

Федеральное агентство по образованию
ГОУ ВПО «Уральский государственный технический университет – УПИ»



А.В. Некрасов

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ МОДЕЛИРОВАНИЯ РАБОТЫ И УПРАВЛЕНИЯ СИСТЕМАМИ ГИДРОПРИВОДА

Учебное электронное текстовое издание
Подготовлено кафедрой «Гидравлика»
Научный редактор: доц., канд. техн. наук В.Ю. Энгель

Методические указания и задания для самостоятельной работы в лаборатории гидроавтоматики для студентов специальности 150802 – «Гидравлические машины, гидроприводы и гидропнеумоавтоматика» всех форм обучения.

Рассмотрены программы для моделирования и управления работой систем гидропривода, установленные на компьютерах лаборатории гидроавтоматики кафедры гидравлики: FluidSim, STEP7-Micro/WIN 32 и Microcomputing. Приведены примеры их использования.

© ГОУ ВПО УГТУ–УПИ, 2006

Екатеринбург
2006

Часть 1.

РАЗРАБОТКА СХЕМ ГИДРОПРИВОДОВ С ЭЛЕКТРОУПРАВЛЕНИЕМ

Цель работы — ознакомиться с общими принципами составления и моделирования работы схем гидропривода с ручным и электрическим управлением с использованием программы FluidSim; сборка и настройка схем гидропривода на экспериментальном стенде.

Общее описание программы FluidSim и принципов составления схем

Программа FluidSim разработана компанией Festo Didactic и предназначена для конструирования схем гидропривода и гидроавтоматики с ручным, электрическим и электронным управлением. Она позволяет исследовать работу составленных схем в различных режимах с использованием мультипликации. Имеется аналогичная программа для моделирования работы пневмопривода.

Используемая нами программа является демонстрационной, что накладывает ряд ограничений на ее использование: невозможно распечатать и сохранить на диске результат работы, отсутствуют справочные материалы фирмы Festo.

Русской версии программы пока не существует, однако это не является серьезной преградой для ее использования. Для построения схем используются стандартные обозначения элементов, а интерфейс программы интуитивно понятен.

Демонстрационная версия программы бесплатная и каждый желающий может установить ее на своем компьютере. Установочная программа имеет объем около 3,5 Мб. Ее можно получить у преподавателя или найти на Интернет-сайте компании Festo по адресу www.festo.com/didactic.

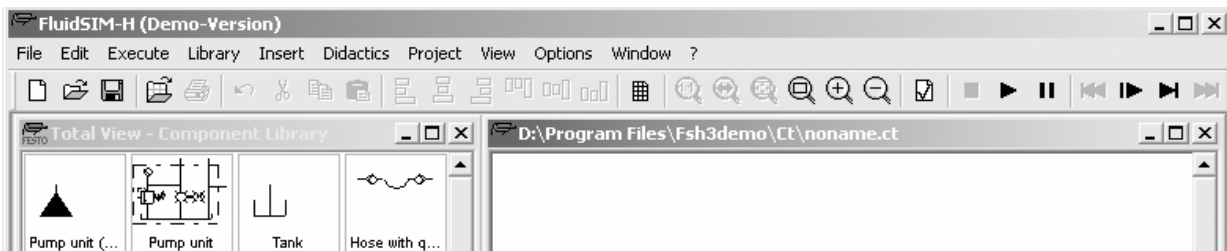


Рис. 1.1. Главное меню FluidSim

После установки программы ее запуск осуществляется из главного меню компьютера (пункт FluidSim-H Demo). Интерфейс программы такой же, как и боль-

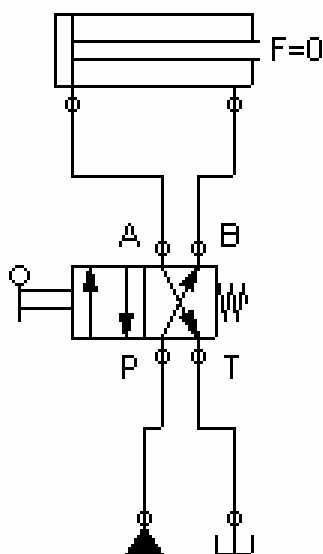


Рис. 1.2. Ручное управление направлением движения гидроцилиндра

шинства приложений Windows. Работа начинается с выбора пункта главного меню программы File-New. На экране размещено два окна - библиотека компонентов (Component Library) и окно проекта (рис. 1-1). Для того, чтобы поместить тот или иной элемент библиотеки в окно проекта, достаточно выделить его левой клавишей мыши и, удерживая ее, перетащить в окно проекта.

В качестве первого проекта составим схему для обеспечения возвратно-поступательного движения гидроцилиндра с ручным изменением направления движения. Результат показан на рис. 1-2. Построение этой схемы осуществляется в следующей последовательности.

1. Найдите в библиотеке компонентов и последовательно установите в окне проекта гидроцилиндр двойного действия (Double acting cylinder), распределитель (4/n way valve), насос (Pump unit) и гидробак (Tank). Особое внимание обратите на распределитель. В тот момент, когда Вы установите его в окне проекта изображение будет отличаться от приведенного на рис. 1-2. Двойным щелчком левой клавиши на изображении распределителя откройте окно для уточнения его конструкции и вида управления (рис. 1-3).

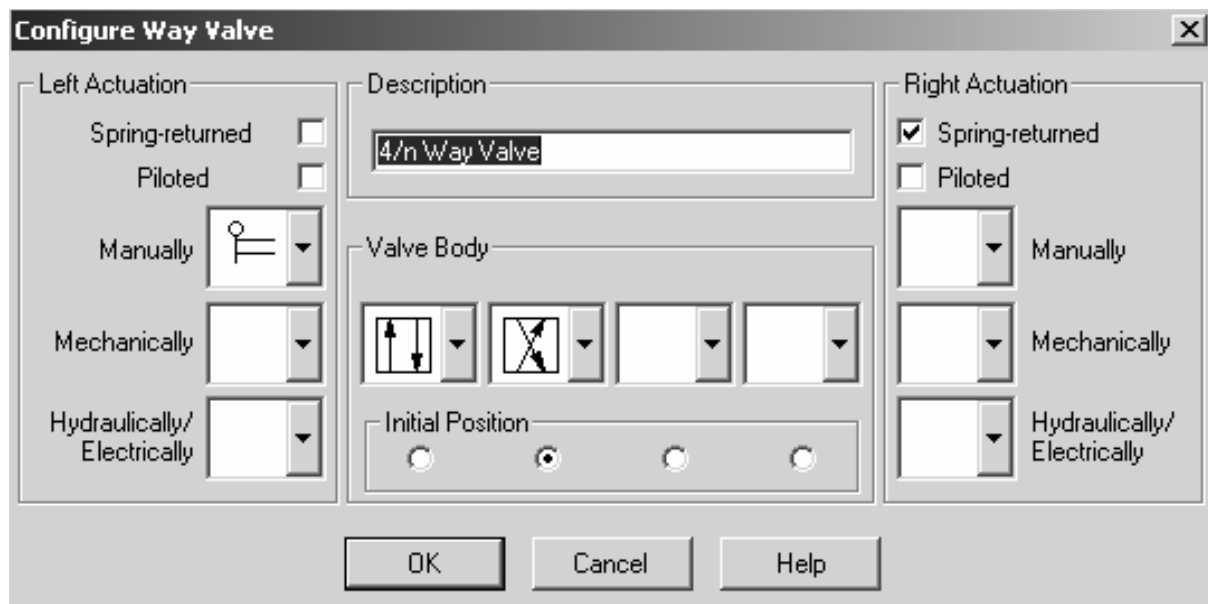


Рис. 1.3. Окно конфигурирования распределителя

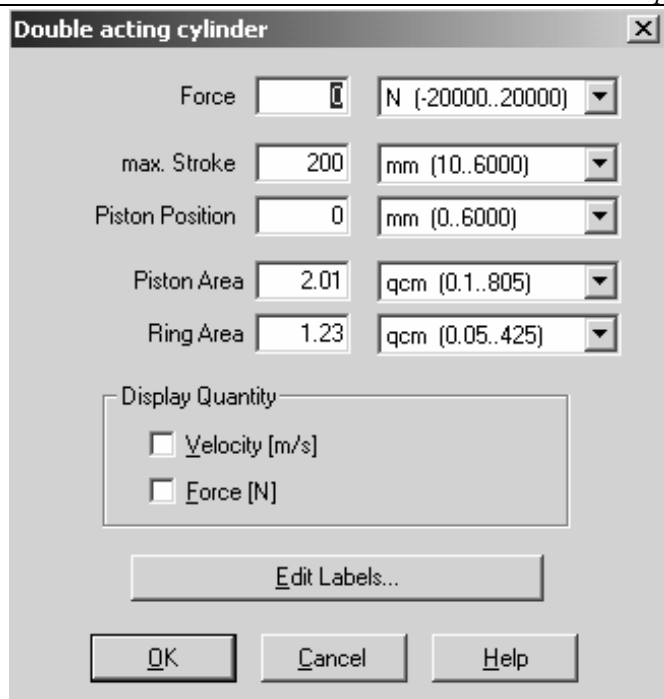


Рис. 1.4. Окно параметров цилиндра двойного действия

можно отметить необходимость вывода на экран текущих значений величины силы (Force) и скорости (Velocity) движения. С помощью кнопки Edit Labels можно открыть дополнительное окно, где при необходимости задаются метки промежуточных положений поршня.

4. Двойным щелчком по изображению насоса вызывается окно, где можно задать его рабочее давление и величину подачи (рис. 1-5).

5. Для изображения гидролинии достаточно подвести курсор к точке какого-либо элемента схемы, помеченной окружностью (при этом она меняет цвет), нажать левую клавишу мыши и, удерживая ее, подвести курсор к другой аналогичной точке или гидролинии (при этом форма курсора изменяется).

Для проверки работы схемы следует нажать кнопку . Программа перейдет в режим анимации, демонстрируя работу схемы. В это время вы можете управлять переключением распределителей, подводя к ним курсор и нажимая левую клавишу мыши. Для остановки работы нажмите кнопку . Скорость анимации можно уменьшить, увеличив значение параметра Slow-motion factor

2. С помощью однотипных меню, расположенных слева и справа, задайте тип управления. В центральной части окна определите позиции распределителя и начальную позицию (Initial Position).

3. Двойным щелчком по изображению гидроцилиндра откройте окно для задания параметров поршня (рис. 1-4). Имеется возможность задать внешнее усилие на шток поршня (Force), ход поршня (max. Stroke), начальное положение поршня (Piston Position), полную (Piston Area) и кольцевую (Ring Area) площади поршня. В разделе Display Quantity

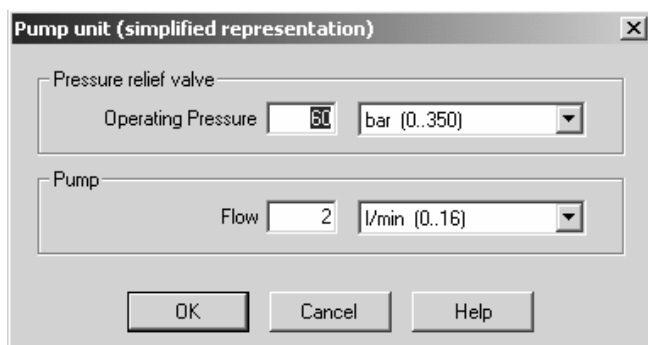


Рис. 1.5. Окно параметров насоса

(рис. 1-6). Доступ к этому окну осуществляется через пункт главного меню Options-Simulation.

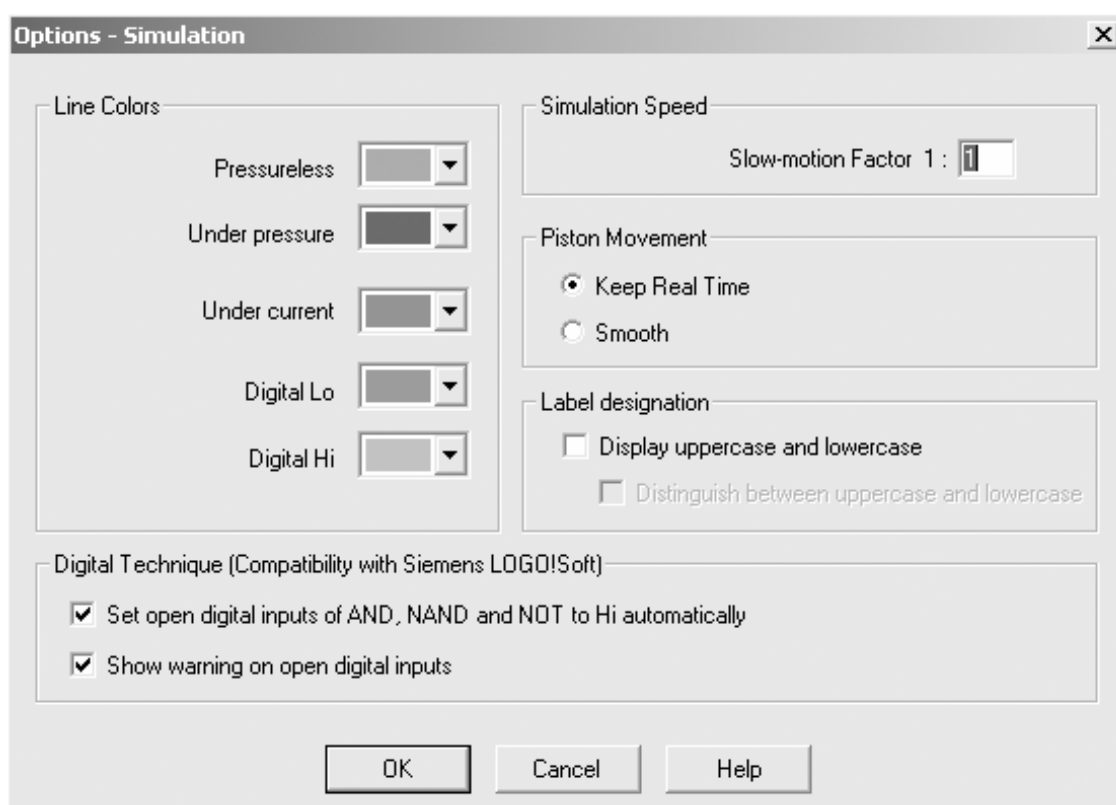


Рис. 1.6. Окно установки параметров режима анимации

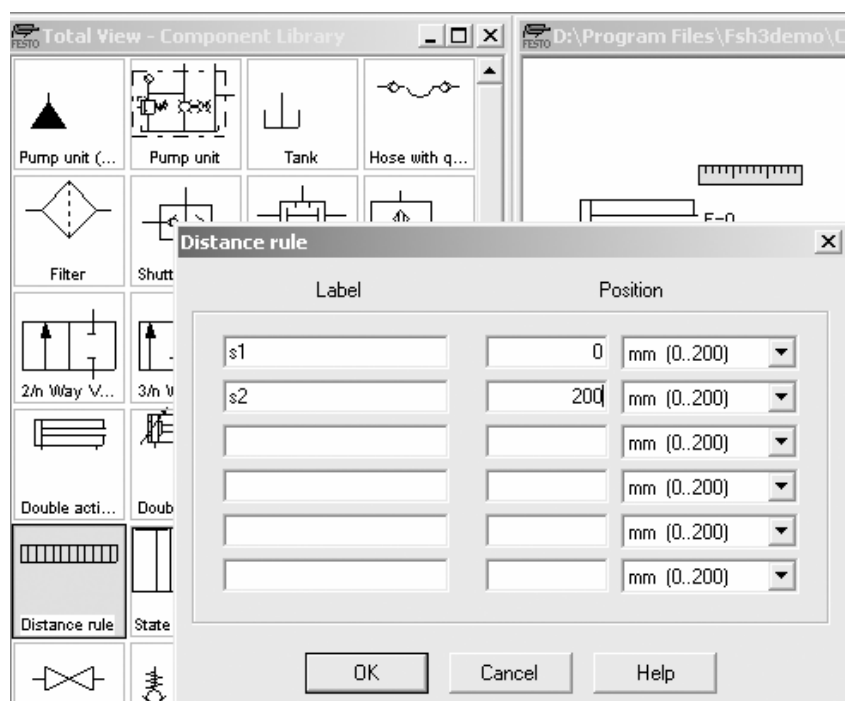


Рис. 1.7. Задание имен и координат датчиков

Рассмотрим особенности разработки схем с электроуправлением. В качестве первого примера построим схему, реализующую возвратно-поступательное движение цилиндра. Прежде всего установить линейку с датчиками (Distance rule),

двойным щелчком откройте окно ее свойств (рис. 1-7). Здесь нужно задать имена датчиков и их положение на линейке.

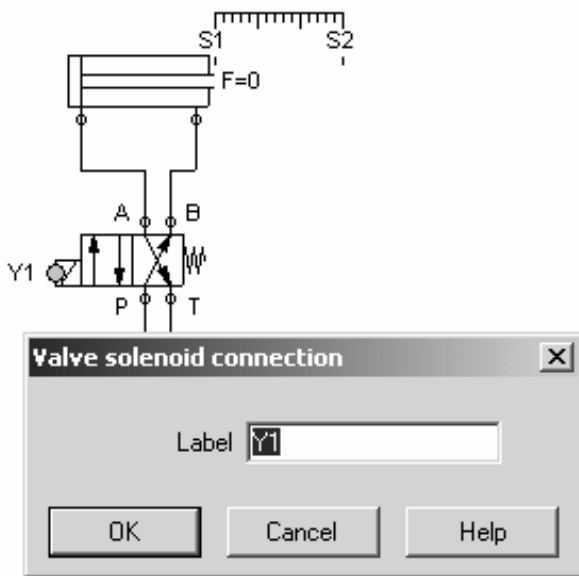


Рис. 1.8. Задание имени управляющего соленоида распределителя

Задайте электрический тип управления распределителем и имя, под которым в электрической схеме будет фигурировать его соленоид — Y1(рис. 1-8).

Теперь приступим к проектированию электрической схемы. Установите контакты для подачи питания на схему и реле (рис. 1-9). По аналогии с соленоидом задайте имя реле. Забегая вперед, заметим, что это имя будут иметь и контакты реле.

Установите два разомкнутых контакта и соедините элементы схемы так, как показано на рис. 1-10. Один из контактов является контактом реле K1, другой – датчиком S1 (рис. 1.11). Обратите внимание, что в

момент задания имени контакта датчика S 1, программа покажет его на схеме замкнутым (об этом говорит значок Π рядом с контактом), поскольку шток цилиндра находится именно под ним. В нижней части схемы мы увидим контакты реле. Уже в таком виде схема вполне работоспособна. Мы можем убедиться в этом, включив режим анимации. Конечно, в этом случае переключать распределитель нужно будет вручную. Обратите внимание, что в момент начала движения штока контакт S1 размыкается, т.е. он реагирует на положение конца штока.

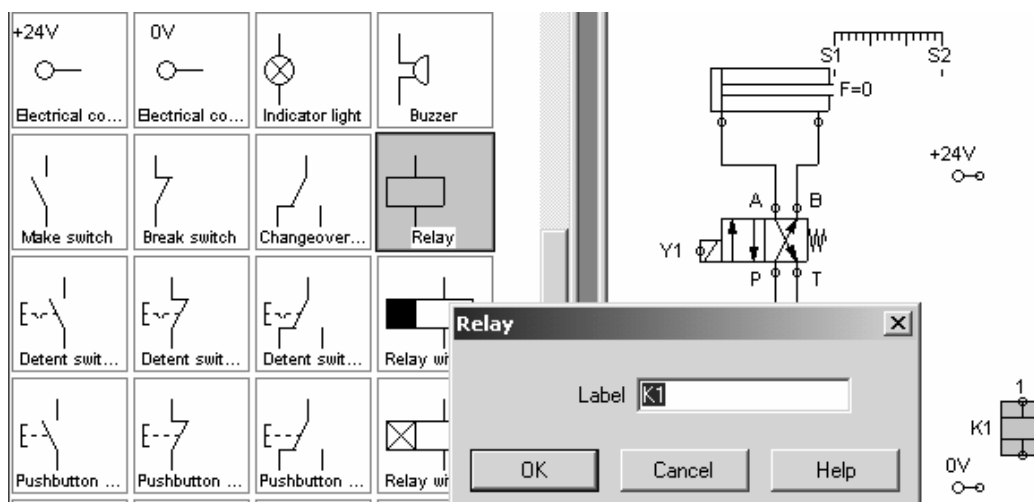


Рис. 1.9. Установка реле

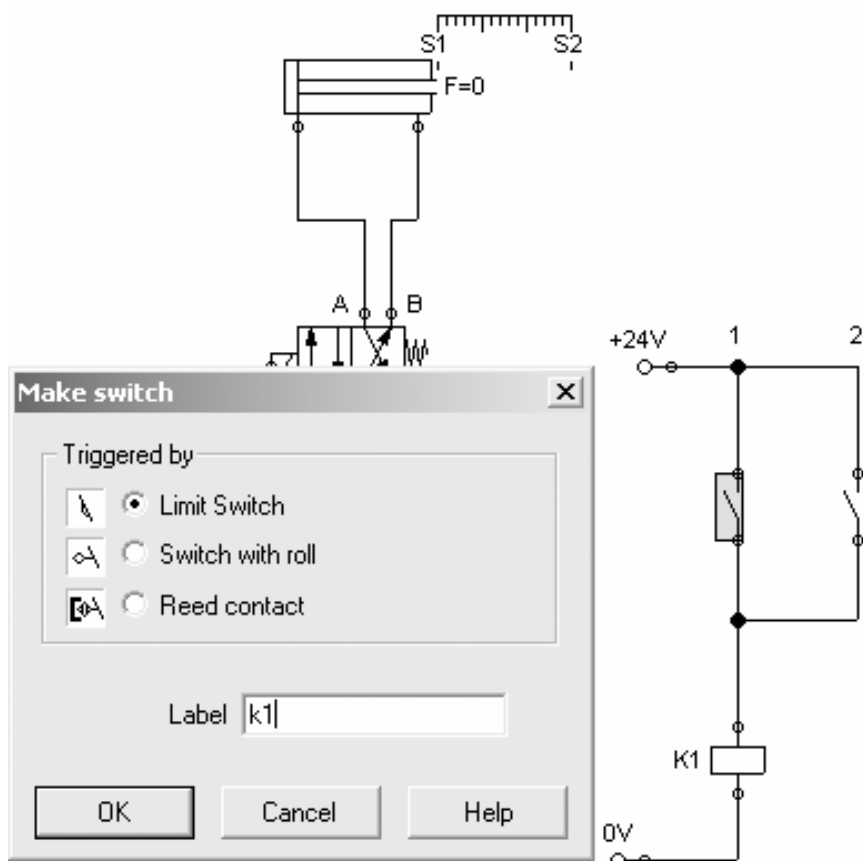


Рис. 1.10. Задание имени контакта

Теперь установите второй датчик S2. Пусть его контакт работает на замыкание (в позиции, показанной на рис. 1-10, он замкнут). Схема показана на рис. 1.12. Рекомендуем снова проверить ее работу, обратив внимание на отличия от предыдущей.

Остается разместить на схеме обмотку соленоида распределителя. Она должна включаться при срабатывании реле. Схема показана на рис. 1.13. Обратите внимание, что имя этого элемента на схеме совпадает с именем на гидравлической схеме (у нас – Y1). Убедитесь, что схема работает верно.

Программа позволяет строить графики изменения параметров привода (перемещение поршня, давление в какой-либо точке и т. д.). Для построения графика сначала разместите в окне проекта заготовку для статической диаграммы (рис. 1.15). Далее просто перетащите элемент привода, например, поршень на диаграмму (рис. 1.16). При работе программы в режиме анимации мы увидим график изменения перемещения поршня во времени. На одной диаграмме можно строить сразу несколько графиков.

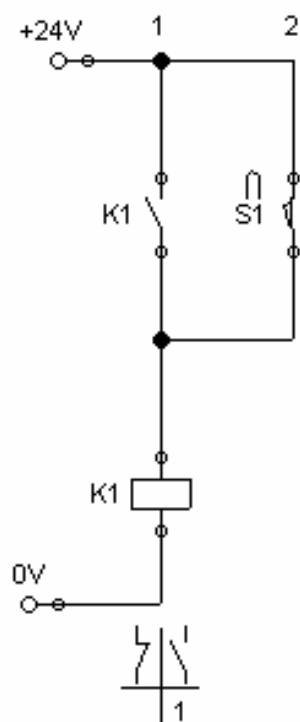


Рис. 1.11. Схема включения реле K2

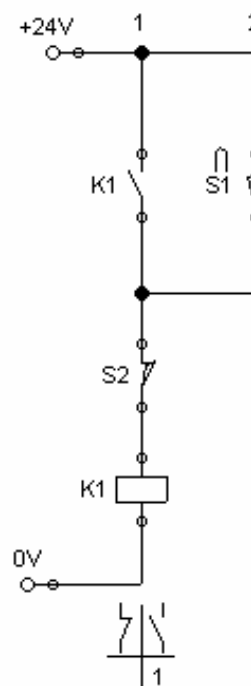


Рис. 1.12. Схема включения датчика S2

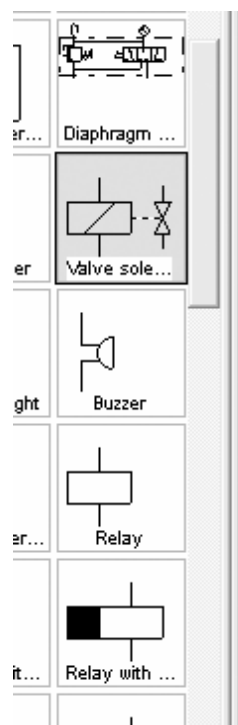


Рис. 1.13. Схема включения соленоида распределителя

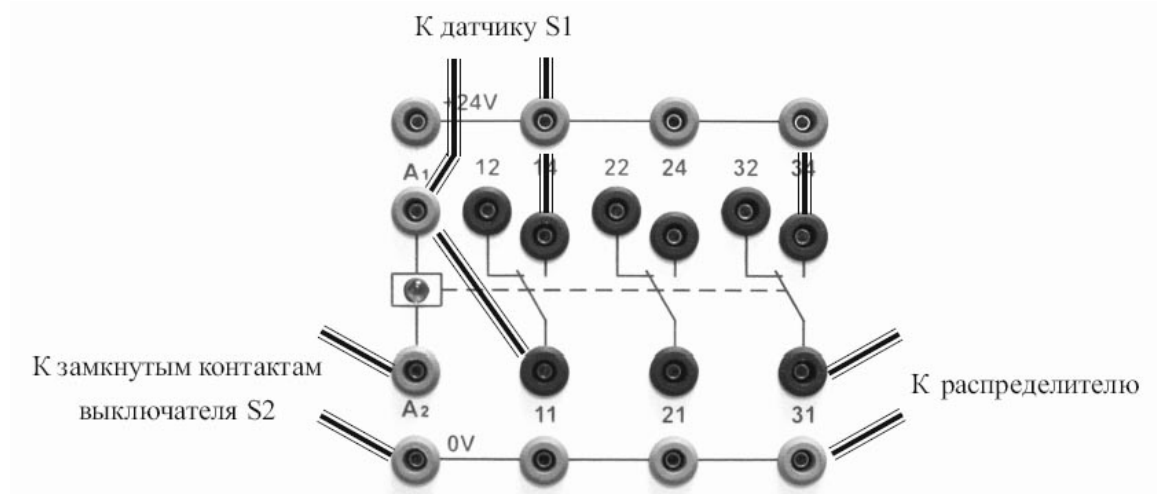


Рис. 1.14. Монтажная схема

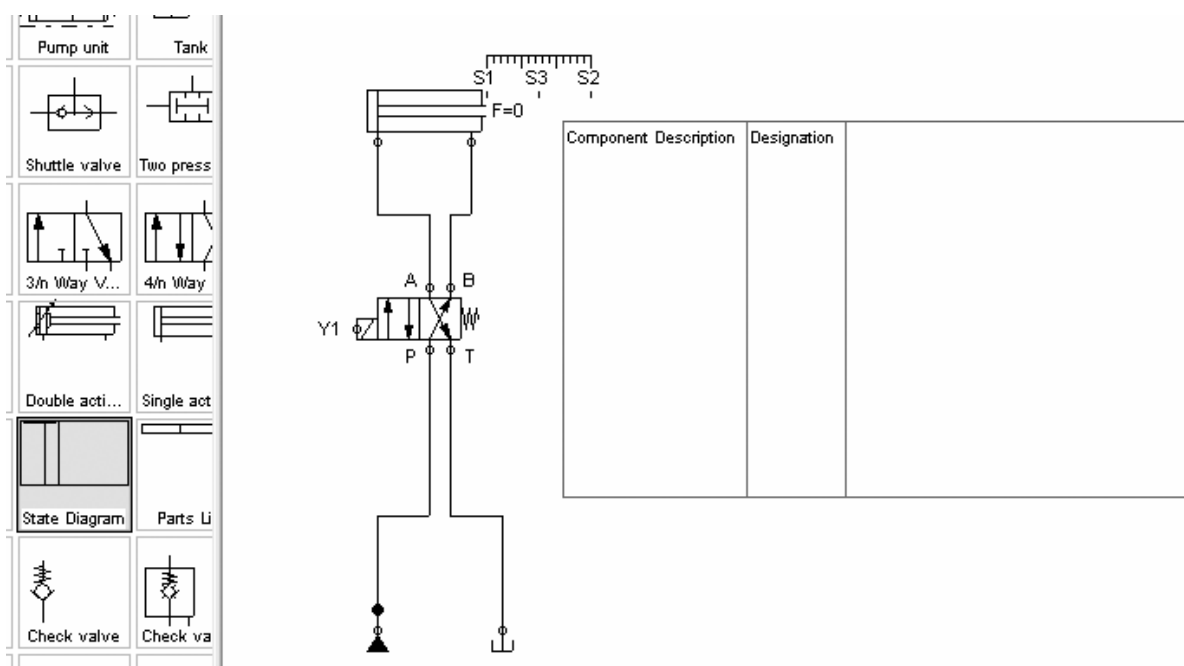


Рис. 1.15. Размещение статической диаграммы в окне проекта

Указания по сборке схемы на стенде

Соберите гидравлическую схему на экспериментальном стенде. В качестве датчика S1 используйте индуктивный или емкостный датчик. Положение датчика по вертикали при необходимости следует отрегулировать, учитывая, что зона его чувствительности не превышает 5–7 мм. В качестве датчика S2 нужно использовать электромеханический концевой выключатель. Убедитесь, что при работе гидроцилиндра датчики не создадут преграды движению его поршня, в противном случае датчики будут повреждены.

Соберите электрическую схему, используя монтажную схему. При присоединении емкостного или индуктивного датчика следует иметь в виду, что в мо-

мент его срабатывания происходит замыкание цепи между красным и зеленым кабелями.

Включение установки производится только по разрешению преподавателя.

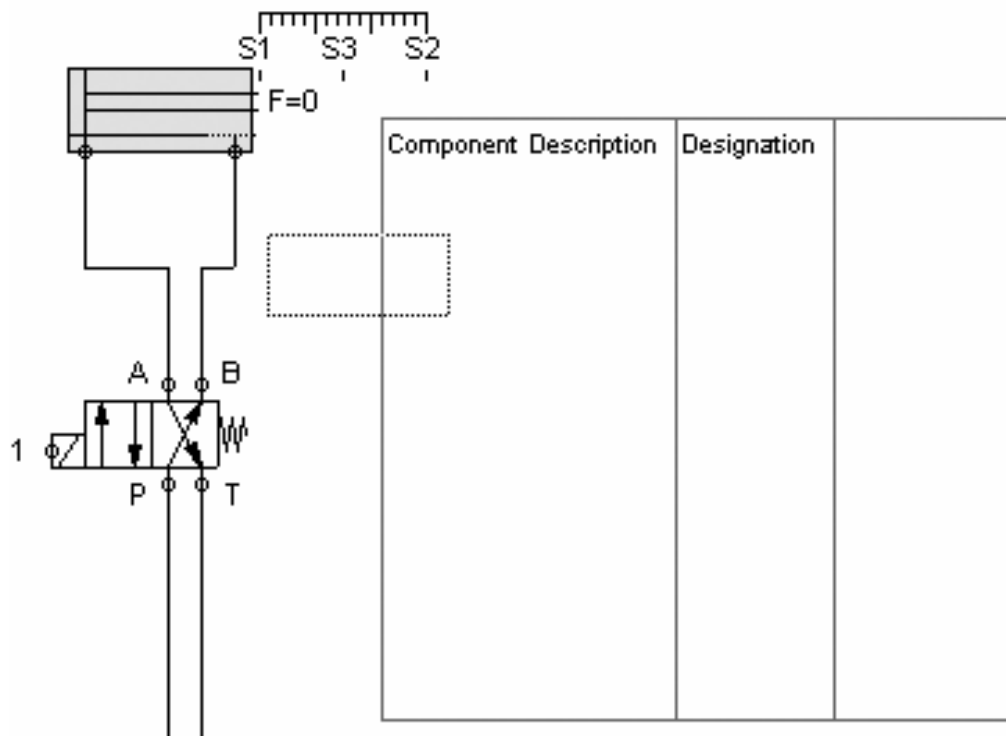


Рис. 1.16. Перемещение поршня в область диаграммы

Задания для самостоятельной работы

Задание 1.1

Предложите схему управления, реализующую возвратно-поступательное движение гидроцилиндра, в случае, когда оба концевых выключателя (датчика) S1 и S2 (см. пример) работают на замыкание. Соберите гидравлическую и электрическую схемы привода на стенде и испытайте в работе.

Состав отчета: задание, гидравлическая, принципиальная и монтажная электрические схемы.

Задание 1.2

Разработайте схему электроуправления гидроприводом с дроссельным регулированием, работающего по схеме: быстрый подвод, рабочий ход, быстрый отвод (рис. 1-17). Определите степень открытия дросселя, при которой скорость рабочего хода будет в два раза меньше скорости подвода.

Быстрый подвод осуществляется на расстоянии 100 мм от датчика S1 до S3; рабочий ход — на расстоянии 100 мм от датчика S3 до S2. Все датчики работают на замыкание.

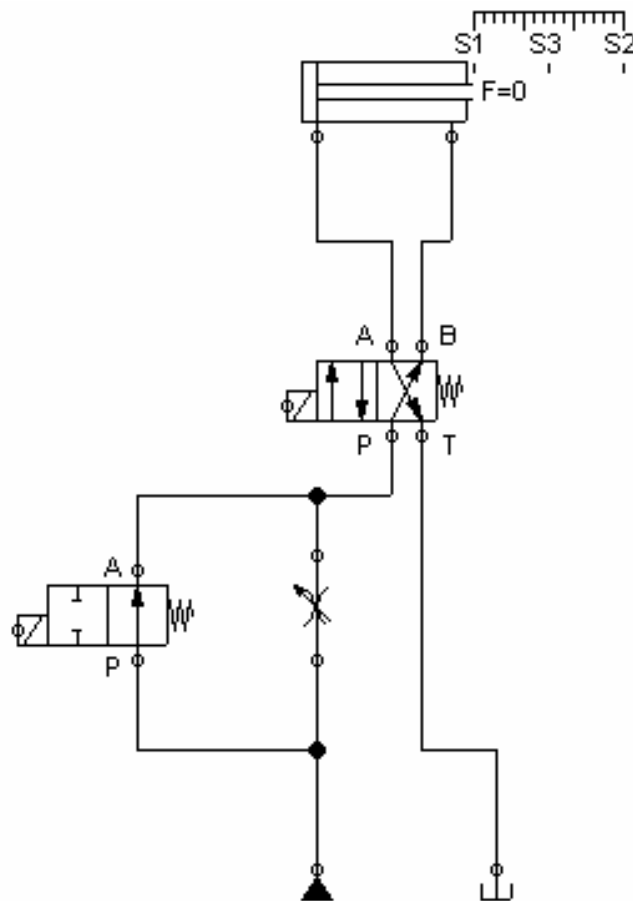


Рис. 1.17. Гидравлическая схема к заданию 1.2

Соберите гидравлическую и электрическую схемы привода на стенде и испытайте в работе, при этом экспериментально подберите требуемую степень открытия дросселя.

Состав отчета: задание, гидравлическая, принципиальная и монтажная электрические схемы, расчетный график зависимости скорости рабочего хода от степени открытия дросселя.

Задание 1.3

Разработайте схему электроуправления гидроприводом с дроссельным регулированием, работающего по схеме: быстрый подвод, рабочий ход, быстрый отвод. Дроссель должен быть включен параллельно гидроцилиндру (рис. 1-18). Определите степень открытия дросселя, при которой скорость рабочего хода будет в два раза меньше скорости подвода.

Быстрый подвод осуществляется на расстоянии 100 мм от датчика S1 до S3; рабочий ход – на расстоянии 100 мм от датчика S3 до S2. Все датчики работают на замыкание.

Соберите гидравлическую и электрическую схемы привода на стенде и испытайте в работе, при этом экспериментально подберите требуемую степень открытия дросселя.

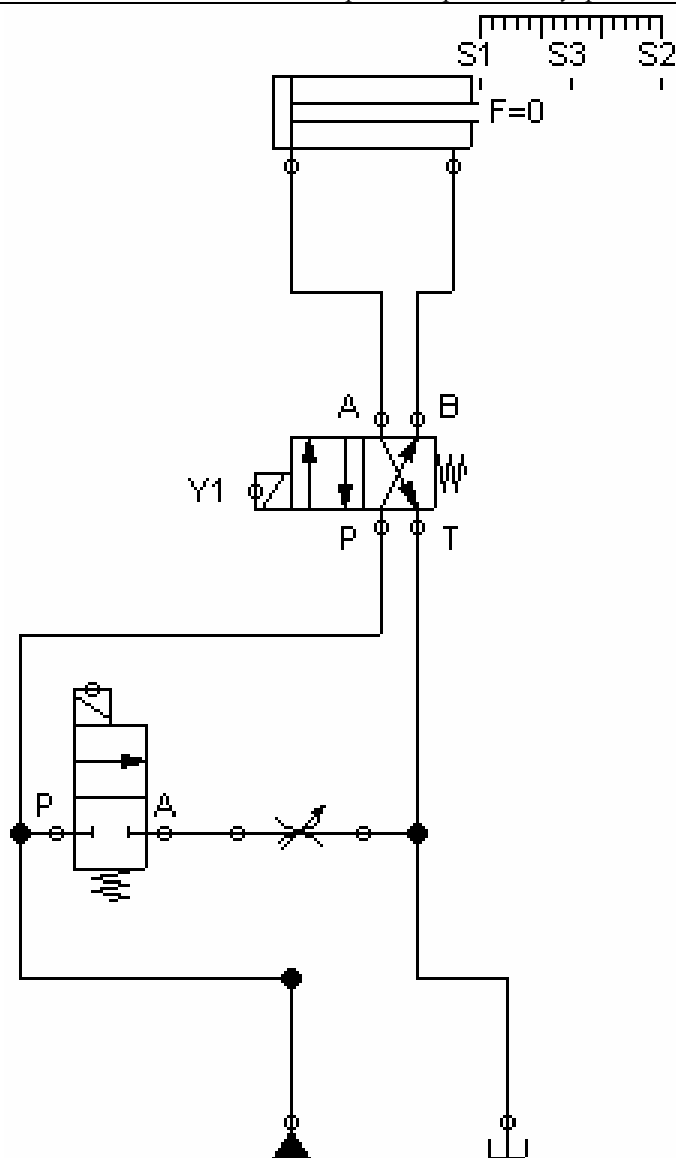


Рис. 1-18. Гидравлическая схема к заданию 1.3

Состав отчета: задание, гидравлическая, принципиальная и монтажная электрические схемы, расчетный график зависимости скорости рабочего хода от степени открытия дросселя.

Часть 2.

РАЗРАБОТКА СХЕМ ГИДРОПРИВОДОВ С ПРОГРАММНЫМ УПРАВЛЕНИЕМ

Цель работы — знакомство с общими принципами составления программ управления гидроприводом с использованием контроллеров S7-222 и программы STEP7-Micro/WIN 32; сборка и настройка схем гидропривода на экспериментальном стенде.

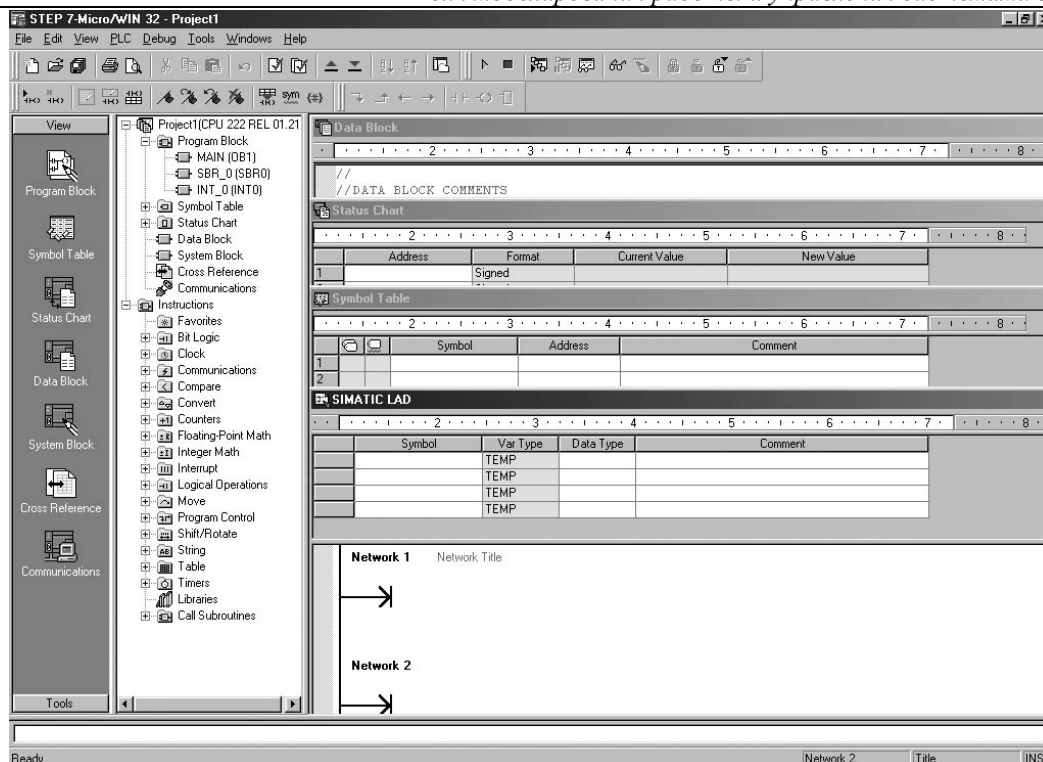


Рис. 2-1. Вид главного окна программы STEP7-Micro/WIN 32

Общее описание программы STEP7-Micro/WIN 32

Программа STEP7-Micro/WIN 32 разработана компанией Siemens. Она предоставляет пользователю широкие возможности, включая создание, редактирование и отладку прикладных программ, их распечатку на принтере, установку связи с контроллером для загрузки или выгрузки программы и наблюдения за процессом ее исполнения.

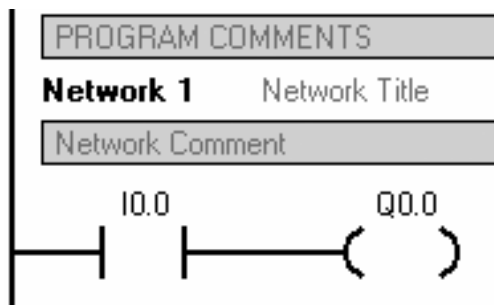


Рис. 2-2. Ваша первая LAD-программа

Пакет STEP7-Micro/WIN 32 разработан для использования на персональных компьютерах с операционной системой Windows и поддерживает программирование на языках релейно-контактных схем (LAD), функциональных схем (FBD) и мнемонического кодирования (STL).

После запуска пакета на экран монитора выводится графическая оболочка программы, реализованная в традиционном стиле Windows (рис. 2.1).

В верхней части окна размещаются главное меню и две панели инструментов в виде кнопок, которые реализуют как стандартные операции Windows, так и собственные функции пакета. Пользователь может настраивать эти панели по своему усмотрению.

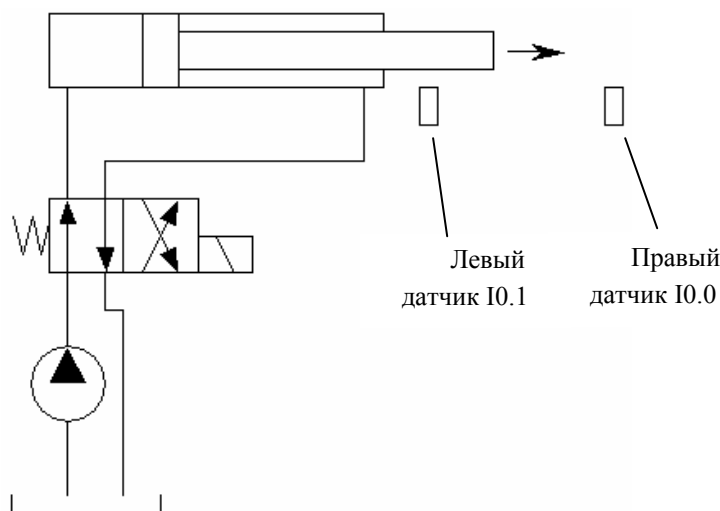


Рис. 2-3. Схема для программирования возвратно-поступательного движения поршня

В левой части главного окна располагается панель управления, объединяющая две группы управляющих кнопок, реализующих определенные функции пакета.

- Группа View («Вид») — содержит кнопки вывода на экран блока программы, таблицы символов, диаграммы состояний, блока данных, блока системных параметров, перекрестных ссылок и коммуникационных параметров;

- Группа Tools («Сервис») — содержит кнопки вызова Мастера инструкций (Instruction Wizard) и Мастера текстового дисплея. Текстовый дисплей – это специальное устройство (например, TD 200), устанавливаемое часто в непосредственной близости от объекта управления. С его помощью можно контролировать его параметры и давать команды контроллеру, предусмотренные программой. Мастер инструкций используется при программировании сложных инструкций, например инструкций коммуникационного обмена данными или непрерывного регулирования.

В окне дерева инструкций, расположенном справа от панели управления, отображаются в виде древовидной структуры все объекты данного проекта, а также представлены все инструкции, доступные в используемом редакторе языка (LAD, FBD, STL). Здесь же представлены папки всех компонентов проекта, каждую из которых можно открывать и изменять; можно также добавлять в дерево новые папки. Инструкции, которые можно использовать в ходе работы над проектом, сгруппированы по категориям в папках, открывающихся при выполнении двойного щелчка левой кнопки мыши (далее просто «двойной щелчок») по названию папки (или при выполнении щелчка по значку «+» слева от названия соответствующей папки). При работе с графическим редактором (LAD или

FBD) нужную инструкцию можно вставить в разрабатываемую программу с помощью мыши методом «drag and drop» («перетянуть и отпустить»).

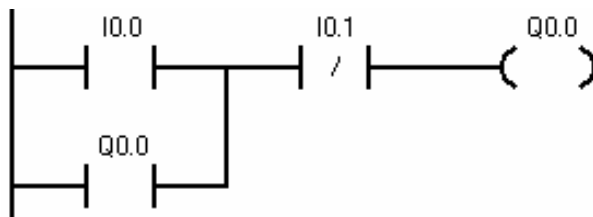


Рис. 2-4. LAD-программа возвратно-поступательного движения поршня

Для создания нового проекта и открытия существующих проектов предназначены команды New («Создать») и Open («Открыть»), входящие в меню File.

Для удобства работы с пакетом STEP 7-Micro/WIN доступ к поддерживаемым функциям реализуется также посредством панелей инструментов, представляющих собой набор специализированных кнопок.

- Стандартная панель инструментов (File Toolbar) — содержит кнопки стандартных функций приложений Windows и кнопки общих функций STEP7-Micro/WIN, из которых наиболее важными для знакомства с программным пакетом являются функции компиляции прикладной программы и загрузки ее в память контроллера.

- Панель инструментов отладки (Debug Toolbar) — включает кнопки пуска и остановки программы, мониторинга значений переменных, а также ряд кнопок специальных функций отладки.

- Панели инструментов редакторов языков программирования LAD, FBD и STL — отображаются в главном окне при выборе соответствующей команды в меню View, содержат кнопки функций, реализуемых конкретным редактором, включая кнопки инструкций (панель инструментов редактора STL содержит только кнопки для работы с метками).

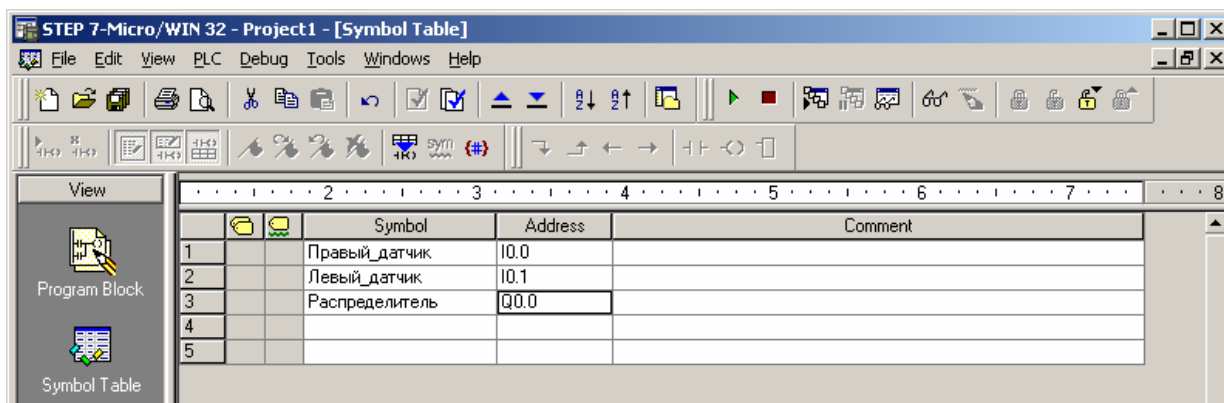


Рис. 2-5. Таблица символов

Работу над проектом можно осуществлять как при подключенном к компьютеру контроллере, так и без него.

Проверим работоспособность контроллера. Для это необходимо включить блок питания контроллера и с помощью кнопки  главного меню осуществить его запуск. Кнопка  останавливает контроллер. На Ваши действия контроллер должен реагировать переключением светодиодов RUN и STOP. Если Ваш опыт закончился успешно, можно переходить к разработке первой очень простой программы.

Рассмотрим реакцию контроллера на единственный входной сигнал от емкостного датчика. Для этого присоедините датчик к входным контактам. Кабели

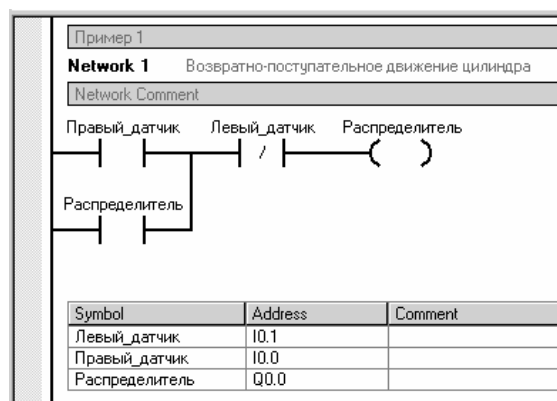


Рис. 2-6. LAD-программа с символьными переменными

питания (красный и синий) можно подключить к любым клеммам соответствующих цветов, а зеленый кабель — к зеленому гнезду I0.0.

Процесс создания программы проиллюстрируем, взяв за основу язык программирования LAD. Работа над новой программой начинается с выбора команды New в меню File главного меню. Далее предлагаем Вам выполнить следующую последовательность действий:

1. Откройте окно редактора LAD, если он еще не открыто.
2. Поместите курсор в первую позицию цепочки Network 1.
3. Щелкните по кнопке с пиктограммой контакта  на панели инструментов редактора.
4. В раскрывающемся списке выберите нормально разомкнутый контакт.
5. Щелчком мыши активизируйте поле идентификатора контакта (операнда) и впишите в него значение I0.0.
6. Нажмите клавишу Enter.
7. Щелкните по кнопке с пиктограммой выхода цепочки  на панели инструментов редактора.
8. В раскрывающемся списке выберите обмотку реле.
9. Щелчком мыши активизируйте поле идентификатора обмотки (операнда) и впишите в него значение Q0.0.
10. Нажмите клавишу Enter.

Результат Вашей работы должен быть таким, как показано на рис. 2.2. Теперь поместим в программу в память контроллера с помощью кнопки  на панели инструментов. После проверки правильности синтаксиса программы и ее за-

грузки в контроллер вам будет выдано соответствующее сообщение. Запустите контроллер с помощью кнопки . При прикосновении к датчику на контроллере должны загораться два светодиода I0.0 и Q0.0

Программа возвратно-поступательного движения гидроцилиндра

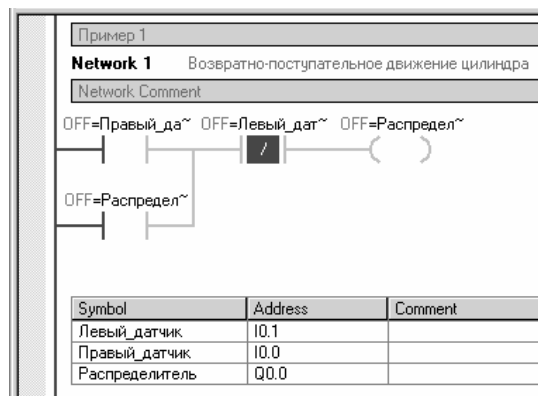


Рис. 2-7. Step7 в режиме Program Status

Следующий пример — программа, обеспечивающая возвратно-поступательное движение гидроцилиндра. Гидравлическая схема и расположение датчиков показаны на рис. 2.3. Назначим адреса входных и выходных переменных. Для удобства их целесообразно записывать в виде таблицы (табл. 2-1). Программа для контроллера достаточно проста (рис.2-4). Если контакт I0.0 замыкается (срабатывает правый датчик), а контакт I0.1 при этом не замкнут (левый датчик не срабатывает), то включается механизм Q0.0 (распределитель). Цилиндр начинает движение влево. Тем не менее, отключения распределителя (Q0.0) не происходит, поскольку соответствующий ему контакт Q0.0 теперь уже замкнут. Отключение произойдет только при замыкании контакта I0.1 (срабатывает левый датчик). Цикл повторяется до остановки контроллера.

Таблица 2.1

Адреса переменных

№	Датчик	Адрес
1	Правый датчик положения	I0.0
2	Левый датчик положения	I0.1
№	Исполнительные механизмы	Адрес
1	Гидрораспределитель	Q0.0

Для улучшения восприятия программы и ее анализа можно вместо адресов переменных использовать их символьные имена. Иными словами, вы можете создать таблицу типа табл. 2-1 в самой программе. Для этого необходимо открыть окно таблицы символов с помощью кнопки и заполнить ее (рис. 2-5).

После возврата в окно программы (Program Block) и загрузки программы в контроллер заданные имена переменных отразятся на экране (рис. 2-6). При необходимости вы можете снабдить программу комментариями.

Для того чтобы проверить работу программы, запустите контроллер и переведите STEP7 в режим Program Status кнопкой . В этом режиме программа в реальном времени будет показывать на экране все изменения переменных (рис. 2-7).

Счетчики импульсов

Счетчики являются обязательной принадлежностью любого контроллера. Они позволяют реализовать управление объектом в режиме реального времени. Для нас с вами это всего лишь программные средства, обращение к которым осуществляется с помощью специальных инструкций.

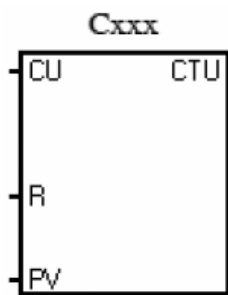


Рис. 2.8. Инструкция суммирующего счетчика

Рассмотрим следующую задачу. Пусть имеется один датчик (например, емкостный) и один исполнительный механизм. Пусть для определенности это обычная электролампа. Требуется составить программу, которая будет включать лампу после, например, третьего срабатывания датчика. Назначим 10.0 как адрес входного сигнала датчика, а Q0.0 как выходной сигнал для включения электролампы.

Познакомимся с инструкцией одного из счетчиков SIMATIC (всего имеется три вида счетчиков), который будет использоваться при решении поставленной задачи. На рис. 2.8 показано изображение в LAD-программе так называемого суммирующего счетчика. В верхней части указывается номер счетчика Sxxx, где xxx — целое число от 0 до 255 (именно столько счетчиков можно использовать в программе). Внутри прямоугольника (бокса) указывается тип счетчика. В нашем случае — STU — суммирующий по возрастанию. CU — вход счетчика. Значение счетчика будет увеличиваться на единицу всякий раз, когда вход изменяет свое состояние с 0 на 1. R — входная величина, управляющая сбросом счетчика в нулевое состояние. Сброс осуществляется при переключении R в состояние 1. PV — так называемая уставка счетчика (в нашем примере — 3). Именно в тот момент, когда значение счетчика станет равным уставке переменная Sxxx меняет свое значение с 0 на 1. Максимальное значение суммирующего счетчика 32767. Программа, которую Вам предстоит ввести показана на рис. 2.9. Ее ввод начнем с цепочки Network 1. В первой позиции курсора установите контакт 10.0. Далее с помощью кнопки  установите бокс. При этом в контекстном меню следует выбрать необходимый нам счетчик STU. Задайте номер счетчика, например, S31 и значение уставки — 3. Далее переведите кур-

сор непосредственно под ранее введенный контакт и установите контакт C31, который будет управлять сбросом счетчика.

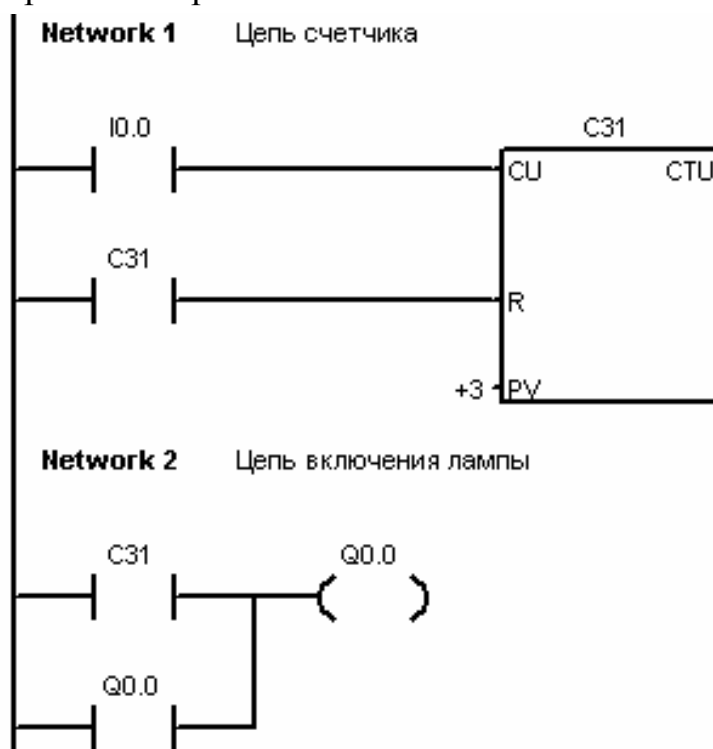


Рис. 2.9. LAD-программа включения исполнительного механизма со счетчиком

Таймеры

Таймеры предназначены для формирования заданных интервалов времени путем подсчета определенных фиксированных интервалов — дискрет. В контроллерах S7-222 таймеры могут иметь дискреты 1, 10 и 100 мс. На рис. 2.10. показана инструкция одного из таймеров (всего их четыре типа). Как видно, в верхней части бокса имеется обозначение таймера Txxx, где xxx – целое число от 0 до 255. Именно это число определяет дискрету таймера. Например, таймеры с T101 по T255 имеют дискрету 100 мс, с T97 по T100 – 10 мс, T32 и T96 – 1 мс. Таймер останавливается при выключении контроллера, или при достижении 32767 с.

Поскольку мы будем использовать так называемый таймер с задержкой включения (Timer On-Delay, TON), рассмотрим его работу подробно. Отсчет времени (счет дискрет) начинается, когда на вход (IN) поступает значение 1. Счет заканчивается, когда будет достигнуто заданное значение уставки PT (заданное число дискрет). В этот момент бит (переменная) таймера, имя которого совпадает с обозначением таймера, становится равным 1.

Сброс бита таймера и накопленного значения счета на ноль происходят в момент сброса входного сигнала независимо от того, отсчитан назначенный интервал или нет.

В качестве примера рассмотрим предыдущую задачу, поставив дополнительное условие — лампа должна гореть 10 секунд.

Окончательный вид программы показан на рис. 2.11. Прежде всего введите цепочку Network 1 для включения счетчика. Она будет точно такой же, как и в предыдущем примере.

Ранее бит C31 мы использовали для включения лампы. Было бы логично предположить, что теперь его следует использовать и для включения таймера.

Однако мы уже знаем, что единичное значение этого бита существует лишь мгновение, а поэтому применять его для включения счетчика невозможно. Однако контроллеры предоставляют нам возможность использовать при составлении программ внутренние переменные, которые называются переменными памяти или меркерами. В нашей программе такой переменной будет M0.1. Введите цепочку Network 2 так, как показано на рисунке. Будьте внимательны — при вводе инструкции бита памяти M0.1 в контекстном меню кнопки следует выбрать букву S, что соответствует функции SET, т.е. оператору присвоения.

Таким образом, для включения таймера будет использоваться переменная M0.1, которая будет сохранять единичное значение до тех пор, пока мы сами его не сбросим на ноль. Введите цепочку Network 3 для включения таймера. В контекстном меню кнопки установки бокса следует выбрать пункт TON, что соответствует нужному нам таймеру. Установите таймер T37. Его дискрета составляет 0,1 с, поэтому в качестве уставки следует ввести 100.

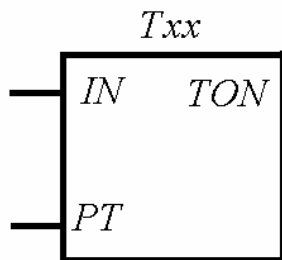


Рис. 2.10. Инструкция таймера с задержкой включения

Естественно, что переменная M0.1 должна использоваться и для включения лампы, что и отражается в цепочке Network 5.

Как только заданный интервал времени будет отсчитан, переменная счетчика T37 примет значение 1. В этот момент значение M0.1 нужно сбросить, одновременно погаснет и лампа. Инструкция сброса бита M0.1 с помощью функции R (RESET) показана в цепочке Network 4.

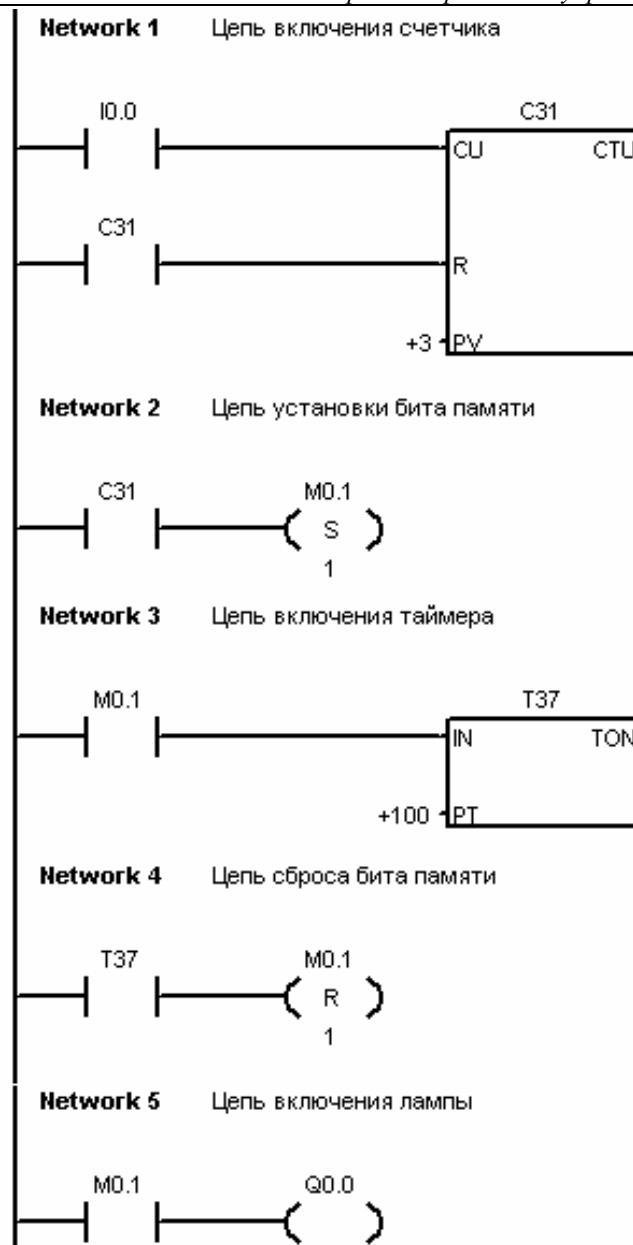


Рис. 2.11. LAD-программа включения исполнительного устройства на 10 с после третьего срабатывания датчика I0.0

Задания для самостоятельной работы

Привод работает следующим образом. Жидкость под высоким давлением поступает в поршневую или штоковую полость цилиндра в соответствии с алгоритмом работы привода. Коммутация полостей осуществляется гидрораспределителем по сигналам контроллера. Положение гидроцилиндра отслеживается путевыми выключателями (датчиками), которые устанавливаются на макетном столе. Управление автоматизированной системой осуществляется оператором с помощью кнопок.

Отчеты по каждому заданию должны включать: задание, таблицу переменных (см. пример табл. 2.1), LAD-программу для контроллера.

Задание 2.1

Привод работает в режимах единичного цикла (однократном) и непрерывного цикла (многократном). Выбор режима осуществляется кнопкой «Режим» с фиксацией положения. В однократном режиме при нажатии кнопки «Пуск» шток гидроцилиндра выдвигается и по истечении 5 секунд возвращается в исходную позицию, обеспечивая выполнение только одного цикла. В многократном режиме при нажатии кнопки «Пуск» шток гидроцилиндра выдвигается и возвращается в исходное положение циклически, обеспечивая непрерывно повторяющийся рабочий цикл. Чтобы прекратить циклическую работу пневмоцилиндра и затем переключиться на однократный режим, необходимо нажать кнопку «Стоп». Гидравлическая схема показана на рис. 2.3.

Задание 2.2

После нажатия кнопки «Пуск» в течение 1 секунды горит одна из сигнальных ламп, а еще через 1 секунду загорается другая лампа, и шток гидроцилиндра начинает циклическое возвратно–поступательное движение. После выполнения заданного числа циклов выдвижения–втягивания штока (например, пяти) лампа гаснет, и шток возвращается в исходную позицию. Нажатие кнопки «Стоп» прекращает процесс движения штока, и он останавливается в исходном положении. Гидравлическая схема показана на рис. 2.3.

Задание 2.3

Привод включается кнопкой «Пуск» и работает в двух режимах. В первом режиме шток гидроцилиндра совершает возвратно-поступательное циклическое движение с задержкой в конце каждого хода на 2 секунды. Во втором режиме циклическое движение штока происходит без задержек. Переключение режимов осуществляется с помощью кнопки «Режим», режимы индицируются соответствующими сигнальными лампами. При нажатии кнопки «Стоп» работа привода прекращается, и шток гидроцилиндра возвращается в исходную позицию. Гидравлическая схема показана на рис. 2.3.

Задание 2.4

Составить программу для осуществления возвратно-поступательного движения гидроцилиндра с одним датчиком положения, установленным в исходном (левом на рис. 2-3) положении. При замыкании этого датчика цилиндр движется вправо в течение двух секунд и возвращается назад.

Задание 2.5

Составить программу для осуществления возвратно-поступательного движения гидроцилиндра с одним датчиком положения, установленным в исходном

(левым на рис. 2-3) положении. При замыкании этого датчика цилиндр движется вправо в течение двух секунд и возвращается назад.

После выполнения трех циклов гидроцилиндр выдвигается вперед, находится в таком положении две секунды и возвращается в исходное положение. Далее весь цикл движения повторяется.

Примечание: уставку таймера задержки гидроцилиндра в крайнем правом положении подберите экспериментально.

Часть 3.

УПРАВЛЕНИЕ РАБОТОЙ СИСТЕМ ГИДРОПРИВОДОВ С ПОМОЩЬЮ КОМПЬЮТЕРА

Управление сложными технологическими процессами с помощью только лишь ПЛК невозможно. Это обусловлено как их ограниченными вычислительными возможностями, так и необходимостью вмешательства в ряде случаев в процесс управления человека (оператора). Такое управление осуществляется при совместном использовании ПЛК и компьютеров, оснащенных специальным программным обеспечением. Это программное обеспечение называется SCADA — Supervisory Control and Acquisition System, т.е. системами сбора данных и оперативного диспетчерского управления.

В общем случае SCADA–система предназначена для выполнения по меньшей мере двух функций:

сбор данных о контролируемом технологическом процессе,

управление технологическим процессом, реализуемое ответственными лицами, на основе собранных данных и правил (критериев), выполнение которых обеспечивает наибольшую эффективность и безопасность технологического процесса.

Полноценные SCADA-системы – сложное и дорогостоящее программное обеспечение. Тем не менее, имеются и довольно простые программы с ограниченными возможностями. Одной из таких программ является MicroComputing, разработанная концерном Siemens. Эта программа разработана специально для ПЛК серии S7-200. Ее демонстрационная версия распространяется бесплатно, но ее использование без контроллера не имеет смысла. Кроме того, во время ее работы на экран периодически выводится сообщение о том, что программа не авторизована, что однако не отражается на ее работоспособности.

Общее описание программы MicroComputing

Запуск программы осуществляется из главного меню компьютера (пункт MicroComputing SoftContainer). Интерфейс программы такой же, как и у большинства приложений Windows (рис. 3-1).

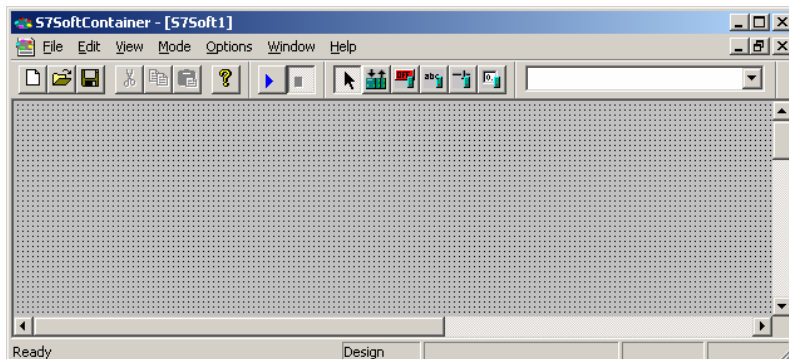


Рис. 3.1. Программа MicroComputing в режиме дизайна

Поскольку мы будем управлять контроллером, то программа для него должна быть составлена заранее. В качестве примера рассмотрим программу циклического движения цилиндра со счетчиком (рис. 3-2). После каждого третьего цикла счетчик сбрасывается на ноль и на 0,5 сек. загорается контрольная лампа. В этой программе входные переменные I0.0 и I0.1 являются соответственно сигналами правого и левого концевых датчиков положения гидроцилиндра. Выходные сигналы Q0.0 и Q0.1 управляют соответственно гидрораспределителем и контрольной лампой.

Загрузив эту программу в контроллер и убедившись в ее правильной работе, можно перейти к разработке проекта в среде MicroComputing.

На рис. 3-3 показан проект, который мы должны создать. При работе программы в центральном окне будет осуществляться счет циклов. После каждого третьего цикла его рамка будет менять цвет с красного на зеленый. В нижней части будет показано состояние распределителя — включен или выключен — с изменением цвета.

Разработка нового проекта начинается с выбора пункта главного меню программы File—New. Сразу после запуска программа будет работать в режиме дизайна, о чем говорит надпись в нижней части окна. Для того, чтобы можно было установить связь будущего проекта с контроллером, в окне проекта следует установить специальный объект Siemens S7Data Control . Найдите и активизируйте его в главном кнопочном меню программы. Далее разместите его в окне проекта, нажав левую клавишу мыши. Положение данного объекта не имеет значения, поскольку при работе программы он виден не будет.

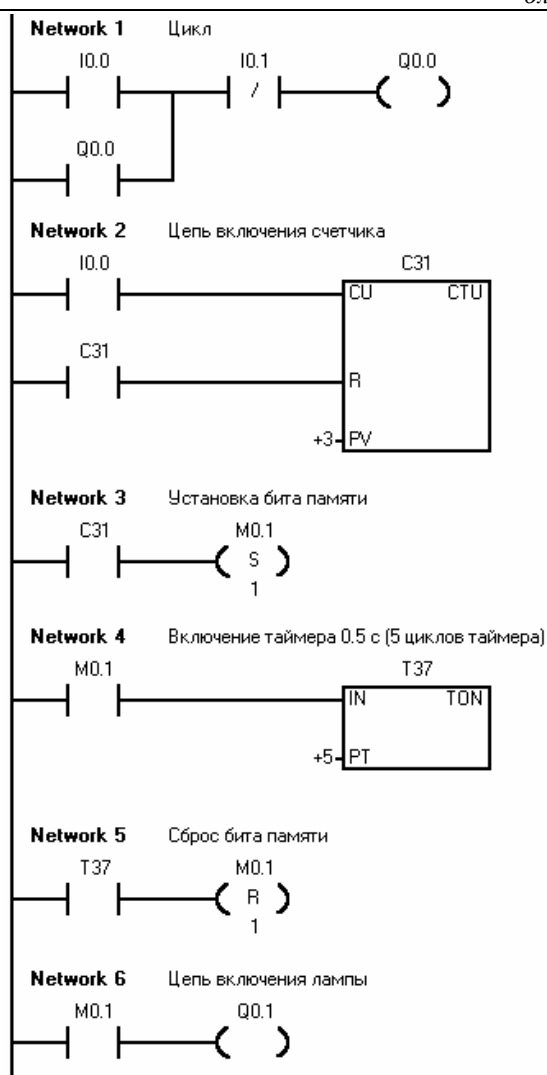


Рис. 3-2. LAD-программа циклического движения цилиндра со счетчиком

Теперь разместим на экране объекты. Объект Label (метка) предназначен для размещения всевозможных текстов. Для его установки следует активизировать пункт главного кнопочного меню . Двойным щелчком мыши открываем окно его свойств (рис. 3-4). Для нас его основным свойством является Caption – именно оно определяет видимый на экране текст. Другие свойства определяют стиль оформления метки. Предлагаем Вам поэкспериментировать с ними. Размер метки может быть любым. Поскольку на ней в качестве фона можно размещать рисунки (вкладка Picture), то возможности оформления экрана достаточно широки.

Следующим шагом разместим кнопки (Button). Для их установки следует активизировать пункт главного кнопочного меню



Двойным щелчком мыши открываем окно свойств объекта (рис. 3-5). Большинство их аналогичны свойствам метки. Обратите внимание на тексты, которые долж-

ны отображаться на кнопке, когда она включена (нажата) и выключена. Перейдите на вкладку Name (имя переменной) и задайте имя объекта (по умолчанию Button1, Button2...). Оно понадобится нам для связи с переменной LAD-программы. Аналогичным образом разместите вторую кнопку, которая в нашем случае выполняет роль фона для объекта Edit. Поля Caption для нее не заполняйте.

После размещения кнопок установите объект Edit (поле редактирования).

Для этого следует активизировать пункт главного кнопочного меню . Окно свойств этого объекта показано рис. 3-6.

После размещения всех объектов следует установить связь их имен с конкретными переменными программы контроллера. Для этого двойным щелчком мыши откройте окно свойств объекта Siemens S7Data Control и перейдите на вкладку Connection (соединения), показанную на рис 3-7. Здесь в окне Controls

следует отыскивать имена установленных ранее объектов (например, Button1 для первой кнопки) и изменить (установить) значения свойства Value на адреса переменных контроллера (рис. 3-8). Объект Edit следует связать с переменной счетчика C31.

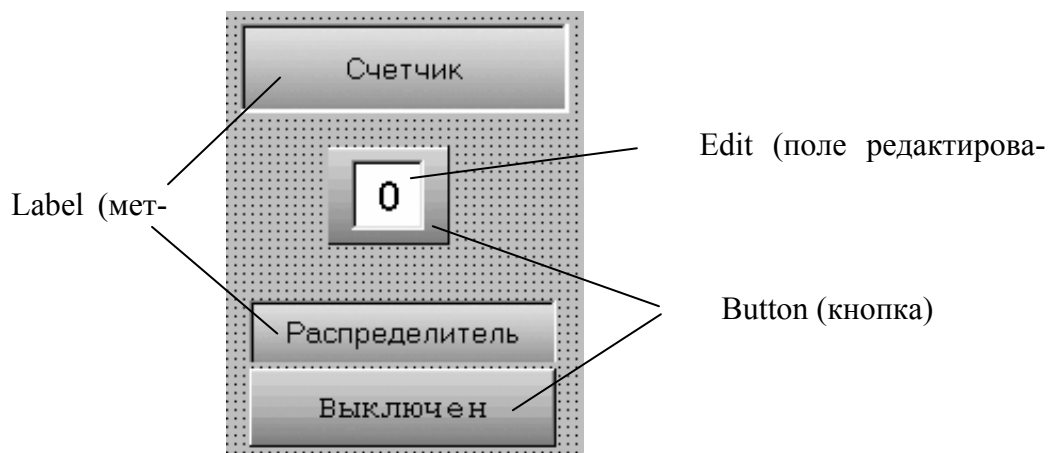


Рис. 3-3. Размещение объектов в окне проекта

После завершения этого этапа работы можно запустить созданный проект, нажав кнопку в главном кнопочном меню. Контроллер в этот момент должен быть включен. Завершение работы проекта осуществляется кнопкой .

После завершения этого этапа работы можно запустить созданный проект, нажав кнопку  в главном кнопочном меню. Контроллер в этот момент должен быть включен. Завершение работы проекта осуществляется кнопкой .

Управление контроллером с помощью компьютера

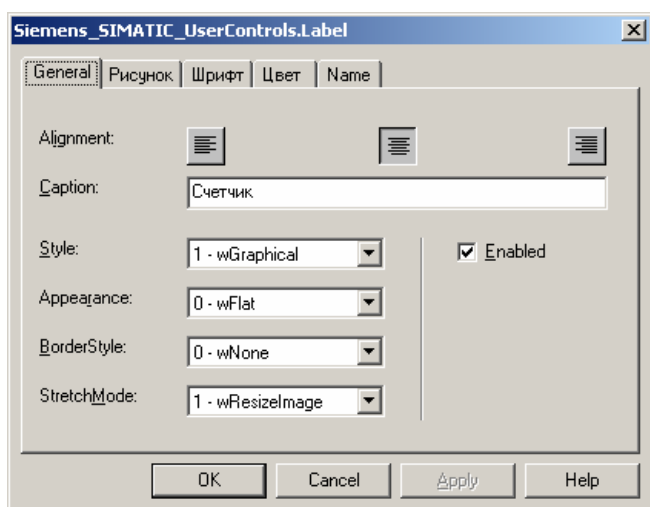


Рис. 3-4. Свойства объекта Label (метка)

MicroComputing позволяет формировать управляющие сигналы. На рис. 3-9а показана программа, состоящая всего из одной вычислительной цепочки. Здесь кнопка, связанная с контактом I0.0, управляет исполнительным механизмом Q0.0. Вы сможете легко убедиться, что изменение значений переменных с помощью программы MicroComputing невозможно. Однако, изменение переменной I0.0 на M0.0 (рис. 3-9б) позволяет достичь желаемого эффекта, но в этом случае контроллер уже не смо-

жет работать автономно, поскольку программно переменная M0.0 не изменяется. На рис. 3-10 показана программа, реализующая возвратно-поступательное движение гидроцилиндра с помощью одного датчика положения, связанного с переменной I0.1. В момент срабатывания датчика одновременно включаются распределитель и таймер. Таймер настроен на 10 циклов. По истечении этого времени распределитель отключается, и гидроцилиндр возвращается в исходное положение.

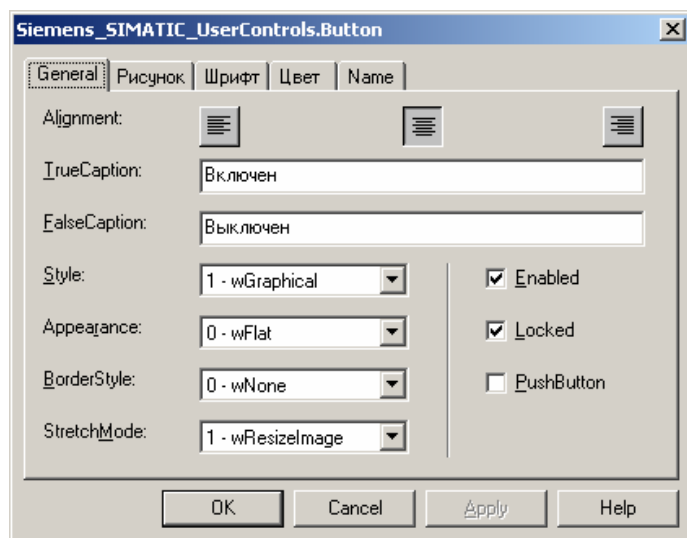


Рис. 3-5. Свойства объекта Button (кнопка)

Network 5. Эта цепочка включает контакт, связанный с переменной SM0.1 и блок присвоения. Переменная SM0.1 – системная. Она принимает единичное значение только один раз – в момент запуска программы. Блок MOV_W выполняет операцию присвоения VW100=10. Рекомендуем убедиться, что работа этой программы не отличается от предыдущей.

Поскольку теперь уставка таймера не константа, а переменная, ее можно менять с помощью программы MicroComputing. Изменять значение переменной VW100 можно с помощью средств Edit (поле редактирования) или Slider (ползунковый регулятор). Для этого данную переменную необходимо свя-

Внесем в программу изменения, которые позволят управлять таймером, а следовательно и временем включения распределителя. Новая программа показана на рис. 3-11. В ней всего два отличия от предыдущей. Прежде всего, обратите внимание на таймер. Вместо фиксированного значения уставки (PT) указана переменная VW100. Начальное значение этой переменной задается в цепочке

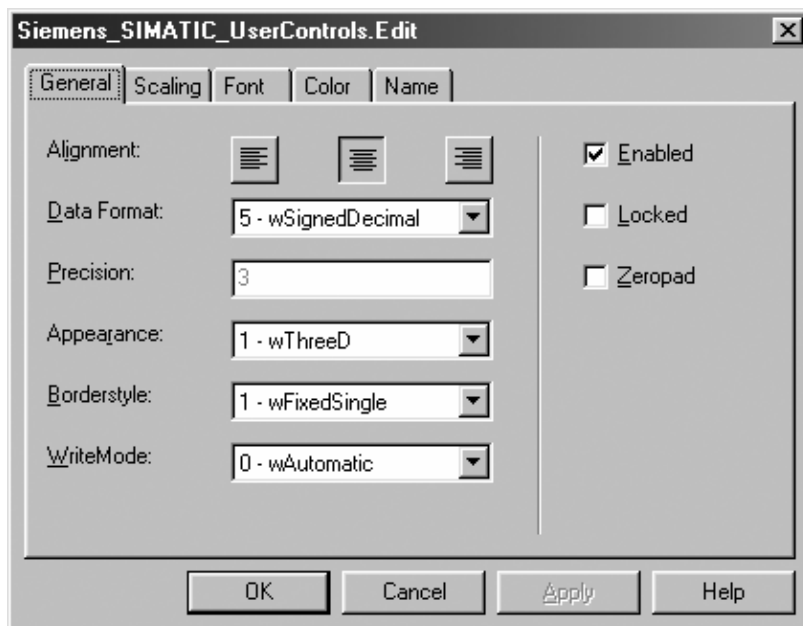


Рис. 3-6. Свойства объекта Edit (поле редактирования)

затем с их свойством Value (рис. 3-12). Вам предоставляется возможность задать возможный диапазон значений переменной (рис. 3-13) и параметры отображения регулятора на экране (рис. 3-14).

Результат построений может быть приблизительно таким, как на рис. 3-15.

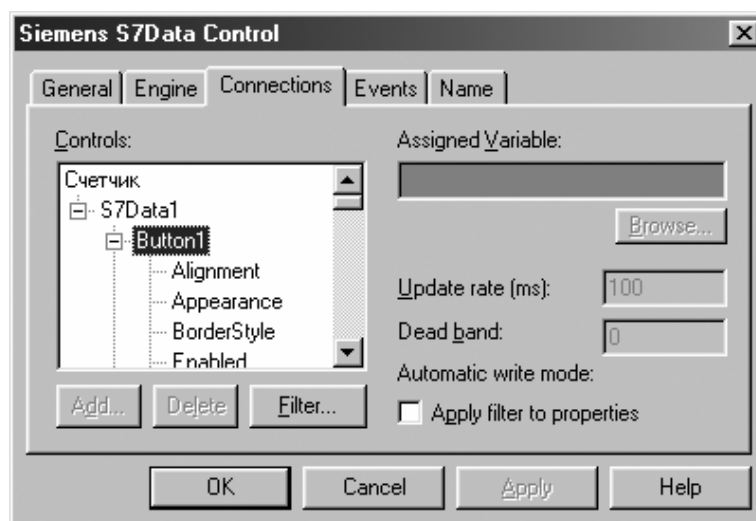


Рис. 3-7. Выбор объекта

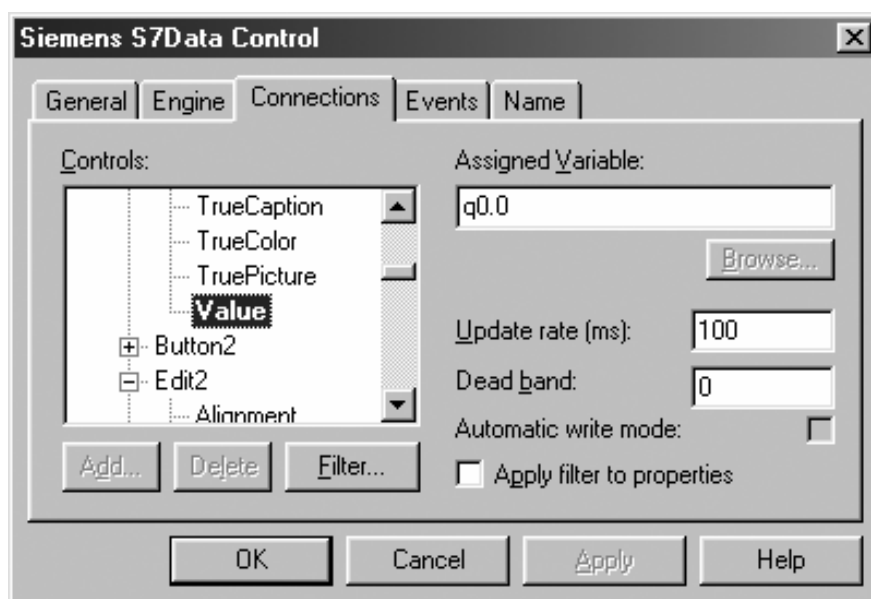


Рис. 3-8. Определение свойства Value (значение)

Задания для самостоятельной работы

Разработайте системы управления контроллером для программ согласно заданиям предыдущей части, поместив на экран компьютера указанные элементы управления (кнопки), сигнальные лампы, сообщения о режиме работы привода и состоянии исполнительных механизмов (распределителей).

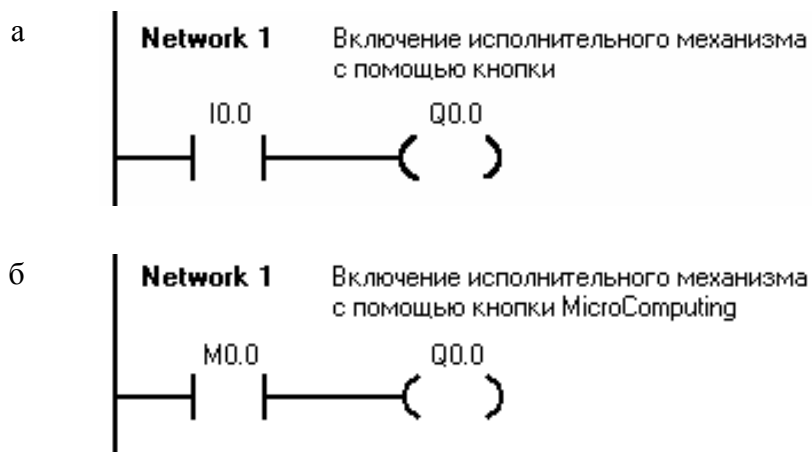


Рис. 3-9. Программа включения исполнительного механизма с помощью электрокнопки (а) и кнопки среды MicroComputing (б)

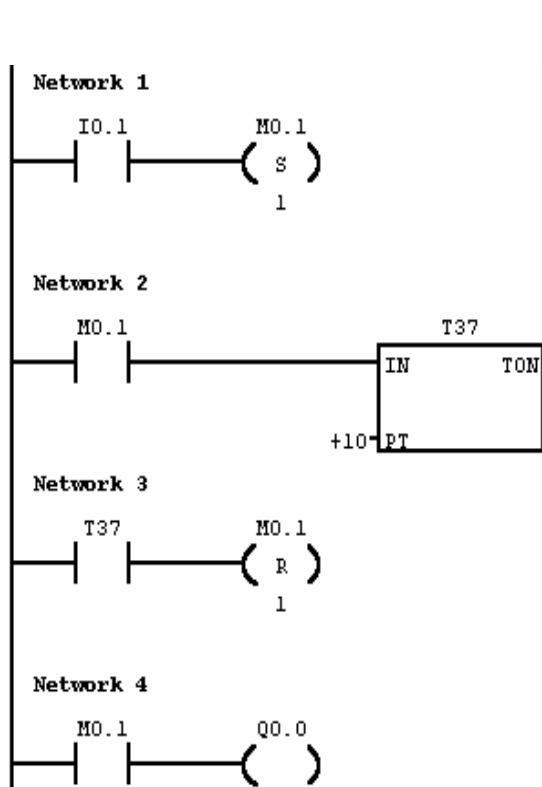


Рис. 3-10. Управление распределителем по времени

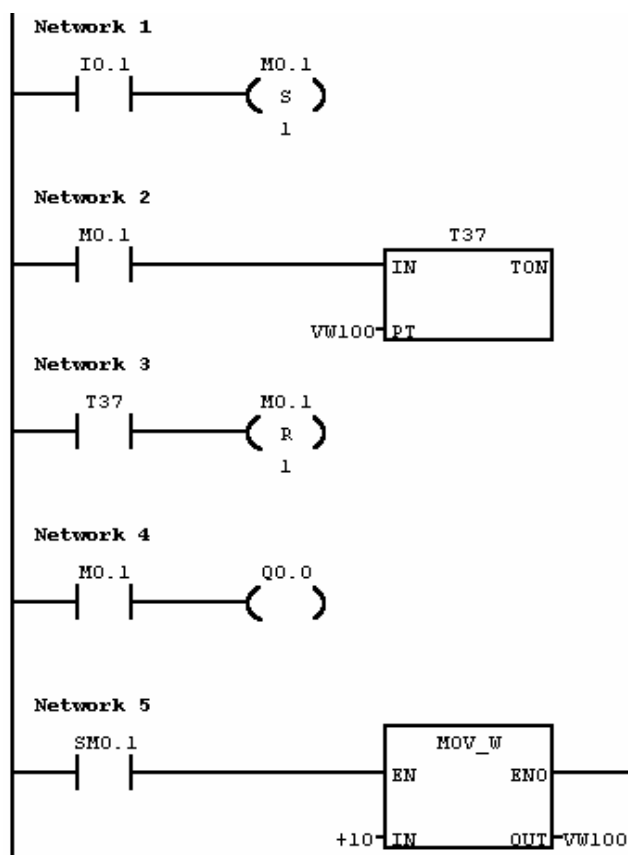


Рис. 3-11. Модификация программы управления распределителем по времени



Рис. 3-12. Задание значения свойства Value для объекта Slider (Edit)

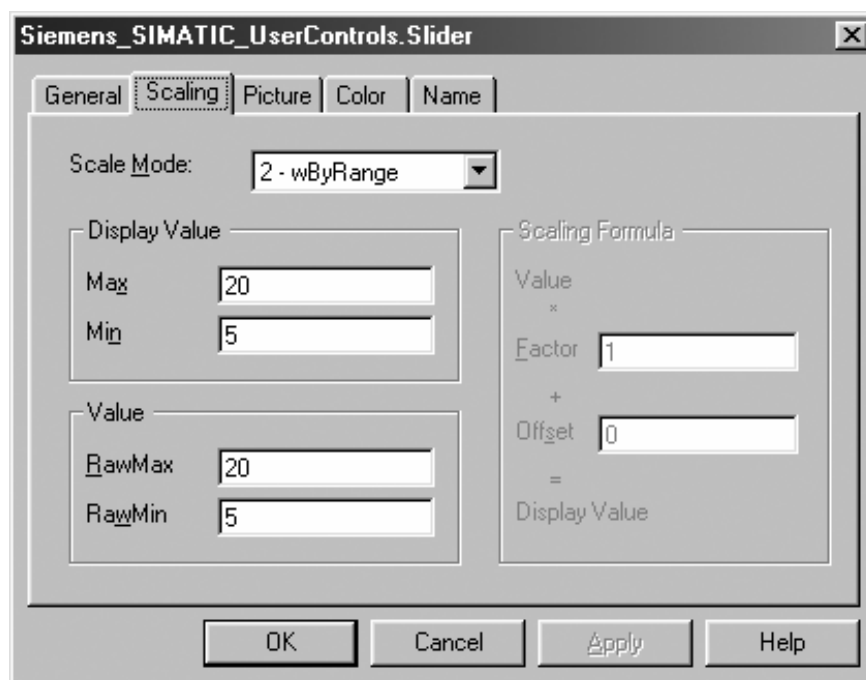


Рис. 3-13. Задание диапазона значений переменной

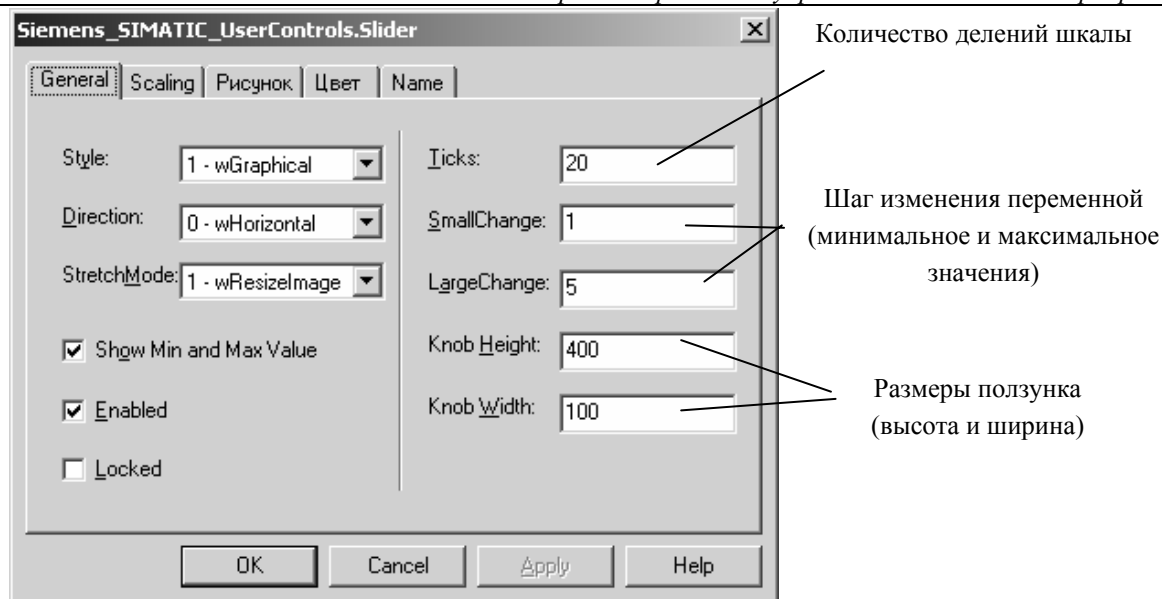


Рис. 3-14

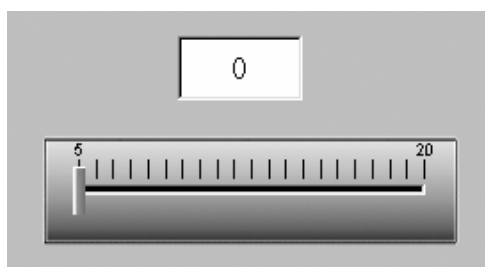


Рис. 3-15. Поле редактирования и регулятор

Учебное электронное текстовое издание

Некрасов Александр Васильевич

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ДЛЯ МОДЕЛИРОВАНИЯ РАБОТЫ И УПРАВЛЕНИЯ
СИСТЕМАМИ ГИДРОПРИВОДА**

Редактор

Компьютерная верстка *А.А. Гребеничикова*

Рекомендовано РИС ГОУ ВПО УГТУ-УПИ

Разрешен к публикации 26.12.06.

Электронный формат – PDF

Формат 60×90 1/8

Издательство ГОУ ВПО УГТУ-УПИ

620002, Екатеринбург, ул. Мира, 19

e-mail: sh@uchdep.ustu.ru

Информационный портал

ГОУ ВПО УГТУ-УПИ

<http://www.ustu.ru>