

С.О. Новиков Ю.Н. Петренко

ПРОГРАММНОЕ УПРАВЛЕНИЕ ТЕХНОЛОГИЧЕСКИМИ КОМПЛЕКСАМИ

Для студентов
учреждений высшего образования

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, получили название *программируемых логических контроллеров*. В связи с повышенной стоимостью такой аппаратуры ее применение оправданно, 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) равнозначны. Разработка ПЛК стала возможной благодаря развитию следующих основных направлений: релейной бесконтактной автоматики (бесконтактных логических управляющих систем); микропроцессорного управления технологическим оборудованием; систем программирования; элементной базы. Несмотря на значительную 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 дают возможность использовать 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.

Программируемый логический контроллер обеспечивает 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

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 входо-выходов. Обычно ПЛК имеют фиксированную общую сумму входо-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 всех систем управления аналогичного технического оборудования.

Синтез содержит следующие этапы.

1. Разделение всех действующих систем на управляющие входными, выходными и промежуточными сигналами. 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 (AND), ИЛИ (OR), НЕ (NOT). 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 третьего 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 с помощью 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.

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 (ИМ) 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.

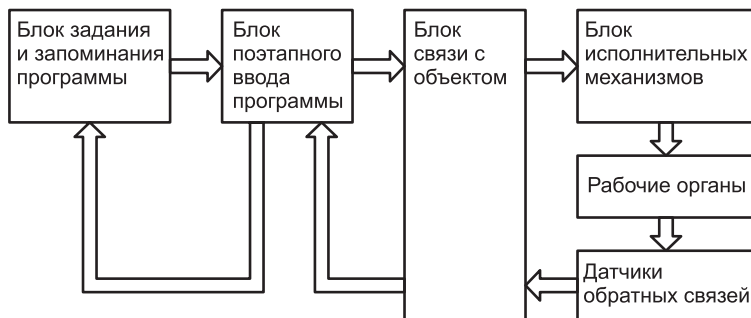


Рис. 2.1. Функциональная схема системы циклового программного управления

В качестве таких устройств широко применяются различные путевые переключатели, в том числе бесконтактные, микропереключатели, по команде которых блок поэтапного ввода программы приступает к опросу программы очередного этапа цикла. Как отдельные устройства системы ЦПУ, так и вся система может строиться не только на электрических компонентах, но и на пневматических, гидравлических, хотя электрические компоненты наиболее распространены. Блок задания и запоминания программы может быть выполнен в виде штекерной панели, панели кнопок или переключателей.

Блок поэтапного ввода программы реализуется обычно в виде счетно-распределительной схемы, построенной на шаговых искателях, электромагнитных реле или логических элементах. Есть также примеры выполнения этого блока в виде барабанных командоаппаратов, в которых имеются барабаны с фиксированными в гнездах кулачками, воздействующими на переключатели по мере их вращения.

Согласование блока ввода программы с исполнительными элементами оборудования и датчиками обратных связей выполняет блок связи с объектом. Здесь же производится усиление (ослабление) и потенциальная развязка сигналов. Исполнительный механизм находится непосредственно на технологическом оборудовании и осуществляет обработку заданных программой этапов цикла. Под исполнительным механизмом понимаются электрический, гидравлический, пневматический или иной привод рабочих органов оборудования, муфты (электромагнитные, гидравлические и др.), электромагниты, гидрозолотники и т.п.

Датчики обратных связей, осуществляющие контроль окончания обработки этапов цикла, обычно также монтируются непосредственно на технологическом оборудовании. Рабочие органы – это различные механизмы технологического оборудования, участвующие в реализации управления (суппорты станков, поворотные столы, насосы, механизмы роботов и манипуляторов и т.д.).

Как указывалось, контроль обработки программы наиболее часто осуществляется с помощью путевых переключателей. Контроль в *функции времени* с помощью реле времени применяется в тех случаях, когда использование других способов затруднительно и когда время обработки этапа программы колеблется мало. Примером может служить включение привода периодической подачи круга плоскошлифовального станка.

Контроль в функции давления удобно применять для механизмов перемещения рабочих органов, приводимых в движение гидроцилиндрами.

Контроль в функции тока можно использовать в электро-механических зажимных устройствах с приводом от электродвигателей.

Для ряда механизмов применяется также контроль в функции числа операций.

Цикловые системы являются простейшим видом программного управления и характеризуются тем, что информация о величинах перемещений рабочих органов задается посредством относительно трудоемкой настройки кулачков-упоров, воздействующих на путевые переключатели. В отличие от них в системах ЧПУ размерная информация о перемещениях задается на программоносителе с помощью одного из видов цифровых кодов. Таким образом, система ЧПУ является более гибкой и универсальной. Следует также отметить, что в системах ЧПУ применяются элементы циклового управления в основном для задания движения вспомогательных механизмов. Например, в гибких производственных модулях цикловое управление используется для задания цикла работы механизма смены инструмента, ориентации приспособлений-спутников, поворотных столов и т.д. При этом программа цикла вводится в систему ЧПУ, а его отработка определяется настройкой конечных (путевых) переключателей. В таких системах перенастройка переключателей производится крайне редко или вообще не производится. Гибкость цикла определяется возможностями ЧПУ. Вместо путевых выключателей применяются также кодовые метки, индуктивные и оптические датчики перемещений.

2.2. Способы формализации работы технологического оборудования

Функционирование объекта управления (технологического оборудования) – это взаимодействие его рабочих органов в соответствии с его структурой и АУ.

Под алгоритмом управления понимается совокупность правил выработки управляющих воздействий к исполнительным элементам объекта управления, обеспечивающих его функ-

ционирование с целью решения поставленной перед ним задачи. Выработку указанных воздействий осуществляет устройство управления на основе программы, введенной в том или ином виде оператором или поступившей от ЭВМ верхнего уровня по каналам локальной вычислительной сети и сигналов датчиков, поступающих от объекта управления.

Структура и содержание АУ зависят от особенностей того объекта, которым предстоит управлять. Многообразие объектов управления не позволяет сформулировать единый подход к построению алгоритмов управления ими.

Предполагая, что все технологические последовательности объекта управления в конечном итоге представляют собой повторяющиеся циклы, выделяют два существенно отличных вида моделей: *комбинационный* и *последовательный*. В первом случае последующее функционирование объекта определяется только его состоянием при выполнении предшествующих операций; во втором – последовательностью смены предшествующих операций.

Общий подход при формализации технологического цикла предполагает следующие этапы.

1. Составление содержательного описания, в котором в произвольной форме описывается технологический цикл при его нормальном протекании и в аварийных ситуациях. Способы описания и детализация зависят прежде всего от целей описания. Само описание также различается по степени формализации – от использования естественного языка до построения формализованной математической модели.

2. Разбиение цикла на этапы (такты), характеризующиеся неизменным состоянием исполнительных приводов и контролируемых параметров.

3. Анализ переходов от одного такта к другому при нормальных и аварийных ситуациях для выявления причин переходов, т.е. выявление изменения состояния командных и исполнительных органов, являющихся причиной перехода.

4. Установление причинно-следственных и логических ситуационных связей между входами и выходами объекта управления, обусловленных требованиями технологии.

5. Составление формализованного графического представления алгоритма функционирования в виде таблицы, циклограммы, схемы и т.п.

2.2.1. Таблицы истинности

Для формального описания комбинационных моделей, в которых дальнейший ход цикла определяется состоянием входов и выходов объекта управления только в данном такте, часто используют таблицы истинности, отражающие однозначное соответствие дискретных состояний входов и выходов.

Активное (включенное) или пассивное (отключенное) состояние исполнительного элемента или уровень контролируемого сигнала (высокий, низкий) может обозначаться любыми символами. Обычно для этих целей используются дискретные сигналы «1» и «0». При числе n входных сигналов возможно $N = 2^n$ сочетаний комбинаций их единичного и нулевого уровней. Поскольку последовательность смены комбинаций в данном случае роли не играет, в таблице истинности их удобно располагать в виде кодов натурального ряда двоичных чисел. Следует также отметить, что все комбинации состояний входов и датчиков могут реально иметь место.

Пример. В зависимости от размера деталей, который может принимать значения L , $2L$ и более $3L$, произвести их сортировку открыванием соответствующей заслонки бункера-накопителя $K1$, $K2$ или $K3$. Контроль размеров деталей можно осуществлять тремя датчиками – $SQ1$, $SQ2$, $SQ3$, установленными поперек роликового транспортера (рис. 2.2).

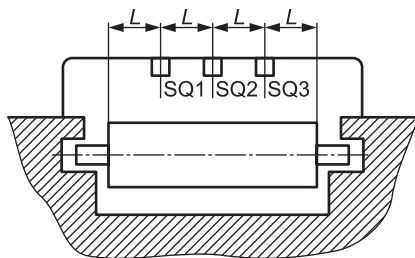


Рис. 2.2. Схема сортировки деталей

Выделим следующие ситуации (табл. 2.1):

- размер детали равен L – включен один из датчиков ($SQ1$, $SQ2$ или $SQ3$), открыт бункер $K1$ (такты 1, 2, 4);
- размер детали равен $2L$ – включены два датчика ($SQ1$, $SQ2$ или $SQ2$, $SQ3$), открыт бункер $K2$ (такты 3, 6);

- размер детали более $3L$ – включены все три датчика, открыт бункер К3 (такт 7).

Таблица 2.1. Таблица истинности

Номер комбинации (такта)	Состояние					
	входов			выходов		
	SQ1	SQ2	SQ3	K1	K2	K3
0	0	0	0	0	0	0
1	0	0	1	1	0	0
2	0	1	0	1	0	0
3	0	1	1	0	1	0
4	1	0	0	1	0	0
5	1	0	1	0	0	0
6	1	1	0	0	1	0
7	1	1	1	0	0	1

Таблица истинности составляется для всех возможных комбинаций датчиков. Число таких комбинаций $N = 2^n = 8$. Состояние датчика равно 1 при наличии детали; выходной сигнал принимает значение 1 при открытом бункере; противоположные состояния соответствуют 0. Незаполненные клетки состояний выходов (такт 5) соответствуют нереальной ситуации, когда датчики SQ1 и SQ3 включены, а SQ2 отключен. Это может случиться лишь при неисправном датчике SQ2. Данные состояния заполняются нулями, и при его появлении должен быть подан аварийный сигнал.

В отличие от комбинационных моделей при составлении последовательных моделей необходимо отражать однозначное соответствие состояний выходов комбинациям состояний входов как в данном такте, так и в предыдущих, которое показывает, что одна и та же комбинация входов в данном такте может вызвать переход в разные новые состояния в зависимости от того, каким было предшествующее состояние. Таким образом, в модели должны быть отражены не только данный такт, но и предыстория.

При большом числе состояний применяются таблицы состояний, а также графические способы: циклограммы, структурные схемы алгоритмов и граф-схемы алгоритмов.

Ниже рассматривается применение этих методов на примерах.

2.2.2. Таблицы состояний

Число строк таблицы соответствует числу состояний, число столбцов – числу возможных комбинаций переменных; крайний столбец слева фиксирует номера исходных состояний. Над таблицей приводится мнемодиаграмма. В клетках таблицы проставляются номера состояний, обусловленных исходным состоянием и возникшей комбинацией управляющих переменных.

Пример. Составить таблицу состояний для подъемника, перемещающегося с одного уровня на другой реверсивным приводом с помощью контакторов КВ и КН. Пуск привода подъемника осуществляется по команде от этажных кнопок SB1, SB2, SB3 и SB4. Для ограничения хода служат конечные выключатели SA1 и SA2, для контроля закрытия дверей шахты – SQ1–SQ4. Перегрузка контролируется с помощью реле. Все входные переменные сводятся к следующим: пуск вверх ПВ – нажаты кнопки «Вверх» SB1, SB2, SB3 на первом, втором или третьем этажах; пуск вниз ПН – нажаты кнопки «Вниз» SB4, SB5, SB6 на четвертом – втором этажах; разрешение движения вверх РВ – все двери закрыты, не нажат конечный выключатель SA1, нет перегрузки; разрешение движения вниз РН – закрыты все двери, не нажат конечный выключатель SA2, нет перегрузки.

Число возможных состояний равно трем: 1 – движение вверх; 2 – движение вниз; 0 – кабина неподвижна.

Таблица состояний (табл. 2.2) содержит три строки и $2^4 = 16$ столбцов, число которых равно числу комбинаций входных переменных.

В первой строке отражается движение вверх цифрой 1 в тех столбцах, где есть сигнал РВ; во второй – соответственно вниз (цифра 2) там, где есть сигнал РН; в остальных столбцах проставлены нули. В третьей строке указывается исходное состояние кабины – «0», который проставляется в столбцах 0–3, поскольку здесь отсутствуют сигналы РВ и РН, в столбцах 7, 8, 15 – так как нет вызова (отсутствуют ПВ и ПН), в столбцах 4 и 14 – так как здесь разрешение и вызов противоречат друг другу. В столбцах 6, 9 проставляется цифра 1, в столбцах 11, 12 – цифра 2, так как разрешение соответствует вызову. Состояние входных переменных для столбцов 5, 10, 13 противоречит

Оглавление

Список основных сокращений	3
Предисловие	5
Введение	8
Глава 1. Проектирование систем автоматизации	15
1.1. ПЛК как универсальное средство автоматизации	15
1.2. Порядок синтеза систем автоматизации	18
<i>Контрольные вопросы и задания</i>	20
Глава 2. Цикловое программное управление технологическими комплексами	20
2.1. Функциональный состав цикловых систем программного управления	20
2.2. Способы формализации работы технологического оборудования	23
2.2.1. Таблицы истинности	25
2.2.2. Таблицы состояний	27
2.2.3. Циклограммы	28
2.2.4. Граф-схемы алгоритмов	32
2.2.5. Блок-схемы алгоритмов	33
<i>Контрольные вопросы и задания</i>	34
Глава 3. Программное обеспечение ПЛК	34
3.1. Обзор языков программирования	34
3.2. Язык РКС	44
<i>Контрольные вопросы и задания</i>	50
Глава 4. Расширение функциональных возможностей языка РКС: LAD и FBD	50
4.1. Операции бинарной логики	50

4.1.1. Последовательные и параллельные схемы в LAD	50
4.1.2. Операции бинарной логики в FBD	56
4.1.3. Программирование с учетом состояния контактов датчиков	62
4.2. Функции для работы с памятью	64
4.2.1. Катушки (элемент присваивания результата) в LAD	64
4.2.2. Блочные элементы в FBD	70
4.2.3. Коннекторы	76
4.2.4. Оценка фронта импульса	79
4.3. Функции передачи	83
4.4. Таймеры	87
4.4.1. Программирование таймера	88
4.4.2. Запуск таймера	90
4.5. Счетчики	94
4.5.1. Программирование счетчика	94
4.5.2. Операции со счетчиком	97
4.6. Функции сравнения	101
<i>Контрольные вопросы и задания.</i>	105
Глава 5. Языки программирования STL и SCL	111
5.1. Базовые функции языка STL	111
5.1.1. Двоичные логические операции	111
5.1.2. Операции с памятью	123
5.1.3. Функции таймеров	128
5.1.4. Функции счетчиков	139
5.2. Операторы управления в языке SCL	144
<i>Контрольные вопросы и задания.</i>	150
Глава 6. Язык S7-Graph	150
<i>Контрольные вопросы и задания</i>	157
Глава 7. Автоматизация производства на основе промышленных сетей ПЛК	157
7.1. Промышленная сеть Profinet	157
7.2. Распределенная автоматизация	167
7.3. Коммуникации в Profinet	171
7.4. Инсталляция сети	176
7.5. Топологии сетей	177
7.6. Технология Profinet: кабели, розетки, коннекторы и коммутаторы	180
7.7. Интеграция с IT (Internet Technologies)	182
7.8. Интеграция систем полевых шин	185
<i>Контрольные вопросы и задания</i>	189

Глава 8. Проектирование дискретной системы управления силовым столом. Общая компоновка станка для многопозиционной обработки	189
8.1. Циклограмма работы силового стола	190
8.2. Таблица функциональных назначений	192
8.3. Управляющая программа в символах РКС	194
8.4. Схема подключения входов и выходов	196
<i>Контрольные вопросы и задания</i>	201
Глава 9. CoDeSys – система программирования ПЛК в стандарте МЭК 61131-3	201
9.1. Запуск CoDeSys	201
9.2. Написание первой программы	203
9.3. Визуализация	211
9.4. Контроль качества выполненных работ	213
<i>Контрольное задание</i>	216
Глава 10. Данные и переменные в CoDeSys	216
10.1. Типы данных	216
10.2. Элементарные типы данных	217
10.2.1. Целочисленные типы	217
10.2.2. Логический тип	219
10.2.3. Действительные типы	220
10.2.4. Интервал времени	220
10.2.5. Время суток и дата	221
10.2.6. Строки	222
10.2.7. Иерархия элементарных типов	223
10.3. Пользовательские типы данных	223
10.3.1. Массивы	223
10.3.2. Структуры	225
10.3.3. Перечисления	227
10.3.4. Ограничение диапазона	228
10.3.5. Псевдонимы типов	228
10.4. Переменные	230
10.4.1. Идентификаторы	230
10.4.2. Распределение памяти переменных	230
10.4.3. Прямая адресация	232
10.4.4. Поразрядная адресация	234
10.4.5. Преобразование типов	235
10.4.6. Особенности вычислений	236
10.4.7. Венгерская запись	240
10.4.8. Формат BCD	242

Глава 11. Языки SFC и CFC стандарта МЭК 61131-3 в CoDeSys .	243
11.1. Проблема программирования ПЛК	243
11.2. ПЛК как конечный автомат	244
11.3. Сети Петри	247
11.4. SFC-диаграммы	248
11.5. Последовательные функциональные схемы SFC	250
11.5.1. Шаги	250
11.5.2. Переходы	251
11.5.3. Начальный шаг	251
11.5.4. Параллельные ветви	252
11.5.5. Альтернативные ветви	253
11.5.6. Переход на произвольный шаг	254
11.6. Функциональные блоки и программы SFC	255
11.6.1. Отладка и контроль исполнения	256
11.6.2. Пример реализации последовательного управления по времени (PRG LD, SFC)	258
11.6.3. Пример управления реверсивным приводом	259
11.7. Графический редактор CFC	265
11.7.1. Наиболее важные команды редактора CFC	265
11.7.2. CFC в режиме Online	268
11.8. Примеры реализации в CoDeSys элементов систем управления	270
11.8.1. Регулятор	270
11.8.2. Широтно-импульсный модулятор на базе таймера	277
<i>Контрольные задания</i>	280

Глава 12. Имитационные и управляющие модели систем управления электроприводами	281
12.1. Имитационная модель управления шаговым двигателем	281
12.1.1. Описание шагового двигателя	281
12.1.2. Построение программных моделей	291
12.1.3. Результаты работы программы	299
12.1.4. Руководство по эксплуатации	301
12.1.5. Практический пример системы управления шаговым двигателем на основе микроконтроллера семейства AVR	303
12.2. Обзор возможностей системы CoDeSys 3.0	306
12.2.1. Система расширения реального времени	306
12.2.2. Работа с файлами в системе CoDeSys 3.0	308
12.2.3. Объектно-ориентированные расширения МЭК 61131-3	308
12.3. Имитационная модель управления ДПТ	315
12.3.1. Основные формулы, используемые при управлении ДПТ	315

12.3.2. Система управления ДПТ с ПИД-регулятором	316
12.3.3. Подключение силового ключа к выходу ШИМ	318
12.3.4. Реализация и анализ системы управления ДПТ	319
Глава 13. Автоматизация работы оборудования энергосистемы на основе ПЛК	332
13.1. Автоматическое повторное включение оборудования	333
13.2. Электрическое АПВ однократного действия	336
13.3. Автоматический ввод резерва	339
13.3.1. Основные требования к схемам АВР	339
13.3.2. Принцип действия АВР	341
13.4. Модель автоматизированной системы управления технологическим процессом подстанции в программной среде CoDeSys	344
<i>Контрольные задания к главам 12, 13</i>	<i>348</i>
Задачи и упражнения	349
Примеры решения задач	357
Литература	359

Новиков, С. О.

Н73 Программное управление технологическими комплексами : учебное пособие / С. О. Новиков, Ю. Н. Петренко; под ред. С. О. Новикова. – Минск : Вышэйшая школа, 2019. – 366 с.: ил.

ISBN 978-985-06-3004-9.

Приводятся сведения о порядке разработки систем автоматизации и управления технологическими комплексами применительно к объектам промышленного назначения. Рассматривается реализация проектов систем управления технологических комплексов на основе ПЛК. Описаны языки программирования ПЛК: RSL, LAD, FBD, STL, SCL, Grafset, SFC, CFC.

Представлены промышленные компьютерные сети, объединяющие ПЛК, интеллектуальные датчики и исполнительные устройства, а также реализация управления в реальном времени.

Изложены приемы и правила работы в системе программирования CoDeSys.

Для студентов учреждений высшего образования по специальности «Автоматизация технологических процессов и производств». Будет полезно специалистам, занимающимся разработкой дискретных систем управления технологических комплексов.

УДК 681.51:004.42(075.8)

ББК 32.965я73