите болт.

Практическая работа .







Создание статического и динамического изображения в Trace Mode

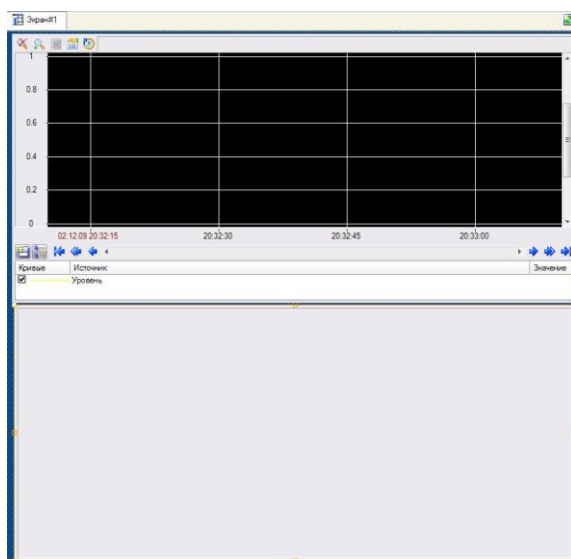
Задание:

1. создать новый проект. В данном проекте создать генератор, привязанный к каналу, значение которого определяется уровнем продукта в емкости;
2. создать экран, расположить на нем тренд, строящий зависимость уровня продукта во времени.
3. создать статическое изображение емкости в разрезе, насоса, трех труб, по одной трубе продукт поступает в емкость, по другой-вытекает из нее. Вторая труба соединена с третьей через насос.
4. создать динамический объект, имитирующий заполнение емкости, используя графический файл.


















Ход работы

1. **Создайте канал, который будет пропорционален уровню продукта в емкости.** Назовем данный канал как уровень, для лучшего восприятия.
2. **Создайте генератор синусоидального сигнала.**
3. **Произведите привязку созданного генератора к созданному каналу.**
4. **Создайте экран.**
5. **Создайте тренд, настройте кривую и произведите привязка к созданному каналу .**
6. **Создание статического изображения.**

a. **Создание рамки.** Для создания рамки щелкните левой клавишей мыши по иконке . Если нет указанной иконки на панели инструментов, то щелкните правой клавишей мыши по одной из иконок: , , или . Среди предложенных объектов выберите рамку . Щелчком левой клавишей мыши задайте противоположные углы рамки. Для перехода в режим редактирования щелкните левой клавишей мыши по иконке .



Размещение рамки

b. **Создание емкости.** Щелкните левой клавишей мыши по иконке . Если данной иконки нет на панели инструментов, то щелкните правой кнопкой мыши по одной из иконок: , , , , , , , , , , , или . Среди предложенных объектов выберите емкость . После выбора инструмента поместите емкость на экране. Задайте противоположные углы емкости щелчком левой клавиши мыши. щелкните левой клавишей мыши по иконке . Емкость примет вид, изображенный на рис. 3.33. Выделите созданную емкость или дважды щелкните по ней левой клавишей мыши. Откроется окно свойств объекта. В поле *толщина стенок* задайте толщину больше 0. Емкость будет изображена в разрезе. В поле *верхний* и *нижний край* выберите необходимый вид края. К примеру, верхний край примет вид , а нижний— . Для задания материала емкости раскройте раздел материал, дважды щелкнув левой клавишей мыши по подчеркнутой строчке *материал*, если данный раздел не раскрыт. В поле *выбрать из списка* выберите значение *true*. В поле *материал* выберите необходимый материал, к примеру,




хром. В поле *стандартная текстура* выберите необходимую текстуру, к примеру, гравировку. При необходимости можно добавить другие объемные фигуры.



Создание емкости






Задание толщины стенок емкости

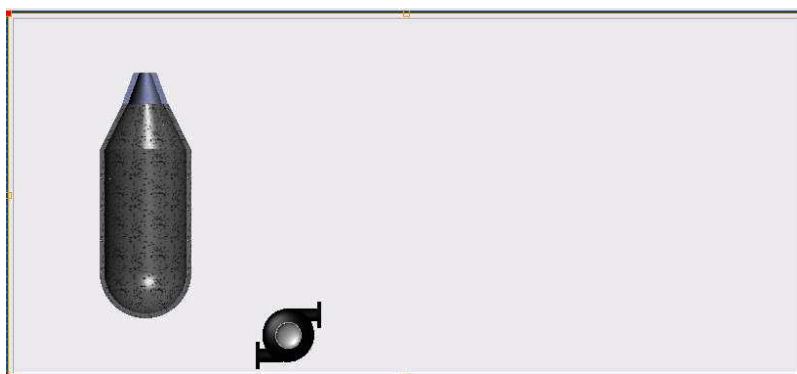
Для размещения конуса щелкните правой клавишей мыши по иконке  и среди предложенных инструментов выберите . Разместите на экране конус, задав противоположные углы прямоугольника в который будет вписан конус щелчком левой клавиши мыши. Щелкните левой клавишей мыши по иконке . Выделите созданный конус. Если не открылось окно свойств конуса, дважды щелкните левой клавишей мыши по нему. В поле *толщина стенок* задайте ту же толщину, что и у емкости. Раскройте раздел *материал* двойным щелчком мыши по подчеркнутой строчке *материал*, если слой еще не раскрыт. Выберите значение *true* в поле *выбрать из списка*. Выберите требуемый материал в поле *материал*, к примеру, олово. В поле *стандартная гравировка* задайте гравировку, к примеру, шлифовку. Емкость примет вид






Пример емкости

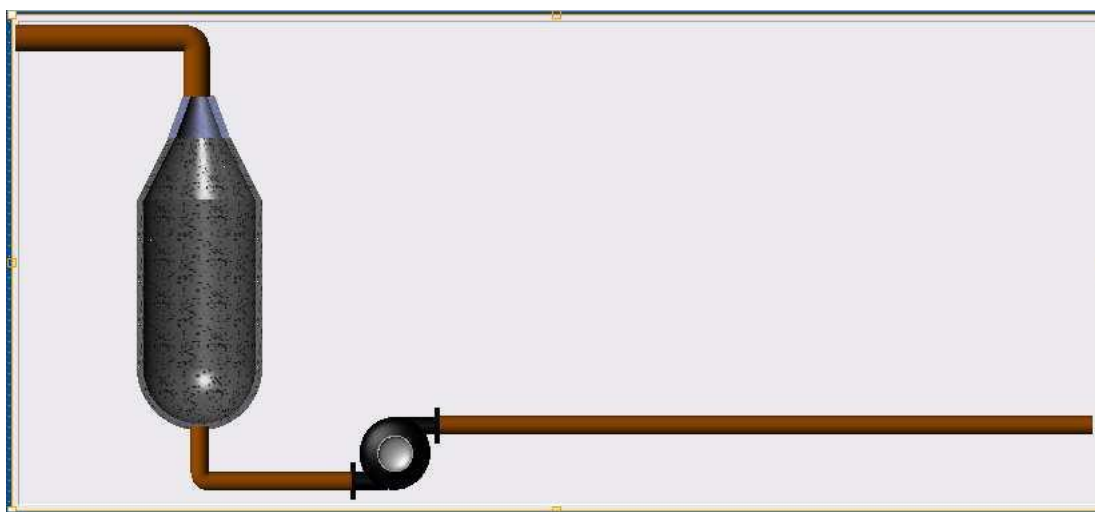
с. **Создание насоса.** Щелкните правой клавишей мыши по иконке . Среди предложенных инструментов выберите . Для размещения насоса на экране задайте противоположные углы прямоугольника, в который будет вписан насос щелчком левой клавиши мыши. Щелкните левой клавишей мыши по иконке . Выделите насос. Если не открылось окно свойств насоса, дважды щелкните левой клавишей мыши по созданному насосу. Раскройте раздел *материал*, двойным щелчком по подчеркнутому тексту *материал*, если он не раскрыт еще. В поле *выбрать из списка* установит значение *true*. В поле

материал выберите необходимый материал, к примеру, пластик черный. В поле **форма насоса** выберите нужную форму насоса. Статическое изображение примет вид



Размещение насоса

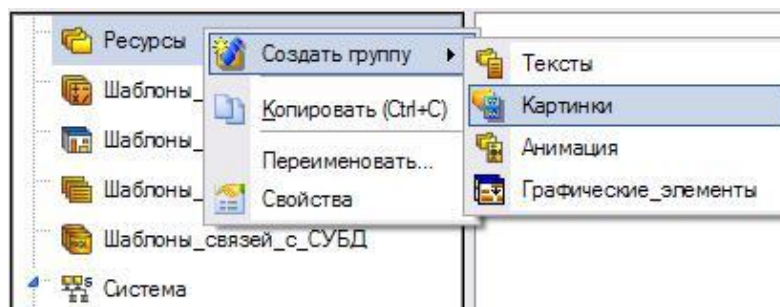
d. **Создание труб.** Щелкните правой клавишей мыши по иконке . Среди предложенных инструментов выберите . Создайте трубу по которой продукт поступает в емкость и по которой из емкости течет в насос. Для этого щелчком левой клавиши мыши отметьте местоположение начала трубы. Переведите курсор мыши в положение изгиба трубы и снова щелкните левой клавишей мыши. Таким образом, отмечаются все точки изгиба трубы. Когда курсор переведен в положение, где размещается конец трубы, щелкните правой кнопкой мыши, завершая создание текущей трубы. Создайте аналогично трубу, по которой продукт поступает в насос и вытекает из него. Для редактирования свойств каждой трубы выделите трубу. Если не открылось окно свойств, дважды щелкните левой клавишей мыши по трубе. В поле толщина подберите толщину каждой трубы, которая лучше будет подходить для рисунка. В поле базовый цвет выберите необходимый цвет, к примеру . Статическое изображение примет вид,



Создание труб

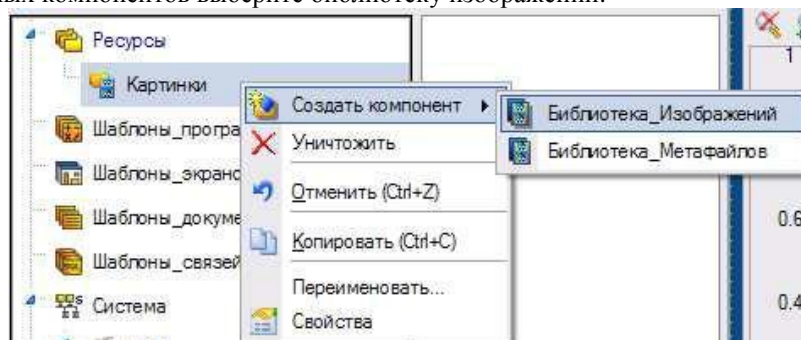
7. **Создание динамического изображения.**

a. **Импорт изображения.** Выделите строку **ресурсы** навигатора проекта. Вызовите контекстное меню. Выберите строку **создать группу**. Среди предложенных групп выберите **картинки**.





Создание группы картинки

Выделите созданную группу картинки, вызовите контекстное меню, выберите **создать компонент**. Среди предложенных компонентов выберите библиотеку изображений.




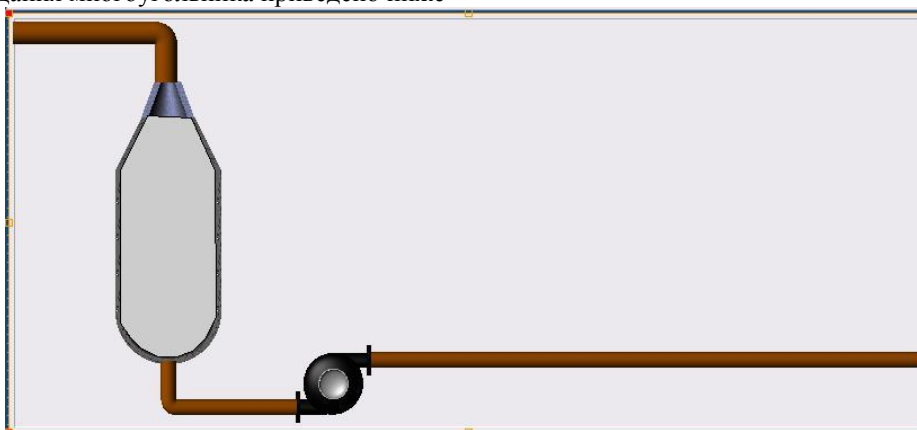
Создание библиотеки изображений

Для импортирования изображения дважды щелкните левой клавишей мыши по созданной **библиотеке изображений**. Откроется пустое окно. В данном пустом окне вызовите контекстное меню и выберите **импортировать**. Откроется диалоговое окно для открывания графического файла. Откройте графический файл изображения, которое будет использоваться в дальнейшем.

в. Создание динамической заливки. Создайте многоугольник. Щелкните левой клавишей мыши по иконке . Если на инструментальной панели нет иконки , щелкните правой клавишей мыши по одной из иконок:



Среди предложенных инструментов выберите . Для размещения многоугольника щелкните левой клавишей мыши там, где должен располагаться один из углов многоугольника. Задайте точки излом щелчком левой клавишей мыши там, где будут располагаться другие углы многоугольника. Последний угол многоугольника следует отмечать щелчком правой клавиши мыши. Результат создания многоугольника приведено ниже



Создание многоугольника для заливки

Выделите созданный многоугольник. Если окно свойств не открылось, дважды щелкните левой клавишей мыши по созданному многоугольнику. Раскройте раздел **заливка** двойным щелчком левой клавиши мыши, если он не раскрыт. В появившемся поле **стиль** выберите без заливки.

Перейдите на закладку динамическая заливка . Поставьте флажок **разрешено**. Раскройте раздел «**слой**». Для этого дважды щелкните левой клавишей мыши на тексте **слой**. В появившемся поле **имя** введите имя для слоя. Произведите привязку слоя к аргументу, посредством которого произведена привязка тренда к канала, хранящему значение уровня продукта в емкости. В поле **тип заливки** выберите

изображение. Щелкните в поле **изображение**. Откроется окно, предлагающее выбор изображений, хранящихся в библиотеке изображений, созданной ранее. Выберите необходимое изображение и щелкните левой клавишей мыши по кнопке **готово**. В поле **Макс.** установите значение верхней границы диапазона значений, хранимых в канале, созданном ранее, а в поле Мин.— нижней границы.



Выбор изображения для заливки

8. **Запуск проекта.** Произведите запуск проекта, аналогично первой работе. Пример результата исполнения созданного проекта приведен ниже

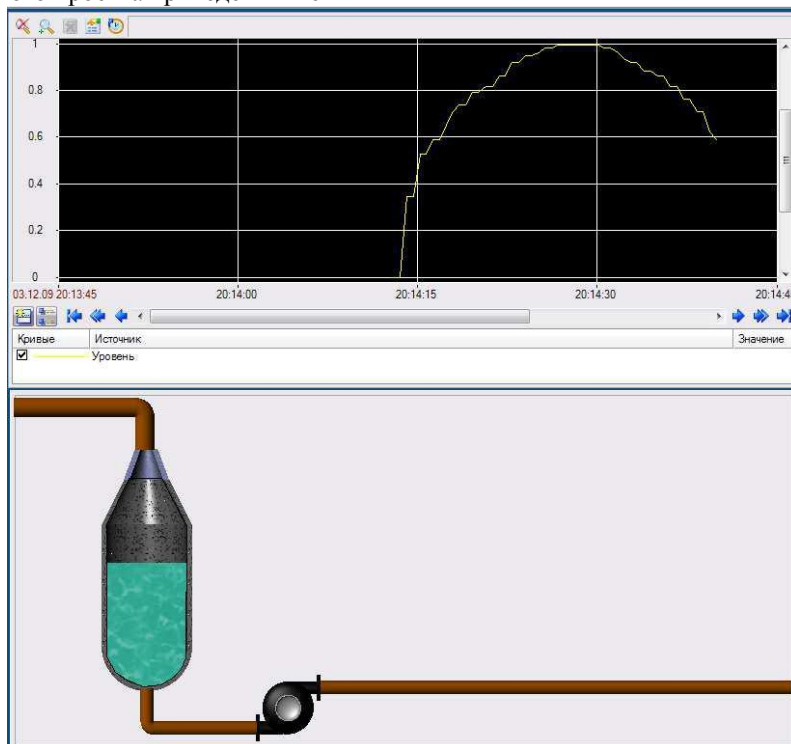


Рис. 3.42 Пример результата запуска

Вопросы для самоконтроля:

1. Опишите различие входных и выходных каналов.
2. Что такое привязка, зачем она нужна, как производится?
3. Что такое навигатор проекта?
4. Как импортируется изображение?
5. Как создаются компоненты базы каналов в Trace Mode?
6. Как размещается объект на экране?
7. Зачем нужно окно свойств объекта, что оно дает?

8. Чем отличается статическое изображение от динамического?
9. Как создается статическое изображение?
10. Как создается динамическое изображение?
11. Какие элементы изображения могут быть динамическими в Trace Mode?

Практическое занятие

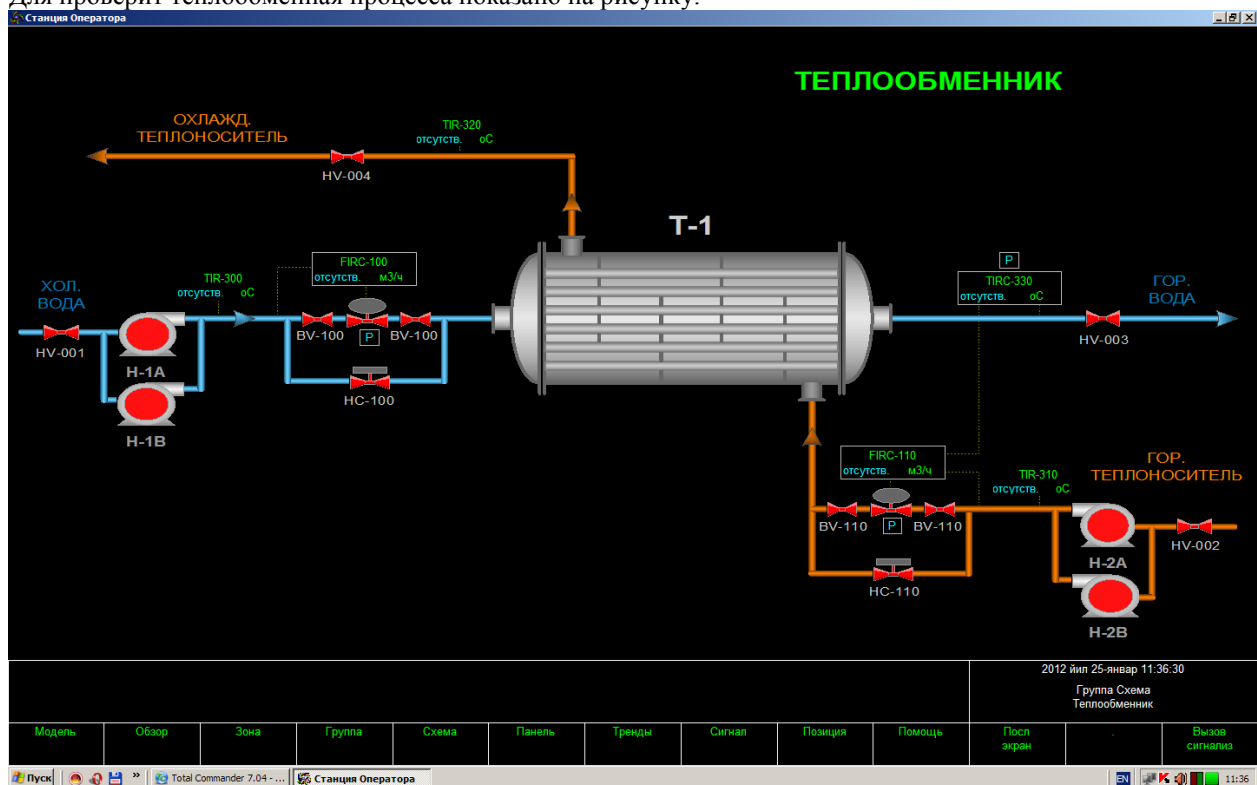
Цель задачи: Принцип работ тренажерного комплекса.

Основная часть: Внедрение современной АСУТП не может быть полноценным без организации тренажерной подготовки эксплуатационного персонала. При этом наибольший эффект по обучению и сокращению затрат на освоение персоналом новой техники может быть достигнут, если тренажёр будет введен в действие ещё до начала внедрения реальной АСУТП на объекте.

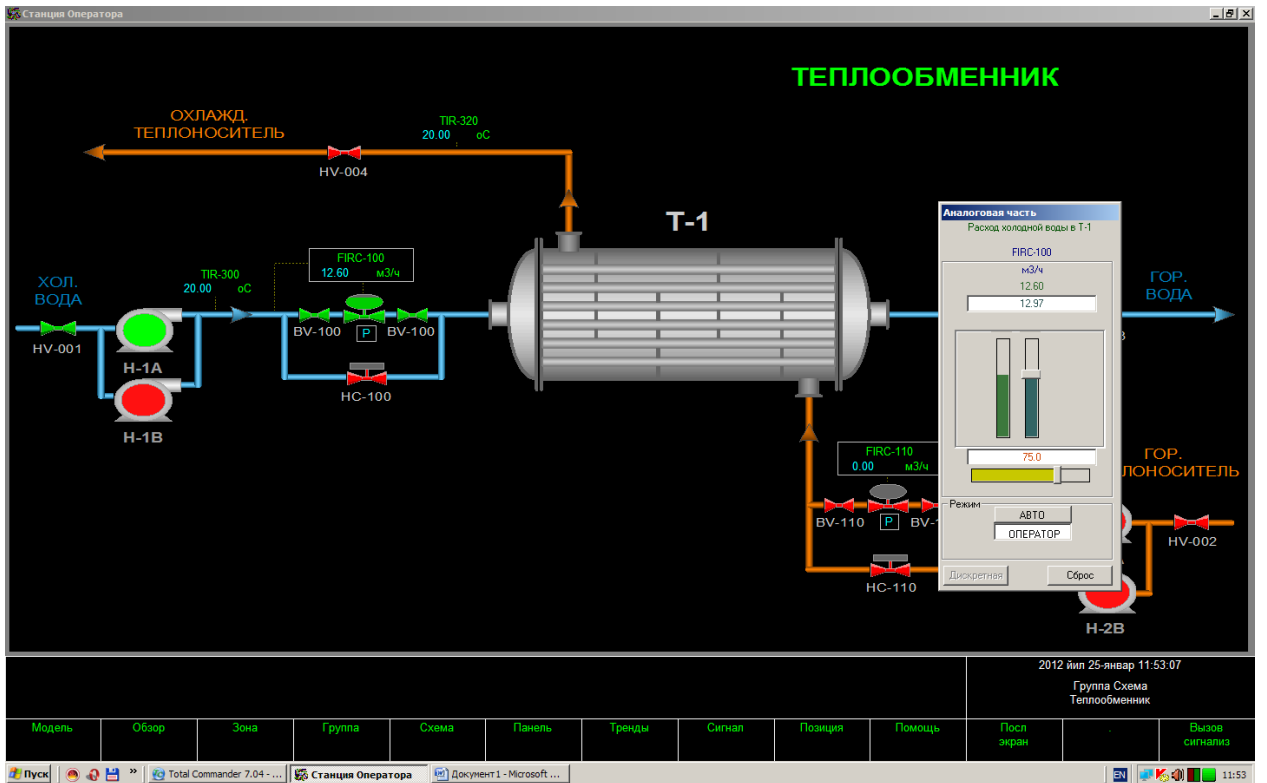
Компьютерный тренажёр состоит из следующих составных частей: математической модели объекта управления (например, энергоблока), модели системы управления и специального программного обеспечения инструктора, обеспечивающего организацию тренировочного процесса.

Совершенно очевидно, что оптимальным решением для современных тренажеров является отказ от создания имитационной модели системы управления и замена её реальной АСУТП с использованием как технических средств, так и программного обеспечения, полностью соответствующих АСУТП энергоблока-прототипа.

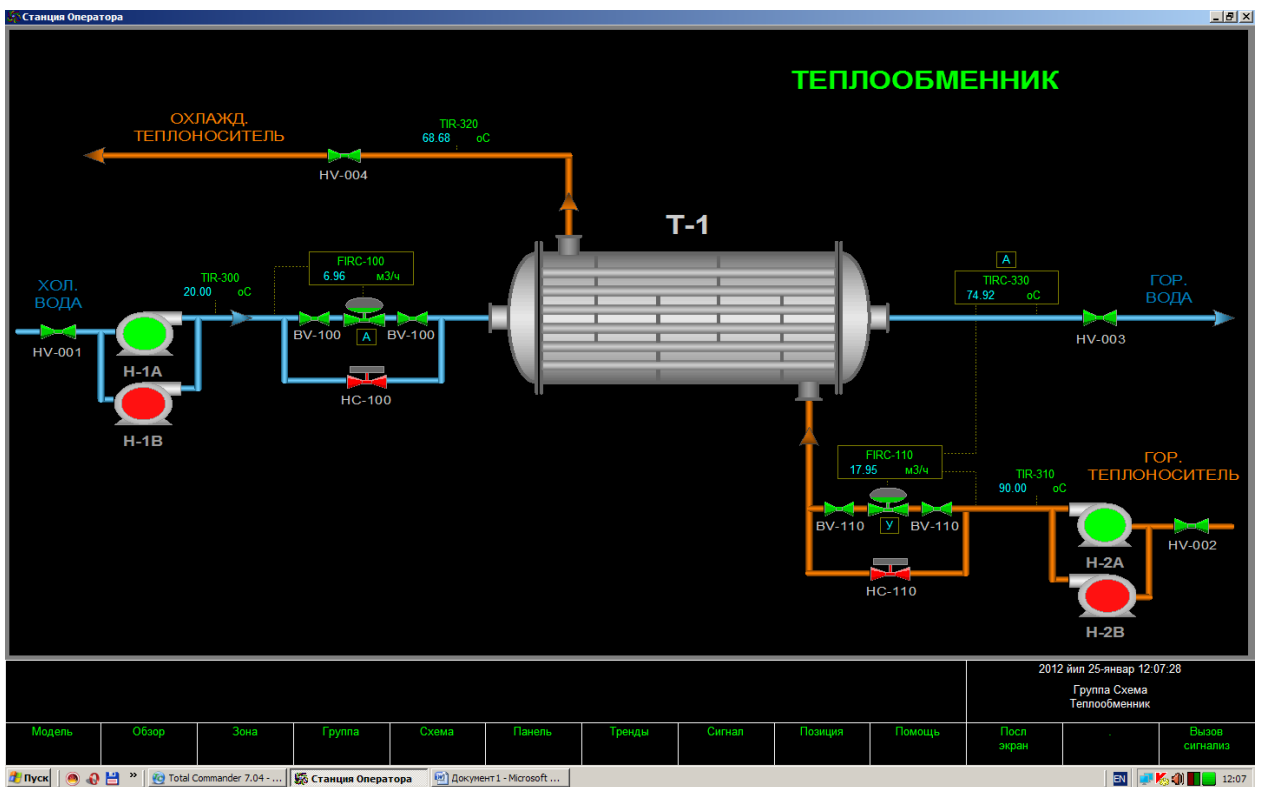
КТК-М тренажерного комплекс состоит из двух част экрана 1 экран оператора 2 экран инструктору. Для проверит теплообменная процесса показано на рисунку.



Панель оператора.



Исправит ошибку процесса.



Процесс в норме

Практическое занятие

Программное Обеспечение Функций Управления «Control Builder»

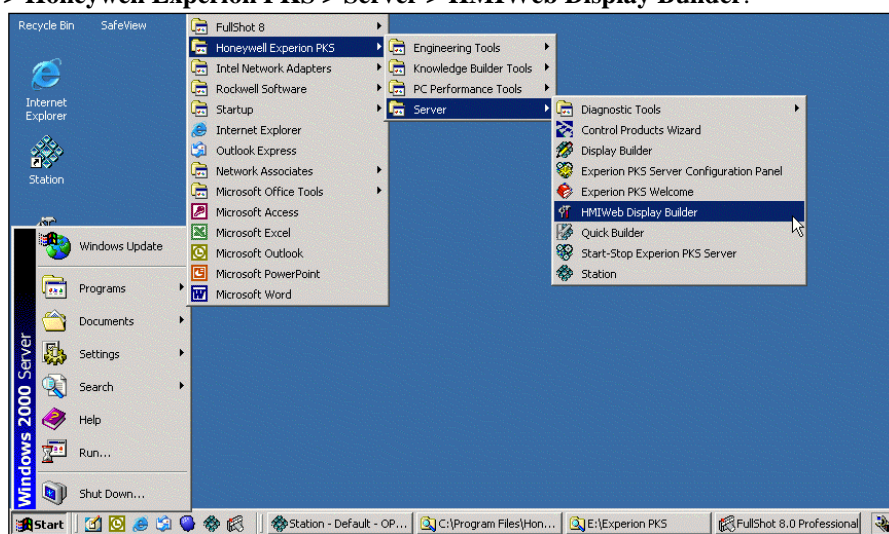
Программирование контуров измерения, регулирующих контуров, контуров безопасного управления осуществлено при помощи программы «Control Builder».

Control Builder (графический построитель алгоритмов управления) - эта программа работает в среде Windows 2000 либо на отдельной операторской станции, либо на сервере и обращается непосредственно к базе данных. Эта программа поддерживает системную архитектуру, документацию и контроль для контроллеров ведения процесса Experion PKS C200.

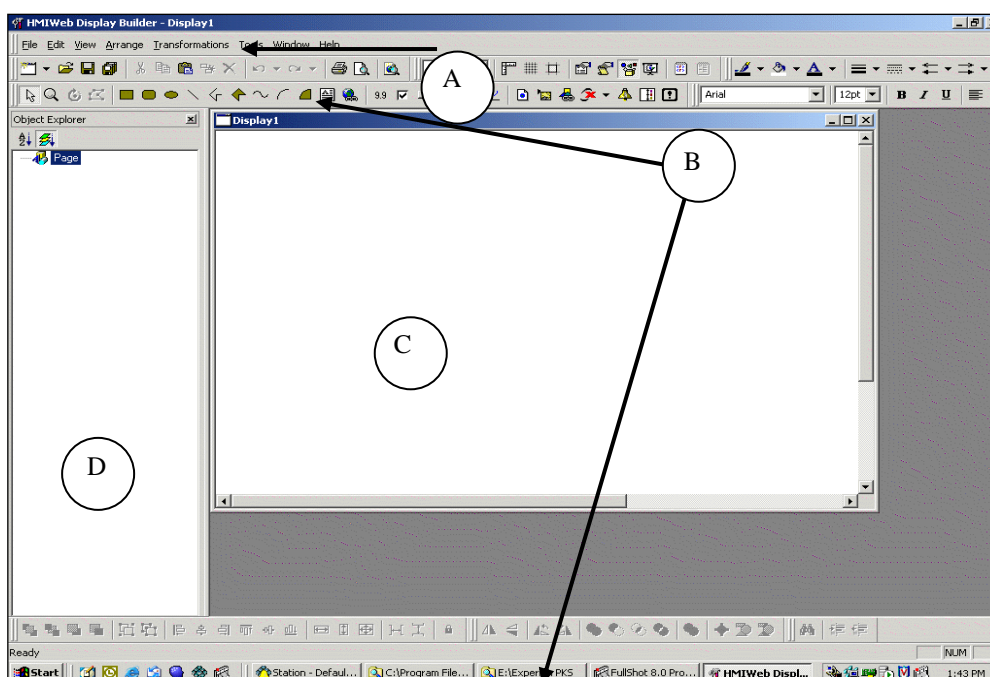
Среда исполнения дисплеев HMIWeb Display Builder.

Графический редактор построителя дисплеев (HMI Web Display Builder) это удобный в использовании инструмент, позволяющий создавать пользовательские дисплеи для станции системы Experion PKS. С помощью HMIWeb Display Builder, инженер создает фон схемы, анимацию, а так же другие активные функции, которые графически показывают данные процесса или позволяют предпринимать действия. Схемы привязываются к системным параметрам точек, которые в свою очередь представляют собой заводские датчики, регуляторы и т.д. HMIWeb Display Builder позволяет пользователю создавать дисплеи с использованием полной цветовой палитры, анимации, трехмерных объектов и фотографии.

1. Для запуска приложения **HMIWeb Display Builder** на **Сервере**: Из меню **Start**, выберите **Start > Programs > Honeywell Experion PKS > Server > HMIWeb Display Builder**.



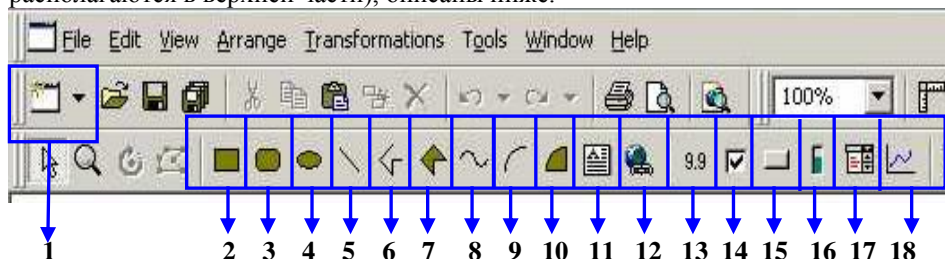
2. HMIWeb Display Builder откроется с пустым экраном, как показано ниже.



- A→ ПАНЕЛЬ МЕНЮ
- B→ ПАНЕЛИ ИНСТРУМЕНТОВ
- C→ ЗОНА ДИСПЛЕЯ\РИСОВАНИЯ
- D→ ПРОВОДНИК ОБЪЕКТОВ

3. B→ Понятие ПАНЕЛЕЙ ИНСТРУМЕНТОВ

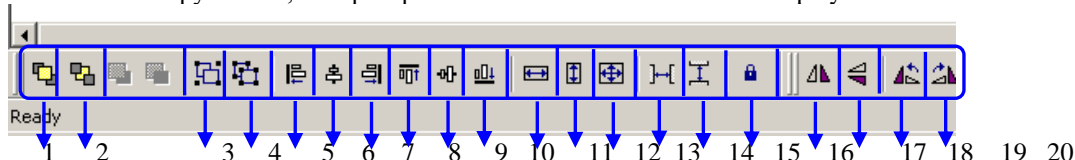
Наиболее часто используемые средства панели инструментов HMIWeb Display Builder (которые располагаются в верхней части), описаны ниже.



1. **New displays:** открывает новые дисплеи.
2. **Rectangle tool button:** используется для рисования прямоугольников.
3. **Rounded rectangle:** используется для рисования скругленных прямоугольников.
4. **Oval:** используется для рисования кругов или овалов.
5. **Line:** используется для рисования линий.
6. **Polyline:** используется для рисования незамкнутых многосегментных линейных объектов.
7. **Polygon:** используется для рисования замкнутых многосегментных линейных объектов.
8. **Bezier curve:** используется для рисования криволинейных объектов.
9. **Arc:** используется для рисования арок.
10. **Wedge:** используется для рисования ¼ окружности.
11. **Textbox:** используется для вставки текста в дисплей.
12. **Hyperlink:** используется для создания ссылки на интернет-сайт.
13. **Alphanumeric:** используется для вставки значения в дисплей.
14. **Checkbox:** используется показа выбора некоторых опций.
15. **Pushbutton:** используется для вызова другого дисплея.
16. **Indicator:** используется для показа индикации уровня.
17. **Combo box:** используется для вставки списка (режимов/состояний точки) в дисплей.
18. **Chart:** используется для создания трендов.

4. B→ Понятие ПАНЕЛЕЙ ИНСТРУМЕНТОВ

Описания инструментов, которые располагаются в нижней части Display Builder.



1. **Bring to front:** перемещает выбранный объект на передний план.
2. **Send to back:** перемещает выбранный объект на задний план.
3. **Group:** группирует выбранные объекты.
4. **Ungroup:** разгруппировывает выбранный объект.
5. **Align left:** выравнивает выбранные объекты по левому краю объекта, выбранного первым
6. **Align center:** выравнивает выбранные объекты по центру объекта, выбранного первым
7. **Align right:** выравнивает выбранные объекты по правому краю объекта, выбранного первым
8. **Align top:** выравнивает выбранные объекты по верхнему краю объекта, выбранного первым
9. **Align middle:** выравнивает выбранные объекты по середине объекта, выбранного первым
10. **Align bottom:** выравнивает выбранные объекты по нижнему краю объекта, выбранного первым
11. **Make same width:** изменяет ширину выбранных объектов на ширину объекта, выбранного первым
12. **Make same height :** изменяет высоту выбранных объектов на высоту объекта, выбранного первым
13. **Make same width and height:** изменяет ширину и высоту выбранных объектов на ширину и высоту объекта, выбранного первым
14. **Even horizontal spacing:** распределяет выбранные объекты равномерно по горизонтали.
15. **Even vertical spacing:** распределяет выбранные объекты равномерно по вертикали.
16. **Lock object:** устанавливает/снимает запрет на изменение выбранного объекта.
17. **Flip horizontal:** отражает выбранный объект слева направо.
18. **Flip vertical:** отражает выбранный объект сверху вниз.

19. Rotate left: поворачивает выбранный объект на 90° влево.

Rotate right: поворачивает выбранный объект на 90° вправо.

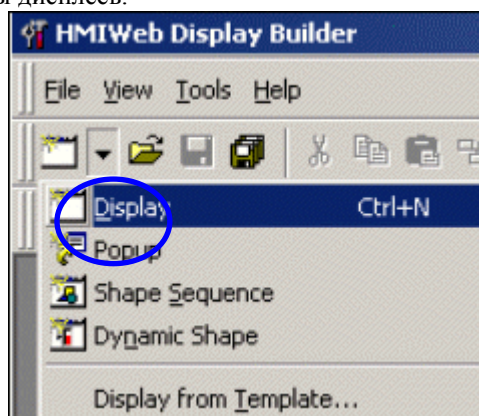
5. C → ЗОНА ДИСПЛЕЯ\РИСОВАНИЯ

По умолчанию, во время запуска HMIWeb Display Builder, открывается пустой дисплей. HMIWeb Display Builder присваивает дисплею имя по умолчанию. Позже, во время сохранения дисплея, вы можете поменять имя на необходимое.

Существует 4 типа дисплеев, которые можно создать, используя приложение HMIWeb Display Builder.

Нажмите выпадающую кнопку со стрелкой “New Display”.

Выпадающее меню покажет типы дисплеев.



Display: стандартный дисплей.

Popup: всплывающий дисплей является вторичным окном, которое появляется, когда пользователь щелкает по объекту, к которому прикреплен Popup.

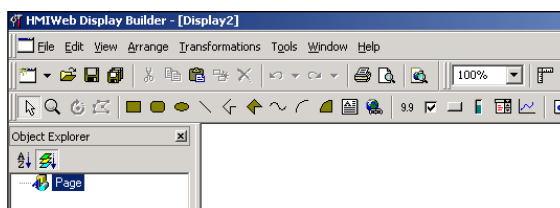
Shape Sequence: Последовательность Форм является, на самом деле, "пользовательским объектом", который используется в стандартных дисплеях как индикатор состояния или как анимация.

Dynamic Shape: Динамические Формы также представляют собой "пользовательский объект" который используется в стандартных дисплеях для представления сложных динамических данных.


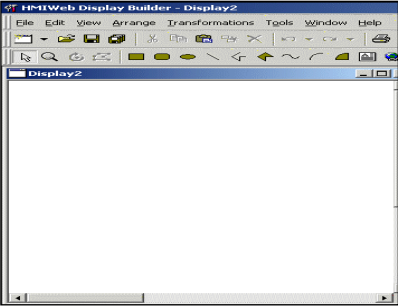
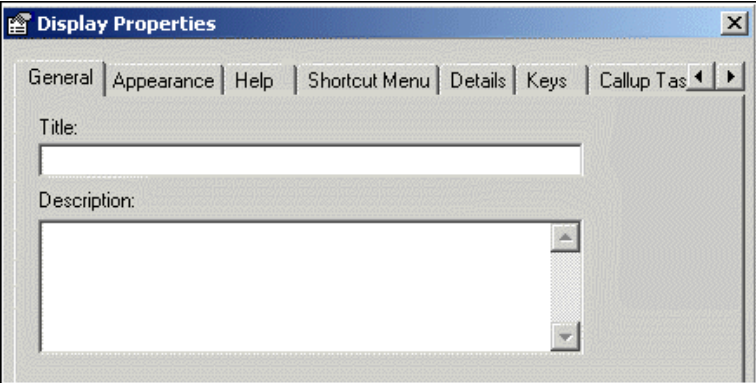
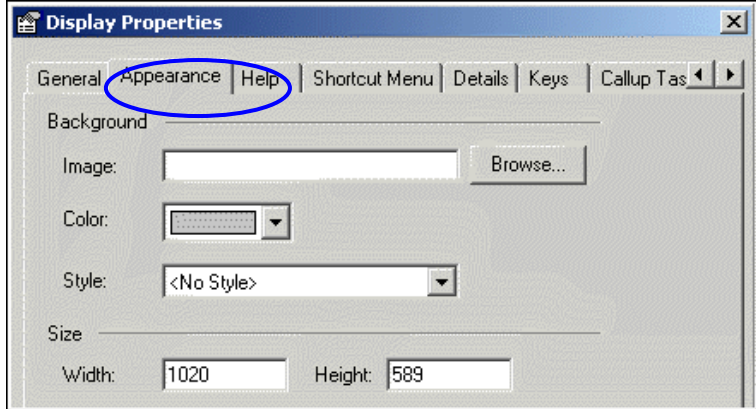
Display from Template: создание нового дисплея из шаблона для того, чтобы поддерживать визуальное постоянство. Шаблон – это стандартный дисплей, который хранится в папке Templates для HMIWeb Display Builder, для повторного использования.


6. D → ПРОВОДНИК ОБЪЕКТОВ

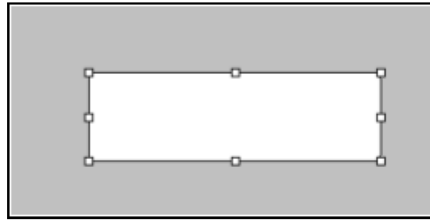
Проводник Объектов – это средство, используемое для управления и редактирования объектов. Оно показывает каждый объект на текущем дисплее и, в случае сгруппированных объектов, показывает их перекрестные связи.



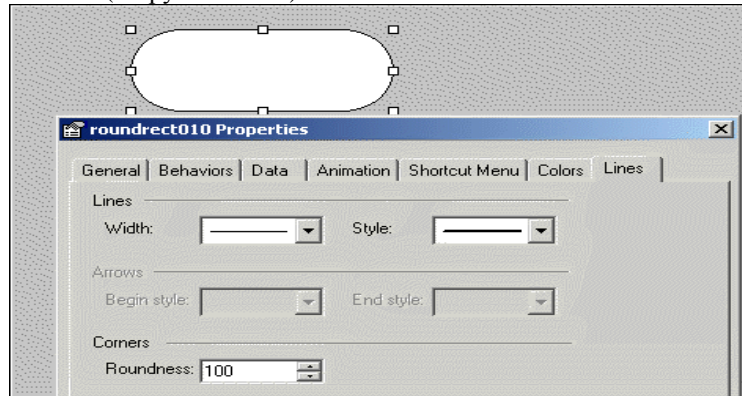
Практическое занятие
Построение Статических Объектов Дисплея

✓	Шаг	Действие
		<p>Откройте новый дисплей в HMI Web Display Builder:</p> <p>Нажмите кнопку  New Display (Новый Дисплей).</p> <ul style="list-style-type: none"> Откроется пустой дисплей, как показано ниже.  <p>Примечание: Каждому новому дисплею присваивается, по умолчанию, имя (Display#).</p>
		<p>Установка цвета фона и размера дисплея:</p> <p>Дважды кликните на дисплей, для вызова Properties Window (Окно Свойств) как показано ниже.</p>  <p>Выберите вкладку Appearance (Внешний Вид).</p> 
		<p>Установите размер дисплея 1276 x 894 (данный размер выбран как основной в конфигурации). Измените цвет фона по умолчанию на цвет silver (серебряный). Оставьте поля image (отображение) и style (стиль) пустыми. Закройте Properties Window (Окно Свойств).</p>

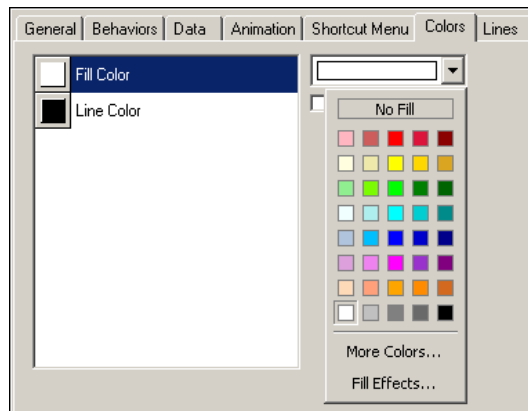
Выберите Rounded Rectangle (Скругленный Прямоугольник) на панели инструментов . Перетащите курсор диагонально по дисплею, до нужного размера объекта, который вы создаете.


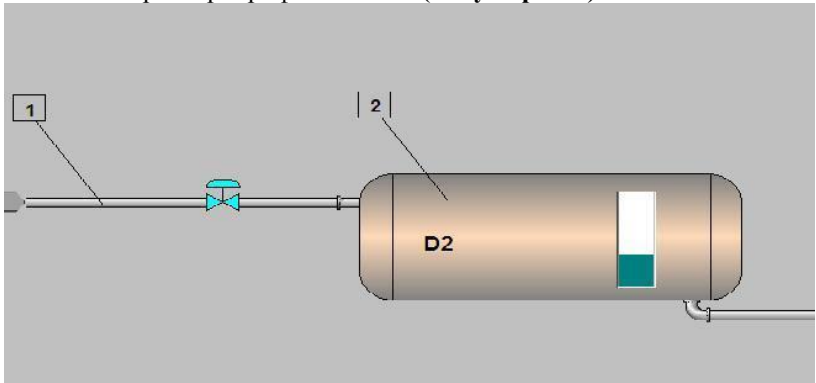



Откройте Properties window (Окно свойств) для объекта rounded rectangle (скругленного прямоугольника). Выберите вкладку Lines (Линии). Измените roundness (закругленность) на **100**.



Выберите вкладку **Colors (Цвета)**. Нажмите на стрелку выпадающего меню для изменения **Fill Effects (Эффекты заливки)**.



		<p>Создание трубопровода, клапана и емкост D2:</p> <p>Выберите кнопку  Textbox (Текст) на панели инструментов. Начертите текстовый объект в середине прямоугольника, как показано ниже. Замените слово “Text” на слово D2. Установить размер шрифта 8 и bold (полужирный).</p>  <p>Сохраните ваш дисплей под именем Lab1.htm в папке “abstract”, (путь: C:\Program Files\Honeywell\Experion PKS\Client\Abstract)</p>
		<p>Выберите Edit > Insert shape, и выберите шайпы LoopLabPid.sha, KlapLab.sha и IndiLab.sha и вставьте их на дисплей. Ваш дисплей должен выглядеть примерно так, как показано ниже.</p> 

2. Введение в Control Builder.

- Control Builder – средство инжиниринга системы Experion PKS.
- Control Builder можно запускать на Сервере или на Инженерной Рабочей Станции. Разрешается подключать к Серверу Experion PKS одновременно максимум четыре клиента Control Builder.
- Устройства, так же как стратегии управления, создаются в Control Builder и загружаются в Контроллер C200 и Сервер Experion PKS.
- Для входа в Control Builder требуется пользователь:
 - Профиль входа в систему, который использует Control Builder, - один из профилей Оператора, который использует Сервер Experion PKS.
- Предъявление пароля пользователем содержит уровень доступа пользователя, например, LVL1, LVL2, OPER, SUPV, ENGR или MNGR.
 - LVL1 и LVL2 → только Просмотр
 - OPER и SUPV → Позволяет производить операции
 - ENGR и MNGR → Позволяет конфигурировать
- Профиль Оператора Experion PKS может использовать идентификатор и пароль учетной записи Windows (Интегрированные Учетные Записи).

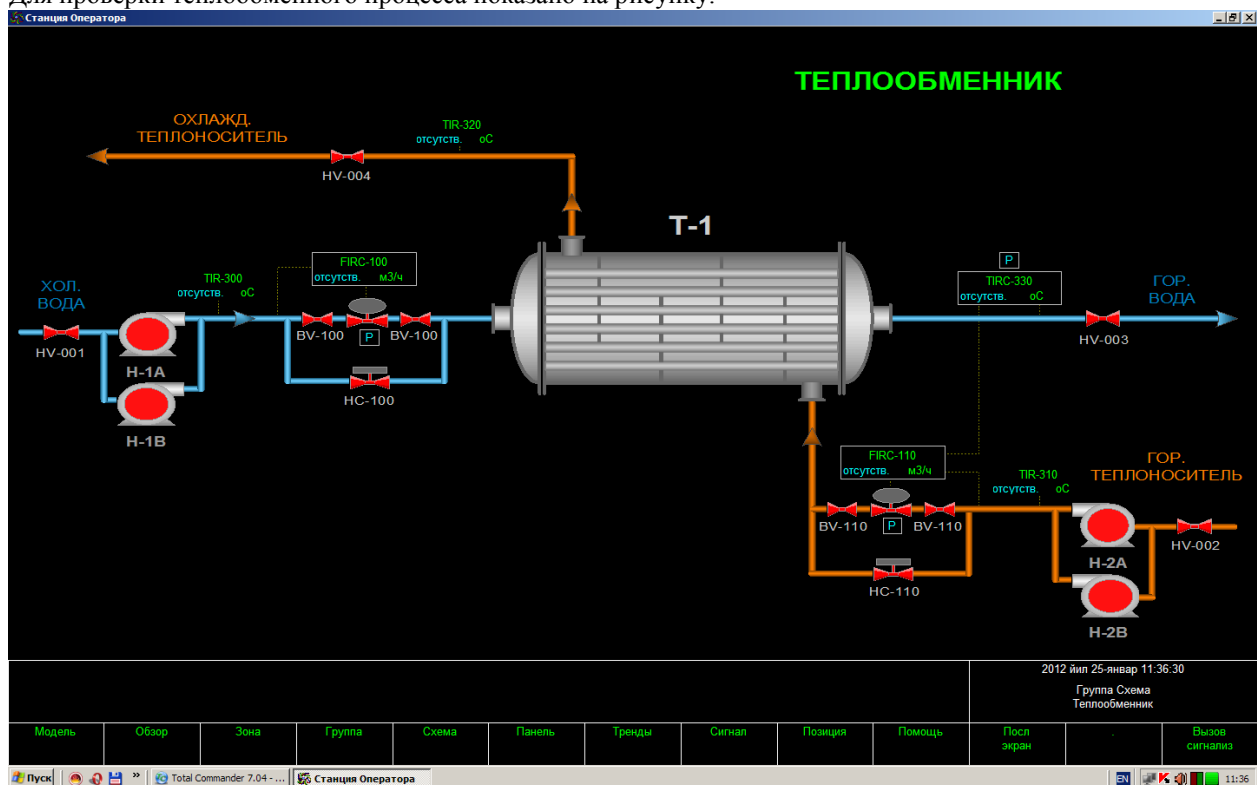
Цель задачи: Принцип работ тренажерного комплекса.

Основная часть: Внедрение современной АСУТП не может быть полноценным без организации тренажёрной подготовки эксплуатационного персонала. При этом наибольший эффект по обучению и сокращению затрат на освоение персоналом новой техники может быть достигнут, если тренажёр будет введен в действие ещё до начала внедрения реальной АСУТП на объекте.

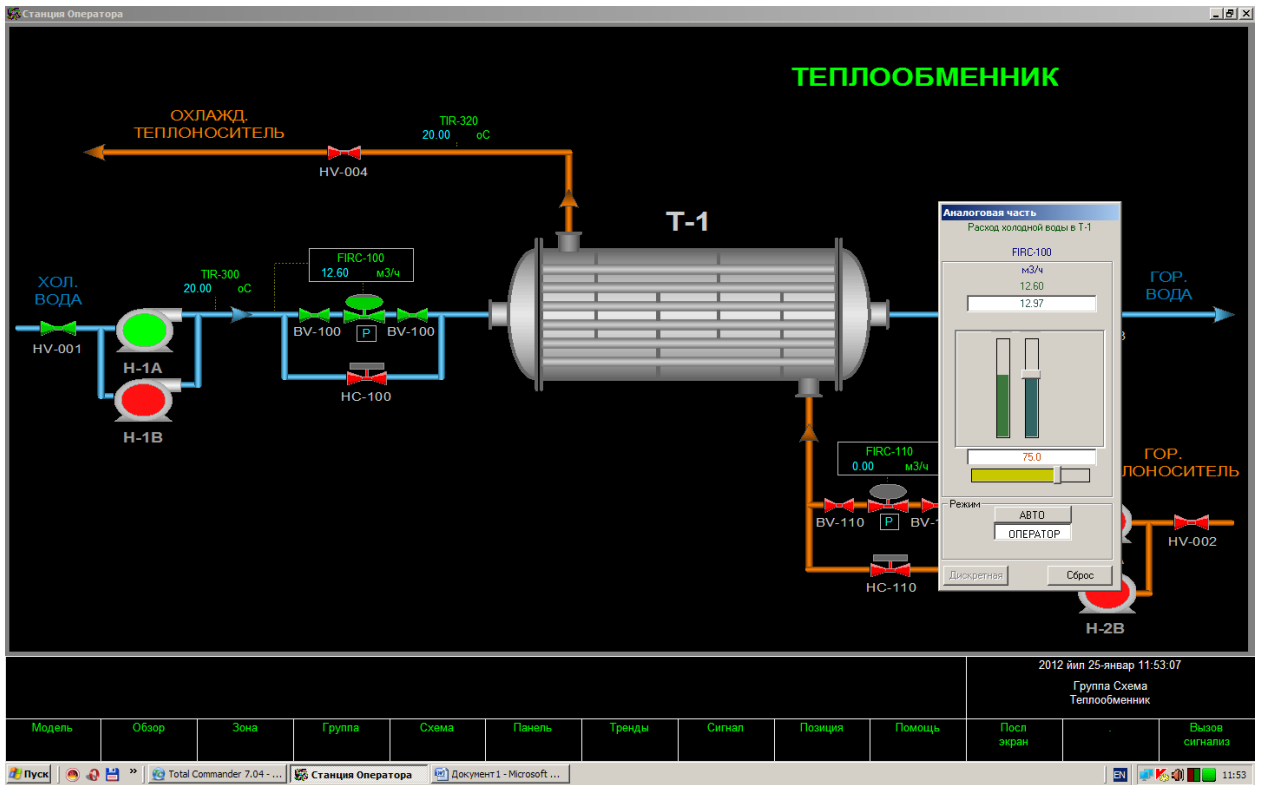
Компьютерный тренажёр состоит из следующих составных частей: математической модели объекта управления (например, энергоблока), модели системы управления и специального программного обеспечения инструктора, обеспечивающего организацию тренировочного процесса.

Совершенно очевидно, что оптимальным решением для современных тренажёров является отказ от создания имитационной модели системы управления и замена её реальной АСУТП с использованием как технических средств, так и программного обеспечения, полностью соответствующих АСУТП энергоблока-прототипа.

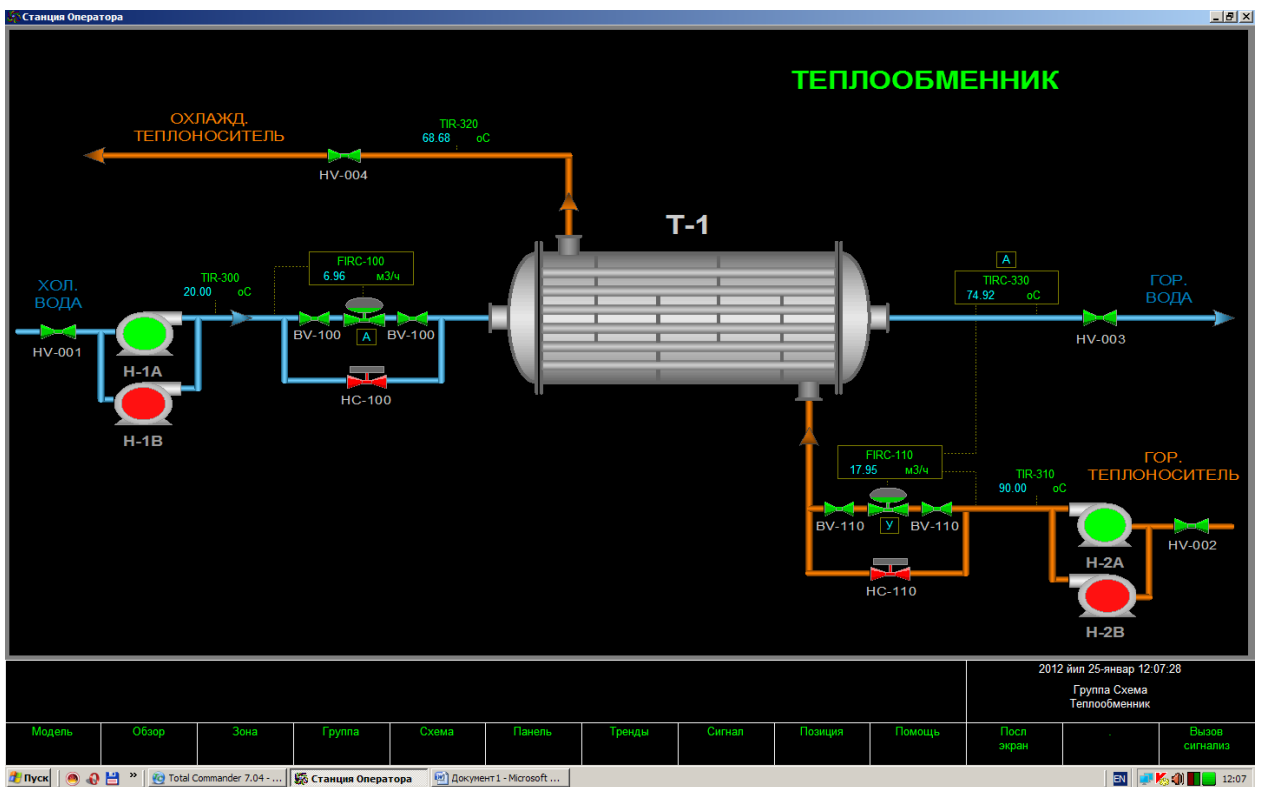
КТК-М тренажерного комплекса состоит из двух частей экрана 1 экран оператора 2 экран инструктору. Для проверки теплообменного процесса показано на рисунку.



Панель оператора.



Исправит ошибок процесса.



Процесс в норме

**Практическая работа.
Логические компоненты STEP7**

Цель работы: Ознакомиться с основами программирования на языках STEP7.

1. Краткие теоретические сведения

Для решения задачи автоматизации необходимо использовать языки программирования STEP 7. В них:

- Входные переменные обозначаются I(E). Адресация по байтам (IB 0...127) – это переменные которые находятся в оперативной памяти контроллера и называются областью отображения входных переменных.
- Выходные переменные Q(A). QB (0...127) – это область оперативной памяти, в которую записываются значения выходных сигналов периферийных модулей, перед тем как они переносятся в модуль вывода DC.
- Дискретные переменные F(M). FB (0...255) – находятся во внутренней памяти процессора.

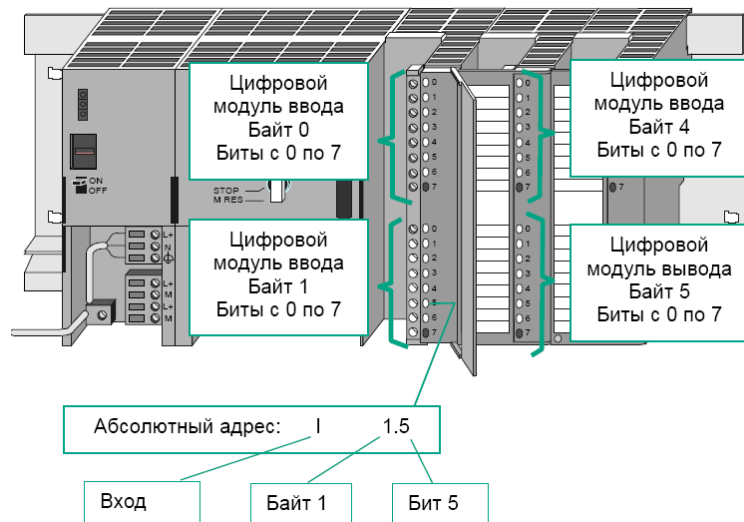


Рис. 1.
контроллерах

**Адресация в
S7-300**

В STEP 7 создаются на

программы S7
стандартных языках

программирования: контактный план (LAD), список операторов (STL) или функциональный план (FBD).

1.1. Элементарные операции

Элементарные операции STL:

A – логическое «И»,

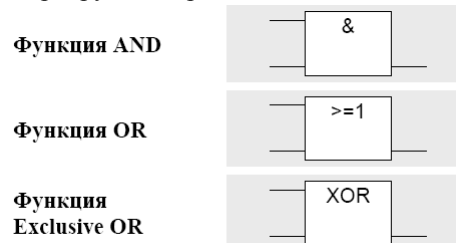
O – логическое «ИЛИ»,

X – логическое «исключающее ИЛИ»,

C инверсией соответственно AN, ON, XN, то есть N – инверсия.

Элементарные операции бинарной логики

FBD использует бинарные функции AND (И), OR (ИЛИ) и Exclusive OR (Исключающее ИЛИ). Все функции могут иметь (теоретически) любое количество функциональных входов (входов функции). Если вход ведет напрямую к функциональному элементу, то сигнальное состояние сканируемого операнда непосредственно используется в логической операции; если вход снабжен знаком отрицания (кружок), то сигнальное состояние сканируемого операнда инвертируется перед выполнением логической операции (см. ниже).



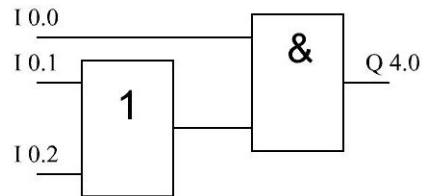
Кружок на входе или выходе символа функции инвертирует результат логической операции. Вы можете использовать инвертирование (отрицание).

Пример 1:

Программирование простейших логических операций операция «И» перед «ИЛИ»

Постановка задачи:

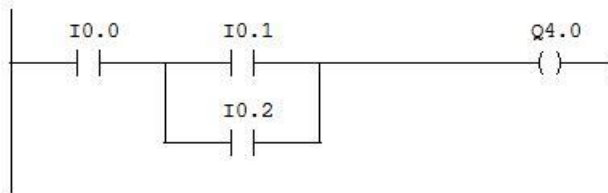
Написать в STL и в LAD следующую схему управления:



Решение (STL):

```
O    I    0.1    // опросить входы I0.1 и I0.2
O    I    0.2    // для операции «ИЛИ»
A    I    0.0    // опрос входа I0.0 для операции «И»
=    Q    4.0    // установить в 1 выход Q4.0
```

Решение (LAD):

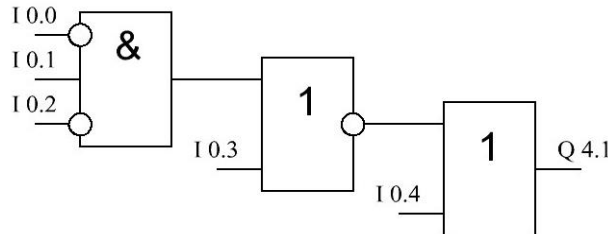


Пример 2:

Прямой и инверсный опрос, операция присваивания, инверсия флага логического результата

Постановка задачи:

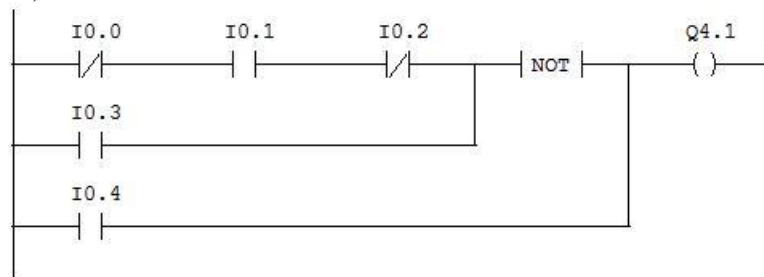
Написать в STL и в LAD следующую схему управления:



Решение (STL):

```
AN   I    0.0    // опросить инверсный вход I0.0 для операции «И»
A    I    0.1    // опросить вход I0.1
AN   I    0.2    // опросить инверсный вход I0.2
ON   I    0.3    // опрос входа I0.3 с инверсией, чтобы не создавать инверсию
                    // на выходе операции «ИЛИ»
O    I    0.4    // опрос входа I0.4 для операции «ИЛИ»
=    Q    4.1    // установить в 1 выход Q4.1
```

Решение (LAD):

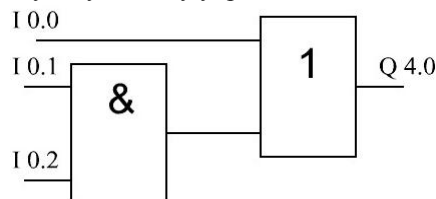


Задачи для самостоятельного решения:

Задача 1: Программирование простейших логических операций

Постановка задачи:

Написать в STL и в LAD следующую схему управления:

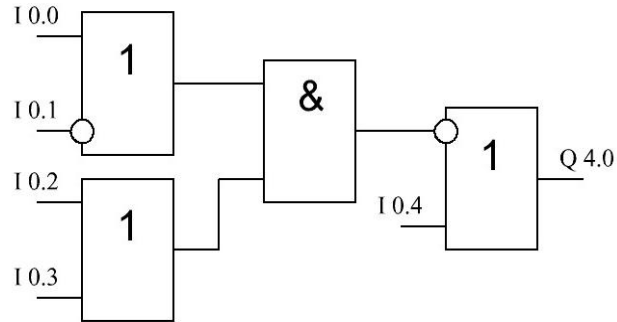


Посмотреть работу в отладчике.

Задача 2: Прямой и инверсный опрос, операция присваивания, инверсия флага логического результата

Постановка задачи:

Написать в STL и в LAD следующую схему управления:



Посмотреть работу в отладчике.

1.2. Операции с памятью

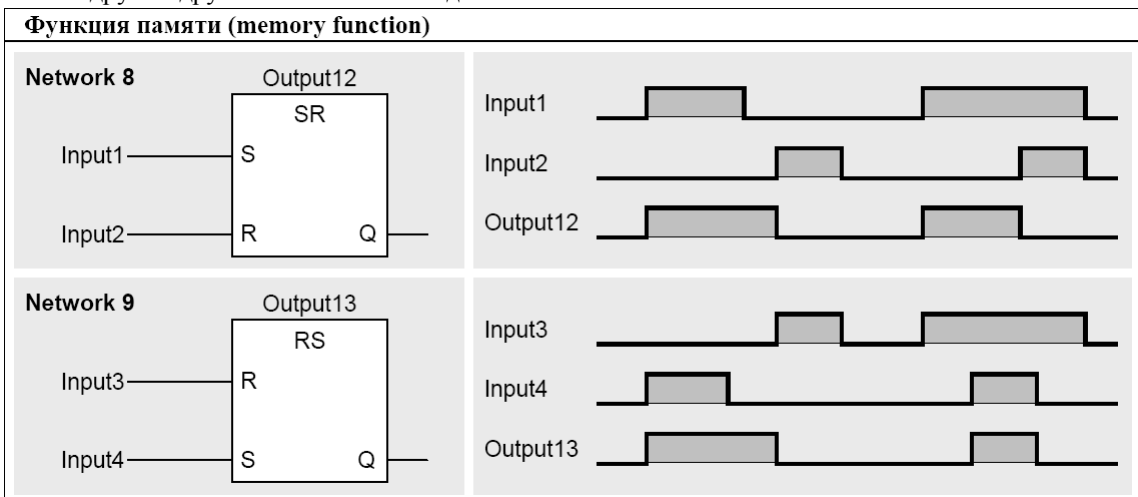
Область, называемая памятью меркеров, содержит объекты, рассматриваемые как «вспомогательные контакторы» контроллера. Память меркеров используется главным образом для хранения бинарных сигнальных состояний. Память меркеров расположена в области системной памяти CPU и поэтому доступна в любой момент времени. Количество битов в меркерной памяти определяется CPU. Память меркеров используется для хранения промежуточных результатов, которые имеют силу вне границ блока и обрабатываются более чем в одном блоке. Для хранения промежуточных результатов доступны:

- Временные локальные данные (temporary local data), доступные во всех блоках, но действующие для вызова только текущего блока;
- Статические локальные данные (static local data), которые доступны только в функциональных блоках, но имеют силу для вызовов множества блоков.

Возможности доступа:

M10.6	бит
MB4	байт
MW20	слово
MD36	двойное слово

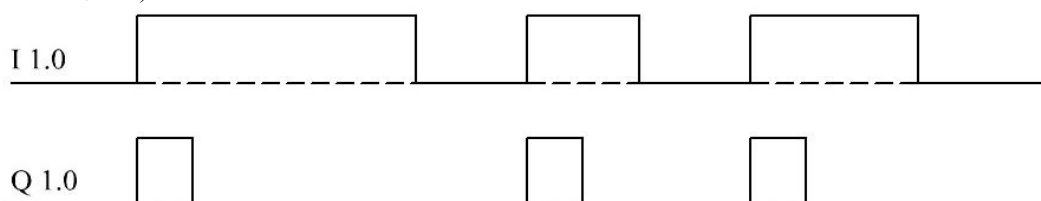
Предоставляются два варианта функции, работающих с памятью: в виде блочного элемента SR (приоритет сброса) и в виде блочного элемента RS (приоритет установки). Кроме обозначения элементы также отличаются друг от друга компоновкой входов S и R.



Пример 3: Выделение положительного фронта

Постановка задачи:

Написать функцию, осуществляющую выделение положительного фронта сигнала, т.е. функцию, реализующую следующую диаграмму работы (длительность импульсов на выходе Q1.0 = длительности 1-го программного цикла)

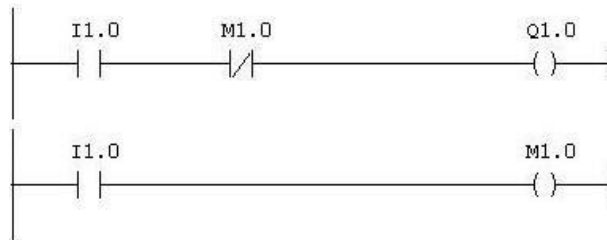


Решение (STL):

```

A    I    1.0    // проверить вход I1.0 на единицу
AN   M    1.0    // проверить меркер M1.0 на ноль
=    Q    1.0    // на Q1.0 появляется 1 на один программный цикл по
                    // положительному фронту
A    I    1.0    // проверить вход I1.0 на единицу
=    M    1.0    // записать в меркер M1.0 значения RLO
Решение (LAD):

```



Команды для выделения фронта:

1. FP – выделение переднего фронта
 FP <Бит> (Выделение положительного фронта RLO) определяет нарастающий фронт при смене состояния RLO с "0" на "1", что отображается с помощью RLO = 1.
2. FN – выделение заднего фронта
 FN <Бит> (Выделение отрицательного фронта RLO) определяет падающий фронт при смене состояния RLO с "1" на "0", что отображается с помощью RLO = 1.

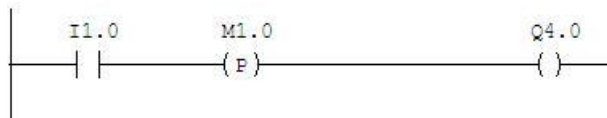
Пример 4: Выделение положительного фронта с использованием команды FP

При определении контроллером положительного фронта на входе I1.0, будет активирован выход Q4.0 в течение одного программного цикла.

```

Решение (STL):
A    I    1.0
FP   M    1.0
=    Q    4.0
Решение (LAD):

```

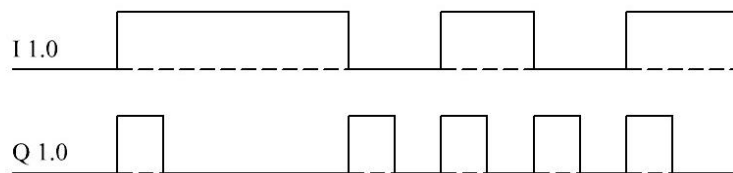


Задачи для самостоятельного решения:

Задача 3: Выделение фронтов сигналов

Постановка задачи:

Переписать программу, представленную в примере 3, осуществляющую выделение обоих фронтов сигнала. Диаграмма показана ниже.



1.3. Таймеры

Таймеры (Timers) позволяют программно реализовать последовательности синхронизации, такие как интервалы ожидания и наблюдения, измерение интервалов или генерирование импульсов. Существуют следующие типы таймеров:

- Импульсные таймеры (Pulse timers);
- Расширенные импульсные таймеры (Extended pulse timers);
- Таймеры задержки включения (On-delay timers);
- Таймеры задержки включения с запоминанием (Retentive on-delay timers);
- Таймеры задержки выключения (Off-delay timers).

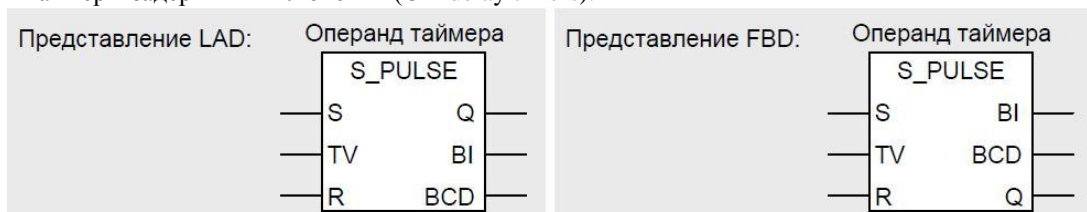


Рис. 2. Таймер в виде блочного элемента (timer box)

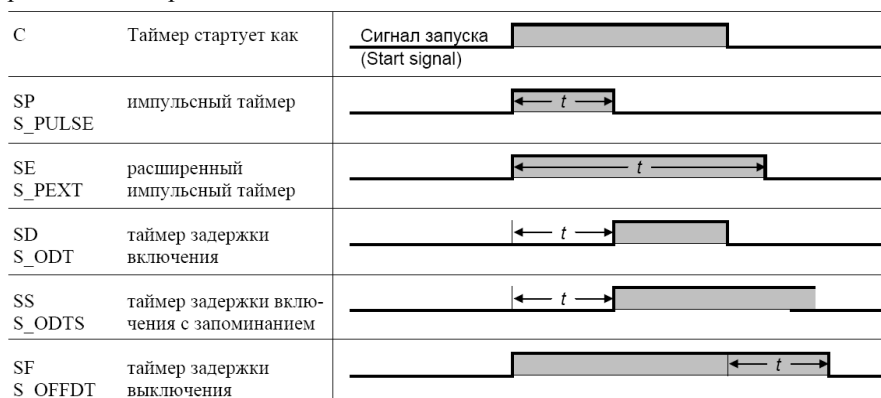
Таблица 1. Параметры таймера в виде блочного элемента

Наименование	Тип данных	Описание
S	BOOL	Вход запуска
TV	S5TIME	Спецификация длительности времени
R	BOOL	Вход сброса
BI	WORD	Текущее значение времени в двоичном представлении
BCD	WORD	Текущее значение времени в BCD-представлении
Q	BOOL	Состояние таймера

Таблица 2. Основные типы таймеров и алгоритм их работы

Таймер	Описание
SP_PULSE Таймер «Импульс»	Максимальное время в течение которого выходной сигнал остается равным 1, совпадает с заданным временем t . Выход сбрасывается раньше, если входной сигнал меняется на 0.
SE_PEXT Таймер «Импульс с памятью»	Выходной сигнал остается равным 1 в течение заданного времени независимо от того, как долго остается равным 1 входной сигнал.
SD_ODT Таймер «Задержка включения»	Выходной сигнал устанавливается в 1 только по истечении заданного времени, при этом входной сигнал все еще должен быть равен 1.
SS_ODTS Таймер «Задержка включения с памятью»	Выходной сигнал устанавливается в 1 только по истечении заданного времени независимо от того, как долго остается равным 1 входной сигнал.
SF_OFFDT Таймер «Задержка выключения»	Выходной сигнал устанавливается в 1, когда устанавливается в 1 входной сигнал, и остается равным 1, пока таймер работает. Отсчет времени начинается, когда входной сигнал меняется с 1 на 0.

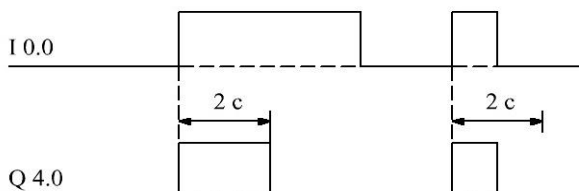
Рабочие характеристики таймеров:



Пример 5: Таймеры

Постановка задачи:

Необходимо написать программу, реализующую следующую диаграмму работы:



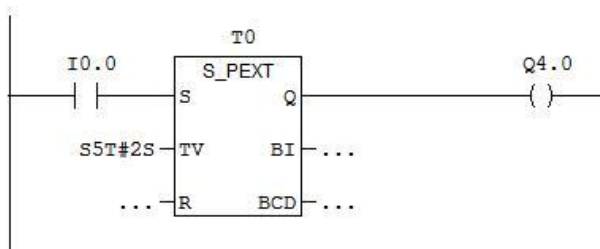
Решение (STL):

```

A    I    0.0    // опрос входа I0.0
L    S5T#2S // задать время работы 2 с
SE   T    0     // тип таймера
A    T    0     // опрос состояние таймера запущен / не запущен (не
                // первичный опрос)
=    Q    4.0   // установить в 1 выход Q4.0

```

Решение (LAD):



Задачи для самостоятельного решения:

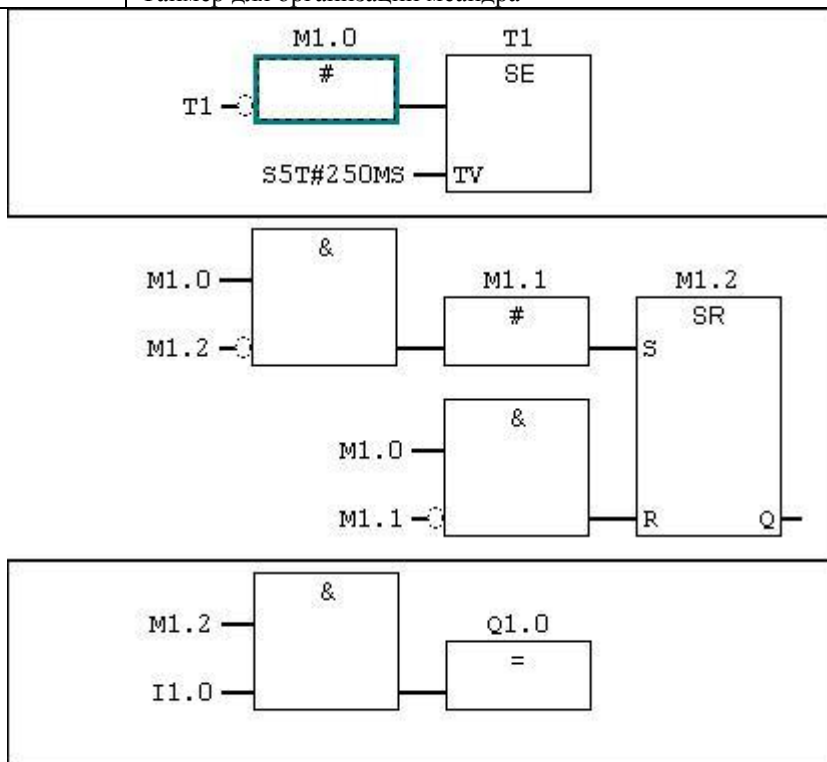
Задача 4: Формирование меандра

Постановка задачи:

Используя абсолютные адреса из таблицы 3 составить в компиляторе схему формирования меандра частотой 2 Гц, которая представлена ниже:

Таблица 3. Список переменных для решаемой задачи.

Абсолютный адрес	Комментарий к переменной
I1.0	Включение / Выключение формирования меандра
Q4.0	Сигнализация частоты меандра
M1.0	Промежуточная переменная для формирования меандра
M1.1	Промежуточная переменная для формирования меандра
M1.2	Сигнал меандра частотой 2 Гц
T1	Таймер для организации меандра



2. Методические указания

- 2.1. Для решения задачи каждого вида необходим отдельный проект.
- 2.2. Для отладки примеров задач их нет необходимости загружать в контроллер, достаточно использовать симулятор S7-PLCSIM.

3. Порядок выполнения работы

- 3.1. Выполнить представленные примеры 5 типов.
- 3.2. По аналогии с примерами, выполнить 4 задачи, предназначенные для самостоятельного решения.
- 3.3. Проверить функционирование каждой задачи, показать преподавателю, сделать скриншот решения, состояния симулятора.

4. Содержание отчета

- 4.1. Цель работы.
- 4.2. Список задач.
- 4.3. Скриншоты решения задач, состояния симулятора.

5. Контрольные вопросы

- 5.1. Какие языки, поддерживаемые STEP 7, поддерживают взаимную конвертацию текстов программ?
- 5.2. Что такое меркеры, для чего они применяются?
- 5.3. Каковы основные типы таймеров в STEP 7, чем они отличаются?
- 5.4. Какие средства предоставляет симулятор PLCSIM для контроля своих входов/выходов?
- 5.5. В каком блоке необходимо записывать программу STL для ее циклического выполнения?
- 5.6. Какие средства проверки значений переменных имеются в STEP 7?
 - 5.7. Какие операторы STL и для чего применены в Ваших программах?

Практическая работа № Знакомство с пакетом программ STEP 7 Win CC. Создание проекта

Цель работы: Ознакомится с пакетом программ STEP 7 Win CC Simatic Manager, освоить методы составления и редактирования программ, представленных бесконтактным планом.

Оборудование: Пакет программ STEP 7 Win CC Simatic Manager.

Теоретические сведения

В зависимости от степени сложности решаемых задач в контроллерах S7-300 может применяться 8 типов центральных процессоров:

- CPU 312 IFM: компактный центральный процессор с набором встроенных входов и выходов для ввода-вывода дискретных сигналов, который может быть использован в качестве автономной системы автоматизации.
- CPU 313: недорогой центральный процессор для построения небольших модульных систем автоматизации.
- CPU 314 IFM: компактный центральный процессор с набором встроенных входов и выходов для ввода-вывода дискретных и аналоговых сигналов, который может использоваться в качестве автономных систем автоматизации с малым временем реакции и выполнением специальных функций обработки сигналов.
- CPU 314: центральный процессор для построения модульных систем автоматизации с высокой скоростью обработки данных.
- CPU 315/CPU 315-2 DP: центральные процессоры для построения систем автоматизации со средним или большим объемом программы, обслуживающих системы локального и распределенного ввода-вывода, подключаемые по PROFIBUS-DP.
- CPU 316-2 DP: центральный процессор для построения модульных систем автоматизации со сложными алгоритмами обработки информации, использующих системы локального и распределенного ввода-вывода, подключаемые по PROFIBUS-DP.
- CPU 318-2 DP: центральные процессоры для выполнения программ большого объема и обслуживания разветвленных конфигураций распределенного ввода-вывода через сеть PROFIBUS-DP.

Основные технические характеристики центральных процессоров:

- Высокое быстродействие. Время выполнения логической команды в CPU 312IFM и CPU 313 составляет 600нс, в CPU 314IFM ... CPU 316 – 300нс, в CPU 318-2 – 100нс.
 - Объемы оперативной памяти, соответствующие классу решаемых задач: от 6 Кбайт в CPU 312IFM до 512 Кбайт в CPU 318-2.
 - Гибкие возможности расширения. CPU 312IFM и CPU 313 допускают подключение до 8 (однорядная конфигурация), остальные центральные процессоры – до 32 модулей (4-рядная конфигурация) ввода-вывода. Для создания программ для контроллеров S7-300 используется пакета STEP 7 Simatic Manager.
- Основными утилитами пакета STEP 7 Simatic Manager, которые доступны из папки SIMATIC - STEP 7, являются:

- SIMATIC Manager;
- LAD, STL, FDB – Programming S7;
- Memory Card Parameter Assignment;
- NetPro – Configuring Networks;
- PID Control Parameter Assignment;
- S7 SCL – Programming S7 Blocks;
- S7-GRAPH – Programming Sequential Control System;
- S7-PDIAG – Configuring Process Diagnostic;
- S7-PLCSIM Simulating Modules;
- Setting the PG-PC Interface;

SIMATIC Manager – это графический интерфейс для редактирования объектов S7 (проектов, файлов пользовательских программ, блоков, оборудования станций и инструментов). Основное окно утилиты показано на рисунке 1.

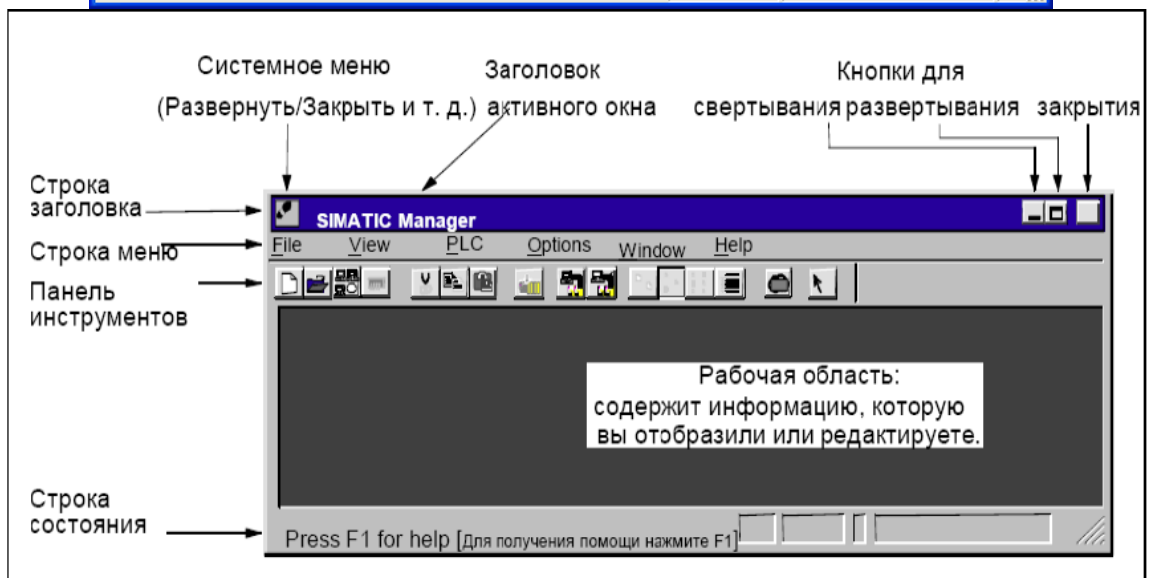
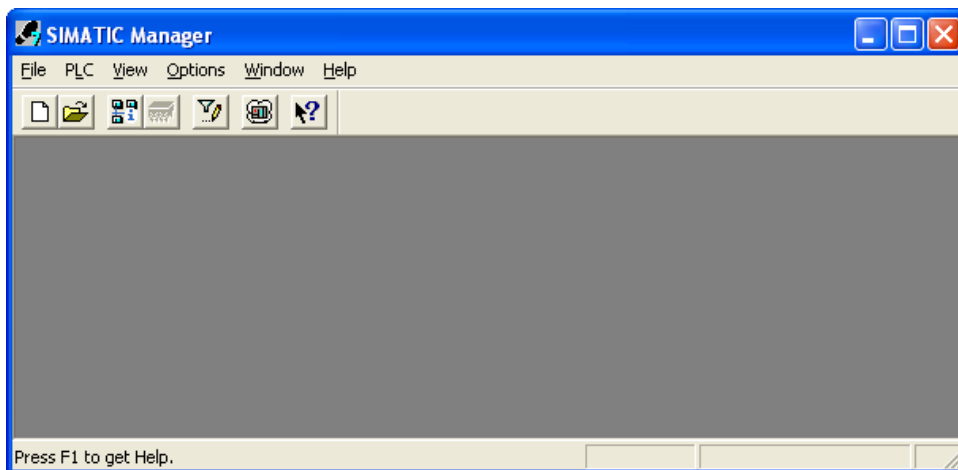


Рисунок 1 - Меню и панель инструментов SIMATIC Manager

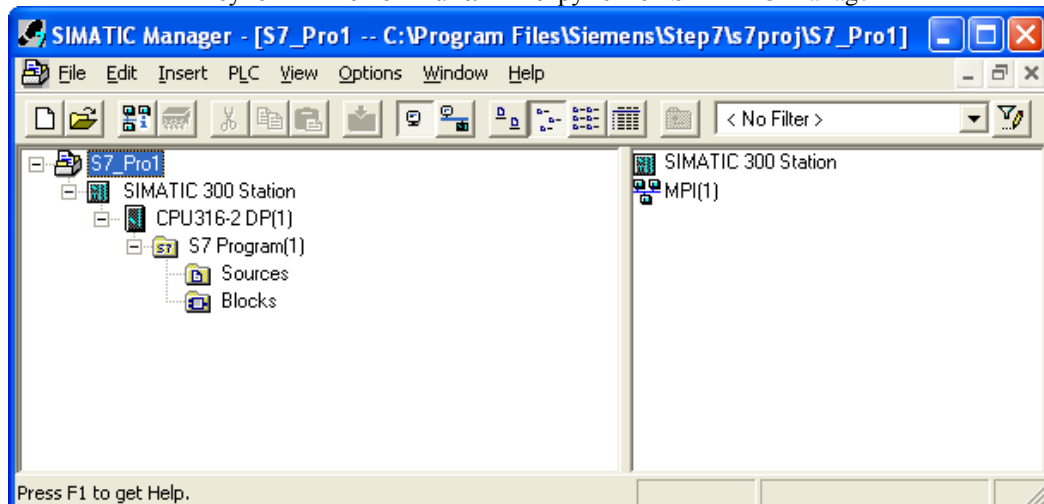


Рисунок 2 - Структура проекта в SIMATIC Manager

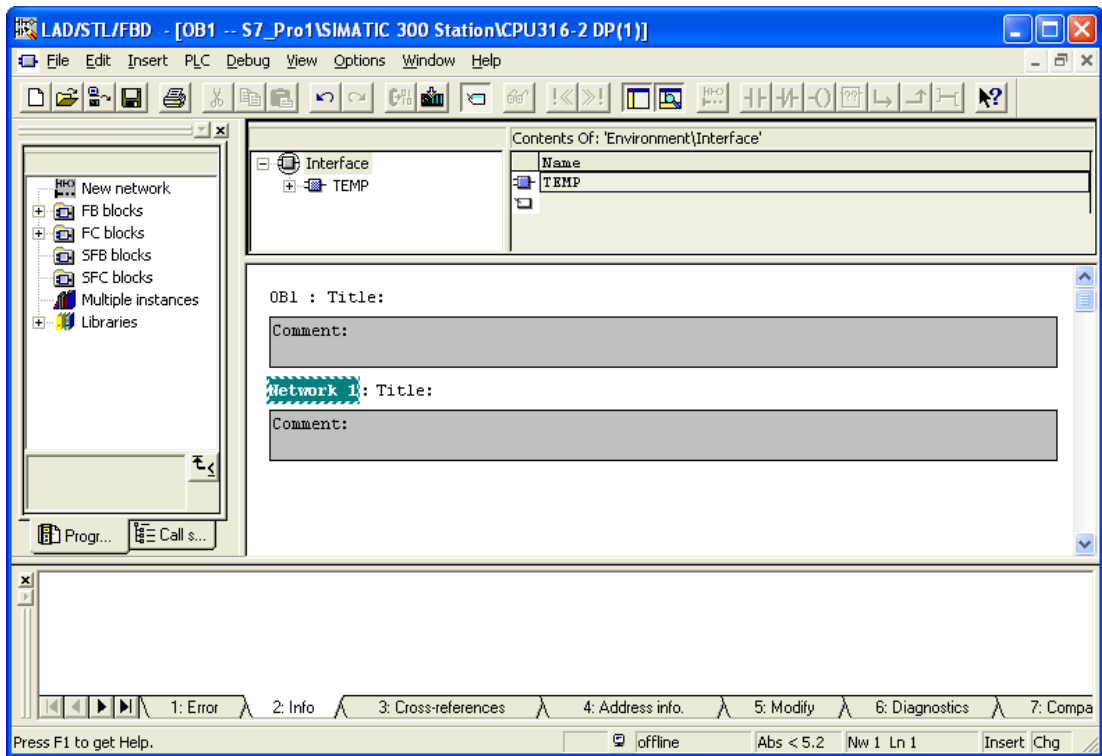
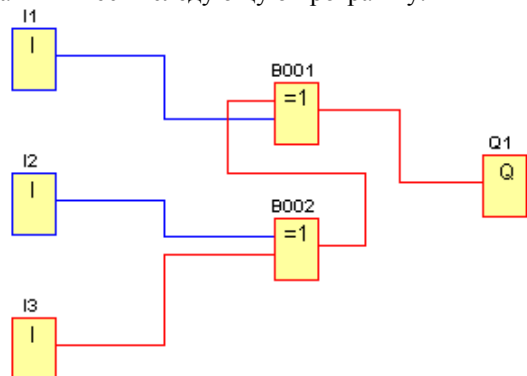


Рисунок 3 - Окно редактора LAD/STL/FBD

1. Порядок выполнения работы:

- 1.1. Загрузить программу SIMATIC Manager, создать проект для CPU 315-2 DP с одним блоком OB1 и ознакомиться с основными элементами окна программы.
- 1.2. Используя средства программы ввести следующую программу:

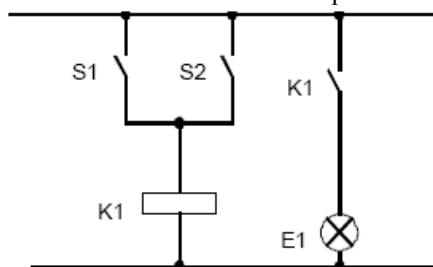


Нажав кнопку симуляция, проверить работу программы и составить таблицу истинности для схемы. Заменить элементы B001 и B002 на элементы И.

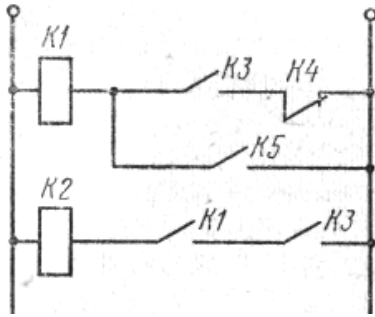
Нажав кнопку симуляция, проверить работу программы и составить таблицу истинности для схемы.

- 1.3. Используя средства программы составить программу на бесконтактных элементах, реализующие следующие алгоритмы работы:

- 1.3.1. Лампа E1 загорается при замыкании любого из выключателей S1, S2



- 1.3.2.



1.4. Используя средства программы составить программу, реализующую следующую функцию:

$$F_z = [(a + d) * b + z] * c$$

Проверить работу программы в режиме эмуляции.

2. Отчет должен содержать

- 2.1. Название работы
- 2.2. Цель работы
- 2.3. Перечень оборудования.
- 2.4. Элементы окна программы и их назначение.
- 2.5. Схемы и программы, реализующие их функцию.
- 2.6. Таблицы истинности для пункта 1.2
- 2.7. Вывод по работе.

3. Контрольные вопросы и задания.

- 3.1. Перечислите основные параметры ПЛК S7-300
- 3.2. Какова область применения ПЛК S7-300
- 3.3. Какие основные элементы окна программы вы знаете, каково их назначение.

САМОСТОЯТЕЛЬНАЯ ОБРАЗОВАНИЯ

ТЕМЫ РЕФЕРАТОВ

- Особенности применение программных пакетов в автоматизации.
- Логические элементы и их построение.
- Классификация пакетов и области их применения.
- Программный пакет MATLAB. Общие понятия и принципы.
- Элементы программирования в MATLAB.
- Построение графиков функций в системе MATLAB.
- Подпрограммы программного пакета MATLAB. Элементы подпрограммы SIMULINK.
- Моделирования систем управления в подпрограмме SIMULINK.
- Программный пакет AutoCAD. Общие понятия.
- SCADA системы. Общие понятия о SCADA системе.
- Архитектура SCADA системы.
- Программный пакет Trace Mode.
- Тренажерные системы. Общие понятия о тренажерных системах.
- Программный пакет Experion.
- Программный пакет UniSim.
- Программируемые контроллеры SIMATIC S7.

Общие понятия и принципы.личностному развитию, повышению квалификации в области пакеты применяемые в автоматизация.

- Особенности применение программных пакетов в автоматизации.
- Логические элементы и их построение.
- Классификация пакетов и области их применения.
- Программный пакет MATLAB. Общие понятия и принципы.
- Элементы программирования в MATLAB.
- Построение графиков функций в системе MATLAB.
- Подпрограммы программного пакета MATLAB. Элементы подпрограммы SIMULINK.
- Моделирования систем управления в подпрограмме SIMULINK.
- Программный пакет AutoCAD. Общие понятия.
- SCADA системы. Общие понятия о SCADA системе.
- Архитектура SCADA системы.
- Программный пакет Trace Mode.
- Тренажерные системы. Общие понятия о тренажерных системах.
- Программный пакет Experion.
- Программный пакет UniSim.
- Программируемые контроллеры SIMATIC S7.

Общие понятия и принципы.личностному развитию, повышению квалификации в области пакеты применяемые в автоматизация.

СБОРНИК ЗАДАЧИ

Составьте таблицы истинности для следующих формул:

№	Логическая формула
19.	$x \wedge (z \leftrightarrow y \vee z) \rightarrow y \rightarrow x \wedge y$
20.	$(y \wedge z \wedge y) \vee x \leftrightarrow z \vee x \rightarrow y$
21.	$z \wedge (x \leftrightarrow y \vee x) \rightarrow z \wedge x \vee y$
22.	$x \wedge z \leftrightarrow y \vee (z \rightarrow y \rightarrow x)$
23.	$y \wedge (z \wedge y) \vee x \leftrightarrow x \wedge z$
24.	$y \wedge x \leftrightarrow y \vee x \rightarrow z \wedge x \wedge y$
25.	$z \wedge y \rightarrow x \leftrightarrow y \vee (x \rightarrow z \wedge x)$
26.	$y \wedge (z \leftrightarrow y \vee x) \rightarrow z \rightarrow x$
27.	$(x \wedge x \wedge y) \vee z \leftrightarrow z \vee y \rightarrow x$

Составьте таблицы истинности и логические схемы для следующих формул:

№	Логическая формула
19.	$x \wedge (z \wedge y \vee z) \vee y \vee x \wedge y$
20.	$(y \wedge z \wedge y) \vee x \wedge z \vee x \wedge y$
21.	$z \wedge (x \wedge y \vee x) \wedge z \wedge x \vee y$
22.	$x \wedge z \wedge y \vee (z \vee y \wedge x)$
23.	$y \wedge (z \wedge y) \vee x \wedge x \wedge z$
24.	$y \wedge x \vee y \vee x \wedge z \wedge x \wedge y$
25.	$z \wedge y \vee x \wedge y \vee (x \vee z \wedge x)$
26.	$y \wedge (z \vee y \vee x) \wedge z \wedge x$
27.	$(x \wedge x \wedge y) \vee z \wedge z \vee y \vee x$

Задачи:

$$\begin{aligned}
 x + y - z &= 36 \\
 x + z - y &= 13 \\
 y + z - x &= 7 \\
 x + 2y + z &= 4 \\
 3x - 5y + 3z &= 1 \\
 2x + 7y - z &= 8 \\
 2x - 4y + 9z &= 28 \\
 7x + 3y - 6z &= -1 \\
 7x + 9y - 9z &= 5 \\
 2x + y &= 5 \\
 x + 3z &= 16 \\
 5y - z &= 10 \\
 x + y + z &= 36 \\
 2x - 3z &= -17 \\
 6x - 5z &= 7 \\
 7x + 2y + 3z &= 15 \\
 5x - 3y + 2z &= 15 \\
 10x - 11y + 5z &= 36 \\
 x + 2y - 4z &= 1 \\
 2x + y - 5z &= -1 \\
 x - y - z &= -2 \\
 2x - y + z &= -2 \\
 x + y + 3z &= -1 \\
 x - 3y - 2z &= 3 \\
 2x - 3y + z &= -1 \\
 5x + 2y - z &= 0 \\
 x - y + 2z &= 3 \\
 3x + 2y + z &= 20 \\
 4x + 0 * y - 10z &= -10 \\
 -x - 2y + 2z &= -1
 \end{aligned}$$

1. Построить графики функции с помощью декартовой системы координат:

1. $f(x) = x^2$; $g(x) = x^3$; $x \in [-1, 1]$.
2. $u(x) = x^4$; $v(x) = x^5$; $x \in [-1, 1]$.
3. $f(x) = \arcsin x$; $g(x) = \arccos x$; $x \in [-1, 1]$.
4. $u(x) = \operatorname{arctg} x$; $v(x) = \operatorname{arctg} 3x$; $x \in [-1, 1]$.
5. $f(x) = \operatorname{sh} x$; $g(x) = \operatorname{ch} x$; $x \in [-1, 1]$.
6. $u(x) = e^x$; $v(x) = e^{-x}$; $x \in [-0.6, 0.6]$.
7. $f(x) = \frac{\sin x}{x}$; $g(x) = e^{-x} \cos x$; $x \in [0, 0.1, 2\pi]$.
8. $u(x) = \sin(\ln(x+1))$; $v(x) = \cos(\ln(x+1))$; $x \in [0, 2\pi]$.
9. $f(x) = x^x$; $g(x) = x^{x^x}$; $x \in [0.1, 1]$.
10. $u(x) = \frac{1}{1+x}$; $v(x) = \frac{1}{1+\frac{1}{1+x}}$; $x \in [0, 1]$.

2. Построить графики функции с помощью декартовой системы координат:

$$f(x) = \begin{cases} |x|, & -2 \leq x \leq 1 \\ \sin \frac{\pi}{2} x, & 1 < x \leq 2 \\ (2-x)^3, & 2 < x \leq 3 \end{cases} \quad 2. \quad f(x) = \begin{cases} (x-1)^2, & -2 \leq x \leq 1 \\ \cos \frac{\pi}{2} x, & 1 < x \leq 3 \\ 1 - e^{3-x}, & 3 < x \leq 8 \end{cases}$$

1.

$$f(x) = \begin{cases} e^x, & -2 \leq x \leq -1 \\ \frac{|x|}{e}, & -1 < x \leq 1 \\ e^{-x}, & 1 < x \leq 2 \end{cases} \quad 4. \quad f(x) = \begin{cases} e^{x+1}, & -2 \leq x \leq -1 \\ x^2, & -1 < x \leq 1 \\ (2-x)^3, & 1 < x \leq 2 \end{cases}$$

3.

$$f(x) = \begin{cases} x^2 \log_2 x, & 1 \leq x \leq 2 \\ x^3/2, & 2 < x \leq 3 \\ x^x/2, & 3 < x \leq 3.5 \end{cases}$$

5.

3. Построить графики в пространстве следующие функции:

- 1) $z(x, y) = (\sin x^2 + \cos y^2)^{xy}$ $x \in [-1, 1]$ $y \in [-1, 1]$
- 2) $z(x, y) = \arctan(x+y)(\arccos x + \arcsin y)$ $x \in [-1, 1]$ $y \in [-1, 1]$
- 3) $z(x, y) = (1+xy)(3-x)(4-y)$ $x \in [0, 3]$ $y \in [0, 4]$
- 4) $z(x, y) = e^{-|x|} (x^5 + y^4) \sin(xy)$ $x \in [-2, 2]$ $y \in [-3, 3]$
- 5) $z(x, y) = (y^2 - 3) \sin \frac{x}{|y|+1}$ $x \in [-2\pi, 2\pi]$ $y \in [-3, 3]$

ГЛОССАРИЙ

Automation Автоматизация Автоматлаштириш	это применение в производстве технических средств, методов и систем управления, освобождающих человека от непосредственного участия в производстве.	this application in the production of technical means, methods and control systems that relieve a person from direct participation in production.	Ushbu qo'llanmani ishlab chiqarishda bevosita ishtirok etishdan ozod qiluvchi texnik vositalar, usullar va nazorat qilish tizimlarini ishlab chiqarish.
CAD САПР АЛС	(англ. CAD, Computer-Aided Design) - программный пакет, предназначенный для проектирования (разработки) объектов производства (или строительства), а также оформления конструкторской и/или технологической документации.	(English CAD, Computer-Aided Design) - a software package designed for the design (development) of production facilities (or construction), as well as design and / or technological documentation.	(Ingliz CAD, Computer Assisted Design) - ishlab chiqarish ob'ektlarini (yoki qurilishi), shuningdek loyihalashtirish va / yoki texnologik hujjatlarni loyihalash uchun ishlab chiqilgan dasturiy ta'minot to'plami.
MATLAB MATLAB MATLAB	(англ. Matrix Laboratory) – среда высокой производительности, предназначенная для технических вычислений, их интегрирования, визуализации и программирование в удобной для использования среде, где задачи и решения выражаются в привычной математической нотации.	(English Matrix Laboratory) is a high performance environment designed for technical computing, integrating, visualizing and programming in an easy-to-use environment where tasks and solutions are expressed in familiar mathematical notation.	(Ingliz Matritsa laboratoriyasi) texnik hisoblash, integratsiya qilish, tasavvur qilish va programmalarga moslashtirilgan qulay muhitda dasturlash uchun mo'ljallangan yuqori ishlash muhiti bo'lib, u erda vazifalar va echimlar tanish matematik notada ifodalangan.
SCADA SCADA SCADA	сокр. от Supervisory Control And Data Acquisition диспетчерское управление и сбор данных, (название класса систем для комплексной автоматизации промышленного производства)	abbr. from Supervisory Control And Data Acquisition dispatching management and data collection, (name of a class of systems for integrated automation of industrial production)	Abbr. Kuzatuv nazorati va ma'lumotlarni to'plash dispetcherlik boshqaruvi va ma'lumotlar yig'ishdan (sanoat ishlab chiqarishni kompleks avtomatlashtirish uchun tizimlar sinfining nomi)
TRACE MODE TRACE MODE TRACE MODE	первая интегрированная информационная система для управления промышленным производством, объединяющая в едином целом продукты класса SOFTLOGIC-SCADA / HMI-MES-EAM-HRM)	the first integrated information system for industrial production management, which unites SOFTLOGIC-SCADA / HMI-MES-EAM-HRM products in a single whole)	SOFTLOGIC-SCADA / HMI-MES-EAM-HRM mahsulotlarini bir butunda birlashtirgan sanoatni ishlab chiqarishni boshqarish bo'yicha birinchi integratsiya axborot tizimi
SIMATIC WinCC SIMATIC WinCC SIMATIC WinCC	(Windows Control Center) - это компьютерная система человеко-машинного интерфейса, работающая под управлением операционных систем Windows и предоставляющая широкие функциональные возможности для построения систем управления различного назначения и уровней автоматизации	(Windows Control Center) is a computer-based human-machine interface system running under Windows operating systems and providing extensive functionality for building management systems for various purposes and automation levels	(Windows Boshqarish Markazlari) Windows operatsion tizimlari ostida ishlaydigan va turli maqsadlar uchun boshqaruv tizimlari va avtomatlashtirish darajalari uchun keng funktsionallikni ta'minlaydigan kompyuterga asoslangan inson-mashina interfeysi tizimidir
Automated Training System Автоматизированная Обучающая Система (АОС) Автоматлаштирилган укув тизимлари	программное средство профессиональной подготовки персонала, состоящее из одного или нескольких автоматизированных учебных курсов (АУК) и набора специализированных локальных	software for professional training of personnel consisting of one or more automated training courses (AUC) and a set of specialized local simulators that allow the formation of professional skills	avtomatlashtirilgan o'quv kurslarida (AUC) va professional ko'nikmalarini qabul qilinishi va boshqaruv yechimlari (xizmat) ob'ekt amalga oshirish shakllantirish uchun imkon

	тренажеров, позволяющих осуществлять формирование профессиональных навыков и умений принятия и выполнения решений по управлению (обслуживанию) объектов, рассматриваемых в содержательной части АУК.	and skills in the adoption and implementation of decisions on management (maintenance) of objects considered in the content part of the AUC.	ixtisoslashgan mahalliy murabbiylar bir qator bir yoki undan ortiq iborat kasbiy ta'lim, dasturiy vositalari, AUC mazmunini ko'rib chiqildi.
Automated Training Course Автоматизированный Учебный Курс (АУК) Автоматлаштирилган уқув курси	программное средство профессиональной подготовки персонала, отвечающее требованиям методики подготовки, реализующее предъявление обучаемому графического и текстового материала нормативно-технической документации конкретного учебного курса и обеспечивающее контроль качества подготовки обучаемых	software for professional training of personnel that meets the requirements of training methods that implements the presentation of the graphic and text material to the trainee of the normative and technical documentation of the specific training course and ensures the quality control of the trainees	Kadrlarni malakali o'qitish uchun maxsus o'quv kursining normativ-texnik hujjatlarini tinglovchilarga grafik va matn materiallarini taqdim etishni o'rgatadigan o'qitish metodlari talablariga javob beradigan dasturiy ta'minot va malaka oshirish kurslarining sifatini nazorat qilish
Independent trainer Автономный тренажер Автоном тренажер	тренажер оператора системы «человек-машина», функционирующий без системы «человек-машина»	The trainer of the operator of the "man-machine" system, functioning without the "man-machine" system	"man-mashina" tizimi bo'lmagan holda ishlaydigan "man-mashin" tizimi operatorining murabbiysi
Adaptive simulator Адаптивный тренажер Адаптив тренажер	тренажер оператора системы «человек-машина», обеспечивающий автоматическую оптимизацию управления процессом подготовки оператора системы «человек-машина» с учетом результатов выполнения им учебных задач	the trainer of the operator of the "man-machine" system, which provides automatic optimization of the management of the process of training the operator of the "man-machine" system, taking into account the results of the performance of the training tasks	"Man-mashina" tizimining operatorini o'qitish jarayonini boshqarishning avtomatlashtirilgan optimallashtirishni ta'minlaydigan "man-mashina" tizimining operatori o'qituvchisi, mashg'ulot vazifalarini bajarish natijalarini hisobga olgan holda
Built-in simulator Встроенный тренажер Ажралмас тренажер	тренажер оператора системы «человек-машина», функционирующий совместно с системой «человек-машина»	The simulator of the operator of the "man-machine" system, functioning in conjunction with the "man-machine" system	"inson mashinasi" tizimi bilan ishlaydigan "inson-mashina" tizimining operatori simulyatori
Group simulator Групповой тренажер Грухли тренажер	тренажер оператора системы «человек-машина», предназначенный для одновременной подготовки операторов взаимосвязанных систем «человек-машина»	the trainer of the operator of the system "man-machine", intended for simultaneous preparation of operators of the interconnected systems "person-machine"	"Man-mashina" tizimining operatorlarini bir vaqtning o'zida ishlab chiqarishga mo'ljallangan "odam-mashin" tizimining operatori,
Information model of "man-machine" system Информационная модель системы «человек-машина» «Инсон-машина» тизимининг информатик модели	условное отображение информации о состоянии объекта воздействия, системы «человек-машина» и способов управления ими	Conditional display of information on the state of the impact object, the "man-machine" system and the ways to manage them	ta'sir ob'ekti holati, "odam-mashina" tizimi va ularni boshqarish usullari to'g'risida shartli ma'lumotlarni ko'rsatish
The reliability of the system operator "man-machine" Надежность оператора системы «человек-машина» «Инсон-машина» оператор тизимининг	свойство человека- оператора системы «человек-машина» сохранять работоспособное состояние в течение требуемого интервала времени	The property of the person-operator of the "man-machine" system to maintain an operational state for the required time interval	"Man-mashina" tizimining shaxs-operatorining mulki talab qilinadigan vaqt oralig'ida operatsion holatni saqlab qolish uchun

ишончлилиги			
Tension system operator "man-machine" Напряженность оператора системы «человек-машина» «Инсон-машина» тизими операторининг диккати	работоспособное состояние оператора системы «человек-машина», определяемое особенностью и интенсивностью психо-физиологических процессов, обеспечивающих выполнение деятельности оператора системы «человек- машина»	The operable state of the operator of the human-machine system, determined by the peculiarity and intensity of the psycho-physiological processes that ensure the performance of the operator of the human machine system	Inson mashinasi tizimining operatorini ishlashini ta'minlaydigan psixofizik jarayonlarning o'ziga xosligi va intensivligi bilan aniqlangan inson-mashinasozlik tizimining operator holati
Assimilation of the "man-machine" system Освояемость системы «человек-машина» «Инсон-машина» тизимининг тушунарлилиги	свойство системы «человек-машина», обуславливающее приспособленность ее технических средств и алгоритмов деятельности к освоению человеком-оператором системы «человек-машина».	the property of the "man-machine" system, which determines the adaptability of its technical means and activity algorithms to the human-machine man-operator development by the human operator.	Inson mexanizmi tizimining o'ziga xos xususiyatlarini va uning inson mexanizmi operatori tomonidan inson mexanizmi ishlab chiqarishini rivojlantirish algoritmlarini moslashuvchanligini belgilovchi xususiyatidir.
System management body "man-machine" Орган управления системы «человек-машина» «Инсон-машина» тизимининг бошқариш органлари	техническое средство в системы «человек-машина», предназначенное для передачи управляющих воздействий от оператора системы «человек-машина» к машине	technical means in the "man-machine" system, designed to transfer control actions from the operator of the "man-machine" system to the machine	"man-mashina" tizimining operatoridan mashinaga boshqaruv faoliyatini o'tkazish uchun mo'ljallangan "man-mashina" tizimidagi texnik vositalar
Error Operator "man-machine" system Ошибка оператора системы «человек-машина» «Инсон-машина» тизимида оператор хатоси	неправильное выполнение или невыполнение оператором системы «человек-машина» предписанных действий	wrong execution or failure by the operator of the "man-machine" system of prescribed actions	Belgilangan xatti-harakatlarning "man-mashinasi" tizimi operatorining noto'g'ri bajarilishi yoki bajarilmasligi
Serviceability "man-machine" system Обслуживаемость системы «человек-машина» «Инсон-машина» тизимининг хизмат курсатилиши	свойство системы «человек-машина», обуславливающее приспособленность ее технических средств к обслуживанию, ремонту и подготовке к применению человеком- оператором системы «человек-машина»	the property of the "man-machine" system, which determines the fitness of its technical means for maintenance, repair and preparation for use by the human operator of the "man-machine" system	"Inson mashinasi" tizimining insonparvarlik operatori tomonidan texnik xizmat ko'rsatish, ta'mirlash va tayyorlashga texnik vositalarining mosligini belgilaydigan "man-mashina" tizimining mulki
Remote Management System "Machine Man" Пульт Управления Системы «Человек-Машина» «Инсон-машина» тизимининг пульт бошқаруви	элемент рабочего места оператора системы «человек-машина», на котором размещены средства отображения информации и органы управления системы «человек-машина»	an element of the workplace of the operator of the "man-machine" system, on which information display means and controls of the "person-machine" system are located	"Man-mashina" tizimining axborot apparati va boshqaruvlari mavjud bo'lgan "man-mashina" tizimining operatori ish joyi elementi
Citect SCADA Citect SCADA Citect SCADA	– программный продукт, представляющий собой полнофункциональную систему визуализации и мониторинга, управления и сбора данных.	- software product, which is a full-featured system of visualization and monitoring, management and data collection.	Vizualizatsiya qilish va monitoring qilish, boshqarish va ma'lumotlar yig'ishning to'liq xususiyatli dasturiy mahsuloti.
InTouch SCADA system	– это достаточно мощная среда разработки визуализации и	Is a powerful visualization and control environment for	Texnologik jarayonlar va dispatcherlik boshqaruvini

SCADA система InTouch SCADA InTouch тизими	управления для промышленной автоматизации технологических процессов и диспетчерского контроля, применяется для создания DCS (распределенных систем управления) и других АСУ ТП.	industrial automation of technological processes and dispatch control, it is used to create DCS (distributed control systems) and other process control systems.	sanoat avtomatizatsiyasi uchun kuchli vizualizatsiya va boshqarish muhiti, DCS (tarqalgan boshqaruv tizimlari) va boshqa jarayonlarni boshqarish tizimlarini yaratish uchun ishlatiladi.
Simatic Step 7 Simatic Step 7 Simatic Step 7	— программное обеспечение фирмы Siemens для разработки систем автоматизации на основе программируемых логических контроллеров Simatic S7-300/S7-400/M7/C7 и WinAC.	- Siemens software for the development of automation systems based on the programmable logic controllers Simatic S7-300 / S7-400 / M7 / C7 and WinAC.	- Simatic S7-300 / S7-400 / M7 / C7 va WinAC dasturlashtiriladigan mantiqiy boshqaruv qurilmalari asosida avtomatlashtirish tizimlarini rivojlantirish uchun Siemens dasturiy ta'minoti.
Training apparatus Тренажёр Тренажер	(от англ. Train — воспитывать, обучать, тренировать) — механическое, электрическое либо комбинированное учебно-тренировочное устройство, искусственно имитирующее различные нагрузки или обстоятельства (ситуацию).	(From the English Train - to educate, train, train) - a mechanical, electrical or combined training device, artificially imitating various loads or circumstances (situation).	(Ingliz poezdidan o'qituvchi, poezd, poezddan) - mexanik, elektr yoki birlashtirilgan o'quv qurollari, turli xil yuklarni yoki vaziyatlarni (vaziyatni) sun'iy ravishda taqlid qilish.
System Software Системное ПО Тизимили дастурий таъминот	– совокупность программ, обеспечивающих общее управление функционированием вычислительной системы и выполнение функций по ее обслуживанию.	- a set of programs that provide overall management of the functioning of the computer system and the performance of the functions for its maintenance.	- kompyuter tizimining ishlashi ustidan umumiy nazoratni ta'minlaydigan va unga xizmat ko'rsatish funksiyalarini bajaradigan dasturlarning to'plami.
Tool Software Инструментальное ПО Ускунали дастурий таъминот	- совокупность средств, обеспечивающих процесс функционирования прикладных программ, т.е. обеспечивающих перевод кода программ, написанных пользователем с помощью высокоуровневых языком программирования на языки более низкого уровня – в машинный код.	- a set of tools that provide the process of functioning of application programs, i.e. providing the translation of the code of programs written by the user with the help of high-level programming language to languages of a lower level - into machine code.	- dastur dasturlarining ishlashini ta'minlaydigan vositalar majmuasi, ya'ni. foydalanuvchi tomonidan yuqori darajadagi dasturlash tilining past darajadagi tillarga tarjima qilingan kodlari tarjimasini mashina kodiga tarjima qilish.
Application Software Прикладное ПО Амалий дастурий таъминот	– совокупность программ, обеспечивающих использование ЭВМ при решении частных задач, необходимых конкретным группам пользователей.	- a set of programs that ensure the use of computers in solving particular problems required by specific groups of users.	- muayyan foydalanuvchilar guruhlari tomonidan muayyan muammolarni hal qilishda kompyuterdan foydalanishni ta'minlaydigan dasturlar to'plami.

ПРИЛОЖЕНИЯ

ТИПОВАЯ ПРОГРАММА

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ

Рўйхатга олинди
№ БД – 5311000 – 3.17

2016 йил « 8 » оғ



МУХАНДИСЛИК ДАСТУРЛАРИ

ФАН ДАСТУРИ

Билим соҳаси:	300 000 – Ишлаб чиқариш техник соҳа
Таълим соҳаси:	310 000 – Мухандислик иши
Таълим йўналиши:	5311000 – Технологик жараёнлар ва ишлаб чиқаришни автоматлаштириш ва бошқариш (кимё, нефть-кимё ва озиқ-овқат sanoati);

Тошкент – 2016

Ўзбекистон Республикаси Олий ва ўрта махсус таълим вазирлигининг 2016 йил « 8 » 08 даги « 3 » -сонли буйруғининг 2 -илоvasи билан фан дастури рўйхати тасдиқланган.

Фан дастури Олий ва ўрта махсус, касб-хунар таълими йўналишлари бўйича Ўқув-услубий бирлашмалар фаолиятини Мувофиқлаштирувчи Кенгашининг 2016 йил « 25 » 08 даги 355- сонли баённомаси билан маъқулланган.

Фан дастури Тошкент давлат техника университетида ишлаб чиқилди.

Тузувчилар:

- Гулямов Ш.М. - Тошкент давлат техника университети «Ишлаб чиқариш жараёнларини автоматлаштириш» кафедраси профессори, т.ф.д.;
- Мухитдинов Д.П. - Тошкент давлат техника университети «Ишлаб чиқариш жараёнларини автоматлаштириш» кафедраси профессори, т.ф.д.;
- Раджабова М.А. - Тошкент давлат техника университети «Ишлаб чиқариш жараёнларини автоматлаштириш» кафедраси доценти, т.ф.н.

Такризчилар:

- Абдуллаев М.М. - Тошкент давлат техника университети «Мехатроника ва робототехника» кафедраси мудирини, т.ф.н., доцент;
- Адилов Ф.Т. - «Химавтоматика» МЧЖ нинг бош директори, т.ф.д., профессор

Фан дастури Тошкент давлат техника университети Кенгашида кўриб чиқилган ва тавсия қилинган (2016 йил « 6 » 07 даги 12 -сонли баённома).

Фаннинг долзарблиги

Ушбу дастур саноат корхоналарида ишлатиладиган дастурий таъминотлар, уларнинг таснифи, тузилиши, фан тарихи ривожининг тенденцияси, истиқболи ҳамда республикамиздаги ижтимоий-иқтисодий ислохотлар натижаларини автоматлаштириш соҳасида ишлатиладиган дастурий пакетлар истиқболига таъсири масалаларини қамраб олади. Бу пакетларни “Мухандислик дастурлари” фани ўтади, шунинг учун ҳам бу фан долзарбдир.

Фаннинг ўқув режадаги бошқа фанлар билан ўзаро боғлиқлиги ва услубий жиҳатдан узвийлиги

Мухандислик дастурлари умумқасбий фанлардан бири ҳисобланиб, 2,3- семестрда ўқитилади. Дастурни амалга ошириш ўқув режасида режалаштирилган математика, ахборот технологиялари, физика ва бошқа қатор фанлар билан мантиқий боғлиқдир.

Фаннинг илм-фан, иқтисодиёт ва ишлаб чиқаришдаги ўрни

Ҳозирги замонавий технологиялар ривожланган даврда, саноат корхоналаридаги технологик жараёнлар автоматлаштирилган мураккаб дастурий воситалар асосида бошқарилмоқда. Шунинг учун мухандислик дастурларига алоҳида талаблар қуйилади. Дастурий пакетлар ишлаб чиқаришни автоматлаштиришда юқори сифат кўрсаткичи, юқори даражадаги аниқликда ишлашни, хавфсизликни юқори даражада таъминлашни тартибга солади. Шунинг учун ушбу фан асосий умумқасбий фанлардан бири ҳисобланиб, ишлаб чиқаришнинг ажаралмас бўғинидир.

Фанни ўқитишдаги замонавий ахборот ва педагогик технологиялар ҳамда ўқув машғулотларини лойиҳалаш

Талабалар «Мухандислик дастурлари» фанини ўзлаштиришлари учун ўқитишнинг илғор ва замонавий усулларида фойдаланиш, янги информацион-педагогик технологияларни тадбиқ қилиш муҳим аҳамиятга эгадир. Фанни ўзлаштиришда дарслик, ўқув ва услубий қўлланмалар, маъруза матнлари, тарқатма материаллар, электрон материаллар, виртуал стендлар, кейс-технологиялардан ҳамда ишлаб чиқаришдаги намуналар, имитацион моделлар ва макетлардан фойдаланилади. Маъруза, амалий ва лаборатория ишларида ўқитишнинг интерактив усуллари (визуал, муаммоли, муаллифлик маърузалари, икки томонлама таҳлил, Инсерт, кластер, “Венна”, Синквейн ва бошқалар)дан фойдаланилади.

Фан ўқитувчиси томонидан педагогик ва модулли технология тамойиллари асосида “Мухандислик дастурлари” фани ўқув машғулотларининг лойиҳалари ишлаб чиқилади.

Фан модулининг дастури (module syllabus)

Намуна

Ўқув курсининг тўлиқ номи:	Мухандислик дастурлари	
Курснинг қисқача номи:	МД	Код:МД
Кафедра:	“Ишлаб чиқариш жараёнларини автоматлаштириш”	
Ўқитувчи ҳақида маълумот:	Раджабова Махфуза Азимовна	maxfuza4717@mail.ru
Семестр ва ўқув курсининг давомийлиги	2,3-семестр, 18 ҳафтадан	
Ўқув соатлари ҳажми:	жами:	188
	шунингдек:	2-семестрда 3-семестрда
	Маъруза	36 18
	амалий машғулот	9 9
	лаборатория иши	9 27
	мустақил таълим	40 40
	Семестр бўйича жами	94 94
Ўқув курсининг статуси	Умумқасбий фанлар блоки	
Дастлабки тайёргарлик:	Курс «Ахборот технологиялари», «Математика» ва «Физика» фанларининг керакли қисмларини ўзлаштирилган билимларига асосланади.	
<p>Фаннинг предмети ва мазмуни: – талабаларга “Мухандислик дастурлари” фанининг асосий тушунчалари ва турлари, қонуниятлари ўргатиш ҳамда амалда татбиқ этиш кўникмасини ҳосил қилишнинг назарий ва амалий томонларини ўрганишга йўналтирилган.</p> <p>Фанни ўқитишдан мақсад –талабаларда автоматлаштиришда қўлланиладиган дастурлар ҳақида тушунча, билим, кўникма ва малакани шакллантириш, асосий тушунчалари ва тамойилларини ўргатиш ҳамда уларни амалда татбиқ этиш кўникмасини ҳосил қилиш.</p> <p>Фанни вазифаси - талабалар «Мухандислик дастурлари» фанини ўзлаштириш жараёнида мантикий элементларнинг тузилиши ва ишлатилиши, дастурий пакетларнинг синфланиши ва тузилиши, дастурий пакетлар ва уларнинг ишлаш принциплари <i>ҳақида тасаввурга эга бўлиши;</i></p> <ul style="list-style-type: none"> - дастурий пакетларни такомиллаштиришнинг конструктив ва технологик ҳамда алгоритмик усулларини; - дастурий пакетларда алгоритмлаш принципларини; - MATLAB тизими ҳақида тушунчаларга эга бўлиши; - автоматлаштиришда тренажер ситемаларининг синфланиши, тузилиши, қўлланилиши ва уларнинг ишлаш принципларини; - SCADA ситемаларининг ишлаш принципи ва уларни созлашни <i>билиши ва улардан фойдалана олиши;</i> - автоматлаштиришда қўлланиладиган дастурий пакетларни ишлата билиш, тренажер ситемаларидаги имитацион моделлардаги носозликларни бартараф этиш бўйича тўғри ечимлар танлаш <i>кўникмаларига эга бўлиши;</i> - SIMATIC S7 дастурланувчи контроллерлари унинг дастурий таъминоти ҳақида <i>кўникмаларига эга бўлиши;</i> - автоматлаштиришда қўлланиладиган дастурий пакетлар ва компьютерли тренажер ситемаларидаги имитацион моделлар билан ишлаш натижасида реал ишлаб чиқариш жараёнларидаги вазиятларни баҳолай олиши ва турли носозликларни бартараф эта олиш <i>малакаларига эга бўлиши керак.</i> 		

Курснинг тематик таркиби ва мазмуни

Т/р	Мавзу	Талабанинг ўқув юкламаси (соатларда)							
		Умумий юклама хажми	Аудитория машғулоти (соатлар)						Мустақил Таълим
			Жами	Маъруза	Амалий машғ.	Лаборатория иши	Семинар	Курс лойиҳаси	
1	Кириш. Автоматлаштиришда қўлланиладиган дастурлар ҳақида қисқача маълумот.	6	2	2	-	-			4
2	Мантикий элементлар ва уларнинг қурилиши.	12	8	2	2	4			4
3	Мухандислик дастурларининг синфланиши ва уларнинг қўлланилиш соҳалари.	6	2	2	-	-			4
4	MATLAB дастурий пакети. MATLAB дастурида арифметик ифодаларни киритиш.	12	8	2	2	4			4
5	MATLAB дастурлаш элементлари ва улар билан ишлаш.	12	8	2	2	4			4
6	MATLAB дастурида функциялар графикларини қуриш. М-файллар.	12	8	2	2	4			4
7	MATLAB дастурий пакетдаги қисм дастурлар. SIMULINK қисм дастуридаги элементлар.	14	8	4	2	2			6
8	SIMULINK қисм дастурида бошқариш системаларини моделлаштириш.	10	6	2	2	2			4
9	AutoCAD дастури. Умумий тушунчалар.	14	10	4	2	4			4
10	SCADA системаси. SCADA системалари тўғрисида умумий маълумот.	10	4	4	-	-			6
11	SCADA системаси архитектураси.	6	2	2	-	-			4
12	SCADA системасининг TRACE MODE дастурий пакети.	10	6	4	-	2			4
13	SCADA системасининг TRACE MODE дастурий пакетда маълумотлар алмашинуви.	12	8	4	2	2			4
14	Тренажер системалари. Тренажер системалари ҳақида умумий тушунча.	10	6	4	-	2			4
15	Expregion дастури ҳақида умумий тушунча.	12	8	4	2	2			4

16	UniSim дастури хақида умумий тушунча.	12	8	4	-	4		4
17	SIMATIC S7 дастурланадиган контроллерлари. Умумий тушунчалар.	10	4	4	-	-		6
18	SIMATIC WinCC. Умумий маълумотлар.	8	4	2	-	2		4
Жами:		188	108	54	18	36		80
Таълим бериш ва ўқитиш услуби:		<i>Маъруза, амалий машгулот, лаборатория иши, мустақил таълим (айлана стол, кейс стади, мастер-класслар)</i>						
Мустақил ишлар:		<i>Ўқув лойиҳалар, гуруҳли тақдимот, рефератлар, кейслар, докладлар, кроссвордлар, постер, проспекти, эссе ва х.з.</i>						
Маслаҳатлар ва топшириқларни топшириш вақти		Кунлар	Вақти		Ауд.			
1.								
2.								
3.								
Билимларни баҳолаш усуллари, мезонлари ва тартиби:								
ЖН ва ОН нинг баллари ишчи дастурда берилди								
Баҳолаш усуллари		Тестлар, ёзма ишлар, оғзаки сўров, презентациялар ва х.з.						
Фан бўйича талабалар билимини назорат қилиш ва баҳолаш		Назорат шакллари						
		Баҳолаш турлари фан хусусиятидан келиб чиққан ҳолда сўровлар, оғзаки савол-жавоб, ёзма иш, тест синовлари ёки бошқа кўринишда ўтказилиши мумкин.						
		Фан бўйича талабалар билимини баҳолаш мезони						
		Балл	Талабанинг билим даражаси					
		86 - 100 балл	– машгулотларга доимо тайёрланган, жуда фаол, дастурий материалларни яхши билади, хулоса ва қарорлар қабул қилади, ижодий фикрлайди, билимларни амалиётда қўллай олади; – ижодий масалаларни ҳал қилиш мобайнида тегишли билимларни қўллаш доирасини мақсадга мувофиқ танлаб, ечимни топишга хизмат қилувчи янги усул ва йўналишларни топа олади, ўқув материални моҳиятини тушунади; – тақдим этилган ўқув масалаларини ечиш йўллари излайди, дастурий материалларни билади ва айтиб бера олад ҳамда тасаввурга эга бўлади.					
		71 - 85 балл	– ўрганилаётган ҳодисалар алоқадорлигини билиш ҳамда объектни тавсифлай олиш қўникмасини биргаликда, қўйилган масалаларни сабаб-оқибат алоқадорлигини очиқ бериши, назарий билимларни амалиёт билан боғдай олади ва мустақил мушоҳала					

		қилади; – билим ва кўникмалар мазмунини тадбик этиши, бир типдаги масалаларни еча олиш, ёзиб олиш ва эслаб қолиш фаолиятини амалга оширади, билимларни амалиётда қўллай олади; – машғулотларга тайёрланган, дастурий материалларни билади, моҳиятини тушунади ва тасаввурга эга.
	55 - 70 балл	– эшитганлари, уларга берилган намуналар, тақдим этилган алгоритм ва кўрсатмалар асосида топшириқларни бажара олади, моҳиятини тушунади; – катор белгилар асосида маълум объектни фарқлаш билан биргаликда унга таъриф бера олади ва ўқув материалини тушунтириб бера олади ва тасаввурга эга.
	0 - 54 балл	– тасаввурга эга эмас; – дастурий материалларни билмайди.

АСОСИЙ ҚИСМ

Фаннинг назарий машғулотлари мазмуни

- 1-модул. Кириш. Автоматлаштиришда қўлланиладиган дастурлар ҳақида қисқача маълумот.
- 2-модул. Мантикий элементлар ва уларнинг қурилиши.
- 3-модул. Мухандислик дастурларининг синфланиши ва уларнинг қўлланилиш соҳалари.
- 4-модул. MATLAB дастурий пакети. MATLAB дастурида арифметик ифодаларни киритиш.
- 5-модул. MATLAB дастурлаш элементлари ва улар билан ишлаш.
- 6-модул. MATLAB дастурида функциялар графикларини қуриш. М-файллар.
- 7-модул. MATLAB дастурий пакетидаги қисм дастурлар. SIMULINK қисм дастуридаги элементлар.
- 8-модул. SIMULINK қисм дастурида бошқариш системаларини моделлаштириш.
- 9-модул. AutoCAD дастури. Умумий тушунчалар.
- 10-модул. SCADA системаси. SCADA системалари тўғрисида умумий маълумот.
- 11-модул. SCADA системаси архитектураси.
- 12-модул. SCADA системасининг TRACE MODE дастурий пакети.
- 13-модул. SCADA системасининг TRACE MODE дастурий пакетида маълумотлар алмашинуви.
- 14-модул. Тренажер системалари. Тренажер системалари ҳақида умумий тушунча.
- 15-модул. Experion дастури ҳақида умумий тушунча.
- 16-модул. UniSim дастури ҳақида умумий тушунча.
- 17-модул. SIMATIC S7 дастурланадиган контроллерлари. Умумий тушунчалар.
- 18-модул. SIMATIC WinCC. Умумий маълумотлар.

Амалий машғулотларни ташкил этиш бўйича кўрсатма ва тавсиялар

Амалий машғулотларини ўтказишда қуйидаги дидактик тамойилларга амал қилинади:

- амалий машғулотларининг мақсадини аниқ белгилаб олиш;
- ўқитувчининг инновацион педагогик фаолияти бўйича билимларни чуқурлаштириш имкониятларига талабаларда қизиқиш уйғотиш;
- талабада натижани мустақил равишда қўлга киритиш имкониятини таъминлаш;
- талабани назарий-методик жиҳатдан тайёрлаш;
- амалий машғулотлари нафақат аниқ мавзу бўйича билимларни яқунлаш, балки талабаларни тарбиялаш манбаи ҳамдир

Амалий машғулотларнинг тахминий рўйхати

1. мантиқий элементлар устида амаллар бажаришни ўрганиш;
2. MATLAB дастуридаги ишчи ойналарда амаллар бажаришни ўрганиш;
3. Simulink қисм дастуридаги махсус блоклар билан ишлашни ўрганиш;
4. Анимацион жараёнлардаги конструктив параметрларнинг таъсирини ўрганиш;
5. AutoCAD лойиҳалаштиришни автоматлаштирилган пакетини ўрганиш.
6. Ишчи муҳитни яратиш.
7. Схема ва чизмаларни ўрганиш учун буйруқларни ўрганиш. Ўлчам ва ёзувларни тушириш.
8. AutoCAD дастурида турли хил автоматлаштириш воситаларининг техник расмларини рангли бажариш.
9. AutoCAD дастурида уч ўлчамли объектлар билан ишлаш функциялари.
10. SCADA системасининг ишчи блоклари билан танишиш;
11. Expertion системасининг ишчи блоклари билан танишиш;
12. тренажер системасини ишга тушириш кетма кетлигини ўрганиш;
13. тренажер системасида баҳолаш алгоритмининг тузишини ўрганиш;
14. SIMATIC S7 дастурланадиган контроллерларининг асосий блоклари билан танишиш.

Лаборатория ишларини ташкил этиш бўйича кўрсатмалар

Лаборатория ишлари талабаларда автоматик бошқариш тизимларини тузилиши, дастурий пакетларнинг ишлаш принципи, пакетлардан фойдаланган ҳолда дастур тузиш бўйича амалий кўникмалар ҳосил қилишга хизмат қилади.

Лаборатория ишларининг тахминий мавзулари

1. Мантикий элементлар асосида чинлик жадвали ва мантикий схема тузишни ўрганиш.
2. MATLAB дастурий пакетидаги ишчи ойнада амаллар бажариш.
3. MATLAB дастурий пакетида массивлар билан ишлаш.
4. MATLAB дастурий пакетида графиклар қуришни ўрганиш.
5. Simulink қисм дастурида оддий моделлар қуриш.
6. Simulink қисм дастурида структур схемалар тузиш.
7. Simulink қисм дастурида системаларнинг вақт характеристикаларини тадқиқ қилишни ўрганиш.
8. Simulink қисм дастуридаги махсус блоklarдан фойдаланган ҳолда амаллар бажариш.
9. MATLAB дастурий пакетида анимацион ростлаш контурларини қуриш.
10. AUTOCAD тизими билан танишиш. AUTOCAD буйруклари билан танишиш.
11. Қўринишларни ва чизмаларни соzлаш. AUTOCADнинг ососий чизма чизиш буйруклари билан танишиш..
12. Лойихада символлар ва штрихларни қушиш. Ёзувлар ва улчамларни қушиш. Варақлар компоновкаси ва босмага чиқариш.
13. Объектларни қуриш ва тахрирлаш.
14. Автоматлаштиришда фойдаланиладиган функционал схемалардаги содда бошқарув контурларини қуриш
15. SCADA системасидаги жараёнларни текшириш.
16. Trace Mode дастурий пакетида мнемосхемалар тузиш.
17. Trace Mode дастурий пакетида проект яратиш.
18. Trace Mode дастурий пакетида каналлар яратиш.
19. Expertion системасида имитацион моделлар қуриш.
20. Trace Mode дастурий пакетида мнемосхемалар тузиш.
21. UniSIM дастурий пакетида статик объектларни қуриш.
22. Тренажер системасидаги иссиқлик алмаштириш жараёнини текшириш ва уларда машқ жараёнлари.
23. Тренажер системасида насослар ва клапанларни текшириш ва уларда машқ жараёнларини ўрганиш.
24. Step 7 дастурий пакетида мнемосхемалар тузиш

Курс иши (лойихаси) ни ташкил этиш бўйича услубий кўрсатмалар

Таълим йўналиши ўқув режасида курс иши (лойихаси) назарда тутилмаган.

Мустақил таълимни ташкил этишнинг шакли ва мазмуни

Мустақил таълимни ташкил этишда муайян фаннинг хусусиятларини ҳисобга олган ҳолда қуйидаги шакллардан фойдаланиш тавсия этилади ва жорий назорат сифатида баҳоланади:

1) **мавулар бўйича конспект** (реферат, тақдимот) **тайёрлаш**. Назарий материални пухта ўзлаштиришга ёрдам берувчи бундай усул ўқув материалга диққатни кўпроқ жалб этишга ёрдам беради. Талаба конспекти турли назорат ишларига тайёргарлик ишларини осонлаштиради, вақтни тежайди;

2) **ўқитиш ва назорат қилишнинг автоматлаштирилган тизимлари билан ишлаш**. олган билимларини ўзлаштиришлари, турли назорат ишларига тайёргарлик кўришлари учун тавсия этилган электрон манбалар, инновацион дарс лойиҳаси намуналари, ўз-ўзини назорат учун тест топшириқлари в.б;

3) **фан бўйича қўшимча адабиётлар билан ишлаш**. Мустақил ўрганиш учун берилган мавзулар бўйича талабалар тавсия этилган асосий адабиётлардан ташқари қўшимча ўқув, илмий адабиётлардан фойдаланадилар. Бунда рус ва хорижий тиллардаги адабиётлардан фойдаланиш рағбатлантирилади;

4) **ИНТЕРНЕТ тармоғидан фойдаланиш**. Фан мавзуларини ўзлаштириш, курс иши, битирув малакавий ишларини ёзишда мавзу бўйича ИНТЕРНЕТ манбаларини топиш, улар билан ишлаш назорат турларининг барчасида қўшимча рейтинг баллари билан рағбатлантирилади;

5) мавзуга оид масалалар, кейс-стадилар ва ўқув лойиҳаларини ишлаб чиқиш ва иштирок этиш;

6) амалиёт турларига асосан материал йиғиш, амалиётдаги мавжуд муаммоларнинг ечимини топиш, ҳисоботлар тайёрлаш;

7) илмий семинар ва анжуманларга тезис ва мақолалар тайёрлаш ва иштирок этиш;

8) мавжуд лаборатория ишларини такомиллаштириш, масофавий (дистанцион) таълим асосида машғулотларни ташкил этиш бўйича методик кўрсатмалар тайёрлаш ва х.к.

Янги билимларни мустақил ўрганиш, керакли маълумотларни излаш ва уларни топиш йўллари аниқлаш, Интернет тармоқларидан фойдаланиб маълумотлар тўплаш ва илмий изланишлар олиб бориш, илмий тўгарак доирасида ёки мустақил равишда илмий манбалардан фойдаланиб илмий мақола (тезис) ва маърузалар тайёрлаш кабилар талабаларнинг дарсда олган билимларини чуқурлаштиради, уларнинг мустақил фикрлаш ва ижодий қобилиятини ривожлантиради. Вазифаларини текшириш ва баҳолаш амалий машғулот олиб борувчи ўқитувчи томонидан, конспектларни ва мавзунини ўзлаштиришни маъруза дарсларини олиб борувчи ўқитувчи томонидан ҳар дарсда амалга оширилади.

Мустақил ишни ташкил этиш бўйича услубий кўрсатма ва тавсиялар, кейс-стади, вазиятли масалалар тўплами ишлаб чиқилади. Маъруза мавзулари бўйича амалий топшириқ, кейс-стадилар ечиш услуби ва мустақил ишлаш учун вазифалар белгиланади.

Тавсия этилаётган мустақил ишларнинг мавзулари

1. Автоматлаштиришда қўлланиладиган пакетларнинг афзалликлари ва камчиликлари.
2. Манتيкий элементлар асосида 3 дан 2 элементини тузиш ва унинг ишлаш принципи.
3. MATLAB пакетининг афзалликлар ва камчиликлари.
4. Simulink қисм дастуридаги блоklarнинг хусусиятлари.
5. SCADA системаларнинг ишлаш принциплари.
6. SCADA системасидаги Exregion пакетининг тузилиши.
7. Треннажер системасидаги оператор ва инструктор панелларининг ишлаш принциплари.
8. Олинган натижаларга асосан билимларни баҳолаш алгоритми.
9. SIMATIC S7 дастурланадиган контроллерларининг ишлаш принципи.

Дастурнинг ахборот-услубий таъминоти

Мазкур фанни ўқитиш жараёнида таълимнинг замонавий усуллари, педагогик ва ахборот-коммуникация технологиялари қўлланилиши назарда тутилган. Жумладан:

– фаннинг бўлимларига тегишли маъруза дарсларида замонавий компьютер технологиялари ёрдамида презентацион ва электрон-дидактик технологияларидан;

– жараёнга амал қилиш қонуниятлари мавзуларда ўтказиладиган амалий машғулотларда ақлий ҳужум, гуруҳли фикрлаш педагогик технологияларидан;

– машғулотларда кичик гуруҳлар мусобақалари, гуруҳли фикрлаш педагогик технологияларини қўллаш назарда тутилади.

Фойдаланиладиган адабиётлар рўйхати:

Асосий адабиётлар:

1. Stephen J. Chapman MATLAB Programming for Engineers. – New York: CL Engineering USA, 2015. -456p.
2. Keith Stouffer, Joe Falco, Karen Kent. Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security. – Gaithersburg: National Institute of Standards and Technology Gaithersburg, 2006. - 164 p.
3. Т.Дадажонов, М.Мухитдинов. MATLAB асослари. - Тошкент: Фан, 2008. -631 б.
4. A.N. Yusupbekov, F.T. Adilov, V.M. Dozorsev Tipik texnologik obyektlarni imitasion modellashtirish va boshqarish malakali kopyuter treningi. -Toshkent: Tafakkur Bo'stoni, 2016. – 196 b.

5. А.Н.Юсупбеков, Ф.Т.Адилов, В.М.Дозорцев Имитационное моделирование типовых технологических объектов и компьютерный тренинг навыкам управления. -Ташкент: Tafakkur Bo'stoni, 2015. – 204с.

Қўшимча адабиётлар:

1. В.М.Дозорцев Компьютерные тренажеры для обучения операторов технологических процессов. – Москва: Энергия, 2009. -436с.
2. Н.Р. Юсупбеков, Ш.М. Гулямов, У.Т. Мухамедханов. Применение программируемые логических контроллеров SIMATIC S7-200 к решению задач логического управления. –Ташкент: НИИМИТ, 2008. -254с.
3. Потемкин В.Г. Вычисления в среде MATLAB -Москва: Бинном, 2004. - 458с.
4. Н.Р. Юсупбеков, Ф.Т. Адилов, С.Ш. Халилова. Построение компьютерных тренажеров для подготовки операторов химико-технологических процессов и производств. –Ташкент: НИИМИТ, 2004. - 236с.
5. М.Мухитдинов, Т.Дадажонов, Х.Кулматов MATLAB илмий тадқиқот ишларида. – Тошкент: «O'zbekiston», 2016.-256 б.
6. С.Т. Jones STEP 7 in 7 Steps A Practical Guide to Implementing S7-300/S7-400 Programmable Logic Controllers. – USA, 2006.-443 p.
7. Данилов А.И. Компьютерный практикум по курсу Теория управления. SIMULINK – моделирование в среде MATLAB. –М.: МГУИЭ, 2002. -342с.

Интернет сайтлари


1. www.gov.uz – Ўзбекистон Республикаси ҳукумат портали.
2. www.catback.ru - научные статьи и учебные материалы:
3. www.zivonet.uz
4. www.knowledge.allbest.ru
5. www.twirpx.com
6. www.matlab.com
7. www.5ballov.ru

РАБОЧАЯ УЧЕБНАЯ ПРОГРАММА

Министерство высшего и среднего специального образования
Республики Узбекистан
Навоийский горно-металлургический комбинат
Навоийский государственный горный институт

Энерго - механический факультет
Кафедра "Автоматизация и управление"

Зарегистрирован:
№ 3-АУ
«28» 08 2017 г.

«УТВЕРЖДАЮ»
Проректор по учебной части:
 Н.А.Абдуазизов
«28» 08 2017 г.

РАБОЧАЯ УЧЕБНАЯ ПРОГРАММА

по курсу
«ИНЖЕНЕРНОЕ ПРОГРАММИРОВАНИЕ»

для бакалавров

Область знаний	300 000 – Производственно – техническая сфера
Область образования	310 000 – Инженерная дело
Направление образования	5 311 000 – Автоматизация и управление технологических процессов и производств (по отраслям)
Семестр	2
Общих аудиторных часов	108
В том числе:	
Лекция	54
Практические занятия	18
Лабораторные занятия	36
Самостоятельная работа	80
Всего	188

Навои – 2017

Составитель:

Уринов Ш.Р. - к.т.н., доц. кафедры «Автоматизация и управление»

Рабочая учебная программа обсуждена и одобрена на заседании кафедры «Автоматизация и управление» от «25» 08 2017 года. (Протокол № 1)

Заведующий кафедрой



Жумаев О.А.

Рабочая учебная программа обсуждена и одобрена на Совете энергетической факультета "26" 08 2017 года (Протокол № 1).

Декан факультета



Бозорова С.Ж.

Начальник учебно-методического отдела



Каримов И.А.

КРАТКАЯ ИНФОРМАЦИЯ ПО КУРСУ		
Адрес ВУЗа	Навоийский государственный горный институт	г. Навои, ул. Галаба, 176.
Кафедра	Автоматизация и управление каб № 305 сот. телефон: +99890-6464635	Факультет «Электроники и автоматики»
Сфера образования	5311000 – Автоматизация и управлени технологических процессов и производств	Бакалавры 1 го года обучения
Время и место занятий	по расписанию,	здание факультета «ЭМФ»
Информация о преподавателе изучаемого курса	доц. Уринов Ш.Р.	E-mail sh_urinov@mail.ru
Актуальность курса:	<p>Целью изучения дисциплины «Инженерное программирования» является подготовка специалистов, обладающих знаниями об архитектуре современных прикладных программ и способных применять свои знания для установки и сопровождения программных и технических средств, а также для работы с SCADA на начальном уровне.</p> <p>Курс должен помочь сформировать у студентов знания по теории и практики программ используемыми в автоматизации и управления.</p> <p>Для достижения поставленной цели в рамках изучения дисциплины требуется решить следующие задачи:</p> <ul style="list-style-type: none"> – ознакомить с пакетами используемых в автоматизации их классификации, архитектурой и принципами работы многозадачных и работающих в сети; – сформировать представление о структуре современных программных продуктов систем и принципах функционирования основных подсистем; – обеспечить знания и умения, достаточные для управления ресурсами персонального компьютера, их конфигурирования и использования средств интерфейса ОС. <p>Освоение дисциплины «Инженерное программирования» необходимо для дальнейшего изучения профессиональных модулей.</p>	
Требования к знаниям, навыкам и квалификации студентов по дисциплине	<p>Бакалавр в рамках задач, осуществляемых в процессе освоения учебной дисциплины «Инженерное программирования» должен знать:</p> <ul style="list-style-type: none"> -Особенности применение программных пакетов в автоматизации. -Логические элементы и их построение. -Классификация пакетов и области их применения. -Программный пакет MATLAB. Общие понятия и принципы. -Элементы программирования в MATLAB. -Построение графиков функций в системе MATLAB. -Подпрограммы программного пакета MATLAB. Элементы подпрограммы SIMULINK. -Моделирования систем управления в подпрограмме SIMULINK. -Программный пакет AutoCAD. Общие понятия. -SCADA системы. Общие понятия о SCADA системе. -Архитектура SCADA системы. -Программный пакет Trace Mode. --Тренажерные системы. Общие понятия о тренажерных системах. -Программный пакет Exregion. -Программный пакет UniSim. -Программируемые контроллеры SIMATIC S7. <p>Общие понятия и принципы.личностному развитию, повышению квалификации в области пакеты применяемые в автоматизация.</p>	
Роль дисциплины в производстве:	<p>Дисциплина «Инженерное программирования» позволяет выровнять уровень подготовки студентов по программированию и научить «свободно владеть» методами выбора компоненты пакетов, применяемых в автоматизации для построения различных архитектур вычислительных средств; методами и средствами разработки и оформления технической документации ЭВМ, систем и периферийных устройств; основными стадиями загрузки ОС и существующими механизмами восстановления при</p>	

	возникновении сбоев.
Виды занятий:	<ul style="list-style-type: none"> – лекции; – опрос теоретических знаний по темам; – самостоятельная работа; – ведение конспектов; – тестов; – индивидуальные и групповые презентации; – выполнения домашних заданий; – написание рефератов; – практическое занятие и др.

Система оценки освоения и уровня знаний студентов по предмету «Инженерное программирование»

1. Текущий контроль (ТК)	<p>Предусматривает оценивать знаний бакалавров полученные за освоение каждой проведенной темы по данному предмету. Обычно это оценивается на лекционных, практических занятиях. В первую очередь текущий контроль включает в себя уровень освоения бакалавром в аудитории, т.е. активность на занятиях, которое включает в себя следующее:</p> <ul style="list-style-type: none"> – опрос теоретических знаний по темам; – проверка конспектов; – проведение тестов; – проверка выполнения домашних заданий; – практическое занятие и др.
2. Проведение промежуточного контроля (ПК)	<p>По дисциплине «Операционные системы» предусмотрено проводить 2 раза по графику учебного плана. При проведении промежуточного контроля, вопросы и тестирование проводится по ранее пройденным темам. Промежуточный контроль может проводиться письменно, устно или в вид тестирования. При проведении промежуточного контроля в письменной форме, каждый вариант должен содержать не менее 3 вопросов, а в виде тестирования не менее 10 вопросов. Вопросы промежуточного контроля должны быть разработаны преподавателями в начале учебного года, после чего должен быть обсужден и утвержден на заседании кафедры. Утвержденные вопросы промежуточного контроля должны быть заранее предоставлены студентам</p>
3. Итоговый контроль (ИК)	<p>Обычно проводится в конце учебного семестра, с целью оценки полученных бакалавром знаний. Он может проводиться в письменной и др. формах (устно, или письменно) и оценивается в 30 баллов. Контроль уровня знаний, успеваемости и квалификаций бакалавров по предмету «Операционные системы» проводится на основе рейтинговой системы и определяется в баллах.</p> <p>Максимальный балл, который может получить студент – 100 баллов, он распределяется следующим образом: текущий контроль (ТК) – 35 баллов; промежуточный контроль (ПК) – 35 баллов; итоговый контроль (ИК) – 30 баллов. По предмету ТК, ПК, ИК за отведенные баллы по каждому контролю 55 баллов считается проходным баллом. Студент, набравший меньше проходного балла по каждому контролю считается академическим должником.</p>

Содержание занятий:

№	ТЕМЫ	Аудиторные часы: в том числе			Самостоятельная работа	Всего
		Лекции	Практические занятия	Лабораторные занятия		
1.	Обзор пакетов, применяемых в автоматизации.	2	-	-	4	6
2.	Логические элементы и их построение.	2	2	4	4	12
3.	Классификация пакетов и области их применения.	2	-	-	4	6
4.	Программный пакет MATLAB. Общие понятия и принципы.	2	2	4	4	12
5.	Элементы программирования в MATLAB	2	2	4	4	12
6.	Построение графиков функций в системе MATLAB	2	2	4	4	12
7.	Подпрограммы программного пакета MATLAB. Элементы подпрограммы SIMULINK	4	2	2	6	14
8.	Моделирования систем управления в подпрограмме SIMULINK	2	2	2	4	10
9.	Введение в AutoCAD	4	2	4	4	14
10.	SCADA системы. Общие понятия о SCADA системе	4	-	-	6	10
11.	Архитектура SCADA системы	2	-	-	4	6
12.	Особенности проектирования распределенной иерархической АСУТП на базе SCADA системы TRACE MODE	4	-	2	4	10
13.	Обмен данными в SCADA системе TRACE MODE	4	2	2	4	12
14.	Тренажерные системы. Общие понятия о тренажерных системах	4	-	2	4	10
15.	Программный пакет Experion	4	2	2	4	12
16.	Программный пакет UniSim	4	-	4	4	12
17.	Программируемые контроллеры SIMATIC S7. Общие сведения	4	-	-	6	10
18.	SIMATIC WinCC. Общие сведения	2	-	2	4	8
	Всего	54	18	36	80	188

Литература:

1. Stephen J. Chapman. Matlab Programming for Engineers. – New York: CL Engineering USA, 2015. – 2015. – 456 p.
2. Keith Stouffer, Jae Falco, Karen Kent. Guide to Supervisory Control and Data Acquisitions (SCADA) and industrial Control Systems Security. – Gaithersburg: National Institute of Standards and Technology Gaithersburg, 2006. – 164 p.
3. Т.Дадажонов, М.Мухиддинов. Matlab асослари. – Тошкент: Фан, 2008. – 632 б.
4. A.N.Yusupbekov, F.T.Adilov, V.M.Dozorsev. Tipik texnologik obyektlarni imitasion modellashtisih va boshqarish malakali kopmpyuter treningi. – Toshkent: Tafakkur Bo'stoni, 2016. – 196 b.
5. A.N.Yusupbekov, F.T.Adilov, V.M.Dozorsev. Имитационное моделирование типовых технологических объектов и компьютерный тренинг навыкам управления. – Ташкент: Tafakkur Bo'stoni, 2015. – 204 с.

Дополнительные литературы:

1. В.М.Дозорцев Компьютерные тренажеры для обучения операторов технологических процессов. – Москва: Энергия, 2009. -436с.
2. Н.Р. Юсупбеков, Ш.М. Гулямов, У.Т. Мухамедханов. Применение программируемые логических контроллеров SIMATIC S7-200 к решению задач логического управления. –Ташкент: НИИМИТ, 2008. -254с.
3. Потемкин В.Г. Вычисления в среде MATLAB -Москва: Бинном, 2004. - 458с.
4. Н.Р. Юсупбеков, Ф.Т. Адилов, С.Ш. Халилова. Построение компьютерных тренажеров для подготовки операторов химико-технологических процессов и производств. –Ташкент: НИИМИТ, 2004. - 236с.
5. М.Мухитдинов, Т.Дадажонов, Х.Кулматов MATLAB илмий таджикот ишларида. – Тошкент: «O'zbekiston», 2016.-256 б.
6. С.Т. Jones STEP 7 in 7 Steps A Practical Guide to Implementing S7-300/S7-400 Programmable Logic Controllers. – USA, 2006.-443 p.
7. Данилов А.И. Компьютерный практикум по курсу Теория управления. SIMULINK – моделирование в среде MATLAB. –М.: МГУИЭ, 2002. -342с.

Интернет сайтлари

1. www.gov.uz – Ўзбекистон Республикаси ҳукумат портали.
2. www.catback.ru - научные статьи и учебные материалы;
3. www.zivonet.uz
4. www.knowledge.allbest.ru
5. www.twirpx.com
6. www.matlab.com
7. www.5ballov.ru

ПРОМЕЖУТОЧНЫЕ И ИТОГОВЫЕ КОНТРОЛЬНЫЕ ВОПРОСЫ

Лекция 1.

Обзор пакетов, применяемых в автоматизации.

Введение. Автоматизация и потребность в автоматизации

План:

1. Обзор пакетов, применяемых в автоматизации
2. Пакеты, применяемые в автоматизации.
3. SCADA системы
4. Тренажерные системы
5. Matlab

Лекция 2.

Логические элементы и их построение И, ИЛИ, НЕ и их комбинации

План:

1. Элемент И.
2. Элемент ИЛИ.
3. Элемент НЕ.
4. Элемент И-НЕ.

Лекция 3.

Классификация пакетов, применяемых в автоматизации

План:

1. Программное обеспечение современных ЭВМ
2. Прикладные программные пакеты
3. Классификация пакетов, применяемых в автоматизации

Лекция 4.

Программный пакет Matlab. Общие понятия и принципы.

План:

1. Окружающая среда «Matlab»;
2. Командное окно программы в Matlab;
3. Стандартные функции в программе «Matlab»;
4. Математические вычисления в Matlab;

Лекция 5.

Элементы программирования в MATLAB.

План:

1. Основные понятие программирования в Matlab;
2. Виды программирования в Matlab;
3. М-файлы в Matlab;
4. Примеры элементов программирования в Matlab;

Лекция 6.

Построение графиков функции в MATLAB

План:

1. Графические команды и функции в Matlab
2. Элементарные графические функции в Matlab
3. Двумерные и Трехмерные график в Matlab
4. Надписи и пояснения к графикам в Matlab
5. Специальная графика в Matlab

Лекции 7-8.

Подпрограммы программного пакета Matlab. Элементы подпрограммы Simulink.

План:

1. Подпрограммы программного пакета MATLAB.
2. Ознакомление с подпрограммным пакетом SIMULINK.
3. Блоки математических операций вSIMULINK.
4. Решение системы нелинейных уравнений.

Лекция 9.

Моделирование системы управления в подпрограмме Simulink.

План:

1. Моделирование в Simulink;
2. Анализ Simulink моделей;
3. Маскирование подсистем в Simulink;
4. Управление Simulink - моделью из MATLAB.

Лекция 10-11.

Введение в программу AutoCAD

План:

1. Графические возможности .
2. Методы задания координат .
3. Организация и работа пакета AUTOCAD.
4. Простое и сложное редактирование
5. Вывод информации

Лекция 12-13.

SCADA системы. Общие понятия о SCADA системе.

План:

1. Обзор SCADA системе
2. Сеть связи SCADA системы
3. Центральный Компьютера SCADA системы
4. Рабочие станции оператора и компоненты программного обеспечения SCADA системы

Лекция 14.

Архитектура SCADA системы

План:

1. Первое поколение – Монолитный SCADA системы
2. Второе поколение – Распределенная SCADA системы
3. Третье поколение - подключенный к сети SCADA системы

Лекция 15-16.

Особенности проектирования распределенной иерархической АСУТП на базе SCADA-системы Trace Mode

План:

1. Идеология распределенных комплексов с применением SCADA-систем.
2. Архитектура TRACE MODE.
3. Основные понятия системы TRACE MODE.

Лекция 17-18.

Обмен данными в SCADA системе TRACE MODE

План:

1. Обмен Trace Mode с базами данных через механизмы ODBC.
2. Обмен данными в Trace Mode через механизмы OPC.
3. СДКУ системами вентиляции и теплоснабжением

Лекция 19-20.

Тренажерные системы. Общие понятия о тренажерных системах.

План:

1. Основные определения и понятия тренажеро-строения
2. Структура и назначение тренажерных комплексов
3. Концепция тренажера уровня установки
4. История создания тренажеров в промышленности
5. Тенденции развития компьютерных тренажеров
6. Обзор тренажерных комплексов
7. Архитектура компьютерного тренажера
8. Основные характеристики тренажеров
9. Требования к тренажеру
10. Этапы создания тренажера

Лекция 21-22.

Программный пакет Experion PKS

План:

1. Архитектура системы управления Experion PKS
2. Структура системы

3. Программирование задач контроллера и управление
4. Программирование и реализация SCADA-программы
5. Уровни безопасности
6. Безопасность по Операторам

Лекция 23-24.

Программный пакет UniSIM

План:

Структура системы UniSim

Программное обеспечение;

UniSim Operations

UniSim Design

Ключевые особенности UniSim Design

Тренажёрные комплекс КТК-М

Лекция 25, 26.

Программируемые контроллеры Simatic Step 7. Общие сведения.

План:

Обзор STEP 7

Стандартный пакет STEP 7

Расширенное использование стандартного пакета STEP 7

Человеко-машинный интерфейс STEP 7

Проектирование решения задачи автоматизации STEP 7

Описание отдельных функциональных областей STEP 7

Определение требований безопасности STEP 7

Лекция 27.

Simatic WinCC. Общие сведения.

План:

Введение в SIMATIC WinCC

Расширения SCADA

Человеко-машинный интерфейс

Встроенное управление пользователями. SIMATIC Logon

Простое и эффективное проектирование. WinCC Graphics Designer

ИТОГОВАЯ КОНТРОЛЬНАЯ РАБОТА

1 -билет

1. Элемент И.
2. Графические возможности .
3. Основные определения и понятия тренажеро-строения

2 -билет

1. Структура и назначение тренажерных комплексов
2. Методы задания координат .
3. Элемент ИЛИ.

3 -билет

1. Элемент НЕ.
2. Организация и работа пакета AUTOCAD.
3. Концепция тренажера уровня установки

4 -билет

1. История создания тренажеров в промышленности
2. Простое и сложное редактирование
3. Элемент И-НЕ.

5 -билет

1. Окружающая среда «Matlab»;
2. Обзор SCADA системе
3. Тенденции развития компьютерных тренажеров

6 -билет

1. Обзор тренажерных комплексов
2. Сеть связи SCADA системы
3. Командное окно программы в **Matlab**;

7 -билет

1. Стандартные функции в программе «Matlab»;
2. Центральный Компьютера SCADA системы
3. Архитектура компьютерного тренажера

8 -билет

1. Основные характеристики тренажеров
2. Рабочие станции оператора и компоненты программного обеспечения SCADA системы
3. Математические вычисления в **Matlab**;

9 -билет

1. Основные понятие программирования в **Matlab**;
2. Архитектура TRACE MODE.
3. Архитектура системы управления Exregion PKS

10 -билет

1. Структура системы UniSim
2. Основные понятия системы TRACE MODE.
3. М-файлы в **Matlab**;

11 -билет

1. Графические команды и функции в **Matlab**
2. Обмен Trace Mode с базами данных через механизмы ODBC.
3. Стандартный пакет STEP 7

12 -билет

1. Простое и эффективное проектирование. WinCC Graphics Designer
2. Тренажёрные комплекс КТК-М
3. Блоки математических операций вSIMULINK.

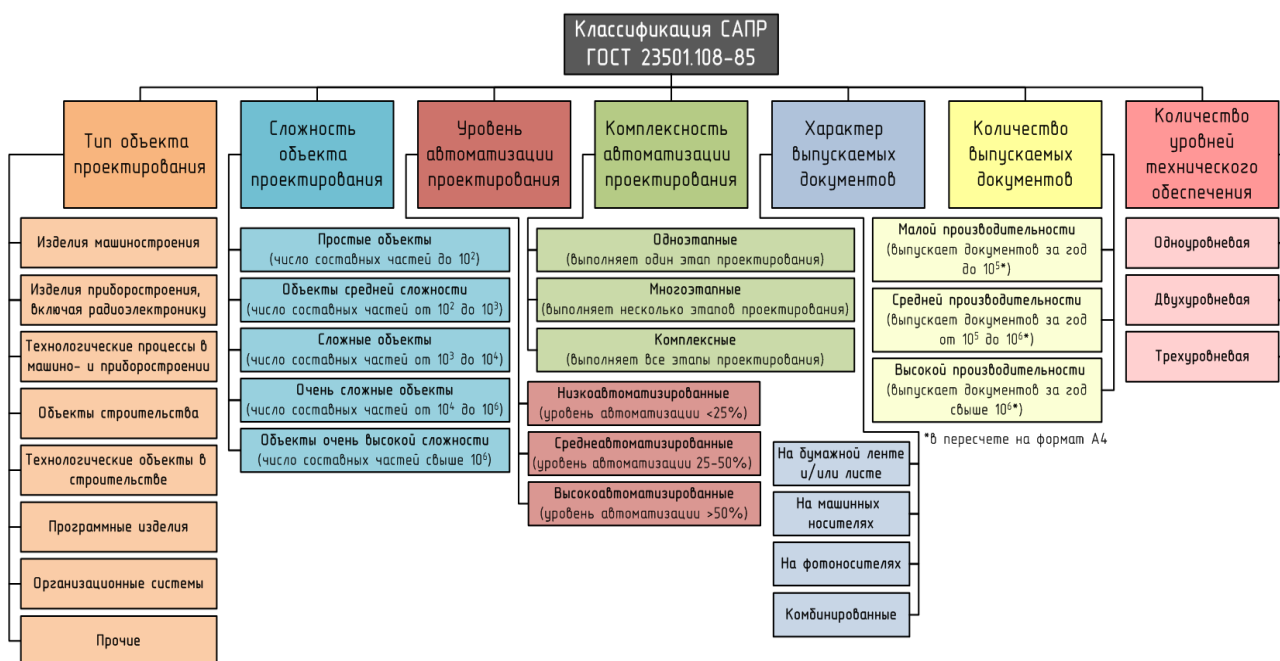
РАЗДАТОЧНЫЕ МАТЕРИАЛЫ

Раздаточные материалы по предмету

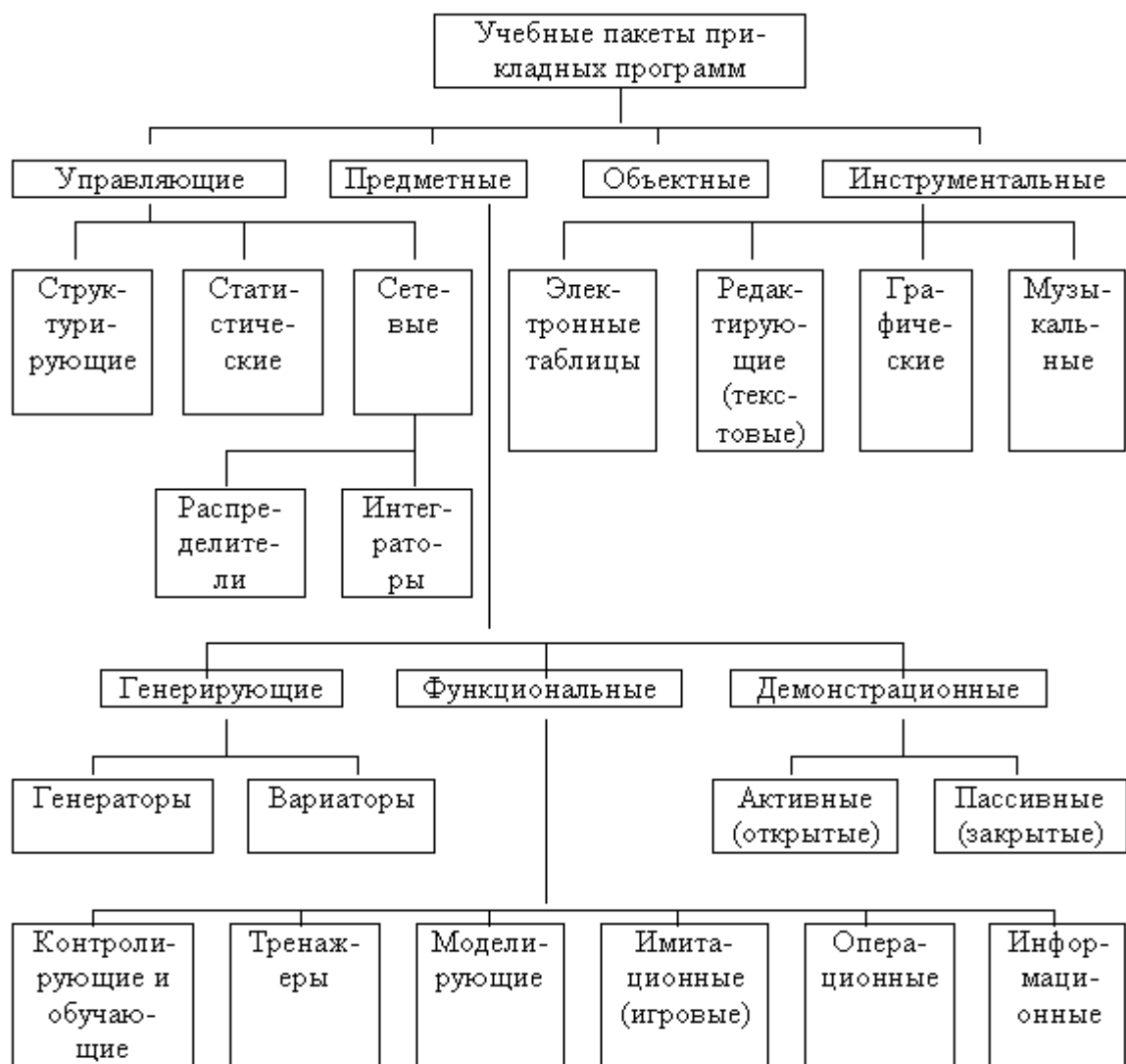




Пример классификации прикладного программного обеспечения.



Классификация САПР



Пример классификации учебных прикладных программ

MATLAB 7.11.0 (R2010b)

File Edit Debug Parallel Desktop Window Help

Current Folder: C:\Program Files\MATLAB\R2010b\bin

Shortcuts How to Add What's New

Current Folder: R2010b > bin

Command Window

```
>> E=ones(3,4)

E =

     1     1     1     1
     1     1     1     1
     1     1     1     1
```

Workspace

Name	Value
E	<3x4 double>
ans	0.0000 + 1.0000i

Command History

```
errorbar(x1, ...
errorbar(x3, ...
%-- 30.04.2016
%-- 01.05.2016
%-- 09.05.2016
%-- 12.05.2016
sqrt(-1)
clc
E=ones(3,4)
```

MATLAB 7.11.0 (R2010b)

File Edit Debug Parallel Desktop Window Help

Current Folder: C:\Program Files\MATLAB\R2010b\bin

Shortcuts How to Add What's New

Current Folder: R2010b > bin

Command Window

```
>> sqrt(-1)

ans =

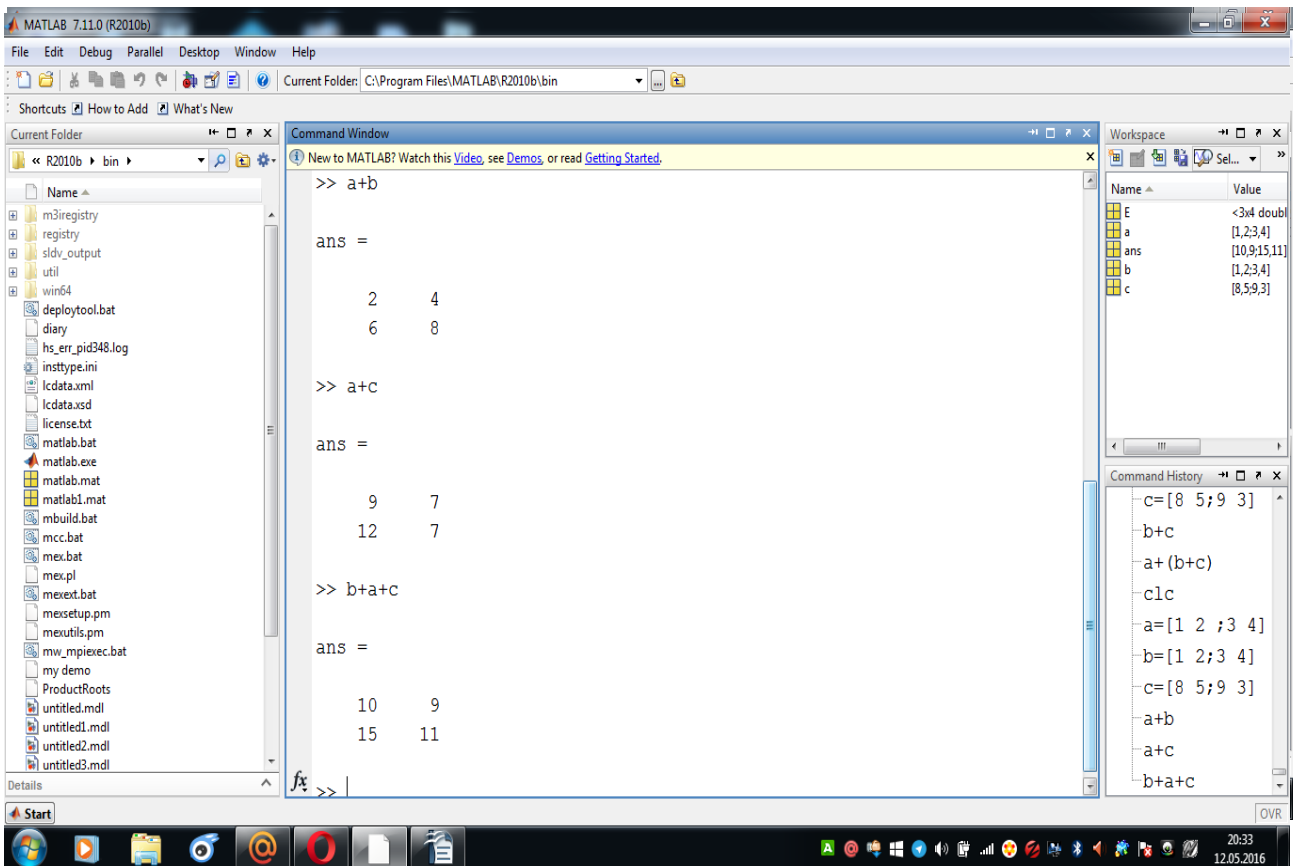
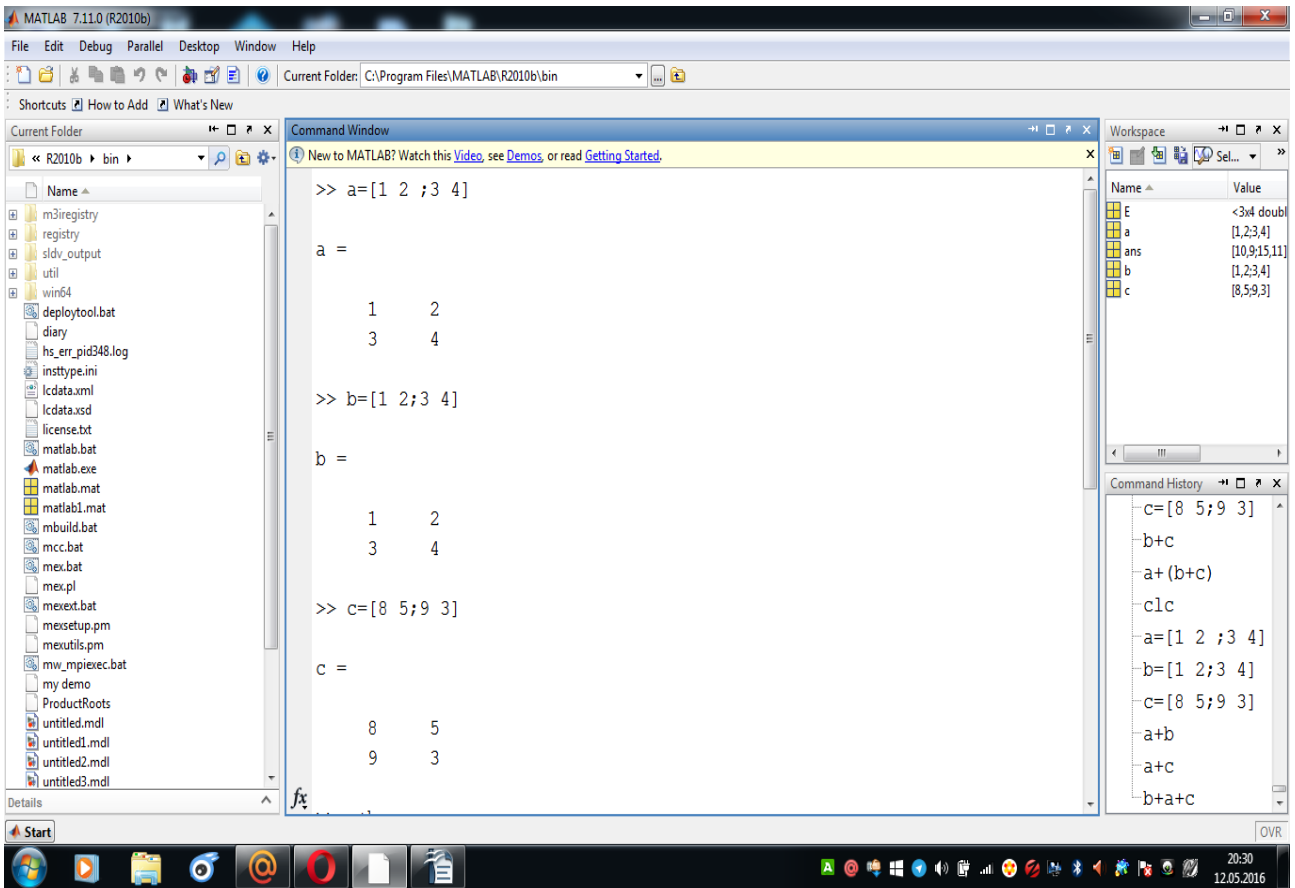
     0 + 1.0000i
```

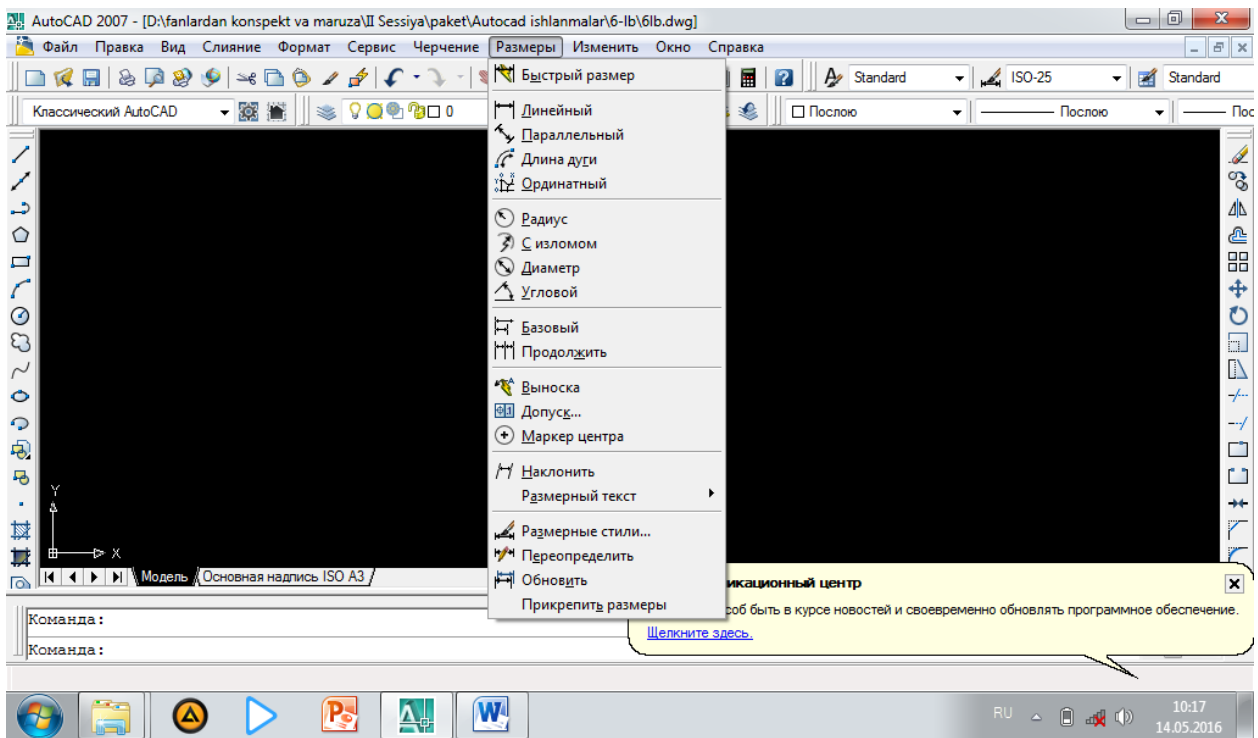
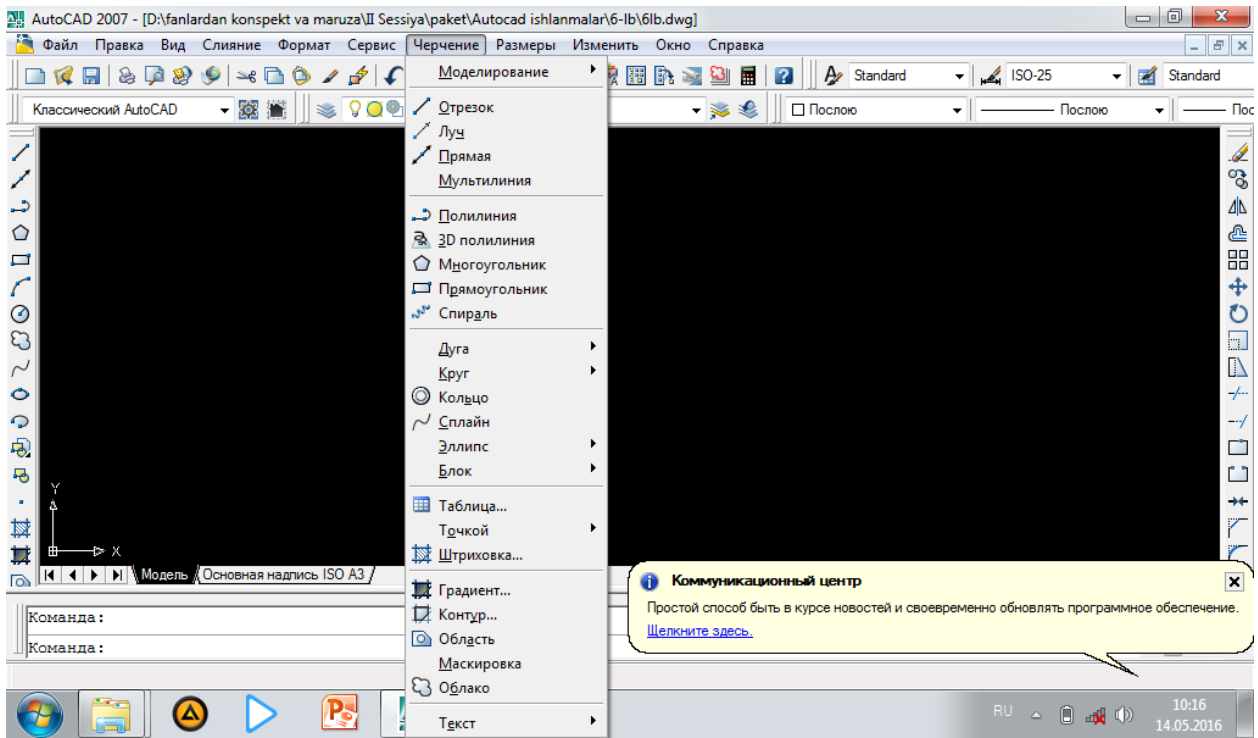
Workspace

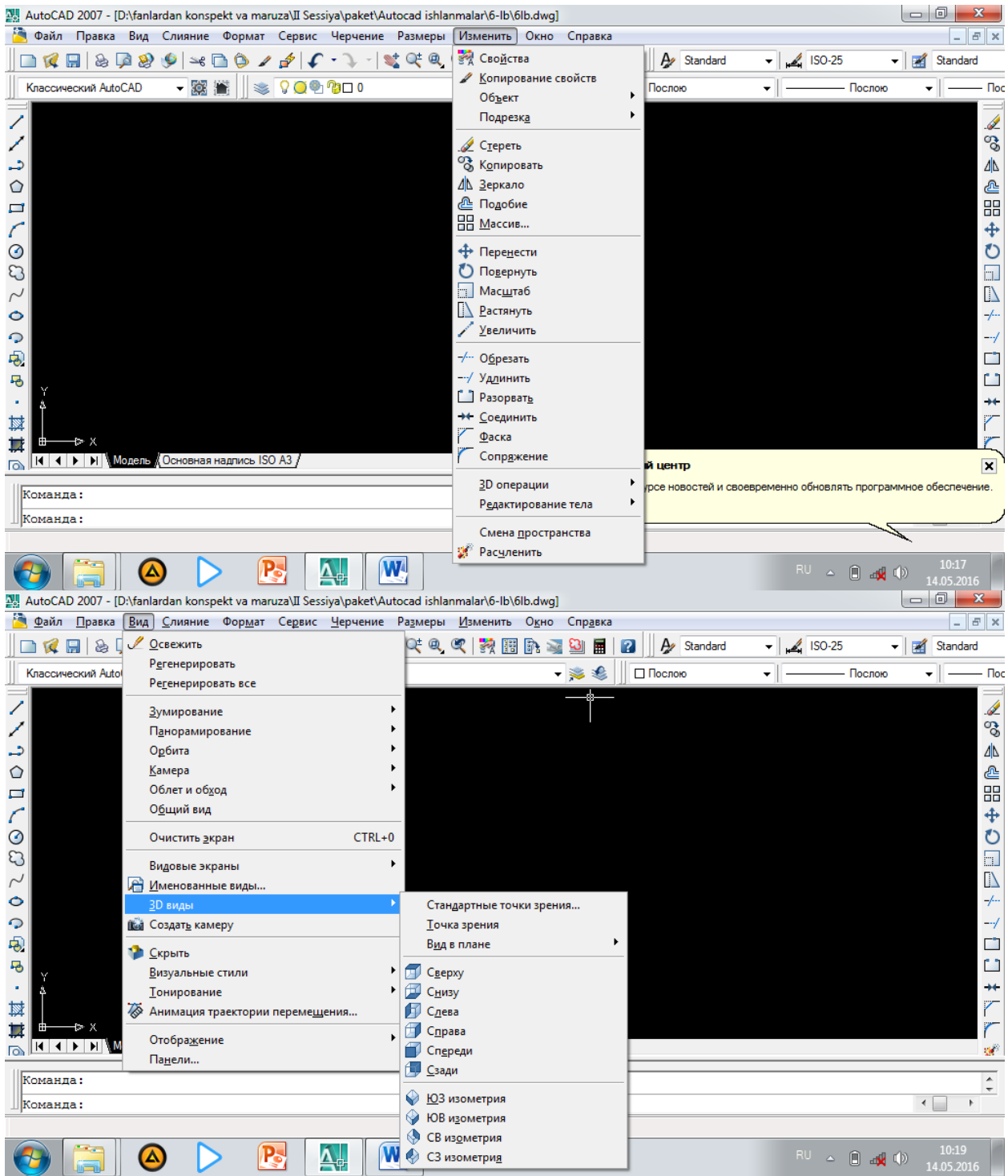
Name	Value
ans	0.0000 + 1.0000i

Command History

```
errorbar(x1, ...
errorbar(x2, ...
errorbar(x1, ...
errorbar(x3, ...
%-- 30.04.2016
%-- 01.05.2016
%-- 09.05.2016
%-- 12.05.2016
sqrt(-1)
```







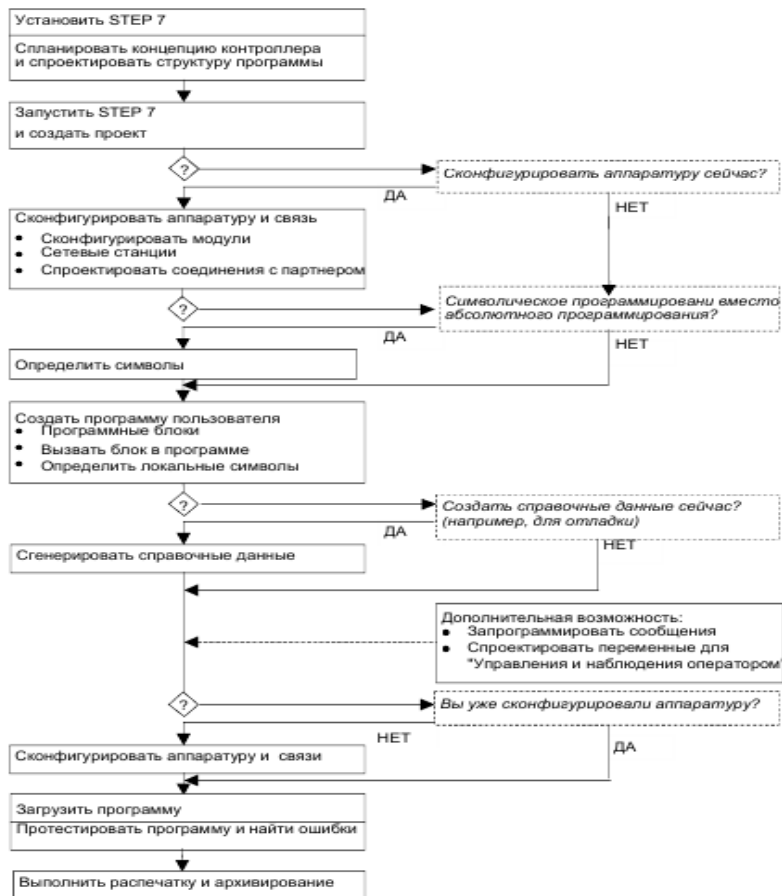


Рис. 1

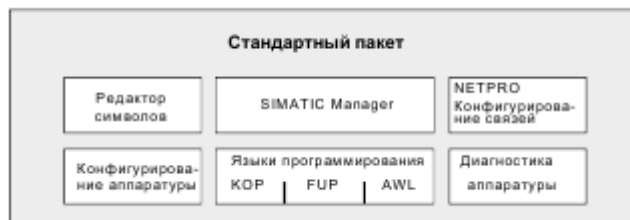


Рис. 2. Состав стандартного пакета STEP 7

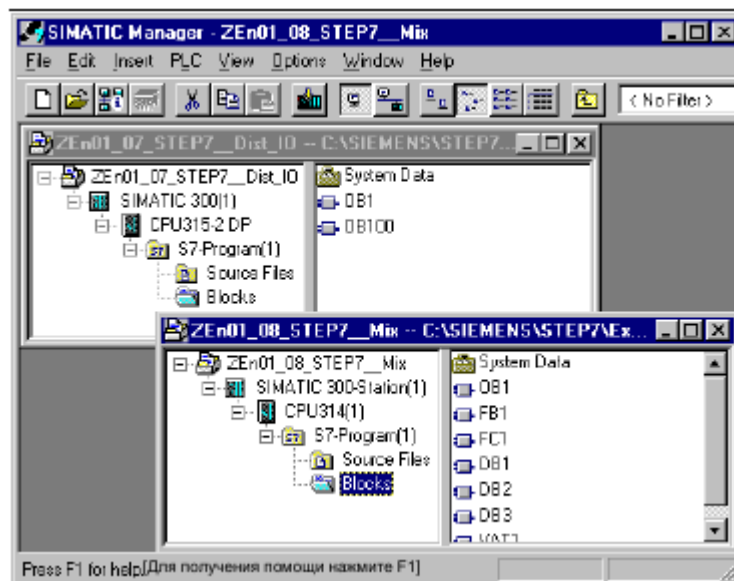


Рис. 3. Автоматически запускаемые инструменты в SIMATIC Manager



Рис. 4. Состав инструментальных средств для проектирования

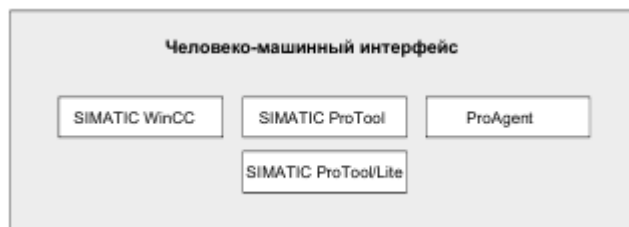


Рис. 6. Состав человеко-машинного интерфейса.

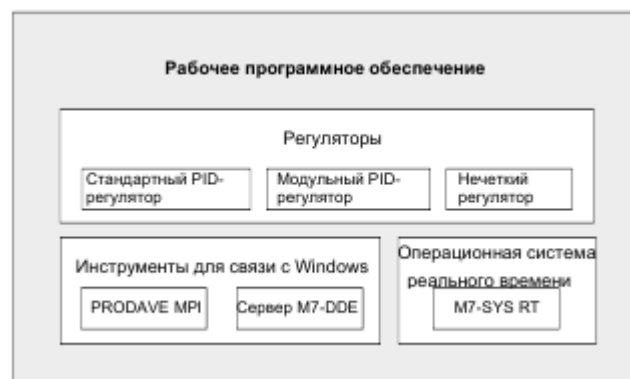


Рис. 5. Состав рабочего программного обеспечения

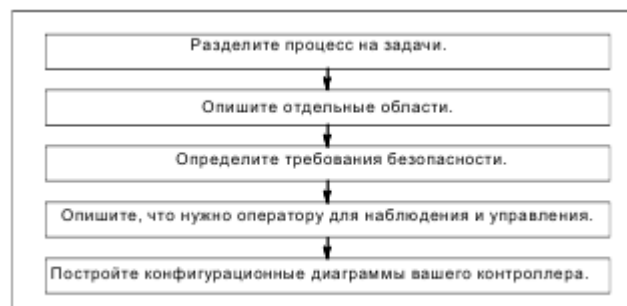


Рис. 7. Основная последовательность действий при планировании проекта автоматизации

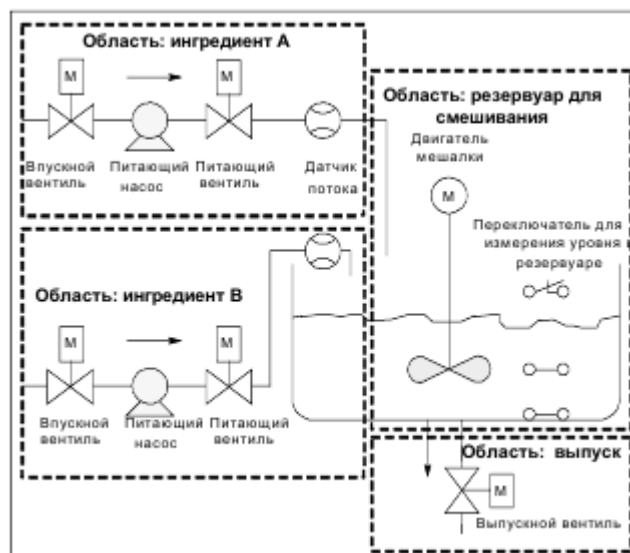


Рис. 9. Разделение процесса на связанные группы областей



Рис. 8. Пример промышленного процесса смешивания

Таблица 1.

Функциональная область	Используемое оборудование
Ингредиент А	Питающий насос для ингредиента А Впускной вентиль для ингредиента А Питающий вентиль для ингредиента А Датчик потока для ингредиента А
Ингредиент В	Питающий насос для ингредиента В Впускной вентиль для ингредиента В Питающий вентиль для ингредиента В Датчик потока для ингредиента В
Резервуар для смешивания	Двигатель мешалки Переключатель для измерения уровня в резервуаре
Выпуск	Выпускной вентиль



Рис. 10. Диаграммы входов и выходов

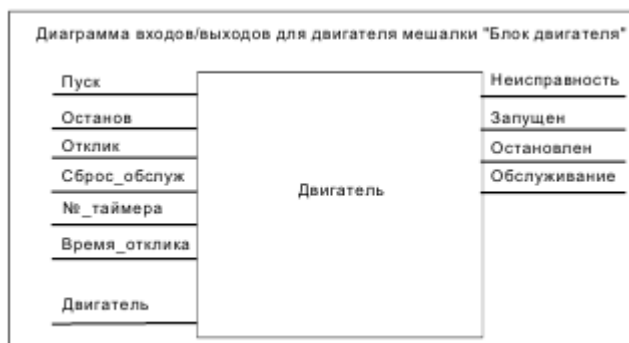


Рис. 11. Диаграмма входов и выходов для двигателя мешалки «Блок двигателя»

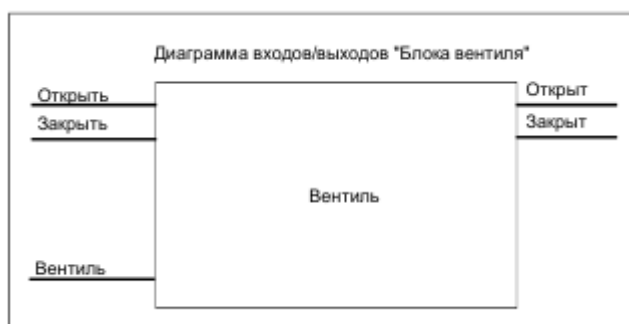


Рис. 12. Диаграмма входов и выходов «Блока вентиля»

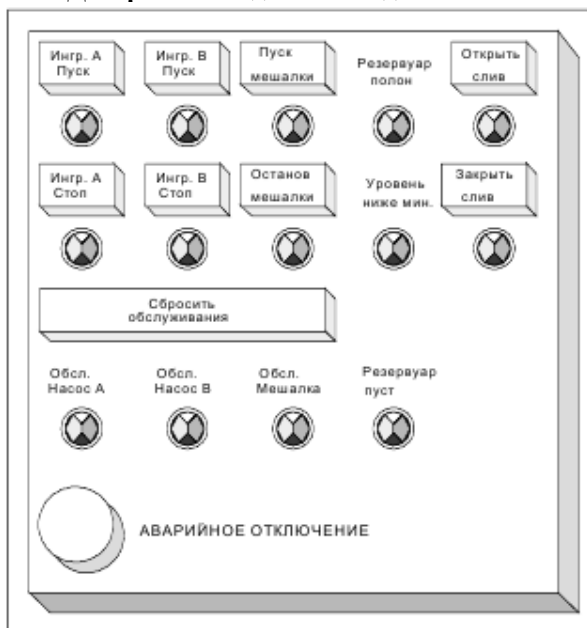
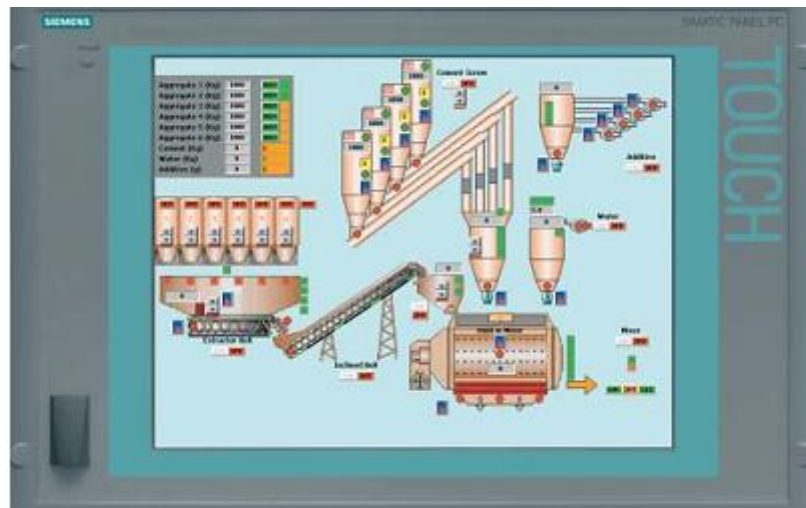
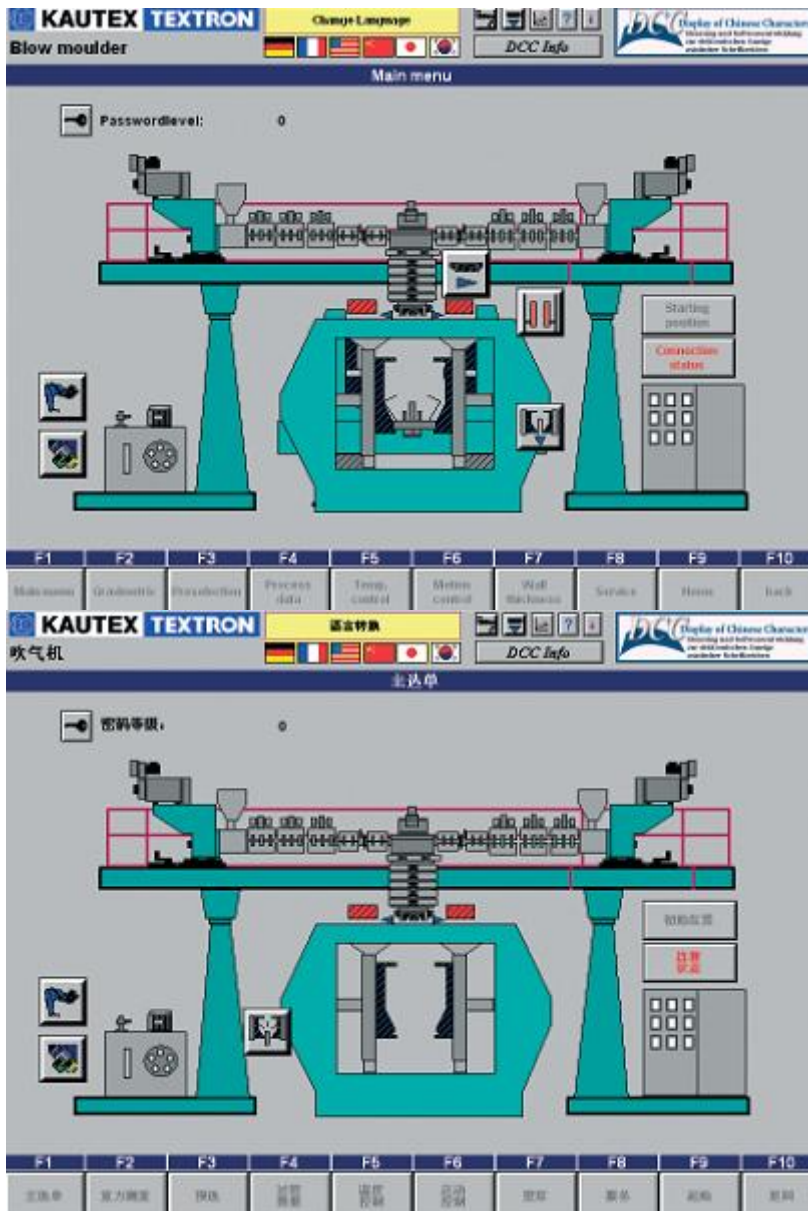


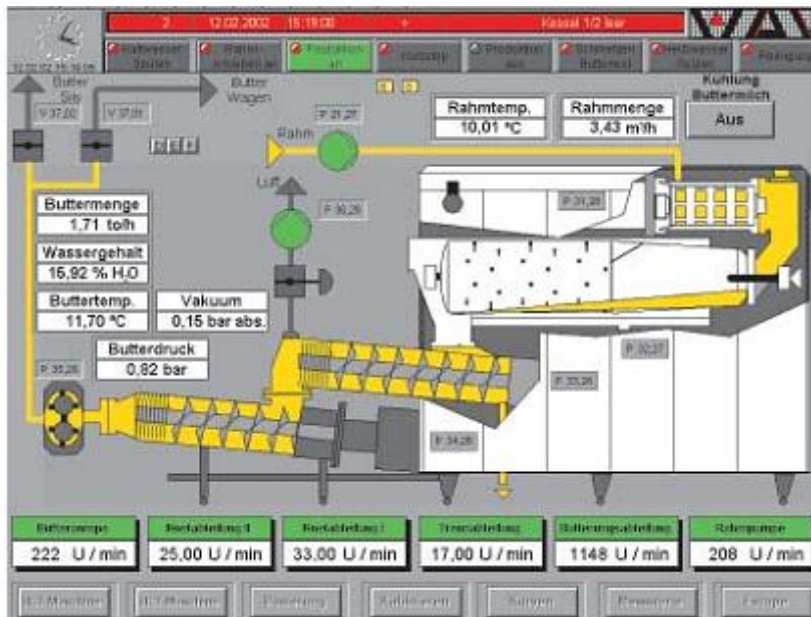
Рис. 13. Пульт оператора



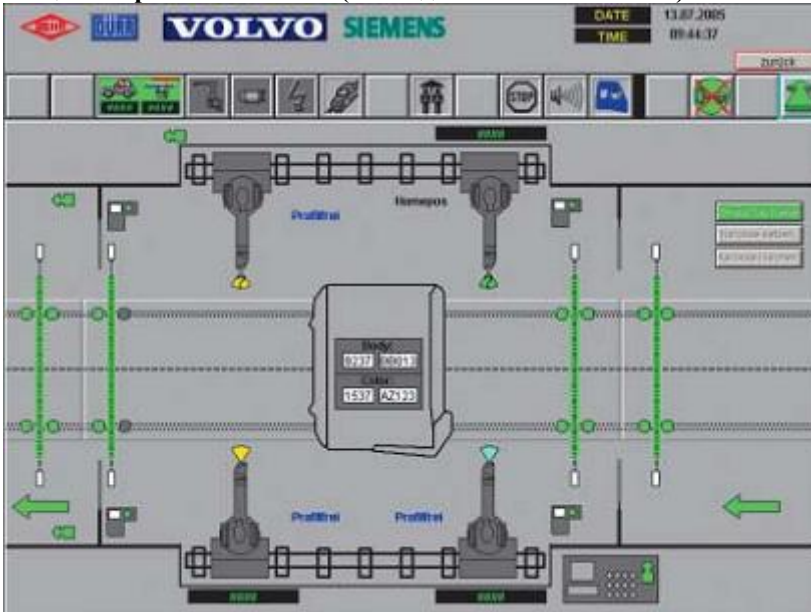
Plant Intelligence и интеграция информационных технологий и бизнеса



Производство пластмасс и переключение языков



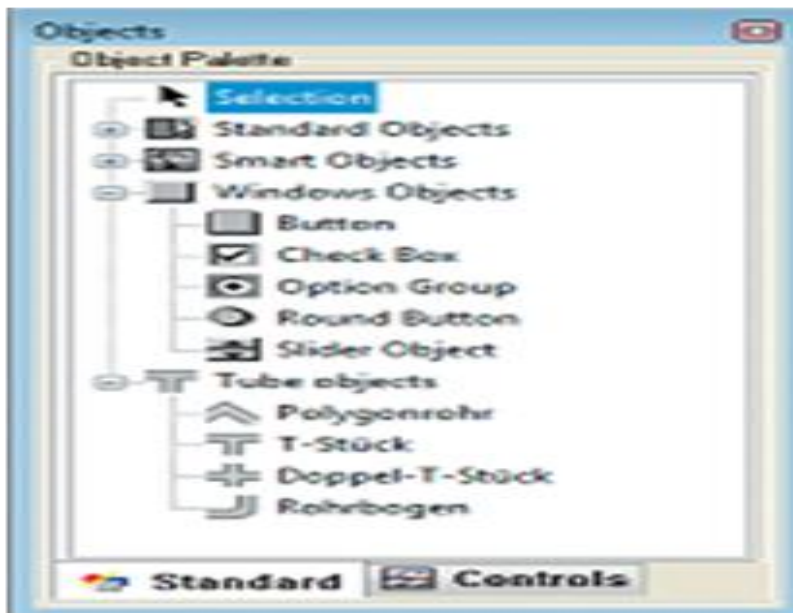
Пищевая промышленность (маслоделательная машина)



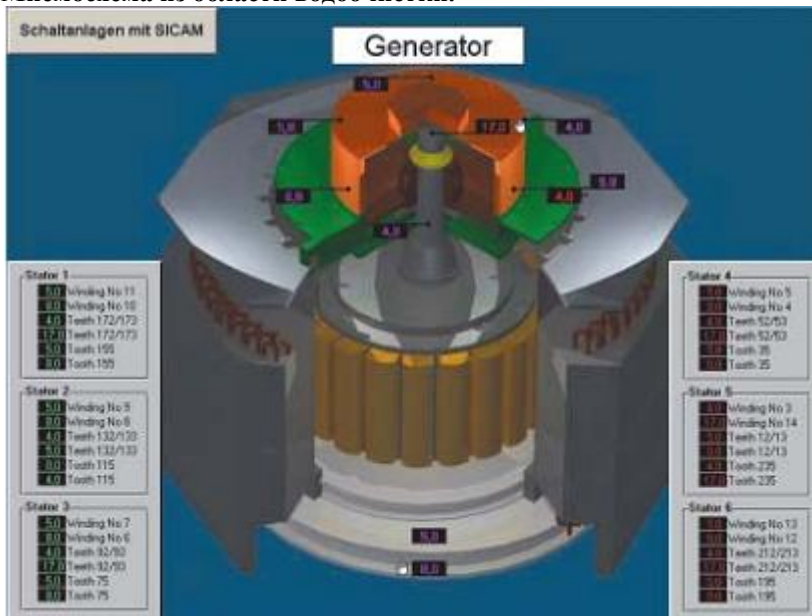
Автомобилестроение (Volvo)



Упаковочная промышленность



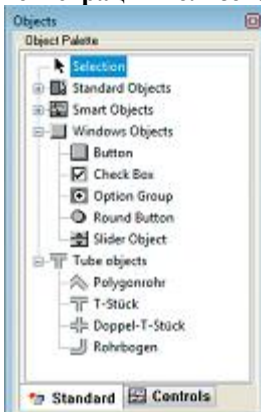
Мнемосхема из области водоочистки.



Распределение энергии (распределительная система SICAM)

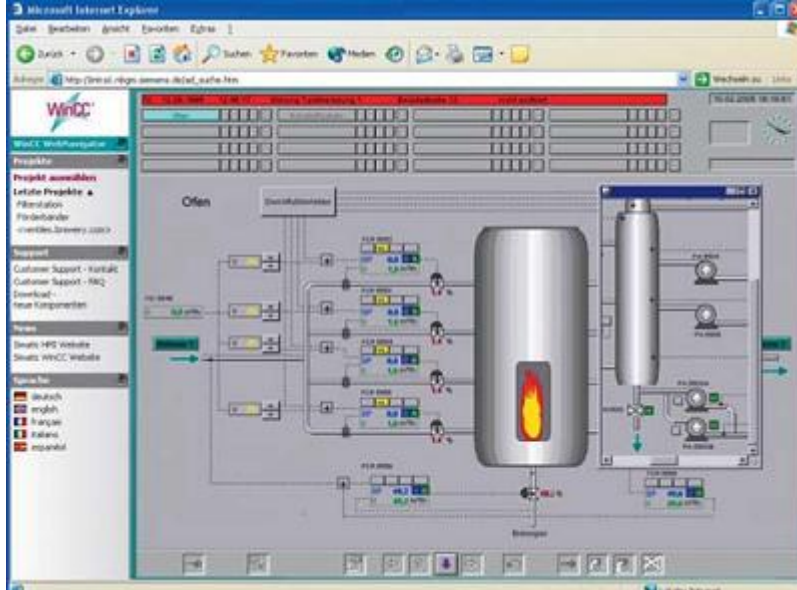


Регистрация пользователя

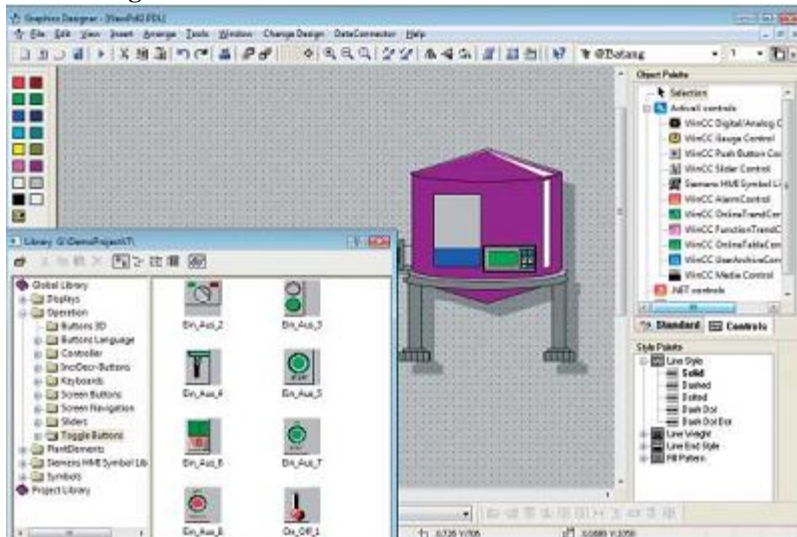


Палитры объектов графического дизайнера WinCC

Статистические функции для архивов значений процесса (Trend Ruler Control)



Мнемосхема системы управления процессом (Basic Process Control) с отображением через WinCC/WebNavigator



Глобальная библиотека объектов SIMATIC WinCC



Проектирование мнемосхем с помощью WinCC Graphics Designer

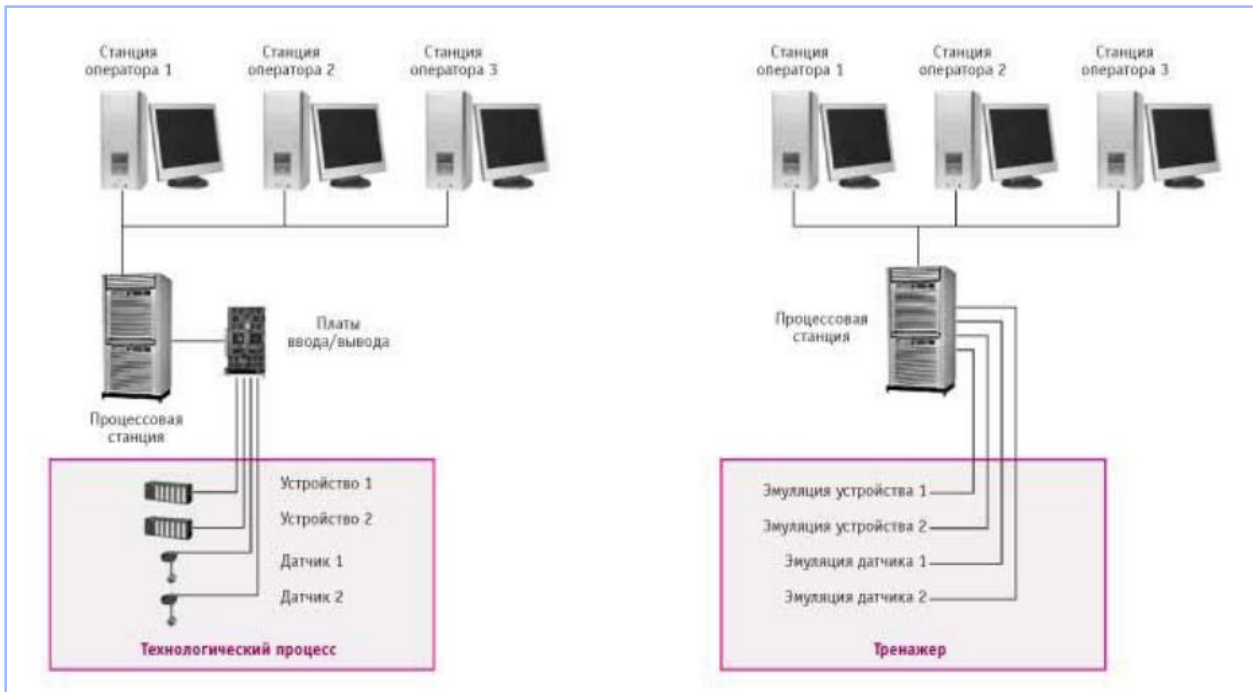
Line	Inst.	Bus.	Bus.	Type	Element Contained	Type	Containing element	Objekt	Property / Action
1	X	K	Tag	-	Action	Applications_AST_illimpf	PLI_Nr=001_vadrom	IssueDisc	
2	X	K	Tag	-	Action	Applications_AST_illimpf	Buton1	IssueDisc	
3	X	K	Tag	-	Action	Applications_AST_illimpf	Buton2	IssueDisc	
4	X	K	Tag	-	Action	Applications_AST_illimpf	m1_vetroz_BIS_retime	IssueDisc	
5	X	K	Tag	Faulty	Property	ASTDevice_General	StatusText4	ForceColor	
6	X	K	Tag	Faulty	Property	ASTDevice_General	StatusText4	Text	
7	X	K	Tag	Faulty	Property	ASTDevice_General	StatusCopy1	Issue	
8	X	K	Tag	Faulty	Action	ASTDevice_General	StatusCopy1	IssueDisc	
9	X	K	Tag	IV	Property	ASTDevice_General	Bar1	Process	
10	X	K	Tag	RSST	Property	VMPQ_Valve_STAIGARD_Fabrik	Alarme01Copy1	Reset	
11	X	K	Tag	RSST	Property	VMPQ_Valve_STAIGARD_Fabrik	Alarme01Copy1	Reset	
12	X	K	Tag	0	Action	Applications_AST_Alarmier	Buton1	IssueDisc	
13	X	K	Tag	1	Action	Applications_AST_Alarmier	Buton1	IssueDisc	
14	X	K	Tag	18	Action	Applications_AST_Alarmier	Buton11	IssueDisc	
15	X	K	Tag	11	Action	Applications_AST_Alarmier	Buton2	IssueDisc	
16	X	K	Tag	15	Action	Applications_AST_Alarmier	Buton3	IssueDisc	
17	X	K	Tag	15	Property	ASTDevice_General	Bar1	Process	
18	X	K	Tag	1877216	Action	Applications_AST_illimpf	Buton3	IssueDisc	
19	X	X	Tag	18	Action	C_322_fabrik_frenza_akt	C_322_fabrik_frenza_akt	ForceOpen	
20	X	X	Tag	18	Property	C_322_fabrik_frenza_akt	Buton1	Visible	
21	X	X	Tag	18_1	Action	C_322_fabrik_frenza_akt	Kochta02	IssueDisc	
22	X	X	Tag	17	Action	C_322_fabrik_frenza_akt	C_322_fabrik_frenza_akt	ForceOpen	
23	X	X	Tag	17	Property	C_322_fabrik_frenza_akt	Buton1	Visible	
24	X	X	Tag	17	Property	C_322_fabrik_frenza_akt	Text_1	OutputValue	
25	X	X	Tag	17	Property	C_322_fabrik_frenza_akt	Text_1	OutputValue	
26	X	X	Tag	17	Property	C_322_fabrik_frenza_akt	Text_1	OutputValue	
27	X	X	Tag	17	Property	C_322_fabrik_frenza_akt	EM-Feed	OutputValue	
28	X	X	Tag	17	Property	C_322_fabrik_frenza_akt	Buton1	Visible	

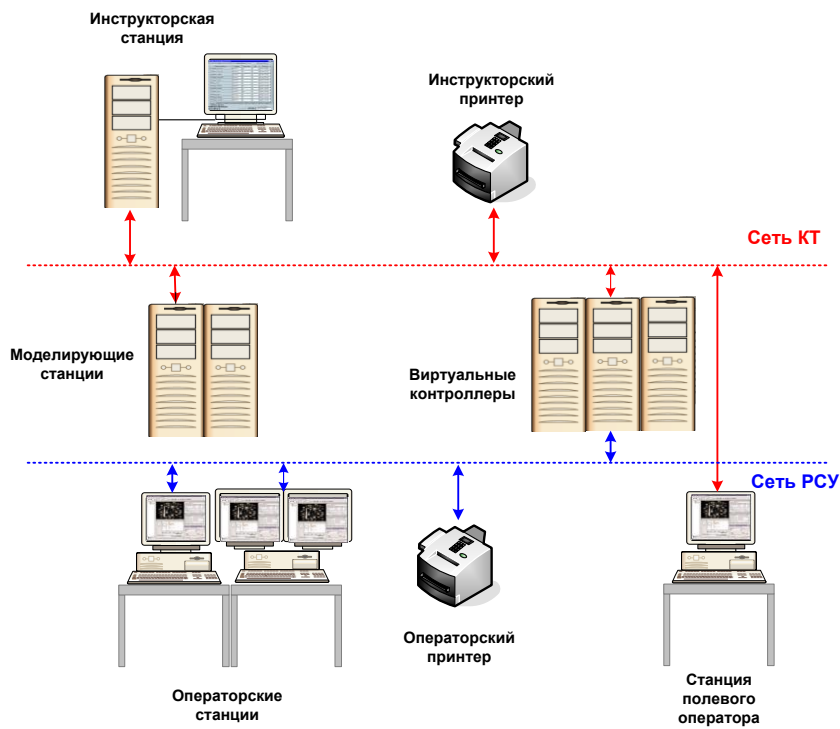
Список перекрестных ссылок для листинга всех используемых переменных, функций и т.д.

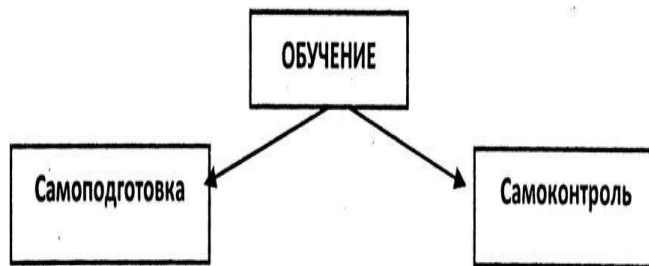
The image shows the WinCC Project Properties dialog box. The 'Data input' section is expanded, showing a table for configuring data sources. A context menu is open over the 'Data input' section, listing options like 'Create project folder', 'Change language', 'Help', 'Info', 'Create WinCC project', 'Establish project connection', 'Add table', 'Statistics', and 'Change RT language'. Below the dialog, a 'Network' window shows a file explorer view of the project files.

Project name	(W1STAVT)WinCC Project
Project type	Multi-user project
Establish connection	Manual
Connection status	Connected
Data input	
Use default values	Yes
-in	68638
message	
	Yes
	Yes
Monitoring	
	Yes
	Yes

Инструментальное средство проектирования для работы с большими проектами







Архитектура системы управления Experion PKS

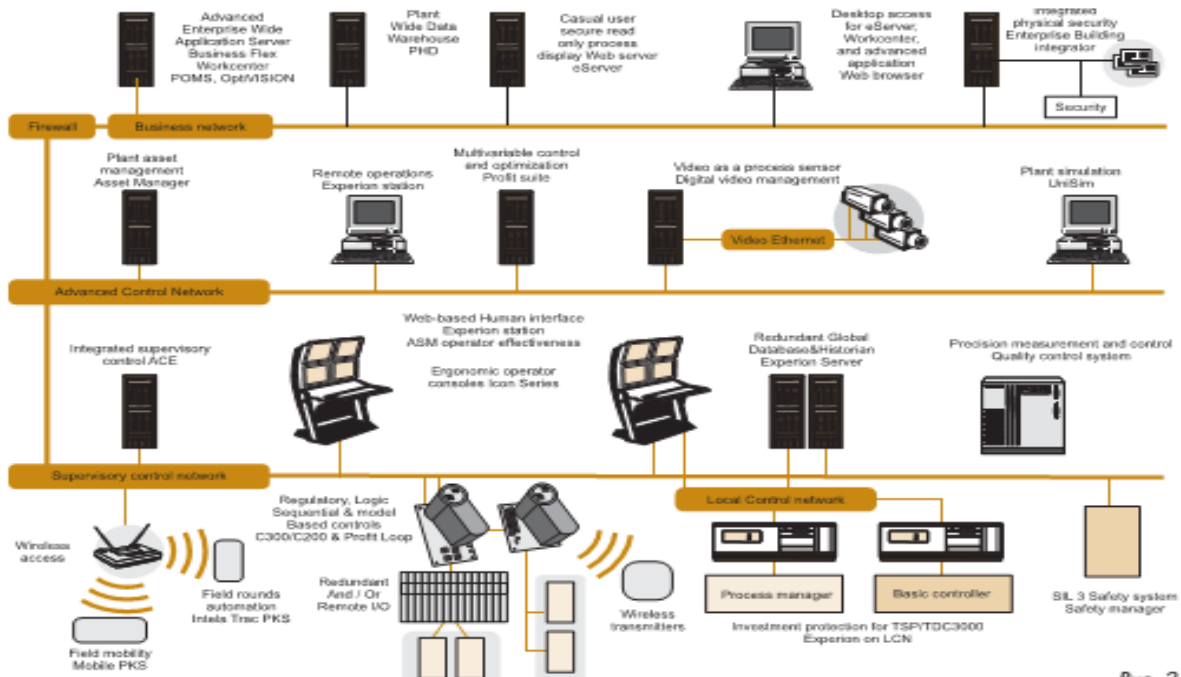
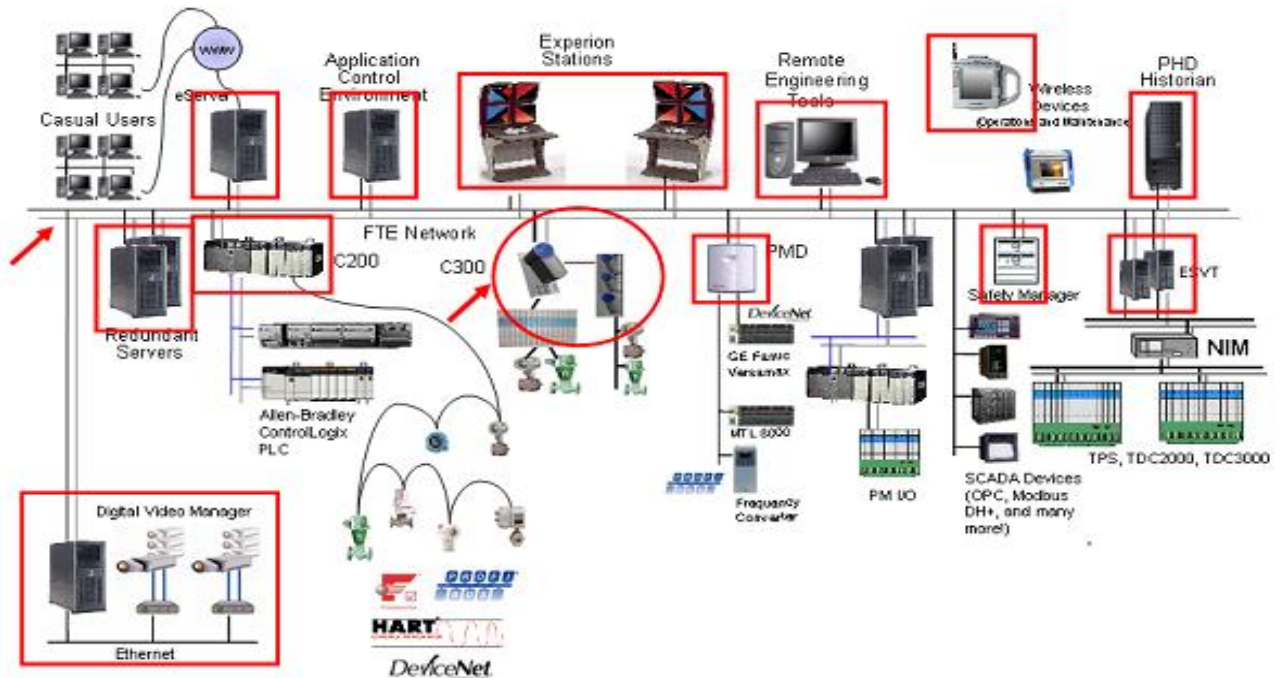


Рис. 2

