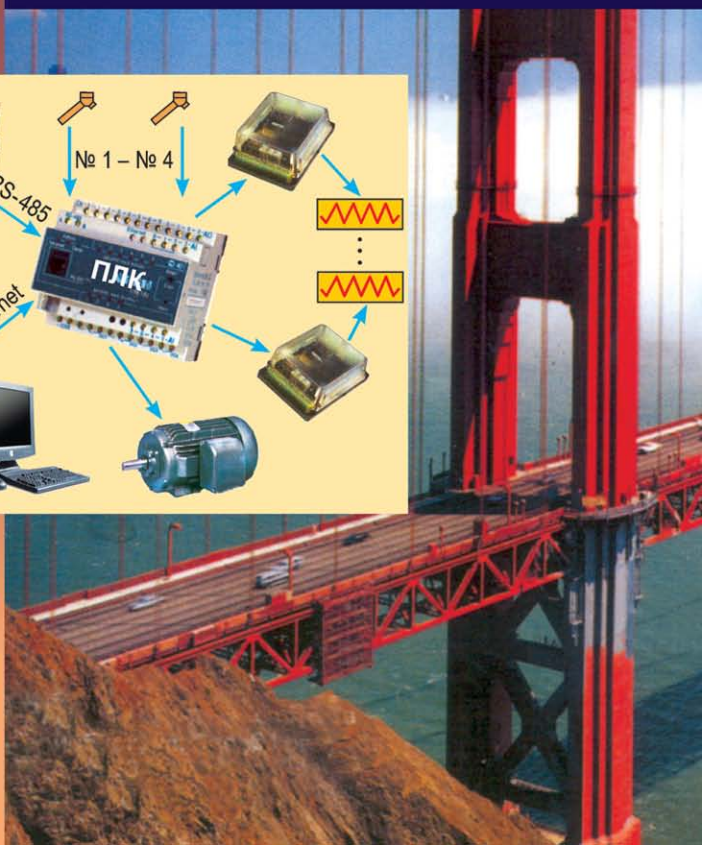
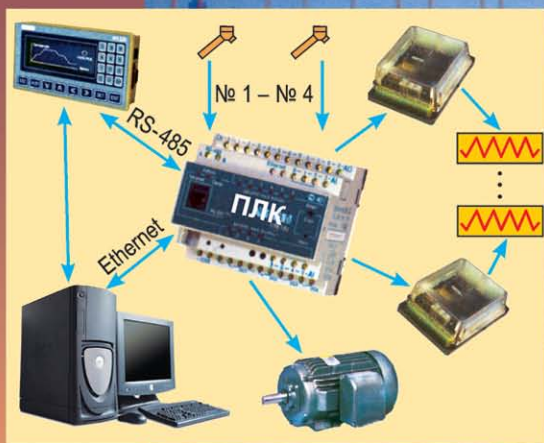




студентам  
учреждений  
высшего  
образования

Ю.Н. Петренко С.О. Новиков  
А.А. Гончаров

# ПРОГРАММНОЕ УПРАВЛЕНИЕ ТЕХНОЛОГИЧЕСКИМИ КОМПЛЕКСАМИ В ЭНЕРГЕТИКЕ



Ю.Н. Петренко С.О. Новиков  
А.А. Гончаров

---

# ПРОГРАММНОЕ УПРАВЛЕНИЕ ТЕХНОЛОГИЧЕСКИМИ КОМПЛЕКСАМИ В ЭНЕРГЕТИКЕ

Допущено  
Министерством образования Республики Беларусь  
в качестве учебного пособия для студентов  
учреждений высшего образования  
по энергетическим специальностям



Минск  
«Вышэйшая школа»  
2013

УДК 621.31:[681.51:004](075.8)

ББК 32.96я73

ПЗ0

Рецензенты: кафедра «Автоматизированный электропровод» УО «Гомельский государственный технический университет им. П.О. Сухого» (заведующий кафедрой кандидат технических наук, доцент *В.С. Захаренко*); проректор по научной работе УО «Белорусский государственный университет информатики и радиоэлектроники» доктор технических наук, профессор *А.П. Кузнецов*

*Все права на данное издание защищены. Воспроизведение всей книги или любой ее части не может быть осуществлено без разрешения издательства*

## **Петренко, Ю. Н.**

ПЗ0 Программное управление технологическими комплексами в энергетике : учеб. пособие / Ю. Н. Петренко, С. О. Новиков, А. А. Гончаров. – Минск : Выш. шк., 2013. – 407 с.: ил.

ISBN 978-985-06-2227-3.

Рассматривается реализация проектов систем управления технологических комплексов на основе ПЛК. Описаны языки программирования ПЛК: RKC, LAD, FBD, STL, SCL, GRAPHSET, SFC, CFC.

Рассматриваются промышленные компьютерные сети, объединяющие ПЛК, интеллектуальные датчики и исполнительные устройства на основе технологий PROFINet, FIELDbus и PROFIBus.

Изложены приемы и правила работы в системе программирования CoDeSys.

Для студентов учреждений высшего образования по энергетическим специальностям.

Будет полезно специалистам, занимающимся разработкой дискретных систем управления технологических комплексов.

**УДК 621.31:[681.51:004](075.8)**

**ББК 32.96я73**

**ISBN 978-985-06-2227-3**

© Петренко, Ю.Н., Новиков С.О.,  
Гончаров А.А., 2013

© Оформление. УП «Издательство  
“Вышэйшая школа”», 2013

## Предисловие

Ускорение научно-технического прогресса на современном этапе развития немислимо без решения задач комплексной автоматизации производства. Быстрое развитие этого направления является одним из определяющих факторов интенсификации экономики. Оно позволяет кардинально повысить производительность труда, качество выпускаемой продукции и эффективность производства. Основной задачей современного направления автоматизации является разработка и внедрение гибких автоматизированных производственных систем различного назначения на базе промышленных роботов, автоматических манипуляторов, контрольно-измерительных и диагностических комплексов и транспортно-складских систем с управлением от ЭВМ, а также их интеграции с системами автоматизированного проектирования и технологической подготовки производства.

Наряду с системами числового программного управления для автоматизации технологических комплексов самого различного назначения с использованием программируемых логических контроллеров широко используются дискретные системы логического управления. Наметился переход от локальной автоматизации отдельных технологических комплексов к распределенному взаимосвязанному управлению оборудованием цеха или предприятия на основе промышленных вычислительных сетей.

Представленная работа посвящена вопросам проектирования и моделирования работы дискретных логических систем управления, область применения которых чрезвычайно широка – комплексы литья под давлением, агрегатные станки, автоматические линии, шлифовальные станки, отдельные циклы станков с числовым программным управлением (например, автоматическая смена инструмента), управление различными энергетическими системами и объектами автоматики и мно-

гое другое. Моделирование дискретных систем управления предполагает применение программируемых логических контроллеров (ПЛК) и соответствующего программного обеспечения – специализированных языков программирования, наиболее распространенные из которых представлены в работе.

Изложены приемы и правила работы в системе программирования CoDeSys, основным назначением которой является использование программ логических контроллеров и промышленных компьютеров в стандарте МЭК 61131-3. Средства CoDeSys позволяют реализовывать законы управления, моделировать схемы управления и исследования их поведения при различных условиях работы в реальном времени.

Авторы благодарят рецензентов за их ценные замечания, которые способствовали улучшению содержания книги.

*Авторы*

## Введение

Информационные технологии характеризуют уровень развития общества, его динамику. Общей задачей является автоматизация – от проектирования и научных исследований до производства готовой продукции.

Традиционно процесс создания сложных технических объектов состоит из нескольких этапов, важнейшим из которых является *проектирование*. Результат данного этапа – проект некоторого изделия, устройства или ТМ.

Первым этапом в создании проекта электрооборудования станков и других ТМ является *разработка циклограмм работы* (алгоритмов управления) на базе циклограмм и схем расположения исполнительных механизмов, полученных от конструкторов – механиков и технологов.

При использовании программируемых контроллеров и ПК для реализации сложных алгоритмов дискретного управления оборудованием этот этап становится определяющим. Хорошо и детально проработанный алгоритм управления (АУ) обеспечивает простоту и эффективность разработки управляющих программ и другой проектной документации. Наряду с этим все неточности и ошибки в алгоритме управления тиражируются и усиливаются на последующих этапах проектирования, изготовления, монтажа и запуска в эксплуатацию.

Основным источником ошибок в АУ являются различные несоответствия между алгоритмом и управляемыми исполнительными механизмами, а также несоответствия в самом АУ.

Общепризнанным современным средством выявления и устранения проектных ошибок является *компьютерное моделирование*. Средства моделирования и проверки проекта до начала аппаратной реализации одинаково актуальны как при традиционном проектировании, так и при использовании САПР электрооборудования.

Компьютерное моделирование вместо дорогостоящего макетирования и натурных испытаний – это обширные программы исследований и разработок.

*Контроллер* – это мозг любой автоматической машины, обеспечивающий ее логику работы. Например: контроллер системы впрыска топлива автомобилей, контроллер управления лифтом, автоматом сборки часов, стиральной машиной и т.д. Естественно, чем сложнее логика работы машины, тем «умнее» должен быть контроллер. Технически контроллеры реализуются по-разному. Это может быть механическое устройство, пневматический или гидравлический автомат, релейная или электронная схема либо даже компьютерная программа.

Часто контроллер встроен в конкретную машину и обладает жесткой логикой работы, заложенной при изготовлении. Проектирование таких контроллеров окупается только для изделий, выпускаемых значительным тиражом. При создании машин, занятых в сфере промышленного производства, как правило, приходится иметь дело не более чем с единицами однотипных устройств. Кроме того, очень существенной здесь является возможность быстрой перенастройки оборудования на выпуск другой продукции. Для уникальных проектов, мелкосерийных изделий и опытных образцов также желательно иметь универсальный свободно программируемый контроллер.

Идея создания *программируемых логических контроллеров* (ПЛК) родилась практически сразу с появлением микропроцессора. ПЛК представляет собой вычислительную машину, имеющую некоторое множество входов и множество выходов (рис. 1В). Контроллер отслеживает изменение входов и вырабатывает программно-определенное воздействие на вы-

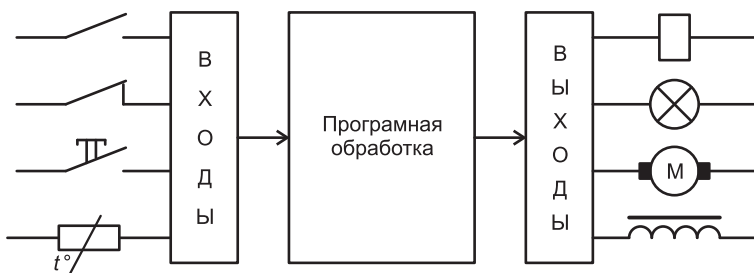


Рис. 1В. Принцип работы ПЛК

ходах. Обладая памятью, ПЛК способен реагировать по-разному, в зависимости от предыстории. Такая модель соответствует широко известным *конечным автоматам*. Однако возможности управления по времени, развитые вычислительные способности, включая цифровую обработку сигналов, поднимают ПЛК на более высокий уровень. Они ориентированы на длительную работу в условиях промышленной среды. Это обуславливает определенную специфику схемотехнических решений и конструктивного исполнения.

Мощное вычислительное ядро современных ПЛК делает их очень похожими на компьютеры. Однако ПЛК – не «железо», а технология. Она включает специфическую аппаратную архитектуру, принцип циклической работы и специализированные языки программирования. Программирование ПЛК осуществляется людьми, хорошо знающими прикладную область, но не обязанными быть специалистами в математике. В первую очередь контроллеры ориентированы на решение задач промышленного производства. Поэтому оценивать их нужно с позиций производственной реальности. Представьте себе, что у вас есть автоматизированный фрезерный станок. Его система управления, представляющая собой не менее полусотни реле и пускателей, выполнена в виде шкафа управления (ШУ). Требуется срочно заменить «еще теплый» шкаф на ПЛК. Есть два варианта:

- вы осмысленно перерисовываете схему ШУ на языке LD и поясняете ее техникам;
- вы пишете программу на языке общего применения и до пенсии связываете свою жизнь с этим станком.

Специализация языков ПЛК заключена в упрощении их применения, приближении к предметной области. На сегодняшний день ПЛК – это на 90 % программный продукт. Контроллер, не обеспеченный средствами визуального прикладного проектирования с поддержкой стандартных языков, использовать очень трудоемко. Один современный ПЛК способен заменить десятки регуляторов, сотни таймеров и тысячи реле.

Для того чтобы разрабатываемая система управления решала поставленные вопросы, необходимо использовать эффективное программно-алгоритмическое обеспечение, которое позволяет реализовывать законы управления, моделировать схему управления и исследовать ее поведение при различных условиях работы.



В качестве такой системы рассмотрена система программирования CoDeSys. Программный комплекс **CoDeSys** разработан компанией 3S-Smart Software Solutions GmbH (3S). Основным его назначением является программирование ПЛК и промышленных компьютеров в стандарте МЭК 61131-3. В нем собран полный набор языков МЭК, предназначенных для программирования ПЛК. CoDeSys является штатным инструментом программирования ПЛК ведущих европейских изготовителей. Некоторые из них используют CoDeSys как базовое ядро собственных систем программирования, известных под собственными торговыми марками.

Как средство программирования ПЛК CoDeSys можно разделить на две части: **среду программирования** и **систему исполнения**. Среда программирования функционирует на персональном компьютере в среде Windows (рис. 2В).

Поскольку CoDeSys дает машинный код, поддержка его исполнения достаточно проста и по минимуму сводится к набору функций поддержки ввода-вывода и отладки. Система исполнения функционирует в ПЛК и обеспечивает загрузку кода прикладной программы, «горячее» обновление кода, от-

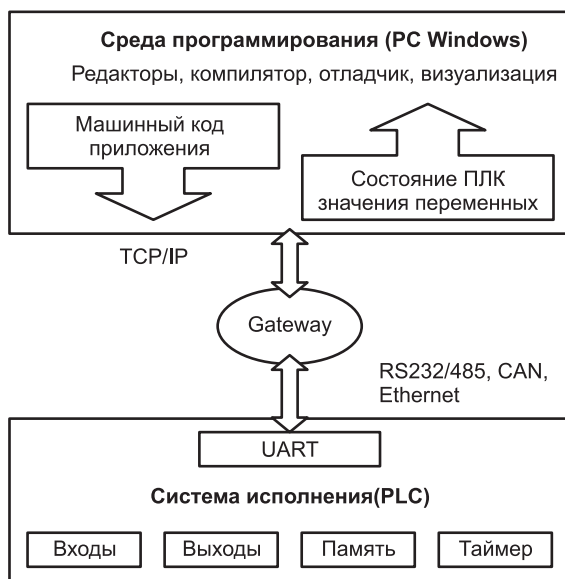


Рис. 2В. Составные части CoDeSys

ладку, управление задачами и ряд сервисных функций. Система исполнения 3S поставляется изготовителям ПЛК в виде **исходных текстов** (ОЕМ). Это позволяет максимально эффективно реализовать поддержку аппаратных средств без каких-либо промежуточных механизмов. Изготовителю оборудования требуется только дописать **аппаратно-зависимые функции** ввода-вывода, возможно, отредактировать функции поддержки канала связи на физическом уровне и функции записи кода прикладной программы в ППЗУ (Flash, диск и др.). Далее он компилирует готовую систему исполнения и помещает ее код в ПЗУ (или на загрузочный диск) своего ПЛК. Теперь ПЛК готов для поставки заказчиком.

Естественно, OEM может добавить в систему исполнения собственные функции и поддержку нестандартной аппаратуры. Например, если контроллер имеет достаточно сложный программно-аппаратный счетчик, его можно полностью встроить в систему исполнения. Пользователю же будут «видны» только переменные, содержащие готовые результаты. Поддержка нестандартного оборудования или сети реализуется, таким образом, прозрачно для пользователя.

Общение системы программирования и системы исполнения происходит через так называемый **Gateway server**. Обычно сверху со средой программирования он соединяется по TCP/IP, с системой исполнения – по последовательному каналу или fieldbus. Среда программирования и Gateway server не обязательно должны функционировать на одной машине. Интерфейсы сервера сверху и снизу открыты, т.е. OEM может применить для связи среды программирования с контроллером собственный протокол либо написать дополнительные программы, использующие протокол CoDeSys. Через тот же Gateway server работают и входящие в комплект **OPC-** и **DDE-**серверы.

Еще один специфичный для оборудования компонент CoDeSys – **ПЛК-браузер**. Для пользователя это окно текстового терминала, оснащенного набором команд, уникальных для данного ПЛК. Команды передаются контроллеру и выполняются там так, как это реализовал OEM, что дает возможность посмотреть версию системного ПО, дампы памяти, выполнить тесты оборудования, настроить определенные параметры и др.

Из отладочных функций CoDeSys интересен инструмент **графической трассировки** (рис. 3В) значений переменных.

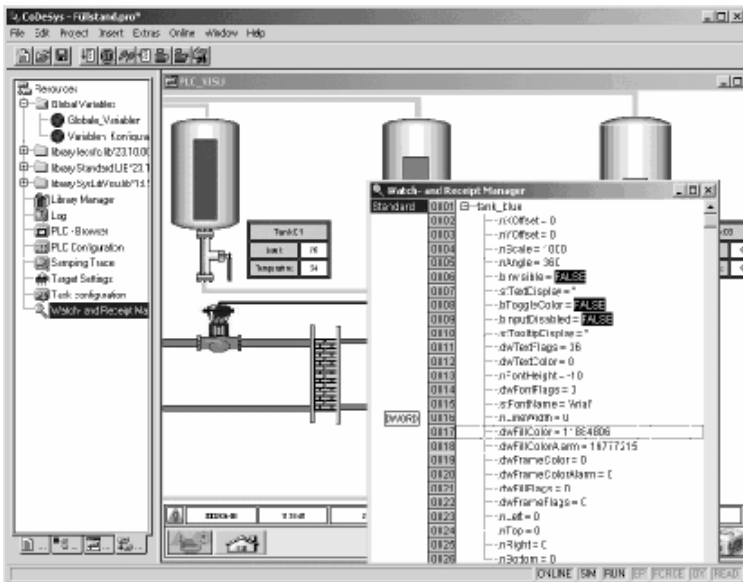


Рис. 3В. Пример графической трассировки наблюдаемых параметров

С его помощью можно проводить отладку не только ПО, но и оборудования, причем **без написания программы**. Весьма удобно и наличие **встроенной системы визуализации**, функционирующей как в инструментальной среде, так и в ПЛК (имеющем дисплей) и Web. На практике пользователи выполняют достаточно сложные проекты автоматизации в CoDeSys без необходимости приобретения SCADA.

Система исполнения CoDeSys может работать как поверх операционной системы (ОС), так и на «голом железе». Естественно, во втором случае адаптация несколько сложнее, поскольку приходится писать собственный начальный загрузчик, системные тесты, обработчик таймера, т.е. типовую минимальную поддержку аппаратуры. Но на 8- и 16-разрядных платформах это дает существенный выигрыш по быстродействию и необходимым ресурсам. Всего существует четыре разновидности систем исполнения CoDeSys: **CSP8**, **CSP16**, **CSP32E** и **CSP32F**. Они предназначены для 8, 16, 32-разрядных процессоров без ОС, соответственно.

В настоящее время CoDeSys поддерживает семейства: Intel 8051, Intel 80x86 / 80186/ Pentium, ARM, MIPS, Motorola

MC68000 / MC68332/ ColdFire, PowerPC, Hitachi SH 2/3/4, H8, Infineon C16x, Infineon TriCore и Texas Instruments TMS32028x.

Особняком стоит система исполнения CoDeSys SP RTE для Windows XP/NT/2000. Она включает собственное ядро жесткого реального времени, функционирующее под Windows. Данная система не требует адаптации. Связь с оборудованием происходит через драйверы. Все это может показаться сложным. Но конечный пользователь всегда получает ПЛК, оснащенный готовой и настроенной системой программирования. Ему необходимо только установить на своем компьютере среду программирования CoDeSys и так называемые файлы поддержки целевой платформы (TSP), благодаря которым CoDeSys получает информацию о типе процессора, распределении памяти, наименованиях и конфигурации модулей ввода-вывода, сети и т.д. Причем все основные настройки уже заданы изготовителем ПЛК, т.е. задача пользователя – просто выбрать нужный ПЛК в списке. TSP может включать описания поддерживаемых в системе модулей, таким образом, пользователь получает возможность конфигурирования модульных систем, включая распределенные. **Специфичные для оборудования файлы справочной системы также интегрируются в CoDeSys.** В итоге пользователь работает в среде программирования так, как будто она специально написана для используемого им ПЛК.

Сложность решаемой задачи и созданной информационной компьютерной среды для ее решения требует привлечения самых разносторонних средств изложения материала данного учебного пособия. Наряду с традиционными средствами формализации и описания задач, методов и алгоритмов широко используются тексты программ на языке высокого уровня. В этих программах основные объекты и переменные представлены на английском языке, это широко используется в практике по той простой причине, что на данном языке имеется документация на ПО. По возможности, использовался русский язык, а операции, процедуры и алгоритмы дополнительно поясняются подробными комментариями.

Таким образом, формализация с привлечением описанных программ практически не требует специальной подготовки читателя в области программирования, одновременно позволяя достаточно глубоко проникнуть в суть изложенных задач моделирования и средств их компьютерной реализации.

# ГЛАВА 1. ПРОЕКТИРОВАНИЕ СИСТЕМ АВТОМАТИЗАЦИИ

## 1.1. ПЛК как универсальное средство автоматизации

Наиболее перспективным направлением совершенствования систем автоматизации является применение гибко программируемых электронных управляющих систем, основанных на использовании управляющих вычислительных машин (УВМ). Характерной чертой УВМ является их способность к работе в реальном времени, т.е. в темпе, диктуемом потребностями объекта управления. Обычная ЭВМ, выполняющая вычислительные операции, не имеет таких жестких рамок реализации алгоритма вычислительного процесса во времени. Специализированные УВМ, предназначенные для управления сложными станками, литейными комплексами, машинами для обработки металлов давлением, автоматическими линиями (АЛ), получили название *программируемых логических контроллеров (ПЛК)*. В связи с повышенной стоимостью такой аппаратуры ее применение оправдано, когда количество релейно-контактных аппаратов эквивалентной схемы более 60–80 и при напряженных циклах работы.

Понятие «*программируемый*» отражает возможность ПЛК воспринимать программу на языке высокого уровня; слово «контроллер» показывает возможность выполнять функции управления сложными технологическими процессами. Встречающиеся в настоящее время термины ПК (РС) и ПЛК (PLC) – программируемый логический контроллер – равнозначны. Разработка ПЛК стала возможной благодаря развитию следующих основных направлений: релейной бесконтактной автоматики (бесконтактных логических управляющих систем); ЦПУ технологическим оборудованием; систем программирования; элементной базы. Несмотря на значитель-

ную общность ПЛК и УВМ, первые имеют ряд характерных с точки зрения пользователя свойств, которые заключаются в следующем.

1. Как правило, ПЛК предназначены для решения логических задач с элементами арифметических операций.

2. Упрощенное программирование, основанное на символах и мнемокодах, легко усваивается персоналом, не имеющим специальной подготовки.

3. Большая помехозащищенность, потенциальная развязка от внешних цепей с помощью оптронов, ферритовых устройств и других, расширенный диапазон допустимых условий эксплуатации в конечном счете дает возможность эксплуатировать ПЛК в цеховых условиях.

4. Модульность конструкции: входы, выходы и объем памяти наращиваются с определенным шагом.

5. ПЛК оснащаются вспомогательными устройствами для выполнения дополнительных функций, к которым относятся таймеры для выдержки времени, счетчики импульсов и др.

Имеется тенденция к расширению функциональных возможностей ПЛК за счет введения арифметических операций и реализации функций регуляторов со стандартными настройками (П, ПИ, ПИД). Изменение уставок регуляторов производится как вручную, так и программным путем.

ПЛК обеспечивают изменение логической программы функционирования с помощью встраиваемого или автономного устройства программирования, что равносильно изменению принципиальной электрической схемы с жесткими связями. При этом отпадает необходимость в отключении объекта управления для переналадки, исключаются трудоемкие процессы, связанные с проектированием новой схемы и обеспечением комплектующими.

Проектирование схемы управления на основе жесткой (аппаратной) логики производится индивидуально для каждой конкретной установки: выбираются электрические элементы и узлы, разрабатывается конструкторская документация на систему управления, производится ее монтаж, отладка и корректировка. Для изготовления схемы необходимо наличие определенной номенклатуры комплектующих изделий. Весь описанный процесс требует значительных затрат времени квалифицированного персонала, что особенно ощутимо при мелкосерийном или индивидуальном характере производства. Стоимость ручного труда в последнее время имеет тенденции к росту, в то время как затраты на оборудование непрерывно

снижаются. Особенности решения логических задач определяют и функциональный состав ПЛК, изображенный на рис. 1.1.1.

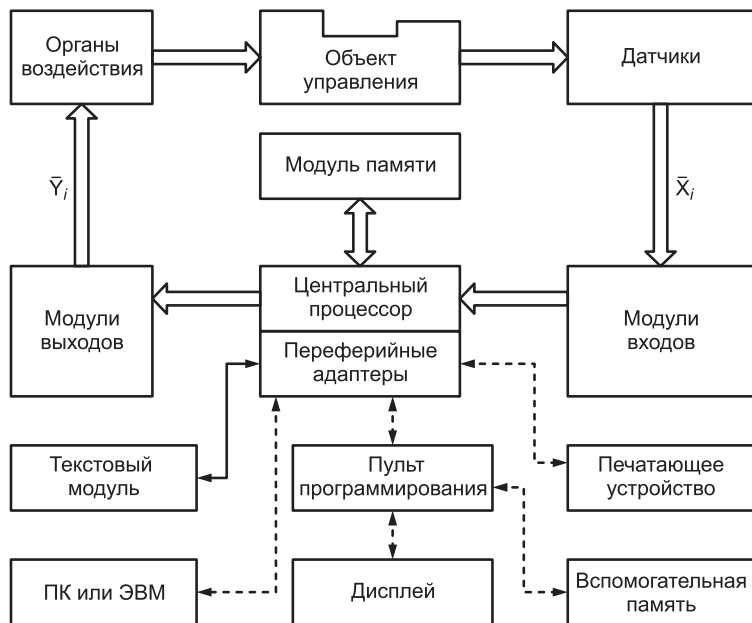


Рис. 1.1. Функциональный состав программируемого логического контроллера:  
 == — постоянные связи; --- — временные; - - - — возможные

Программируемый контроллер состоит из центрального процессора, постоянного запоминающего устройства (ПЗУ), модулей входов-выходов (входного и выходного регистров), обеспечивающих сопряжение с входными датчиками и с исполнительными устройствами объекта управления, и сканирующего устройства, которое производит поочередный опрос входов и выходов. К контроллеру подключается пульт программирования (программная панель), через который производится набор программы.

В некоторых типах контроллеров в период настройки цикла вместо ПЗУ подключается оперативное запоминающее устройство (ОЗУ), которое может осуществлять как считывание, так и запись (и перезапись) информации. После окончательной отладки цикла и внесения всех изменений программа

перезаписывается с ОЗУ на ПЗУ. Часто в контроллерах используются ПЗУ с возможностью перезаписи информации, т.е. перепрограммируемые (полупостоянные) запоминающие устройства (ППЗУ). Помимо ручного ввода программы большинство ПЛК предусматривают возможность связи с другими ПЛК или ЭВМ по компьютерной сети предприятия. Возможно также подключение дисплея, печатающего устройства и модуля вспомогательной памяти.

Содержание вычислительного процесса заключается в решении логических уравнений типа  $y = F(x)$ , с помощью которых описывается алгоритм управления объектом. Набор выходных функций  $y_i(j, k)$  состоит из выходных функций  $Y(i, k)$ , поступающих на объект для осуществления функций управления, и внутренних функций  $Y(i, j)$ , не имеющих реального выхода и ограничивающих область своего действия самой программой; их можно назвать промежуточными выходами. Сюда же относятся сигналы таймеров, счетчиков. Набор входных функций включает сигналы датчиков; внутренние функции: таймеры, счетчики.

Таким образом, в течение цикла своей работы ПК обменивается информацией с внешней средой и осуществляет обработку этой информации в соответствии с записанной в нем программой. Чем короче цикл работы ПК, тем выше его вычислительная мощность, которая обычно оценивается по времени обработки 1 К логических слов. Время, необходимое для осуществления одного цикла опроса, – время сканирования – зависит от объема программы и элементной базы и для современных ПЛК составляет обычно 2–3 мс на 1 К слов памяти.

Разнообразие требований потребителей ПК предопределило модульность их конструкции, предусматривающую широкую номенклатуру взаимозаменяемых модулей, устанавливаемых в одном каркасе. Особенно это касается модулей входов-выходов. Обычно ПЛК имеют фиксированную общую сумму входов-выходов (например, 16, 64, 128, 256, 512, 1024, 2048) с возможностью перераспределения в пределах общей суммы. Перераспределение заключается в изъятии и установке соответствующих модулей.

## 1.2. Порядок синтеза систем автоматизации

Для выполнения синтеза логических управляющих систем промышленной автоматики необходимо владеть исчерпываю-



щей информацией о технологическом процессе, что позволяет сформулировать условия работы системы управления технологическим объектом. С этой целью необходимы предварительное изучение технологического процесса совместно со специалистами-технологами, его детализация вплоть до каждой элементарной операции, сведения о применяемом оборудовании. Устанавливается последовательность операций и необходимые временные задержки для всех режимов работы объекта управления; определяются параметры и показатели, подлежащие контролю и учету в ходе процесса, тип датчиков, их количество и размещение; устанавливаются информационные связи с локальными и центральными пультами управления.

При определении технических условий работы проектируемой системы управления необходимо руководствоваться действующими правилами, нормами и инструкциями. Кроме того, следует изучить принцип работы действия всех систем управления аналогичного технического оборудования.

Синтез содержит следующие этапы.

**1.** Разделение всех действующих систем управления сигналов на входные, выходные и промежуточные. К *входным* относятся те сигналы, которые поступают от кнопок управления, конечных и промежуточных выключателей, датчиков и т.д. Каждому сигналу присваивается буквенное обозначение.

Эту информацию целесообразно представить в виде таблицы, которая должна содержать обозначения сигналов, их адресацию и привязку к входным (выходным) разъемам ПК.

**2.** Составление алгоритма управления механизма, таблиц включения исполнительных органов и соответствующих им элементов контроля.

**3.** Вычерчивание циклограммы работы механизма

**4.** Описание алгоритма (циклограммы) с помощью управлений алгебры логики.

**5.** Вычерчивание релейно-контактного эквивалента алгоритма; при этом управление приводится к базису И, ИЛИ, НЕ. В принципе, данный этап не является необходимым, однако наличие релейно-контактного эквивалента облегчает отладку программы в цеховых условиях и ее чтение. Кроме того, многие ПК содержат обозначения контактной символики на клавишах программаторов и позволяют вычерчивать релейно-контактные алгоритмы на дисплее.

**6.** Составление программы управления ПК, ее отладка и запись в память ПК.

7. Начертание принципиальной схемы электроавтоматики, составление спецификаций аппаратов, их конструктивная встройка в станцию управления, разработка монтажных схем. Этот принцип работы обычно для традиционных методов реализации электроавтоматики с применением «жестких» структур может выполняться после 3-го этапа параллельно с разработкой алгоритма и программы для ПК или даже опережать их, что является одним из важнейших преимуществ ПК, так как позволяет ускорить передачу технической документации в производство.

### ***Контрольные вопросы и задания***

1. В чем состоит отличие ПЛК от ЭВМ общего назначения?
2. Назовите основные функциональные узлы ПЛК.
3. Обладают ли ПЛК достаточным быстродействием? Сравните этот показатель с реализацией автоматизации, например на электромагнитных реле.
4. Какой математический аппарат используется для описания алгоритма управления объекта?
5. В чем заключается отличие реализации алгоритма управления на микросхемах «жесткой» логики и спомощью ПЛК?

## **ГЛАВА 2. ЦИКЛОВОЕ ПРОГРАММНОЕ УПРАВЛЕНИЕ ТЕХНОЛОГИЧЕСКИМИ КОМПЛЕКСАМИ**

### **2.1. Функциональный состав цикловых систем программного управления**

Значительная доля оборудования промышленных предприятий выполняет обработку однотипной продукции с редкой сменой циклов движений. *Циклом* называется совокупность элементарных операций, осуществляемых в определенной последовательности, обеспечивающих выполнение оборудованием своих рабочих функций. Автоматизация работы такого оборудования решается с помощью *систем циклового программного управления* – ЦПУ. Удачно выбранная система ЦПУ оборудованием обеспечивает гибкость его автоматизации, т.е. возможность перехода на выпуск нового изделия

в пределах определенной номенклатуры, и решает противоречия между сроками и стоимостью изготовления специального оборудования и сроками обновления изделия, выпускаемого с его помощью.

*Элементарные операции* разделяются на *рабочие* и *вспомогательные*, они называются *этапами цикла* или *тактом работы системы*. В течение этапа цикла в системе не происходит изменений в состоянии управляющих воздействий. Этапы цикла характеризуются следующими параметрами: скоростью и направлением перемещения рабочего органа, наличием смазки и охлаждения, видом режущего инструмента и т.д. Применительно к металлообработке параметры рабочих этапов цикла определяют режимы обработки. В системах ЦПУ программа содержит в числовом виде только информацию о цикле и режимах обработки, а величину перемещения рабочих органов задают настройкой (обычно вручную) датчиков перемещений (кулачков, упоров и др.).

Кроме управления, в функции пути в оборудовании с ЦПУ широко применяется управление в функции времени или других параметров (давления, температуры, силы зажатия и т.д.).

В обобщенном виде систему ЦПУ можно представить в виде, приведенном на рис. 2.1.1.

Программа работы оборудования в виде этапов цикла формируется в блоке задания и запоминания программы. С помощью блока поэтапного ввода команды программы вводятся в исполнительный механизм через блок связи с объектом. Режим работы системы, т.е. переход от выполненного этапа цикла к следующему зависит от положения рабочего органа, которое контролируется устройствами обратной связи. В каче-

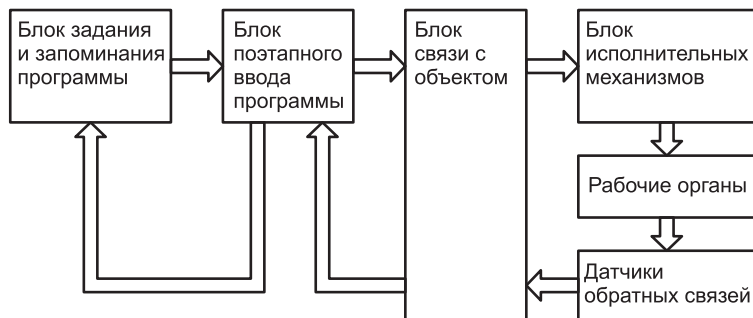


Рис. 2.1.1. Функциональная схема системы циклового программного управления

## Оглавление

Предисловие .....	3
Список сокращений .....	5
Введение .....	7
<b>Глава 1. Проектирование систем автоматизации .....</b>	<b>14</b>
1.1. ПЛК как универсальное средство автоматизации .....	14
1.2. Порядок синтеза систем автоматизации .....	17
<i>Контрольные вопросы и задания</i> .....	19
<b>Глава 2. Цикловое программное управление технологическими комплексами .....</b>	<b>19</b>
2.1. Функциональный состав цикловых систем программного управления .....	19
2.2. Способы формализации работы технологического оборудования .....	22
2.2.1. Таблицы истинности .....	24
2.2.2. Таблицы состояний .....	26
2.2.3. Циклограммы .....	27
2.2.4. Граф-схемы алгоритмов работы .....	31
2.2.5. Блок-схемы алгоритмов .....	32
<i>Контрольные вопросы и задания</i> .....	33
<b>Глава 3. Программное обеспечение ПЛК .....</b>	<b>34</b>
3.1. Обзор языков программирования .....	34
3.2. Язык РКС .....	43
<i>Контрольные вопросы и задания</i> .....	54
<b>Глава 4. Расширение функциональных возможностей языка РКС: LAD и FBD .....</b>	<b>55</b>
4.1. Операции бинарной логики .....	55

4.2. Функции для работы с памятью .....	70
4.3. Функции передачи .....	91
4.3.1. Блочный элемент MOVE .....	92
4.3.2. Перемещение операндов .....	95
4.4. Таймеры .....	95
4.4.1. Программирование таймера .....	97
4.4.2. Запуск таймера .....	99
4.4.3. Последовательность операций с таймерами .....	102
4.4.4. Импульсный таймер .....	103
4.4.5. Таймер задержки включения .....	104
4.4.6. Таймер задержки выключения .....	105
4.5. Счетчики .....	107
4.6. Функции сравнения .....	114
4.7. Арифметические функции .....	119
4.7.1. Обработка арифметической функции .....	120
4.7.2. Вычисления с типом данных INT .....	124
4.8. Функции преобразования .....	125
4.8.1. Обработка функции преобразования .....	125
4.8.2. Преобразование чисел типов INT и DINT .....	129
4.8.3. Преобразование чисел типа BCD .....	130
4.9. Обозначение функций (инструкций) для различных моделей ПЛК .....	131
<i>Контрольные вопросы и задания по языкам LAD – FBD</i> .....	134
<b>Глава 5. Языки программирования STL и SCL</b> .....	141
5.1. Двоичные логические операции в STL .....	141
5.2. Операции с памятью .....	153
5.3. Функции таймеров .....	166
5.4. Функции счетчиков .....	178
5.5. Операторы управления в языке SCL .....	183
<i>Контрольные вопросы и задания</i> .....	189
<b>Глава 6. Язык S7-GRAPH</b> .....	189
<i>Контрольные вопросы и задания</i> .....	197
<b>Глава 7. Автоматизация производства на основе промышленных сетей ПЛК</b> .....	197
7.1. Промышленная сеть PROFINet .....	197
7.1.1. Распределенные полевые устройства .....	203
7.2. Распределенная автоматизация .....	207
7.3. Коммуникации в PROFINet .....	211

7.4. Инсталляция сети .....	217
7.5. Топологии сетей .....	217
7.6. Технология PROFINet: кабели, розетки, коннекторы и коммутаторы .....	220
7.7. Интеграция с IT (Internet Technologies) .....	222
7.7.1. Интеграция систем полевых шин .....	226
<i>Контрольные вопросы и задания</i> .....	230
<b>Глава 8. Проектирование дискретной системы управления силовым столом</b> .....	230
8.1. Общая компоновка станка для многопозиционной обработки .....	230
8.1.1. Циклограмма работы силового стола .....	231
8.1.2. Таблица функциональных назначений .....	233
8.1.3. Управляющая программа в символах РКС .....	235
8.1.4. Схема подключения входов и выходов .....	237
<i>Контрольные вопросы и задания</i> .....	242
<b>Глава 9. CoDeSys – система программирования ПЛК в стандарте МЭК 61131-3</b> .....	242
9.1. Пишем первую программу .....	244
9.2. Визуализация .....	252
9.3. Контроль качества выполненных работ .....	255
<i>Контрольное задание</i> .....	258
<b>Глава 10. Данные и переменные в CoDeSys</b> .....	258
10.1. Типы данных .....	258
10.2. Элементарные типы данных .....	259
10.2.1. Целочисленные типы .....	259
10.2.2. Логический тип .....	261
10.2.3. Действительные типы .....	262
10.2.4. Интервал времени .....	262
10.2.5. Время суток и дата .....	263
10.2.6. Строки .....	264
10.2.7. Иерархия элементарных типов .....	265
10.3. Пользовательские типы данных .....	265
10.3.1. Массивы .....	265
10.3.2. Структуры .....	267
10.3.3. Перечисления .....	269
10.3.4. Ограничение диапазона .....	270
10.3.5. Псевдонимы типов .....	270
10.4. Переменные .....	272

10.4.1. Идентификаторы	272
10.4.2. Распределение памяти переменных	273
10.4.3. Прямая адресация	274
10.4.4. Поразрядная адресация	276
10.4.5. Преобразования типов	277
10.4.6. Особенности вычислений	278
10.4.7. Венгерская запись	282
10.4.8. Формат BCD	284
<b>Глава 11. Языки SFC и CFC стандарта МЭК 61131-3 в CoDeSys</b>	<b>285</b>
11.1. Проблема программирования ПЛК	285
11.2. ПЛК как конечный автомат	287
11.3. Сети Петри	289
11.4. SFC-диаграммы	291
11.5. Последовательные функциональные схемы (SFC)	292
11.5.1. Шаги	292
11.5.2. Переходы	293
11.5.3. Начальный шаг	293
11.5.4. Параллельные ветви	294
11.5.5. Альтернативные ветви	295
11.5.6. Переход на произвольный шаг	296
11.6. Упрощенный SFC	297
11.6.1. Входные и выходные действия	298
11.6.2. Механизм управления шагом	298
11.7. Стандартный SFC	300
11.7.1. Классификаторы действий	301
11.7.2. Действие – переменная	305
11.7.3. Механизм управления действием	306
11.7.4. Внутренние переменные шага и действия	308
<b>Глава 12. Функциональные блоки и программы SFC</b>	<b>309</b>
12.1. Отладка и контроль исполнения SFC	311
12.2. Пример реализации последовательного управления по времени (PRG LD, SFC)	312
12.3. Пример управления реверсивным приводом (SFC)	314
<b>Глава 13. Графический редактор CFC</b>	<b>319</b>
13.1. Наиболее важные команды редактора CFC	319
13.1.1. Соединение элементов	319
13.1.2. Удаление линий соединения	321
13.1.3. Вставка входов-выходов «на лету»	321

13.1.4. Порядок выполнения схемы .....	322
13.1.5. Обратные связи CFC .....	322
13.2. CFC в режиме Online .....	323
13.2.1. Мониторинг .....	323
13.2.2. Контроль потока .....	324
13.2.3. Точки останова .....	324
13.2.4. Метка Return .....	324
13.2.5. Выполнение программы по шагам .....	324
<b>Глава 14. Примеры реализации в CoDeSys элементов систем управления .....</b>	<b>324</b>
14.1. Регулятор .....	324
14.2. Широтно-импульсный модулятор на базе таймера .....	332
<i>Контрольные задания по гл. 10–14 .....</i>	<i>335</i>
<b>Глава 15. Имитационные и управляющие модели систем управления электроприводами .....</b>	<b>336</b>
15.1. Имитационная модель управления шаговым двигателем ..	336
15.1.1. Описание шагового двигателя .....	336
15.1.2. Двигатели с переменным магнитным сопротивлением .....	337
15.1.3. Двигатели с постоянными магнитами .....	338
15.1.4. Гибридные двигатели .....	339
15.1.5. Биполярные и униполярные шаговые двигатели ....	341
15.1.6. Построение программных моделей .....	346
15.1.7. Результаты работы программы .....	354
15.1.8. Руководство по эксплуатации .....	356
15.1.9. Практический пример системы управления шаговым двигателем на основе микроконтроллера семейства AVR ...	359
15.2. Обзор возможностей системы CoDeSys 3.0 .....	362
15.2.1. Система расширения реального времени .....	362
15.2.2. Работа с файлами в системе CoDeSys 3.0 .....	363
15.2.3. Объектно-ориентированные расширения МЭК 61131-3 .....	364
15.3. Имитационная модель управления ДПТ .....	370
15.3.1. Основные формулы, используемые при управлении ДПТ .....	370
15.3.2. Система управления ДПТ с ПИД-регулятором .....	372
15.3.3. Подключение силового ключа к выходу ШИМ .....	374
15.4. Реализация и анализ системы управления ДПТ .....	375
15.4.1. Разработка системы управления ДПТ в среде CoDeSys 3.0. Проектирование структуры системы управления .....	375



15.4.2. Технические характеристики двигателя .....	376
15.4.3. Статическая модель предмета разработки .....	377
15.4.4. Диаграммы изменения основных характеристик ДПТ и управляющих сигналов ПЛК .....	381
Задачи и упражнения .....	388
Примеры задач .....	396
Литература .....	398

Учебное издание

**Петренко Юрий Николаевич**  
**Новиков Сергей Олегович**  
**Гончаров Александр Александрович**

**ПРОГРАММНОЕ УПРАВЛЕНИЕ ТЕХНОЛОГИЧЕСКИМИ  
КОМПЛЕКСАМИ В ЭНЕРГЕТИКЕ**

Учебное пособие

Редактор *А.В. Новикова*  
Художественный редактор *В.А. Ярошевич*  
Технический редактор *Н.А. Лебедевич*  
Корректоры *Т.К. Хваль, О.В. Ракицкая*  
Компьютерная верстка *А.Н. Бабенковой*

Подписано в печать 20.03.2013. Формат 84×108/32. Бумага офсетная.  
Гарнитура «Times New Roman». Офсетная печать. Усл. печ. л. 21,42.  
Уч.-изд. л. 23,2. Тираж 250 экз. Заказ 590.

Республиканское унитарное предприятие «Издательство “Вышэйшая школа”».  
ЛИ № 02330/0494062 от 03.02.2009. Пр. Победителей, 11, 220048, Минск.  
e-mail: [market@vshph.com](mailto:market@vshph.com) <http://vshph.com>

Филиал № 1 открытого акционерного общества «Красная звезда».  
ЛП № 02330/0494160 от 03.04.2009. Ул. Советская, 80, 225409, Барановичи.