

Трушников М.А.

ДИСКРЕТНО-ЛОГИЧЕСКИЕ СИСТЕМЫ УПРАВЛЕНИЯ



Волжский

2017

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ВОЛЖСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

М.А.Трушников

ДИСКРЕТНО-ЛОГИЧЕСКИЕ СИСТЕМЫ УПРАВЛЕНИЯ

Электронное учебное пособие



2017

УДК 681.3(07)

ББК 31я73

Т 804

Рецензенты

Доцент ФГБОУ ВО «ВолгГТУ», канд. тех. наук
А.Г. Алехин,

Преподаватель Волжского политехнического техникума
Б.Г. Севастьянов

Издается по решению редакционно-издательского совета
Волгоградского государственного технического университета

Трушников, М.А.

Дискретно-логические системы управления [Электронный ресурс]: учебное пособие / М.А. Трушников ; ВПИ (филиал) ВолгГТУ.- Электрон. текстовые дан. (1 файл: 1,1МБ). – Волжский, 2017. - Режим доступа: <http://lib.volpi.ru>. – Загл. с титул. экрана.

ISBN 978-5-9948-2715-4

В учебном пособии изложены общие сведения об этапах и основных требованиях к проектированию и эксплуатации автоматизированных систем.

Предназначены для студентов, обучающихся по направлению 15.03.04 "Автоматизация технологических процессов и производств", всех форм обучения

Ил. 38, табл. 5, библиограф.: 11 назв.

ISBN 978-5-9948-2715-4

© Волгоградский государственный
технический университет, 2017
© Волжский политехнический
институт, 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Промышленные контроллеры	4
1.1 Характерные черты и сферы применения контроллеров	7
1.2 Структурные компоненты контроллеров	10
1.3 Исполнение выходных цепей датчиков сигналов	16
1.4 Устройства связи с объектами	19
1.5 Гальваническая изоляция каналов ввода/вывода.....	20
1.6 Системное и прикладное обеспечение промышленных контроллеров	35
1.7 Рабочий цикл плк	36
1.8 Работа циклов на примере контроллера Twido фирмы шнейдер-электрик (Schneider Electric)	40
1.9 Время реакции.....	46
1.10 Классификация контроллеров	47
2 Языки программирования промышленных контроллеров.....	59
2.1 Стандарт МЭК 61131.....	67
2.2 Структура стандарта МЭК 61131	68
2.3 Комплексы программирования МЭК 61131-3	75
2.4 Выбор контроллеров	77
2.5 Современный рынок контроллерных средств.....	80
2.6 Проектирование систем логического управления на базе промышленных контроллеров	81
2.7 Основные этапы проектирования управляющего устройства	83
2.8 Логический синтез управляющих логических устройств с позиционной структурой	90
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	100

ВВЕДЕНИЕ

Дисциплины «Дискретно-логические системы управления» и «Промышленные контроллеры» изучаются в 8-м семестре и являются очень важными при подготовке бакалавров по направлению 15.03.04 "Автоматизация технологических процессов и производств". Также материал данного пособия может использоваться при изучении дисциплины «Автоматизированные системы управления технологическими процессами» при подготовке магистров по направлению 15.04.04 «Автоматизация технологических процессов и производств».

Цель изучения дисциплины – овладение современными техническими средствами для управляющих систем автоматизации, изучение способов и возможностей программирования данных технических средств.

Изучение курса основывается на дисциплинах общетехнического (профессионального) цикла подготовки бакалавров технических наук по направлению 15.03.04 "Автоматизация технологических процессов и производств".

Основное внимание в учебном пособии уделено общим сведениям о физической структуре промышленных контроллеров, способам программирования на языках стандарта МЭК 61131-13.

1 Промышленные контроллеры

Многие системы управления строятся на основе простых не имеющих развитых интерактивных средств микропроцессорных блоках – контроллерах. Контроллер это мозг любой автоматической машины, обеспечивающий логику ее работы.

Термином "промышленный контроллер" обозначают микропроцессорное устройство со встроенным аппаратным и программным

обеспечением, которое используется для выполнения функций управления технологическим оборудованием.

Их развитие идет по двум направлениям – создание специализированных и универсальных контроллеров.

Специализированным контроллером считается устройство, которое разрабатывалось для конкретного применения и не может применяться в иных местах. Разработка специализированного контроллера базируется на идеи минимизации аппаратных и программных средств для того, чтобы в конечном итоге добиться оптимизации таких качественных показателей системы как быстродействие, точность, стоимость, массогабаритных показателей. Такой контроллер может быть встроен только 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 желательно иметь универсальный свободно программируемый контроллер.

Универсальный контроллер строится исходя из концепции размещения в рамках выбранного конструктива максимально возможного набора аппаратных средств избыточного для отдельного конкретного применения.

Каждый пользователь найдет в составе аппаратных ресурсов универсального контроллера средства нужные именно ему. Кроме избыточности аппаратных средств мощным фактором универсализации выступает возможность управлять микропроцессорными устройствами с помощью самостоятельно создаваемых пользователем программ.

На контроллеры в современных системах управления возлагаются самые разнообразные задачи и для их решения требуются самые различные технические средства. В этом смысле концепция универсализации контроллеров (широты их применения) наталкивается на необходимость введения в состав каждого из них большого и избыточного для каждой конкретной задачи набора средств и в конечном итоге эта концепция может потерять смысл.

Выходом из этой проблемы явился модульный подход к реализации контроллеров. При этом подходе каждый из них состоит как бы из двух частей – базовой части, которая включает в себя обязательный и минимальный набор средств, и части, которая может модифицироваться, набираться из нескольких отдельных модулей. Каждый из модулей включает в себя определенный набор средств. Проектировщик как из кубиков набирает тот состав средств, который нужен ему для решения конкретной задачи.

Базовая часть контроллера включает в себя центральный процессорный блок (Central Processors Units – CPU), основными задачами которого является управление периферийными модулями, хранение программы работы и данных, организация обмена данными и командами с центральным управляющим компьютером, обмен данными с прочими контроллерами того же иерархического уровня, проведение диагностики работоспособности узлов.

Модульный принцип набора функций контроллеров как нельзя лучше соответствует общей концепции создания и использования универсальных контроллеров, что способствует расширению сферы их применения и, в конечном счете, увеличению объемов их выпуска.

Кроме рассмотренных очевидных факторов, определяющих широту применения универсальных контроллеров, большую роль играют усилия производителей таких устройств направленные на обеспечение максимального удобства их программирования, отладки систем и возможности включения

контроллеров в состав сетей распределенных (децентрализованных) систем управления.

Важнейшую роль при создании систем на основе универсальных контроллеров играет программное обеспечение, которое четко подразделяется на две группы – программы управления процессом и сервисное программное обеспечение. Диапазон используемых сервисных программных продуктов простирается от средств отладки программ, созданных с использованием конкретного языка программирования того или иного контроллера, использования драйверов аппаратуры до программных комплексов, интегрирующих средства разработки элементов различных уровней системы – от нижнего уровня систем автоматизации до планирования ресурсов предприятия, организации обмена данными между приложениями различного уровня.

При выборе программного обеспечения необходимо принимать во внимание не только опыт и предпочтения пользователя, но и современные тенденции и методы построения систем. Имеется ли необходимость, чтобы программные средства были построены на принципах открытой архитектуры? Требуется ли от разработчиков системы определенная квалификация и опыт в области программирования? Насколько программное обеспечение легко в использовании и сопровождении? Ответы на эти и подобные вопросы в конечном итоге определяют, какое именно программное обеспечение следует выбрать.

1.1 Характерные черты и сферы применения контроллеров

Потребность в применении программируемых логических контроллеров (ПЛК) обозначилась в 60 – х годах прошлого века. Для многих технических задач применяемые средства автоматизации строились преимущественно на

релейно – контактных элементах и за значительный период своего использования обнаружили целый ряд присущих им недостатков:

- для разработки, обслуживания и ремонта таких систем требовались значительные кадровые и экономические ресурсы, так как каждая отдельная схема создавалась под конкретную, и только под неё, задачу;
- переналадка схемы на решение другой задачи была невозможна без полной или кардинальной разборки её, и, если нужно, с возможностью повторного использования компонентов – это трудоёмко и неудобно;
- затруднительно объединение в единую структуру фрагментов системы, территориально удалённых друг от друга;
- практически невозможно построить схему на реле, выдержав её в минимальных габаритах;



Рисунок 1 –Схема взаимодействия с объектом управления

Только с появлением ПЛК, построенных на микропроцессорах, удалось сосредоточить в конструктивно очень компактном модуле сотни и даже тысячи «релейных» элементов, счётчиков, таймеров, пусть даже не существующих физически, а программно воспроизводимых. Это позволило создать гибко перенастраиваемую структуру, способную выполнить любую из очень широкого круга задач.

Возможности программируемых логических контроллеров делают их практически незаменимыми для автоматизации насосных станций, компрессорных установок, котельных, конвейеров, норий, для управления технологическими процессами в комплексе с датчиками самого различного вида, приводными устройствами, клапанами, задвижками и т. д. по схеме взаимодействия с объектом, изображенной на рисунке 1.

Программируемые контроллеры находят применение в различных отраслях промышленности.

Черная и цветная металлургия. Программируемые контроллеры в этих отраслях применяются для управления транспортными операциями на коксовых батареях, загрузке доменных печей, для автоматизации литейных цехов. Их используют также для решения задач, связанных с анализом газов и с контролем качества.

Металлообработка и автомобильная промышленность. Это те отрасли, где ПЛК нашли очень широкое применение. Их можно встретить на автоматических линиях и сборочных конвейерах, на стендах для испытания двигателей, а также на прессах, токарных автоматах, шлифовальных и агрегатных станках, сварочных установках, автоматических станках для резки.

Химическая промышленность. В настоящее время ПЛК используются для управления технологическими установками, устройствами дозирования и смешивания продуктов, очистки отходов химического производства, а также на установках по переработке пластмасс и агрегатах в производстве резины.

Нефтедобыча. Кроме областей применения, аналогичных предыдущей отрасли, ПЛК используется на перекачивающих и распределительных станциях, для управления работой и наблюдения за магистральными трубопроводами.

Транспортные и погрузочно-разгрузочные операции. Программируемые контроллеры используются при сортировке посылок, почтовых отправок, механизированном управлении складскими операциями, упаковке, конвейерной пересылке, комплектовании изделий на поддонах, в лифтовом хозяйстве, грузоподъемных механизмах и др.

Другие области применения. Все случаи использования ПЛК перечислить невозможно. В текстильной промышленности они могут применяться для управления операциями автоматического раскроя тканей и контроля нитей, на транспортных конвейерах. В стекольной промышленности, в производстве хрусталя ПЛК управляют операциями отрезки и упаковки. Устройства логического управления используются при решении задач, связанных с охраной (зданий, заводов) и обеспечении безопасности (ядерная энергетика).

1.2 Структурные компоненты контроллеров

ПЛК представляет собой микропроцессорное устройство, предназначенное для сбора, преобразования, обработки, хранения информации

и выработки команд управления. Каждый из них представляет собой вычислительную машину, имеющую некоторое множество входов и выходов

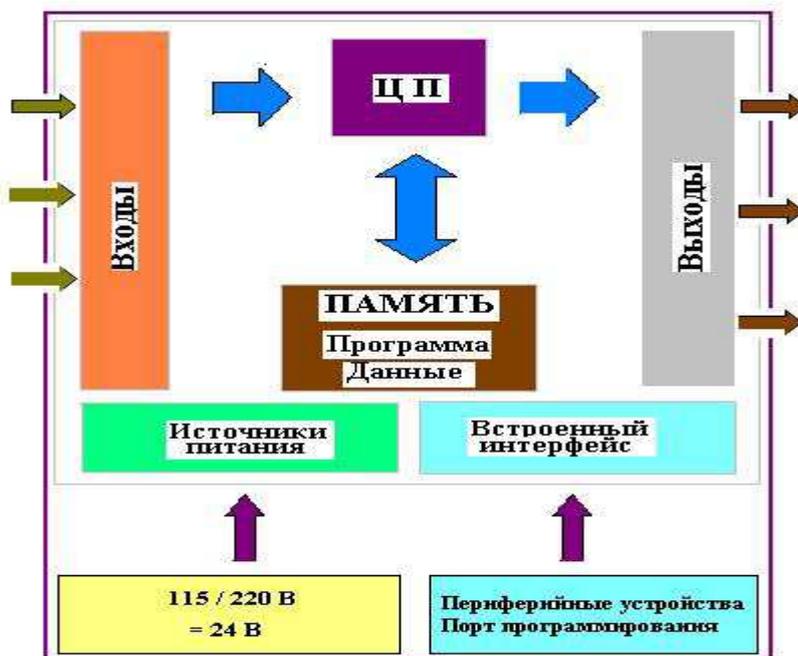


Рисунок 2 - Обобщенная структура контроллера

Контроллер отслеживает изменение входов и вырабатывает программно определенное воздействие на выходах. Обладая памятью, ПЛК в зависимости от предыстории событий способен реагировать по-разному на текущие события. Такая модель соответствует широко известным конечным автоматам. Однако возможности перепрограммирования, управления по времени, развитые вычислительные способности, включая цифровую обработку сигналов, поднимают ПЛК на более высокий уровень.

Первоначально они предназначались для замены релейно-контактных схем, собранных на дискретных компонентах – реле, счетчиках, таймерах, элементах жесткой логики. Принципиальное отличие ПЛК от релейных схем заключается в том, что все его функции реализованы программно. На одном контроллере можно реализовать схему, эквивалентную тысячам элементов жесткой логики. При этом надежность работы схемы не зависит от ее сложности. Связь контроллера с окружающей средой осуществляется через устройства ввода и вывода. Через первые в контроллер вводятся сигналы о параметрах и характеристиках объекта управления, через вторые – выводятся управляющие воздействия на включение / выключение исполнительных устройств. Для прохождения сигналов через порты ввода / вывода эти устройства реализуются не программно, они должны быть физически существующими. Множество внутренних реле, счётчиков, таймеров, необходимых для составления и работы программы, физически не существуют, они моделируются центральным процессором (ЦП) контроллера. Именно благодаря этому удалось в ограниченном объёме «разместить» огромное количество блоков, каждый из которых предназначен для имитации выполнения некоторой вполне конкретной задачи.

Успешность применения ПЛК для задач управления зависит от того, насколько подробно и правильно в контроллер вводится информация о состоянии и поведении объекта. Чтобы грамотно встроить ПЛК в разрабатываемую систему управления (СУ), достаточно выполнить несколько очевидных, но, тем не менее, очень важных правил, основные из которых следующие:

- информация о наиболее важных параметрах и характеристиках объекта, определяющих особенности управления в данной задаче, должна быть введена в контроллер;

- форма представления сигналов должна быть такой, которую контроллер в состоянии правильно воспринять.

Дискретные входы. Один дискретный вход ПЛК способен принимать один бинарный электрический сигнал, описываемый двумя состояниями – включен или выключен. На уровне программы это один бит информации. Кнопки, выключатели, контакты реле, датчики обнаружения предметов и множество приборов с выходом типа «сухой контакт» или «открытый коллектор» непосредственно могут быть подключены к дискретным входам ПЛК.

Состояние некоторых первичных приборов систем промышленной автоматики определяется целым цифровым словом. Для их подключения используют несколько дискретных входов.

Системное программное обеспечение ПЛК включает драйвер, автоматически считывающий физические значения входов в оперативную память. Благодаря этому, прикладному программисту нет необходимости разбираться с внутренним устройством контроллера. С точки зрения прикладного программиста дискретные входы это наборы бит, доступные для чтения в ОЗУ.

Все дискретные входы (общего исполнения) контроллеров обычно рассчитаны на прием стандартных сигналов с уровнем 24 В постоянного тока. Типовое значение тока одного дискретного входа (при входном напряжении 24 В) составляет около 10 мА. В простейшем случае, для подключения нормально разомкнутого контакта, дискретный вход и сам контакт необходимо подключить последовательно к источнику питания в 24 В. Для питания таких внешних датчиков нужен отдельный источник питания. Иногда источник питания внешнего маломощного оборудования включают в состав ПЛК. Дискретные входы некоторых контроллеров рассчитаны на прием уровней сигналов с напряжениями высокого уровня, в том числе переменного тока (например, в 220... 240 В).

Все современные датчики, базирующиеся на самых разнообразных физических явлениях (емкостные, индуктивные, ультразвуковые, оптические и т.д.), как правило, имеют встроенные первичные преобразователи и не требуют дополнительного согласования при подключении к дискретным входам ПЛК.

Аналоговые входы. Аналоговый электрический сигнал отражает уровень напряжения или тока, соответствующий некоторой физической величине, в каждый момент времени. Это может быть температура, давление, вес, положение, скорость, частота и т. д.

Поскольку ПЛК является цифровой вычислительной машиной, аналоговые входные сигналы обязательно подвергаются аналого-цифровому преобразованию (АЦП). В результате, образуется дискретная переменная определенной разрядности. Как правило, в ПЛК применяются 8 - 12 разрядные преобразователи, что в большинстве случаев, исходя из современных требований по точности управления технологическими процессами, является достаточным. Кроме этого АЦП более высокой разрядности не оправдывают себя, в первую очередь, из-за высокого уровня промышленных помех, характерных для условий работы контроллеров.

Для аналоговых входов наиболее распространены стандартные диапазоны постоянного напряжения $-10...+10$ В и $0...+10$ В. Для токовых входов – $0...20$ мА и $4...20$ мА. Особые классы аналоговых входов представляют входы, предназначенные для подключения термометров сопротивления и термопар. Здесь требуется применение специальных технических решений (трехточечное включение, источники образцового тока, схемы компенсации холодного спая, схемы линеаризации и т. д.). Для достижения хороших результатов измерений должно обеспечиваться высокое качество выполнения монтажа внешних аналоговых цепей.

Практически все модули аналогового ввода являются многоканальными. Входной коммутатор подключает вход АЦП к необходимому входу модуля. Управление коммутатором и АЦП выполняет

драйвер системного программного обеспечения ПЛК. Прикладной программист работает с готовыми значениями аналоговых величин в ОЗУ.

Несоответствие физических значений напряжений и токов датчиков уровням входов/выходов контроллера решается применением нормирующих преобразователей или заменой нестандартных датчиков.

Специальные входы. Стандартные дискретные и аналоговые входы ПЛК способны удовлетворить большинство потребностей систем промышленной автоматике. Необходимость применения специализированных входов возникает в случаях, когда непосредственная обработка некоторого сигнала программно затруднена, например, требует много времени.

Наиболее часто ПЛК оснащаются специализированными счетными входами для измерения длительности, фиксации фронтов и подсчета импульсов.

Например, при измерении положения и скорости вращения вала очень распространены устройства, формирующие определенное количество импульсов за один оборот – поворотные шифраторы. Частота следования импульсов может достигать нескольких мегагерц. Даже если процессор ПЛК обладает достаточным быстродействием, непосредственный подсчет импульсов в пользовательской программе будет весьма расточительным по времени. Здесь желательно иметь специализированный аппаратный входной блок, способный провести первичную обработку и сформировать, необходимые для прикладной задачи величины.

Вторым распространенным типом специализированных входов являются входы способные очень быстро запускать заданные пользовательские задачи с прерыванием выполнения основной программы – входы прерываний.

Дискретные выходы. Один дискретный выход ПЛК способен коммутировать один электрический сигнал. Также как и дискретный вход, с точки зрения программы это один бит информации.

Наиболее простым получается соединение с датчиками, обладающими релейными выходами (Рисунок 3). На клеммах 3-4-5 организована контактная группа, гальванически не связанная с остальными компонентами контроллера. Это позволяет подать входной сигнал на ПЛК через эту контактную группу от любого подключенного к ней источника питания.

Использование релейного переключения сигнала может вызвать дополнительные требования по обеспечению импедансных соотношений в прерываемой цепи. Импеданс – это полное или кажущееся сопротивление, учитывающее не только его активную часть, но также индуктивную и емкостную составляющие. Их соотношение оказывает влияние на характер переходных процессов в разрываемой цепи. Это может стать причиной возникновения перенапряжений на элементах схемы и, как следствие, - привести к искрообразованию, что совершенно недопустимо для очень широкого круга применений. Поэтому там, где это необходимо, сопряжение выполняют с применением специальных модулей – барьеров искробезопасности или по особо организованным интерфейсам.

Повышение искробезопасности обеспечивают путём уменьшения значений питающих напряжений, ограничения номинальных величин емкости и индуктивности в разрываемой цепи, применения параметрических (увеличение доли активного сопротивления) и схемных приёмов предотвращения перенапряжений.

Часто встречается такое исполнение выходной цепи датчика,

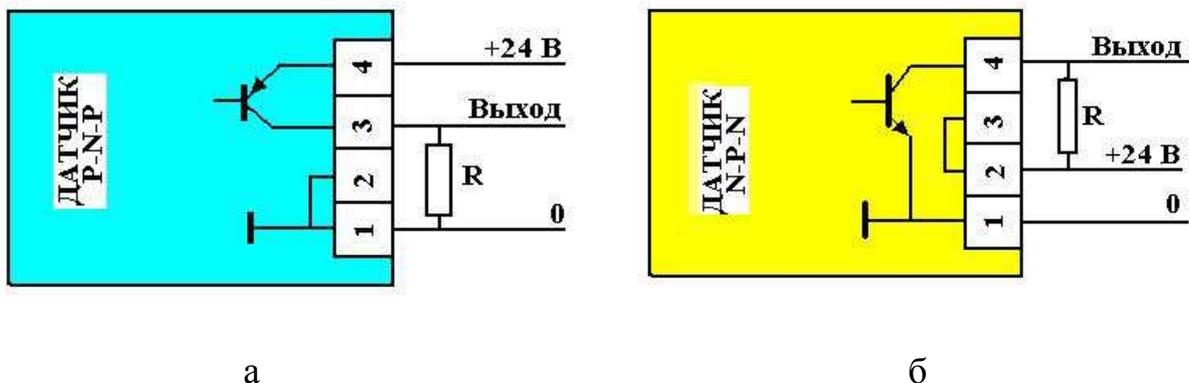
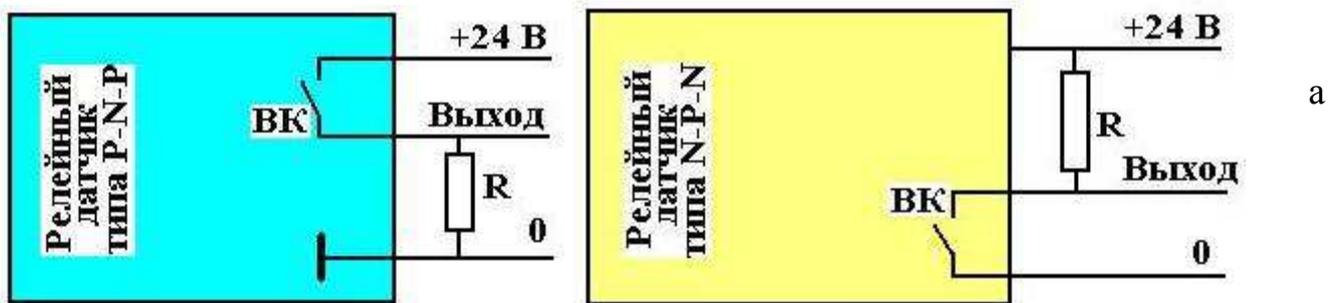


Рисунок 4

при котором переключение выходного сигнала осуществляется бесконтактно, за счёт применения транзистора в выходном каскаде. Распространены два варианта таких выходов $N - P - N$ и $P - N - P$ при трёхпроводном включении (Рисунок 4). Оба варианта построены на каскаде с открытым коллектором, поэтому требуют включения внешнего резистора R . Если иметь в виду подключение датчиков с таким выходом к контроллеру, нужно помнить, что сигнал на вход ПЛК надо подавать плюсом на входную клемму относительно нулевой шины питания. Выполнить это условие можно, если снимать сигнал с R относительно корпуса у датчиков $P - N - P$ типа (Рисунок 4а) или с коллекторно-эмиттерного перехода выходного транзистора у датчиков $N - P - N$ типа (Рисунок 4б). Нетрудно заметить, что включение транзистора в первом случае приводит к выдаче выходного сигнала уровнем **ON**, тогда как во втором – уровнем **OFF**. Получаемая взаимная инверсия снимаемого сигнала с датчиков, обладающих различным исполнением выходов, легко учитывается при составлении программы контроллера.



б

Рисунок 5

Такого же типа выходные цепи широко используются для находящихся широкого применения индуктивных, ёмкостных и других датчиков положения (приближения). Их особенность только в том, что вместо транзистора ключевым элементом может использоваться релейный внутренний контакт ВК.

С предыдущим вариантом (Рисунок 4) это исполнение выхода объединяют похожесть расположения ключевого элемента и одинаковость способа сопряжения с контроллером, а с вариантом, изображенным на рисунке 3, – только то, что ключ выбран контактным.

Обычно в сопроводительной документации на приобретаемые компоненты средств автоматизации указывается вид исполнения входных и выходных портов, приводятся их характеристики. Но номинал резистора R в схемах (Рисунок 4 и 5) скорее всего в этих характеристиках указан не будет. Причина этого в том, что для его определения требуется точно знать, что с чем сопрягается в данном случае и при каком соотношении параметров.

1.4 Устройства связи с объектами

Неотъемлемой частью любой автоматизированной системы управления технологическим процессом являются устройства связи с объектом (далее – УСО), назначение которых заключается в сопряжении датчиков аппаратуры и исполнительных механизмов контролируемого объекта и/или технологического процесса с вычислительными средствами системы. Как правило, на УСО возлагаются следующие функции:

- Нормализация аналогового сигнала – приведение границ шкалы первичного непрерывного сигнала к одному из стандартных диапазонов входного сигнала аналого-цифрового преобразователя измерительного канала.
- Предварительная низкочастотная фильтрация аналогового сигнала – ограничение полосы частот первичного непрерывного сигнала с целью снижения влияния на результат измерения помех различного происхождения. На промышленных объектах наиболее распространены помехи с частотой сети

переменного тока, а также хаотические импульсные помехи, вызванные влиянием на технические средства измерительного канала переходных процессов и наводок при коммутации исполнительных механизмов повышенной мощности.

- Обеспечение гальванической изоляции между источником аналогового или дискретного сигнала и измерительным и/или сетевым каналом системы. В равной степени это относится к изоляции между каналами дискретного вывода системы и управляемым силовым оборудованием. Гальваническая изоляция позволяет снизить влияние на систему помех по цепям заземления за счет их разделения, а также обеспечивает защиту входных и выходных цепей от аварийного попадания на них высокого напряжения. Отсутствие гальванической изоляции допускается только в технически обоснованных случаях.

Помимо перечисленных функций, ряд устройств связи с объектом может выполнять более сложные задачи за счет наличия в их составе подсистемы аналого-цифрового преобразования и дискретного ввода/вывода, микропроцессора и средств организации одного из интерфейсов последовательной передачи данных.

Использование конструктивно законченных типовых модулей устройств связи с объектом, предлагаемых различными производителями, позволяет сократить сроки разработки и ввода в эксплуатацию АСУ ТП, повысить их надежность и упростить обслуживание.

1.5 Гальваническая изоляция каналов ввода/вывода

Основными функциями технических средств для гальванической развязки (нормирующих преобразователей) являются приведение границ параметров

первичного сигнала датчика к одному из стандартных диапазонов, принятому для входного сигнала измерительного средства, и согласование стандартного диапазона выходного сигнала управляющего элемента с видом (ток/напряжение) и диапазоном входного сигнала исполнительного устройства. Дополнительными функциями устройств нормализации, как правило, являются предварительная фильтрация первичных сигналов, а также обеспечение надежной гальванической развязки между первичными измерительными или исполнительными цепями и соответствующими цепями контроля или управления более высокого уровня АСУ ТП.

Нормирующие преобразователи относятся к классу аналоговых модулей УСО. Они должны обладать большой точностью, хорошей линейностью и обеспечивать достаточно высокое напряжение изоляции. Кроме того, желательными являются работа с различными источниками входных сигналов (токи, напряжения, сигналы от терморезисторов, термопар и т. д.), возможность быстрой замены при выходе из строя и низкая стоимость.

Учитывая возросшие возможности современных датчиков, может показаться, что для нормализаторов настало время «уходить со сцены». Однако, это не представляется таким уж однозначным. Перечислим основные причины, которые будут определять востребованность элементов нормализации еще долгие годы.

Во-первых, существует много производств, до сих пор использующих огромный парк первичных датчиков – термопары, терморезисторы и т. д., выходные сигналы которых не унифицированы и нуждаются в согласовании перед использованием в системе управления. При модернизации таких производств с целью повышения точностных характеристик системы управления, экономически оправданной, в первую очередь, оказывается замена именно элементов нормализации, так как тотальная замена первичных датчиков на датчики интегральные ведёт к существенным издержкам, связанным с дорогостоящими монтажными работами (сложный демонтаж, несовпадение

габаритов старого и нового оборудования, необходимость прокладки новых коммуникационных каналов, соответствующих современным спецификациям промышленных интерфейсов и т. д.).

Во-вторых, большинство аппаратуры, используемой в системах автоматизации производства, является универсальной (платы ввода/вывода, устанавливаемые в промышленные компьютеры, контроллеры в распределенных системах управления и т. д.), то есть имеет входы/выходы со стандартными (нормализованными) уровнями и диапазонами сигналов, и при подключении нестандартных сигналов нуждается в использовании определённых средств согласования.

В-третьих, применение нормирующих преобразователей позволяет упростить построение систем с дублированием каналов обработки сигналов от одного первичного датчика при распараллеливании его выходного сигнала.

В-четвёртых, являясь, как правило, одноканальными, элементы нормализации при построении управляющих систем помогают реализовать фундаментальный принцип модульности аппаратуры, минимизируя информационные (и финансовые) потери в системе при выходе нормализатора из строя и облегчая ремонт.

В-пятых, внешний элемент нормализации, выполненный, как правило, на основе специализированной микросхемы или полупроводникового реле для случая дискретного сигнала, по сравнению с платой изолированного ввода/вывода, устанавливаемой в слот компьютера, обеспечивает более качественную гальваническую развязку и более эффективно устраняет возможное влияние сильноточных или высоковольтных входных сигналов на аппаратуру системы.

Характеристики модулей нормализации определяются большим числом параметров. Перечислим и рассмотрим основные из них.

Основными характеристиками нормализаторов, конечно, являются **вид входного и выходного сигнала** – он может представляться как током, так и

напряжением, а также **диапазоны входного и выходного сигнала** – традиционные значения диапазонов по напряжению: 0...10В, 0...1В, 0...5В, ±1В, ±5В, ±10В; по току 0...20мА, 4...20мА.

Кроме этого существует значительное количество характеристик, определяющих точностные характеристики и степень защищенности от помех.

Выходной сигнал инструментального (измерительного) усилителя может быть искажён из-за влияния помех на его входе. Принято различать поперечные помехи, называемые также помехами нормального вида (Noise Normal-Mode), и продольные помехи, именуемые помехами общего вида (Noise Common-Mode). **Поперечные помехи** действуют между входными зажимами измерительного усилителя наряду с входным сигналом. **Продольные помехи** действуют между входными зажимами измерительного усилителя и землей и в общем случае являются следствием электрической связи источника сигнала и измерительного усилителя с землёй через комплексные сопротивления. Разность потенциалов «земель», обусловленная блуждающими токами и определяет возникновение в измерительном контуре дополнительного источника напряжения продольной помехи, суммируемого с напряжением измеряемых сигналов датчика. Напряжение поперечной помехи можно привести к напряжению поперечной помехи – учитывая, что проводники, соединяющие датчик и измерительный усилитель, имеют конечное сопротивление, на входе измерительного усилителя получим напряжение пропорциональное отношению сопротивления проводника к сумме комплексных сопротивлений электрических связей с землёй, внутреннего сопротивления источника продольной помехи и сопротивления проводника. Методом борьбы с продольной помехой является реализация гальванической развязки между входной и выходной цепями измерительного усилителя.

Полоса пропускания (Bandwidth) – это диапазон частот, для которого величина передаточной функции составляет не меньше 70,7% от своего максимального значения.

Время отклика (Response Time) – время, необходимое измерительной системе (прибору), чтобы изменение входного измеряемого сигнала достигло на выходе 90% от своего входного значения.

Нелинейность (Nonlinearity). Линейность системы предполагает, что выходной сигнал прямо пропорционален входному сигналу (чувствительность системы не зависит от значения измеряемой величины), то есть график зависимости выходной величины от входной измеряемой величины представляет собой прямую линию с определённым углом наклона. В реальных устройствах отмечаются отклонения этой зависимости от идеальной линейной характеристики. Разность между реальным значением величины и теоретическим значением, полученным при предположении, что система измерения линейна, определяет такую характеристику, как нелинейность систем.

Подавление помехи нормального вида (NMR – Normal-Mode Rejection). На входе инструментального усилителя на постоянную составляющую сигнала датчика накладывается переменная составляющая помехи. Величина, характеризующая степень уменьшения влияния переменного сигнала помехи нормального вида на значение выходного сигнала и приводимая в децибелах для конкретного значения частоты (например, для частоты силовой цепи питания 50 или 60 Гц), называется подавлением помехи нормального вида.

При реализации в измерительном приборе аналого-цифрового преобразования входного сигнала проблема устранения влияния переменной составляющей помехи нормального вида во входном сигнале решается изменением времени интегрирования (выбирается кратным периоду предполагаемой помехи), а при отсутствии такой возможности – использованием фильтров.

Подавление помехи общего вида (CMR – Common-Mode Rejection) – выраженная в децибелах величина отношения коэффициента передачи дифференциального сигнала к коэффициенту передачи синфазного сигнала

(основной измеряемый сигнал – это дифференциальный сигнал на входе инструментального операционного усилителя, помеха наводится одновременно на оба входа и имеет равную величину). На этой характеристике стоит остановиться более подробно. Основным элементом входной цепи нормализатора является инструментальный операционный усилитель (ОУ). При подаче на оба входа ОУ (прямой и инверсный) напряжений равной величины на выходе должно быть нулевое напряжение. Но это только теоретически; на практике при подключении к входам ОУ синфазного напряжения на его выходе появляется напряжение, отличное от нуля. Существует такая характеристика ОУ, как коэффициент ослабления синфазного сигнала (CMRR), равный отношению коэффициента усиления дифференциального сигнала к коэффициенту усиления синфазного сигнала. Эта характеристика, выраженная в децибелах по отношению к входу и выходу нормализатора, и есть CMR.

Учитывая, что основным видом синфазного напряжения для нормализаторов является синфазная помеха с частотой промышленной сети, показатель ослабления синфазного сигнала приводится для частоты 50 или 60 Гц. Этот показатель очень важен для обеспечения точностных характеристик нормализатора. Чем выше значение CMR, тем лучше параметры инструментального усилителя и, следовательно, точностные характеристики нормализатора в целом.

Трехуровневая (трехсторонняя) изоляция (3-Way Isolation). Нормализаторы проектируются как развязывающие, или изолированные усилители. Подобные требования диктуются областью их применения: например, в медицине, где датчиками служат электроды, прикладываемые к телу человека, большая прочность изоляции необходима по соображениям безопасности, а необходимость изоляции на производстве связана с высокими значениями синфазных напряжений.

Блок-схему развязывающего усилителя можно представить как совокупность входного и выходного каскадов, цепей питания и развязки (рисунок 6).

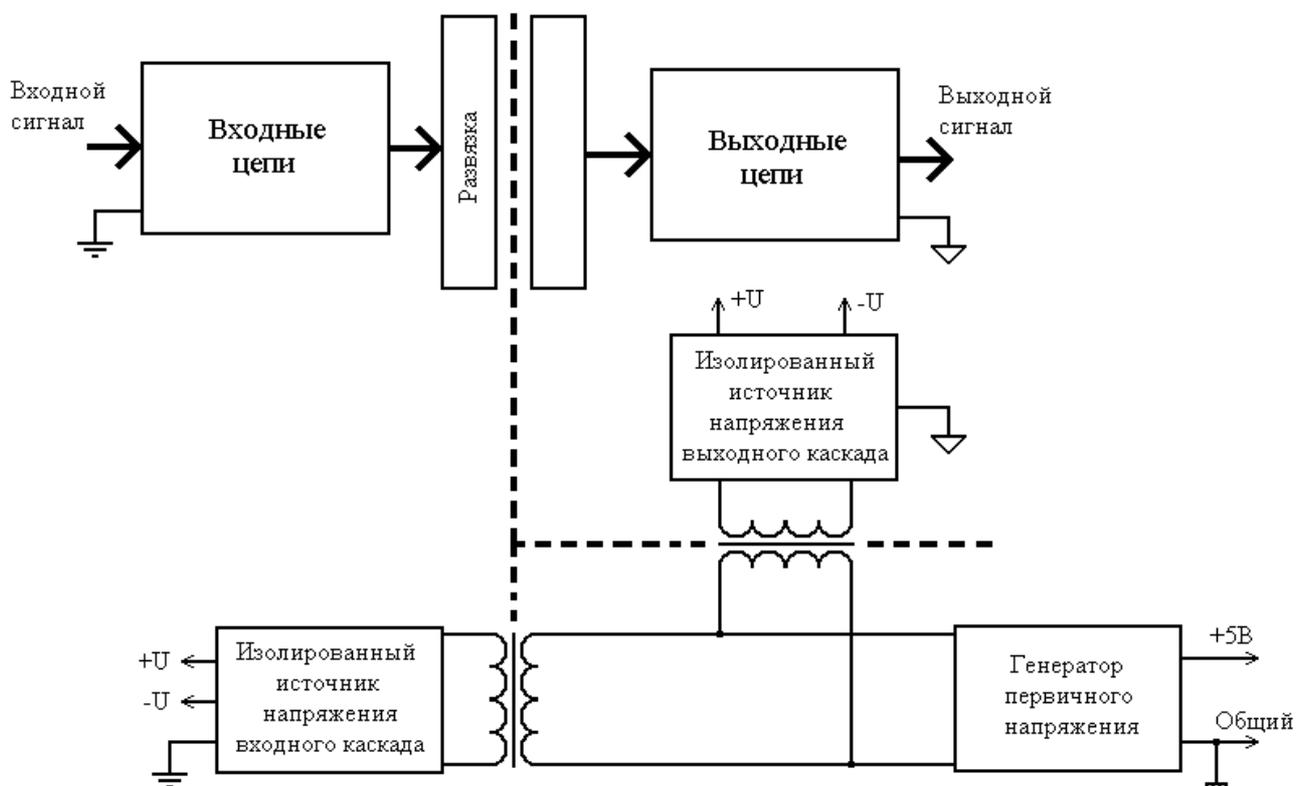


Рисунок 6 - Организация трехуровневой (трехсторонней) изоляции

Отличительной особенностью развязывающих усилителей является обеспечение отдельного питания входного и выходного каскадов, осуществляемого через разделительный трансформатор. При этом сигнал от входного к выходному каскаду может передаваться тремя способами: через трансформаторную, оптическую или ёмкостную связь. Метод передачи сигнала – это, как правило, модуляция/демодуляция (амплитудная, широтно-импульсная или частотная) и линеаризующая обратная связь. При ёмкостной связи, например, модулированный сигнал передается через ёмкость небольшого номинала (порядка единиц пикофарад) для обеспечения ограничения переменного синфазного напряжения. Нормализаторы выполняются в виде

конструктивно законченных типовых модулей, которые, как правило, устанавливаются в специализированные платы, имеющие клеммные соединители для подвода внешних цепей. Такие платы называют монтажными панелями или оптопанелями. Конструктивные особенности модулей и монтажных панелей дают возможность быстро производить диагностику и замену вышедших из строя модулей.

Дальнейшее рассмотрение подходов к построению нормализаторов, их параметров, конструктивов будет вестись на примере конкретных изделий фирм, специализирующихся на выпуске таких устройств.

Одним из таких производителей является американская компания Dataforth. Для использования в системах промышленной автоматизации она предлагает широкий набор изолированных модулей нормализаторов аналоговых сигналов серий SCM5B, SCM7B, DSCA и DSCT. Внешний вид нормализаторов серий SCM5B, SCM7B и монтажных панелей для них показан на рисунке 7.

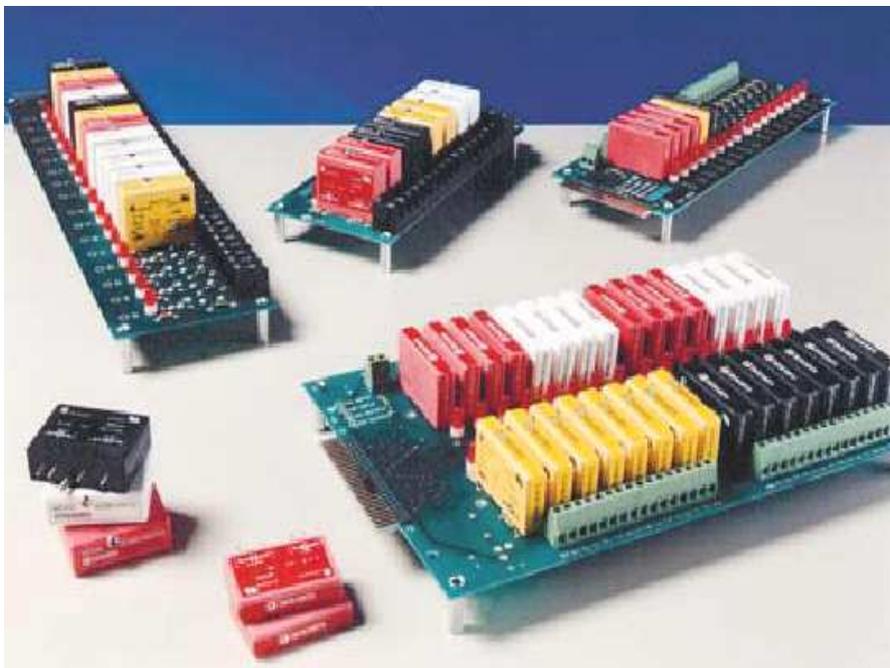


Рисунок 7 - Пример внешнего вида нормализаторов и монтажных панелей

Каждый такой модуль поддерживает отдельный канал изолированного аналогового ввода или вывода. Все модули Dataforth помещены в прочный пластиковый корпус. Форм-фактор этих модулей и функциональная эквивалентность позволяют использовать их вместо или совместно с аналогичными изделиями других производителей. Кроме самих модулей, данные серии включают адресуемые и неадресуемые, одиночные и двойные, 8- и 16-канальные установочные панели, со встроенными температурными датчиками компенсации холодного спая или без них, кабели, металлические монтажные каркасы, универсальные интерфейсные платы, модули для прямой коммутации входа/выхода на установочной панели, предохранители, прецизионные резисторы и т. д.

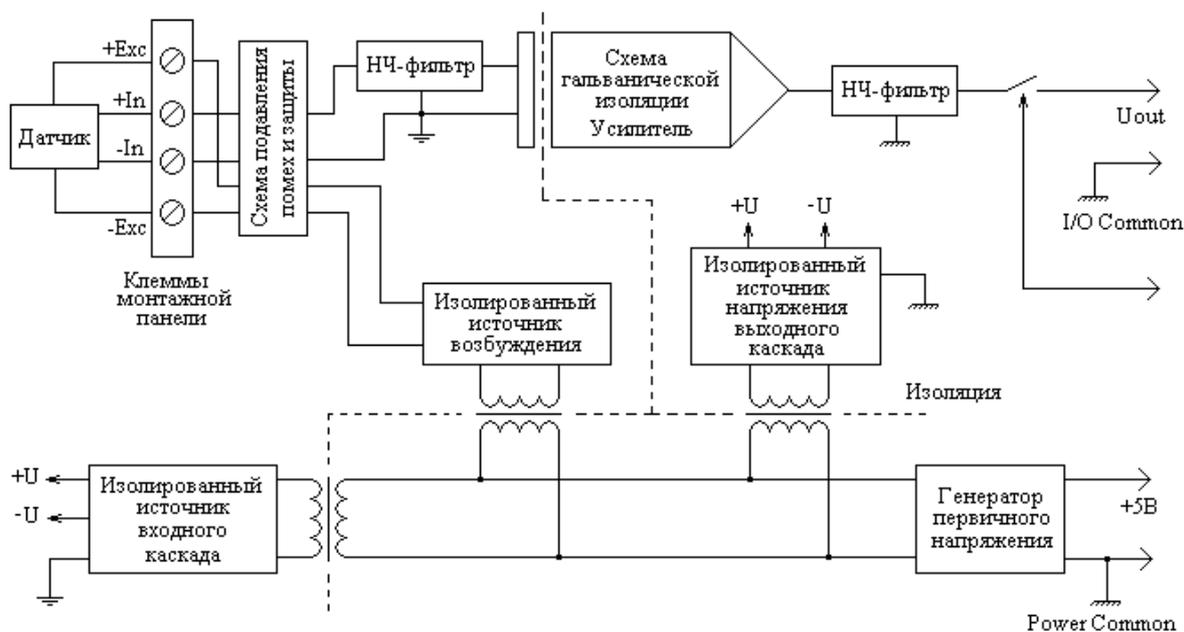


Рисунок 8 - Пример структуры нормализатора с трехуровневой изоляцией (SCM5B43 фирмы Dataforth)

Входные модули обеспечивают интерфейс со всеми типами внешних датчиков. Модули фильтруют, изолируют, усиливают и преобразуют входной сигнал к выходному аналоговому сигналу тока/напряжения с диапазонами изменения, принятыми в измерительной технике. Входные аналоговые сигналы

могут быть представлены напряжением или током с узкой или широкой полосой пропускания, сигналами от термопары, терморезистора, измерительного потенциометра, датчика деформации или частоты.

Модули вывода принимают аналоговый сигнал тока или напряжения от системы управления, буферизуют, изолируют, при необходимости усиливают и обеспечивают выходным управляющим током или напряжением исполнительные устройства.

Структурный состав модулей показан на примере нормализатора SCM5B43 на рисунке 8.

Учитывая высокие точностные показатели, трёхуровневую трансформаторную изоляцию, фильтрацию входного сигнала, диапазоны входного и выходного сигнала, диапазон питающего напряжения (номинал 24 В), систему крепления и малые габаритные размеры, можно определить основную сферу применения данных модулей – функциональные узлы нормализации сигналов в распределенных измерительных системах и системах управления предприятиями с высоким уровнем промышленных помех.

В части электромагнитной совместимости модули соответствуют европейским нормативным требованиям для применения в тяжёлых промышленных условиях. Среднее время безотказной работы, рассчитанное по результатам стресс-теста, составляет от 468000 до 740000 часов. Стопроцентный выходной контроль на производстве дополняется выдачей паспорта на каждое изделие, где указываются характеристики точности конкретного модуля.

Общие характеристики для модулей SCM5B, SCM7B, DSCA и DSCT приведены в таблице 1.

Таблица 1 - Общие характеристики модулей SCM5B, SCM7B, DSCA и DSCT

Гальваническая изоляция	до 1500 В
Типовая точность измерений	0,02-0,05%
Подавление помехи общего вида (CMR)	160 дБ
Подавление помехи нормального вида (NMR) на частоте 60 Гц	80 - 95 дБ
Диапазон рабочих температур	-40...+85°С

Помимо этого, модули данных серий имеют такие общие особенности, как защита от скачков напряжения, низкий уровень шума на выходе, высокая стабильность параметров в течение длительного времени, возможность монтажа установочных панелей с модулями на DIN-рельс.

В состав серий SCM5B, SCM7B, DSCA и DSCT входят следующие типы модулей:

- с потенциальным входом;
- с токовым входом (внешний резистор);
- с токовым входом (внутренний резистор);
- для подключения терморезисторов, линеаризованные, 2- или 3-проводные;
- для подключения терморезисторов, линеаризованные, 4-проводные;
- с потенциометрическим входом;
- для подключения линейных датчиков на основе дифференциального трансформатора;
- для подключения термопар;
- для подключения датчиков деформации;
- с токовым выходом;

- для управления сервоприводом;
- 2-проводные интерфейсные (питание от токовой петли);
- с частотным входом;
- для подключения термопар, линеаризованные;
- с потенциальным выходом.

Линеаризованные модули для подключения термопар используют десяти сегментную аппаратную линеаризацию.

Основное назначение изделий серии DSCL – гальваническая изоляция уже нормализованного выходного сигнала датчика от измерительного входа системы. Наличие такой изоляции предупреждает возможность возникновения проблемы «земляной петли». Отдельные изделия позволяют при этом преобразовать вид входного сигнала (ток-напряжение, напряжение-ток). Характер преобразования вида входного сигнала может определяться заводской установкой или назначаться пользователем посредством позиционирования переключателей.

В серии DSCL представлены как одноканальные, так и многоканальные устройства. Многоканальные модули могут быть использованы как разветвители одного входного сигнала на несколько гальванически развязанных выходов и благодаря этому применяться для построения резервированных систем контроля производственных процессов. Питание изоляторов-преобразователей осуществляется либо от источника входного сигнала (максимальный потребляемый ток 50 мА, максимальное входное напряжение 18/24 В), либо от внешнего источника питания 12...30 В или внешнего универсального источника 24...60 В постоянного тока, от сети переменного тока 185...230 В.

Для динамично перестраиваемых производств, а также тестового и измерительного оборудования, где требуется высокая гибкость, вызванная

частым изменением поддиапазонов измеряемых параметров, заменой датчиков, изменениями измерительного диапазона либо даже типа входного сигнала, фирма Dataforth выпускает преобразователи серий DSCP20, DSCP80, DSCP81, SCTP20.

Главной особенностью этих изделий является перепрограммируемость входных и выходных параметров преобразователей (рисунок 9). Модули DSCP20, DSCP80, SCTP20 реализуют интерфейс со всеми стандартными типами термопар и терморезисторов, модуль DSCP81 предназначен для приема входных сигналов тока и напряжения. Модули DSCP20, DSCP80, SCTP20 отслеживают состояние обрыва и короткого замыкания по входу. Модуль DSCP81 имеет релейный выход с большой нагрузочной способностью (до 2 А при напряжении 250 В переменного тока или 125 В постоянного тока); другой особенностью является то, что светодиод питания может извещать миганием о выходе за пределы измерительного диапазона входного сигнала.

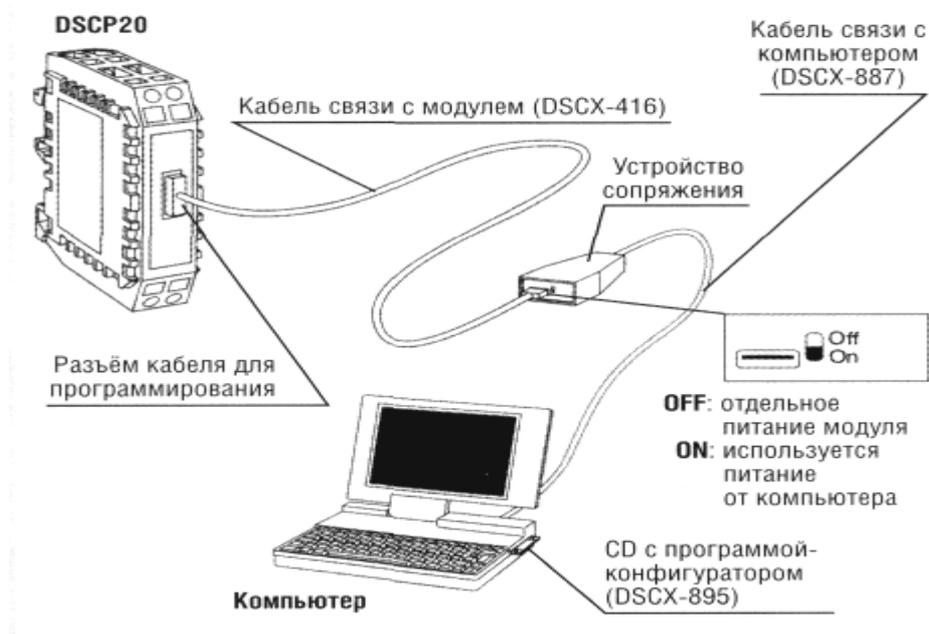


Рисунок 9 - Схема коммутации оборудования для программирования модуля DSCP20

Для реализации возможности программирования поставляются комплект программных продуктов DSCX-895, устанавливаемый на персональном компьютере, для преобразователей DSCP20, DSCP80, SCTP20 и программа-конфигуратор DSCX-557 для DSCP81. Программные продукты имеют удобный пользовательский интерфейс с графическими подсказками о последовательности подключения клемм при различных конфигурациях используемых датчиков, поддерживают опции по редактированию в графическом виде выходных характеристик преобразователей и обеспечивают возможность назначения пароля для блокирования несанкционированного изменения установленных значений. Для соединения преобразователей с компьютером необходимо приобрести два кабеля (рисунок 9). Один из них – для подключения к устройству сопряжения со стороны компьютера (DSCX-887 для изделий DSCP20, DSCP80, SCTP20 и DSCX787 для DSCP81), а второй – для подключения со стороны преобразователя (DSCX-416 для DSCP20 и DSCP80, DSCX-440 для SCTP20, DSCX-587 для DSCP81).

Известным производителем средств для построения систем автоматизации является компания Advantech. Хорошо известны выпускаемые этой компанией модули нормализации и гальванической развязки серии ADAM-3000.



Рисунок 10 - Внешний вид и подключение модулей ADAM-3000

Модули серии ADAM-3000 предназначены для нормализации аналоговых сигналов датчиков и гальванической изоляции каналов аналогового ввода/вывода информационно-измерительных систем и систем управления. Каждый модуль представляет собой функционально законченное устройство, заключенное в пластмассовый корпус и оснащенное клеммными соединителями с винтовой фиксацией для подключения входных и выходных цепей. Внешний вид модуля и способы его установки показаны на рисунке 10. Габаритные размеры модулей показаны на рисунке 11.

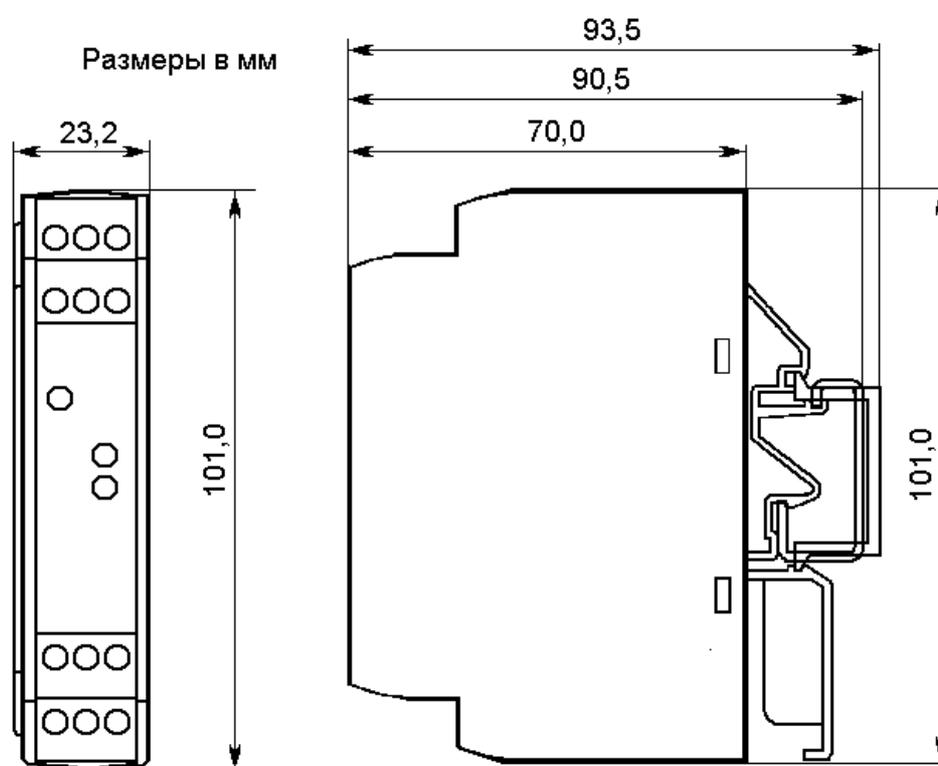


Рисунок 11 - Габаритные размеры модулей серии ADAM-3000

Модули серии ADAM-3000 имеют ряд отличий от изделий аналогичного назначения других производителей:

- для установки модулей не требуется специальных объединительных плат (установка производится на стандартный

несущий DIN-рельс 37,5 мм);

- тип и диапазон входного сигнала задаются при помощи миниатюрных переключателей, расположенных под монтажным кронштейном;

Модули имеют относительно невысокую стоимость. Питание модулей осуществляется напряжением 24 В постоянного тока. Диапазон рабочих температур от 0 до 70°C (кроме ADAM-3011).

В целом фирма Advantech выпускает более двух десятков различных типов модулей нормализаторов. Состав основных модулей серии ADAM-3000 продемонстрирован в таблице 2.

Таблица 2 -Состав серии ADAM-3000

ADAM 3011	Нормализатор сигналов термопар
ADAM 3012	Модуль гальванически изолированного аналогового
ADAM 3013	Нормализатор сигналов термометров сопротивления
ADAM 3014	Нормализатор аналоговых сигналов
ADAM 3016	Нормализатор сигналов тензодатчика
ADAM 3112	Нормализатор сигналов переменного напряжения
ADAM 3114	Нормализатор сигналов переменного тока
ADAM 3021	Модуль гальванически изолированного аналогового

1.6 Системное и прикладное обеспечение промышленных контроллеров

Программное обеспечение универсальных контроллеров состоит из двух частей. Первая часть это системное программное обеспечение. Проводя аналогию с программным обеспечением ЭВМ можно сказать, что оно выполняет функции операционной системы, т.е. управляет работой узлов контроллера, занимается организацией их взаимосвязи, внутренней диагностикой. Системное программное обеспечение ПЛК расположено в постоянной памяти в адресном пространстве центрального процессора и всегда готово к работе. По включению питания, ПЛК готов взять на себя управление системой уже через несколько миллисекунд.

Системное программное обеспечение (СПО) непосредственно контролирует аппаратные средства ПЛК. СПО отвечает за тестирование и индикацию работы памяти, источника питания, модулей ввода-вывода и интерфейсов, таймеров и часов реального времени. Система исполнения кода прикладной программы является составной частью СПО. Система исполнения включает драйверы модулей ввода-вывода, загрузчик кода программ пользователя, интерпретатор команд и отладочный монитор. Код СПО расположен в ПЗУ и может быть изменен только изготовителем ПЛК.

Другая часть программного обеспечения универсальных контроллеров это прикладные программы управления конкретным процессом. Эти программы создаются разработчиком системы управления.

Код прикладной программы размещается в энергонезависимой памяти, чаще всего это электрически перепрограммируемые микросхемы. Изменение кода прикладной программы выполняется пользователем ПЛК при помощи системы программирования и может быть выполнено многократно.

1.7 Рабочий цикл ПЛК

Задачи управления требуют непрерывного циклического контроля. В любых цифровых устройствах непрерывность достигается за счет применения дискретных алгоритмов, повторяющихся через достаточно малые промежутки времени. Таким образом, вычисления в ПЛК всегда повторяются циклически. Одна итерация, включающая замер, обсчет и выработку воздействия, называется рабочим циклом ПЛК. Выполняемые действия зависят от значения входов контроллера, предыдущего состояния и определяются пользовательской программой.

По включению питания ПЛК выполняет самотестирование и настройку аппаратных ресурсов, очистку оперативной памяти данных (ОЗУ), контроль целостности прикладной программы пользователя. Если прикладная программа

сохранена в памяти, ПЛК переходит к основной работе, которая состоит из постоянного повторения последовательности действий, входящих в рабочий цикл.

В общем виде рабочий цикл ПЛК можно представить состоящим из нескольких фаз.

1. Начало цикла.
2. Чтение состояния входов.
3. Выполнение кода программы пользователя.
4. Запись состояния выходов.
5. Обслуживание аппаратных ресурсов ПЛК.
6. Монитор системы исполнения.
7. Контроль времени цикла.
8. Переход на начало цикла.

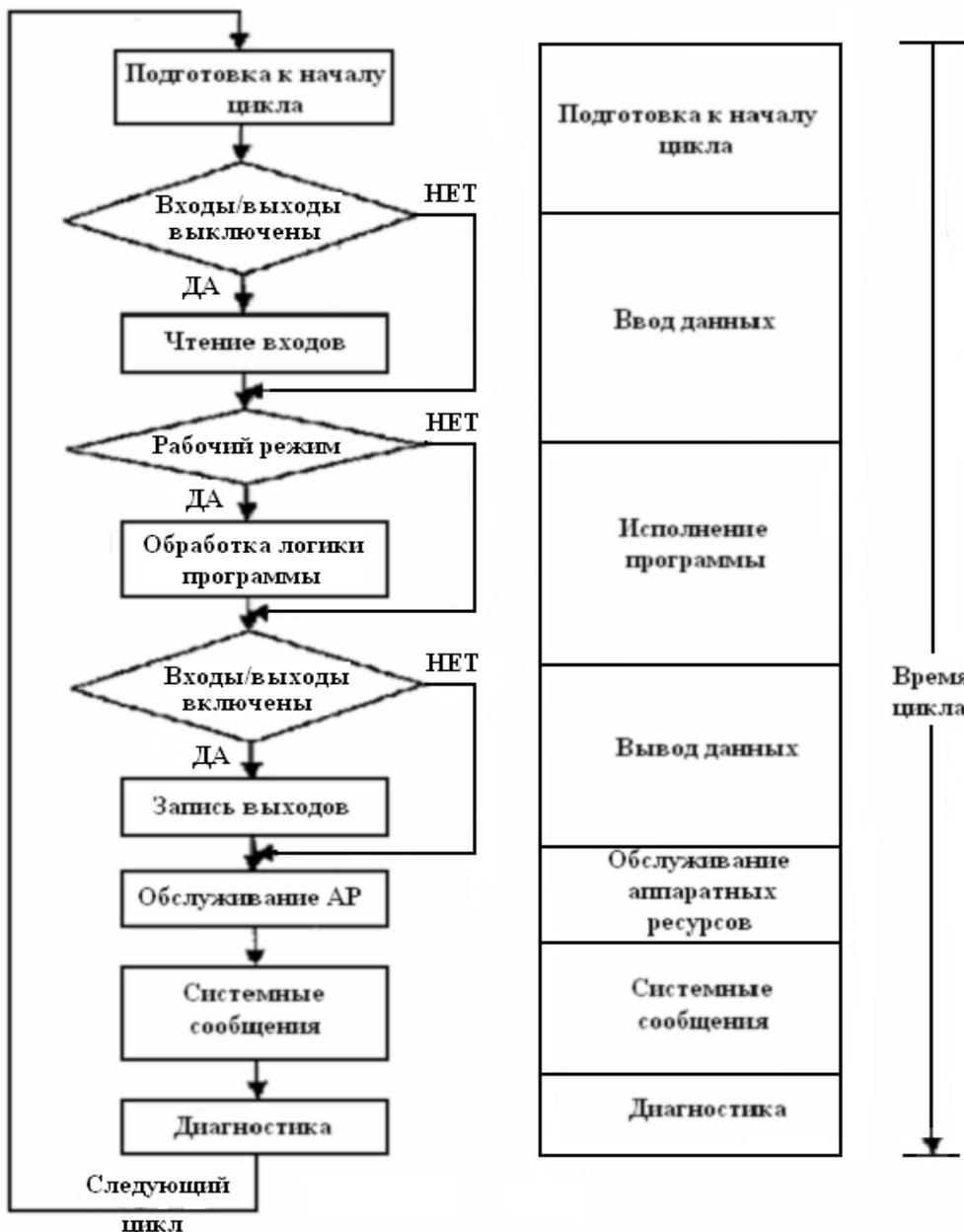


Рисунок 12 - Элементы цикла работы ПЛК

В самом начале цикла ПЛК производит физическое чтение входов. Считанные значения размещаются в области памяти входов. Таким образом, создается полная одномоментная зеркальная копия значений входов.

Далее выполняется код пользовательской программы. Пользовательская программа работает с копией значений входов и выходов, размещенных в оперативной памяти. Если прикладная программа не загружена или остановлена, то данная фаза рабочего цикла, естественно, не выполняется.

Отладчик системы программирования имеет доступ к образу входов-выходов, что позволяет управлять выходами вручную.

После выполнения пользовательского кода физические выходы ПЛК приводятся в соответствие с расчетными значениями (фаза 4).

Обслуживание аппаратных ресурсов подразумевает обеспечение работы системных таймеров, часов реального времени, оперативное самотестирование, индикацию состояния и другие аппаратно-зависимые задачи.

Монитор системы исполнения включает большое число функций, необходимых при отладке программы и обеспечении взаимодействия с системой программирования, сервером данных и сетью. В функции системы исполнения обычно включается: загрузка кода программы в оперативную и электрически перепрограммируемую память, управление последовательностью выполнения задач, отображение процесса выполнения программ. Пошаговое выполнение, обеспечение просмотра и редактирования значений переменных, фиксация и трассировка значений переменных, контроль времени цикла и т. д.

Пользовательская программа работает только с мгновенной копией входов. Таким образом, значения входов в процессе выполнения пользовательской программы не изменяется в пределах одного рабочего цикла. Это фундаментальный принцип построения ПЛК сканирующего типа. Такой подход исключает неоднозначность алгоритма обработки данных в различных его ветвях. Кроме того, чтение копии значения входа из ОЗУ выполняется значительно быстрее, чем прямое чтение входа. Самое существенно это то, что значения входов обновляются автоматически исключительно в начале каждого рабочего цикла.

1.8 Работа циклов на примере контроллера Twido фирмы шнейдер-электрик (Schneider Electric)

Циклическое выполнение.

При циклическом выполнении циклы контроллера последовательно соединяются вместе один за другим. После обновления выходов система выполняет свои определенные операции и затем присоединяет следующий цикл задачи.

Время сканирования пользовательской программы контролируется сторожевым таймером контроллера и не должно превышать 500 мс. В противном случае появляется ошибка, вызывающая немедленный переход контроллера в режим ОСТАНОВ. Выходы в этом режиме устанавливаются в аварийное состояние.

На следующем рисунке показаны фазы интервала сканирования при циклическом выполнении.

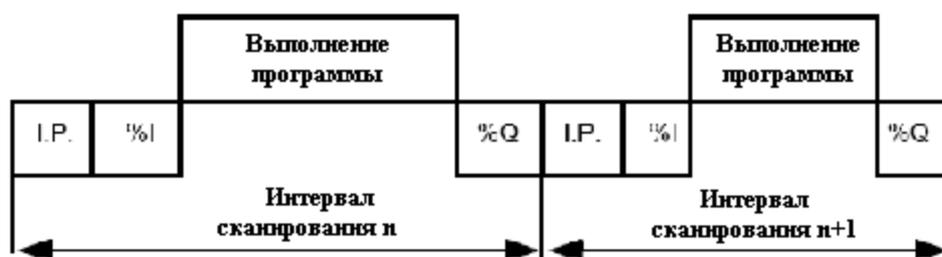


Рисунок 13 - Фазы интервала сканирования при циклическом выполнении

Описание фаз цикла

Адрес	Фаза	Описание
I.P.	Внутренняя обработка	Система полностью контролирует контроллер (управление системными битами и словами, обновление текущих значений таймера, обновление индикаторов статуса, определение изменения

		состояния РАБОТА/ОСТАНОВКА и т. п.). Обработка запросов от ПО.
%I, %IW	Чтение входов	Записывает в память состояние дискретных входов и входов определенных приложением модулей.
-	Обработка программы	Выполняет написанную пользователем программу приложения.
%Q,%QW	Обновление выходов	Записывает выходные биты или слова, связанные с дискретными и определяемыми приложением модулями.

Режимы работы контроллера

Контроллер в режиме РАБОТА, процессор выполняет:

1. Внутреннюю обработку
2. Чтение входов
3. Выполнение программы приложения
4. Обновление выходов

Контроллер в режиме ОСТАНОВКА, процессор выполняет:

1. Внутреннюю обработку
2. Чтение входов

На следующем рисунке показаны рабочие циклы.



Рисунок 14 - Работа циклов при циклическом выполнении

Проверка цикла: сканирование контролируется сторожевым таймером.

Периодическое выполнение

В этом режиме чтение входов, обработка программы приложения и обновление выходов выполняется периодически в соответствии с длительностью периода, определенного во время конфигурации (от 2 до 150 мс).

В начале сканирования таймер, чье текущее значение устанавливается равным длительности периода, определенного во время конфигурации, начинает считать на уменьшение. Сканирование должно закончиться прежде, чем таймер достигнет нулевого значения, которое в свою очередь запускает новое сканирование.

На следующем рисунке показаны фазы интервала сканирования при периодическом выполнении.

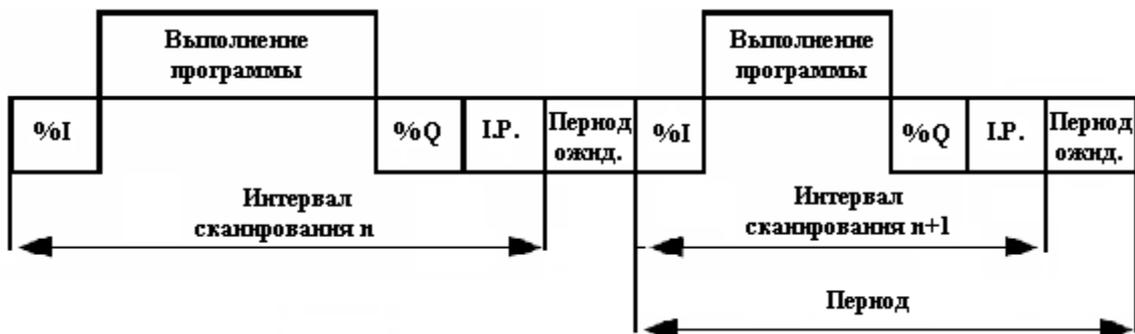


Рисунок 15 - Фазы интервала сканирования при периодическом выполнении.

Описание фаз цикла при периодическом выполнении аналогично описанию фаз цикла при циклическом выполнении.

Режимы работы контроллера

Контроллер в режиме РАБОТА, процессор выполняет:

1. Внутреннюю обработку
2. Чтение входов
3. Выполнение программы приложения
4. Обновление выходов

Если период еще не закончен, процессор выполняет свой рабочий цикл до конца *периода внутренней обработки*. Если время выполнения операций становится более длительным, чем выделено для периода, контроллер указывает на то, что период был превышен, установкой определенного системного бита %S19 в 1.

Системный бит %S19 показывает переполнение периода. Он устанавливается:

- в 1 системой, когда время сканирования больше периода задачи,
- в 0 пользователем.

Процесс продолжается до полного завершения. Он, однако, не должен превышать временной предел сторожевого таймера. Следующее сканирование начинается после полной записи выходов в текущем сканировании.

Контроллер в режиме ОСТАНОВКА, процессор выполняет:

1. Внутреннюю обработку

2. Чтение входов



Рисунок 16 - Работа циклов при периодическом выполнении

На рисунке 16 показаны рабочие циклы.

Проверка цикла: выполняются две проверки

- контроль переполнения периода;
- при помощи сторожевого таймера.

Правильно составленная пользовательская программа не должна содержать бесконечных циклов. В противном случае управление системе исполнения не будет передано, и, соответственно, нормальное функционирование контроллера будет нарушено.

Для преодоления данной проблемы служит контроль времени цикла. Контроль осуществляется при поддержке аппаратно реализованного «сторожевого таймера». Если фаза пользовательского кода выполняется дольше установленного порога, то ее работа будет прервана. Таким образом, достигается предсказуемое поведение ПЛК при ошибках в программе, или при «зависании» по причине аппаратных сбоев.

Обслуживание сторожевого таймера выполняется в рабочем цикле ПЛК. Выполнять эту операцию по прерыванию нельзя, поскольку при «зависании» процессора система прерываний достаточно часто продолжает исправно работать.

При периодическом или циклическом выполнении срабатывание сторожевого таймера приводит к программной ошибке. Приложение переходит в режим ОСТАНОВ и устанавливает системный бит %S11 в 1. Перед перезапуском задачи необходимо при помощи ПО проанализировать причины ошибки, исправить ошибку в приложении и вернуть программу в режим РАБОТА.

Состояние ОСТАНОВ – это состояние, при котором выполнение приложения немедленно прекращается из-за программной ошибки, такой как переполнение при сканировании. Текущие значения данных сохраняются, что позволяет проанализировать причину ошибки. Программа останавливается на текущей инструкции. Связь с контроллером возможна.

1.9 Время реакции

Время реакции – это время с момента изменения состояния системы до момента выработки соответствующей реакции. Очевидно, для ПЛК время реакции зависит от распределения моментов возникновения события и начала фазы чтения входов. Если изменение значений входов произошло непосредственно перед фазой чтения входов, то время реакции будет наименьшим и равным времени сканирования. Худший случай, когда изменение значений входов происходит сразу же после фазы чтения входов. Тогда время реакции будет наибольшим, равным удвоенному времени сканирования минус время одного чтения входов. Иными словами, время реакции ПЛК не превышает удвоенного времени сканирования.

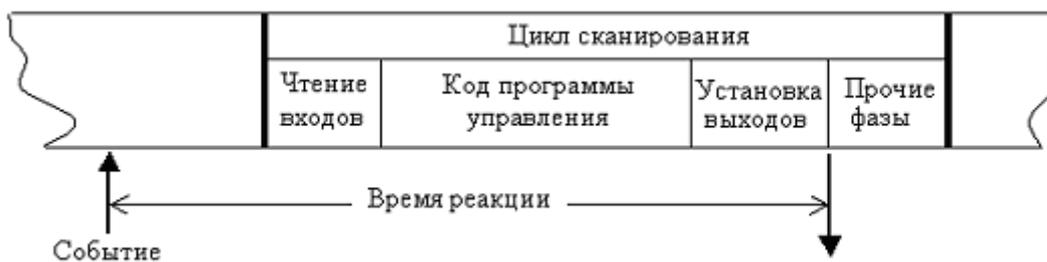


Рисунок 17 - Время реакции ПЛК.

Помимо времени реакции ПЛК, существенное значение имеет время реакции датчиков и исполнительных механизмов, которое также необходимо учитывать при оценке общего времени реакции системы.

Для уменьшения времени сканирующих контроллеров алгоритм программы разбивается на несколько задач с различным периодом исполнения. В наиболее развитых системах пользователь имеет возможность создавать отдельные программы, исполняемые по прерыванию, помимо кода, исполняемого в рабочем цикле. Такая техника позволяет ПЛК существенно форсировать ограничение реакции временем сканирования при небольшом количестве входов, требующих сверхскоростной реакции.

Время цикла сканирования является базовым показателем быстродействия ПЛК. Иногда изготовители приводят несколько значений времени цикла, полученных при работе с переменными различной разрядности.

Ориентировочно о скорости обработки различных типов данных можно судить по тактовой частоте и разрядности центрального процессора. Хотя нет ничего удивительного в том, что восьмиразрядные ПЛК не редко оказываются быстрее 32-разрядных при выполнении битовых операций. Объясняется это тем, что в 8-разрядных микропроцессорах более распространена аппаратная поддержка работы с битами. В PC-совместимых процессорах для выделения бита приходится использовать логические команды и циклический сдвиг.

1.10 Классификация контроллеров

Любой потенциальный заказчик микропроцессорных контроллеров стоит перед нелегким выбором: на нашем рынке контроллеры предлагают десятки отечественных и зарубежных производителей.

Контроллеры можно подразделить по ряду признаков на разные классы по определенным существенным для потребителей свойствам. Определение для каждого контроллера его классификационных особенностей фактически указывает его место среди прочих контроллеров, обозначает его отличия от них и выделяет группы контроллеров разных производителей, близких по большинству классифицируемых признаков.

Мощность. Термин «мощность» специалистами, работающими с контроллерами, введен как сленговый – он не отражает общепринятого понятия этого термина, принятого в целом в технике и физике. В данном случае под определением "мощность" понимается некоторая совокупность характеристик: разрядность и быстродействие центрального процессора, объем разных видов памяти, число портов и сетевых интерфейсов, т.е. это обобщающий термин,

показывающий «превосходство» одного контроллера над другим. Обычно основным показателем, косвенно характеризующим мощность контроллера, является число входов и выходов (как аналоговых, так и дискретных), которые могут быть подсоединены к контроллеру. По этому показателю контроллеры подразделяются на следующие классы:

- наноконтроллеры, имеющие до 15 входов/выходов;
- микроконтроллеры, рассчитанные на 15-128 входов/выходов;
- малые контроллеры, рассчитанные примерно от 100 до 300 входов/выходов;
- средние контроллеры, рассчитанные примерно на 300-2000 входов/выходов;
- большие контроллеры, имеющие примерно от 2000 и более входов/выходов.

Следует иметь в виду, что если контроллер модульный, то речь идет не о количестве встроенных в корпус контроллера входов/выходов, а о потенциальной возможности обслуживания определенного их числа, которое будет набрано путем подключения к центральному модулю необходимого количества периферийных блоков.

Требования автоматизируемого объекта всегда четко фиксируют необходимое число входных и выходных каналов контроллеров, что позволяет точно указать классы контроллеров по мощности, среди которых следует производить их отбор, чтобы не допустить излишних затрат и удовлетворить заданным требованиям.

Производители контроллеров, как правило, разрабатывают и выпускают их сериями. Контроллеры, входящие в одну серию, незначительно отличаются друг от друга отдельными параметрами и характеристиками – количеством входов/выходов, набором встроенных функций, они имеют общие

программное обеспечение и интерфейсы к другим средствам, единую взаимосвязывающую их сетевую структуру. Контроллеры, относящиеся к различным сериям, отличаются друг от друга мощностью существенно.

Эти сведения представляют немалый интерес для разработчиков, если системы автоматизации должны состоять из ряда контроллеров, отличающихся по параметрам, характеристикам, числу подключаемых входов/выходов. Естественно, что для их построения целесообразно и технически, и экономически отбирать контроллеры из одной серии, а каждый отдельный контроллер должен выбираться из условия минимизации избыточных встроенных средств.

Функциональное назначение. Классификация по функциональному назначению позволяет потенциальному заказчику связать требуемые функции контроля и управления с определенным классом контроллеров.

Специализированный контроллер с жестко встроенными функциями. Обычно им является минимальный по мощности контроллер, программа действия которого заранее прошита в его памяти, а изменению при эксплуатации подлежат только параметры программы. Число и набор блоков ввода/вывода определяется реализуемыми в нем функциями. Часто такие контроллеры реализуют различные варианты функций регулирования и их называют регуляторами. Основные области применения: управление каким-либо малым самостоятельным механизмом либо элемент общей системы управления, выделенный из-за специфических требований к отдельной функции. Так, например, из-за высокоскоростного изменения свойств какого-то узла объекта, требуется управлять им с очень малым циклом опроса, что легче и надежнее реализовать отдельным контроллером.

Контроллер, рассчитанный на реализацию логических зависимостей (в основном: на блокировку, программное управление, пуск, останов машин и механизмов). Набор блоков ввода/вывода у такого контроллера рассчитан, в основном, на разнообразные дискретные каналы. Он характеризуется прошитой

в его памяти развитой библиотекой логических функций и функций блокировки типовых исполнительных механизмов. Главная сфера применения такого контроллера – замена релейно-контактных шкафов во всех отраслях промышленности. Часто такие контроллеры называют программируемыми реле. Для его программирования используются специализированные языки типа релейно-контактных схем.

Контроллер, рассчитанный на реализацию любых вычислительных и логических функций. Наиболее распространенный универсальный контроллер, не имеющий ограничений по области применения. Центральный процессор контроллера имеет достаточную вычислительную мощность, разрядность, память, чтобы выполнять как логические, так и математические функции. Иногда для усиления его вычислительной мощности он снабжается еще и математическим сопроцессором. В программные инструментальные средства входят специализированные технологические графические языки и конфигураторы с библиотеками математических и регулирующих функций. В состав блоков ввода/вывода входят блоки на всевозможные виды и характеристики каналов (аналоговых, дискретных, импульсных и т. д.).

Контроллер, реализующий функции противоаварийной защиты. Он должен отличаться от контроллеров других классов:

- особенно высокой надежностью, достигаемой различными вариантами диагностики и резервирования (например, диагностикой работы отдельных компонентов контроллера в режиме on-line, наличием основного и резервного контроллеров с одинаковым аппаратурным и программным обеспечением и с модулем синхронизации работы контроллеров, резервированием блоков питания и коммуникационных шин);
- высокой готовностью, т. е. высокой вероятностью того, что объект находится в рабочем режиме (например, не только идентификацией, но и компенсацией неисправных элементов; не

просто резервированием, но и восстановлением ошибок программы без прерывания работы контроллеров);

- отказоустойчивостью, когда при любом отказе автоматизируемый процесс переводится в безопасный режим функционирования.

В этих контроллерах предусмотрены различные варианты полной диагностики и резервирования как отдельных компонентов, так и всего контроллера в целом. Можно отметить следующие распространенные варианты резервирования:

- горячий резерв всех компонентов и/или контроллера в целом (при непрохождении теста в рабочем контроллере управление безударно переходит ко второму контроллеру);

- тройное выполнение основных компонентов и/или контроллера в целом с "голосованием" результатов обработки сигналов всех контроллеров (выходной сигнал принимается тот, который дало большинство, а контроллер, давший другой результат, объявляется неисправным);

- работа по принципу "пара и резерв" – параллельно работает пара контроллеров с голосованием результатов, а аналогичная пара находится в горячем резерве. При выявлении разности результатов работы первой пары управление переходит ко второй; первая пара тестируется и либо выявляется наличие случайного сбоя, тогда управление возвращается к ней, либо выявляется неисправность и управление остается у второй.

Контроллер, предназначенный для телемеханических систем автоматизации. Данный класс универсальных контроллеров удобен для создания систем диспетчерского контроля и управления распределенными на

местности объектами. Он отличается от прочих классов контроллеров особой проработкой программных и технических компонентов передачи информации на большие расстояния беспроводными линиями связи. В качестве таких линий часто используются УКВ радиоканалы. При этом возможна передача информации от каждого контроллера в диспетчерский центр, а также эстафетная передача информации по цепи от одного контроллера к другому до достижения диспетчерского центра.

Монолитная или модульная архитектура. Этот классификационный признак делит контроллеры по возможностям пространственного расположения его отдельных компонентов. Существуют:

- контроллеры монолитной структуры, у которых состав всех компонентов, включая элементы ввода/вывода, конструктивно зафиксирован в одном корпусе;
- контроллеры модульной структуры, у которых часть или все блоки ввода/вывода могут быть конструктивно отделены от центральной части контроллера

Номенклатуру периферийных модулей составляют:

- модули аналогового ввода/вывода на разные типовые сигналы датчиков и исполнительных механизмов;
- модули дискретного ввода/вывода на разные токи и напряжения;
- аналого-цифровые и цифро-аналоговые преобразователи;
- модули сетевых интерфейсов, обслуживающие разные сети;
- специализированные модули (счетчики, таймеры, модули для управления двигателями и т. д.).

Подключая к центральному процессорному модулю необходимые периферийные модули, каждый проектировщик системы управления получит контроллер с заданными и минимально избыточными аппаратными средствами.

Форма конструктивного объединения центрального процессорного модуля и периферийных модулей может быть различной. В ряде случаев процессорный модуль конструктивно объединяется в одном корпусе с блоком питания и объединительной панелью, к которой и подключаются периферийные модули. Такой объединительный модуль всегда рассчитан на подключение определенного числа периферийных модулей (обычно от 3 - 4 до 8). В других случаях взаимное подключение модулей друг к другу осуществляется с помощью внешних шлейфов или специальных разъемов.

Архитектура некоторых контроллеров позволяет относить периферийные модули от процессорного модуля на значительные расстояния. Соединяются они между собой последовательной шиной или полевой сетью. Такой класс контроллеров может иметь ряд преимуществ перед другими при построении распределенных по производству систем управления. Он позволяет устанавливать блоки ввода/вывода непосредственно возле соответствующих датчиков и исполнительных механизмов, что технически целесообразно, а экономически эффективно.

Есть еще одно важное классификационное разделение блоков ввода/вывода и периферийных компонентов. Они подразделяются на:

- разработанные конкретно для данного контроллера (или данной серии контроллеров);
- стандартизированные, которые могут быть использованы в разных контроллерах.

Стандартизируются габариты периферийного модуля, его присоединительные разъемы, дисциплины обмена с процессорным блоком.

Закрытая или открытая архитектура. По архитектуре контроллеры подразделяются на два класса: контроллеры, имеющие фирменную закрытую

архитектуру, и контроллеры открытой архитектуры, основанной на одном из магистрально-модульных стандартов.

При фирменной архитектуре изменения (модификации) контроллера возможны только компонентами конкретного производителя. Сами изменения достаточно ограничены и заранее оговорены производителем.

При открытой магистрально-модульной структуре, имеющей стандартный интерфейс для связи центрального процессора с другими модулями контроллера, ситуация кардинально меняется:

- открытость и широкая доступность стандарта на шину, соединяющую модули разного назначения, дает возможность выпускать в данном стандарте любые модули разным производителям, а разработчикам контроллеров дает возможность компоновать свои средства из модулей разных фирм;
- возможность любой модификации и переконфигурации средств путем замены в них отдельных модулей, а не замены самих средств, удешевляет эксплуатацию средств;
- сборка контроллеров из готовых модулей позволяет точнее учитывать конкретные технические требования и не иметь в них лишних блоков и элементов, не нужных для данного конкретного применения;
- широкая кооперация разных фирм, поддерживающих данный стандарт на шину и работающих в этом стандарте, позволяет пользователям модулей не быть привязанным к конкретному поставщику и иметь широкий выбор необходимой ему продукции.

РС-подобие. Контроллеры подразделяются на РС-совместимые (PC-based) контроллеры и РС-несовместимые контроллеры. Основные отличия данных классов контроллеров состоят в следующем.

PC-совместимые контроллеры максимально открытые средства. Они имеют архитектуру IBM PC; базируются на той же, что и PC, компонентной базе; работают под одной из операционных систем персонального компьютера; взаимодействуют со всем наработанным для персональных компьютеров программным обеспечением; программируются на любых языках, используемых для программирования PC; в основном ориентированы на информационный обмен с другими средствами, как и PC, через сеть Ethernet.

PC-несовместимые контроллеры существенно более закрыты. Их архитектура большей частью является оригинальной разработкой производителя; их компонентная база отличается от PC и она разная у разных производителей; их специфические операционные системы реального времени точно отслеживают требования динамичных промышленных объектов; они почти не пользуются стандартами Windows в части программного обеспечения и не могут применять наработанные для PC программы; их программирование ведется на специальных технологических языках; в сетевых взаимосвязях они, в основном, ориентированы на стандартные или специальные промышленные и полевые сети.

Характер приведенных свойств рассматриваемых классов контроллеров определяет сравнительные достоинства и недостатки каждого класса.

PC-совместимые контроллеры по сравнению с PC-несовместимыми контроллерами в целом обладают большей вычислительной мощностью, легче стыкуются с различными SCADA-программами и СУБД, открыты для большинства стандартов в областях коммуникаций и программирования, они в среднем дешевле, проще обслуживаются и ремонтируются.

В то же время PC-совместимые контроллеры по сравнению с PC-несовместимыми контроллерами гораздо хуже учитывают требования промышленной автоматики, их операционные системы не полностью соответствуют ряду требований режима реального масштаба времени; они не имеют всех наработанных в промышленности способов диагностики и горячего

резервирования, обеспечивающих повышенную надежность работы контроллеров; в них недостаточно используются возможности связи с промышленными и полевыми сетями.

Приведенное выше краткое перечисление особенностей этих двух классов контроллеров обобщенно определяет рациональные ниши применения каждого из них.

PC-несовместимые контроллеры целесообразнее применять на нижних иерархических уровнях автоматизации отдельных агрегатов и механизмов. На этих уровнях наблюдаются особенно строгие специфически промышленные требования к средствам автоматизации, а необходимость открытости к стандартам программирования и к СУБД, требования больших вычислительных ресурсов почти или совсем отсутствуют;

PC-совместимые контроллеры целесообразнее применять на верхних иерархических уровнях автоматизации участков, цехов и производства в целом. Здесь слабеет строгость промышленной специфики вычислительных средств автоматизации и усиливаются требования к информационной совместимости контроллеров с корпоративными сетями и к использованию наработанного для PC программного обеспечения.

Сегодня мы можем смело утверждать, что архитектура PC распространяется все ниже и ниже в иерархии автоматизированных систем управления. Автомобильная промышленность США, которая контролирует 35% рынка контроллеров, уже приняла решение о переходе на IBM PC совместимые компьютеры и контроллеры. Более того, разработка и начало производства такими компаниями как Intel, AMD, C&T однокристалльных PC привели к тому, что сейчас архитектура PC вторгается на рынок недорогих контроллеров, где раньше безраздельно господствовали микроконтроллеры типа 8051.

Встраиваемые и автономные контроллеры. По конструктивному исполнению контроллеры подразделяются на два класса: встраиваемые и автономные.

Встраиваемые контроллеры выпускаются на раме без специального кожуха, поскольку они монтируются в общий корпус оборудования (агрегат, машину, прибор) и являются его неотъемлемой частью.

Если встраиваемые контроллеры выпускаются без специального кожуха, то автономные контроллеры помещаются в защитные корпуса, рассчитанные на разные условия окружающей среды.

Каркасы и корпуса контроллеров могут быть как собственной разработки производителя (что наблюдается все реже), так и стандартные (обычно стандарта "Евромеханика"). Конструктивное исполнение каркасов – любое (стойка, рама, башня, настольное исполнение, шкаф). Стандартные конструктивы имеют широкий диапазон вариантов исполнения на разные условия окружающей среды: температуру, пыль, влагу, вибрацию, электромагнитные помехи и пр.

Если автономные контроллеры выпускаются как локальные приборы, то в них встраивается или предусматривается возможность подключения к ним панели интерфейса с оператором, состоящей из дисплея и функциональной клавиатуры управления. Если они рассчитаны на работу в сетевом комплексе, состоящем из ряда контроллеров, то они не имеют встроенного пульта или панели, но есть физические порты, соединяющие их с другой аппаратурой, и сетевые интерфейсы, которые через сеть связывают их с другими средствами автоматизации, в частности, с рабочими станциями операторов.

Контроллеры, учитывающие и не учитывающие российскую специфику эксплуатации. Лишь несколько лет назад стало исчезать разделение контроллеров на отечественные и зарубежные. Считалось, что отечественные контроллеры существенно уступают по техническим характеристикам и по надежности, но зато они значительно дешевле

зарубежных контроллеров того же класса. На данный момент у конкурентных контроллеров отечественных производителей эти отличия становятся все менее заметными. Этому способствует то, что все они собираются из зарубежных электронных компонентов, используют близкое по свойствам системное и прикладное программное обеспечение, соответствуют одинаковым международным стандартам. Продукция отечественного производства оказывается вполне конкурентоспособной по сравнению с импортными аналогами. К сожалению, при этом ее стоимость также становится сопоставимой с зарубежными изделиями.

В настоящее время, когда говорят о контроллерах, учитывающих российскую специфику эксплуатации, обычно понимают:

- значительно более высокий, чем на зарубежных предприятиях, уровень промышленных помех;
- более широкий диапазон изменения параметров атмосферной и промышленной сред;
- возможность информационной связи с рядом морально устаревших, но еще находящихся в эксплуатации средств автоматизации, выпуска российских предприятий 80-х годов;
- недостаточную квалификацию обслуживающего персонала;
- низкую культуру оперативного персонала в части общения с вычислительными системами и дисплейными рабочими станциями.

Представляется, что эта специфика тоже имеет тенденцию к исчезновению. В частности, следует отметить, что зарубежные производители предлагают модификации контроллеров под разные виды и диапазоны помех и климатических характеристик (в том числе охватывающие все возможные условия работы контроллеров на российских предприятиях), а с точки зрения

вандалоустойчивости российские и зарубежные контроллеры практически не отличаются.

При недостаточной квалификации персонала любой специальности и при нарушениях технологической дисциплины на производстве следует уделять внимание исключению этих недостатков, а не приспособлению к ним средств управления.

2 Языки программирования промышленных контроллеров

Массовое применение программируемых промышленных контроллеров потребовало услуг высококвалифицированных профессионалов, способных решать вопросы программирования и внедрения этих контроллеров. В работах по внедрению этой техники в реальное производство участвуют программист, электронщик и технолог, досконально знающий автоматизируемый процесс. Однако, затраты времени и средств на эту работу, как правило, находятся в прямой зависимости от способностей программиста.

При создании АСУ ТП любой сложности всегда существовала тяжело решаемая проблема по взаимопониманию программистов и технологов. К сожалению, практика показывает, как трудно добиться от «главного специалиста по...» формализованного описания алгоритма. Пока успешнее всего работы по созданию АСУ ТП идут там, где программисты вникают во все тонкости конкретного технологического процесса. Зачастую при таком подходе программист остается единственным человеком, способным разобраться в своем творении, со всеми вытекающими отсюда последствиями. Абсурдность такой ситуации породила стремление создать некие технологические языки программирования, доступные инженерам и технологам и максимально упрощающими процесс программирования.

За последнее десятилетие появилось несколько технологических языков. Более того, Международной Электротехнической Комиссией

разработан стандарт IEC-1131 (МЭК -1131), концентрирующий все передовое в области языков программирования для систем автоматизации технологических процессов. Этот стандарт требует от различных изготовителей ПЛК предлагать команды, являющиеся одинаковыми и по внешнему виду, и по действию.

Стандарт специфицирует 5 языков программирования:

- Sequential Function Chart (SFC) – язык последовательных функциональных блоков;
- Function Block Diagram (FBD) – язык функциональных блоковых диаграмм;
- Ladder Diagrams (LAD) – язык релейных диаграмм;
- Statement List (STL) – язык структурированного текста;
- Instruction List (IL) – язык инструкций.

STL и IL являются неким программистским «эсперанто» – они вобрали в себя наиболее общие операторы языков типа Pascal и ассемблер. Они обеспечивают совместимость стандарта с ранними версиями программного обеспечения производителей контроллеров. Язык SFC позволяет осуществлять программирование на алгоритмическом уровне, но предполагает конечную реализацию программы на других языках. Язык LAD отдает дань поклонникам стиля, когда программы похожи на электрические схемы релейной логики. Язык FBD напоминает создание схем на логических элементах.

Наибольшее распространение в настоящее время получили языки LAD, STL и FBD. Охарактеризуем их более детально.

Язык релейных диаграмм (язык контактного плана) является графическим языком программирования. При создании программы в форме контактного плана используются графические компоненты, с помощью которых строятся логические сети, сходные с электрическими релейными схемами.

В LAD можно использовать следующие элементы и основные правила:

- контакты: каждый контакт представляет собой переключатель, через который при замкнутом состоянии может протекать ток.
- катушки: каждая катушка представляет собой реле, которое включается при протекании тока.
- блоки: каждый блок представляет собой функцию, которая выполняется, когда к блоку течет ток.
- сети: сеть образует полную цепь тока, который течет от левой шины тока через замкнутые контакты к катушкам или блокам – за счет этого они активизируются.

Графическое представление элементов программы воспринимается более легко, чем текстовое, поэтому этот язык и ему подобные максимально наглядны (особенно для тех, кто привык работать с электрическими схемами), позволяет исключить многие ошибки программирования свойственные «обычным» языкам. Этот язык можно рекомендовать для использования начинающими программистами.

Рассмотрим составление программы с помощью релейных диаграмм на нескольких небольших примерах.

Всем возможным типам переменным в контроллерах присвоены системные имена. Обозначения «I» и «Q» присвоены соответственно дискретным входам и выходам. Для элементов логической схемы, алгоритма предусматриваются специальные обозначения, напоминающие обозначения элементов электрических схем.

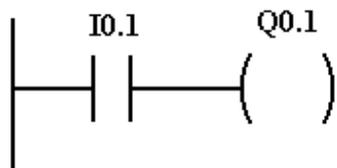


Рисунок 18 - Пример представления одной логической цепи в языке LAD

На рисунке 18 показаны, последовательно включенные, ключ (вход) I0.1 и катушка реле (выход) Q0.1. Вертикальная линия условно определяет собой цепь подачи питания. Выход Q0.1 активизируется тут же, как только появится сигнал на входе I0.1.

Выход Q0.1 в программе на рисунке 19 активизируется тогда, когда появится сигнал на входе I0.3 и сигналы хотя бы на одном из входов I0.1 или I0.2.

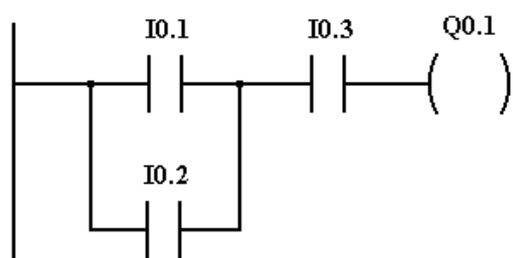


Рисунок 19 - Пример представления в языке LAD цепи реализующей логическую функции ИЛИ-И

На рисунке 20 представлена программа, включающая в себе блок таймера. Указанный в программе таймер, имеет дискретность отсчета 10 мс. Уставка таймера равна 300 и он сработает через 3000 мс после поступления сигнала на вход I0.1. При этом замкнутся управляющий контакт таймера T32 и

выход Q0.1 будет активизирован. Все действия программы выполняются по очереди – шаг за шагом, поэтому программа состоит из двух цепей.

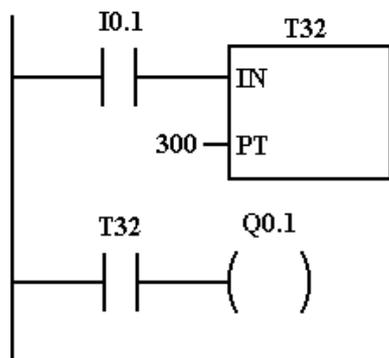


Рисунок 20 - Пример представления в языке LAD программы с таймером

Программа, представленная на рисунке 21, будет по очереди выполнять сложение, умножение и деление соответствующих переменных после поступления сигнала на вход I0.1 (используемые в примере типы переменных и их адресация будет рассмотрена далее).

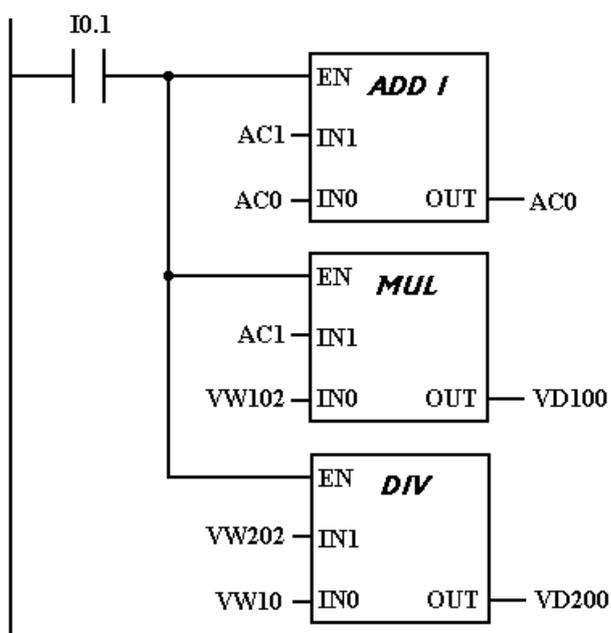


Рисунок 21 - Пример представления в языке LAD программы для выполнения арифметических операций

Язык функциональных блокковых диаграмм FBD (язык функционального плана) описывает функции между входными переменными и выходными переменными функциональных блоков (рисунок 22). Эти функции описываются в виде сочетания элементарных функциональных блоков. Каждый функциональный блок представляется прямоугольником, внутри которого имеется обозначение функции, выполняемой блоком.

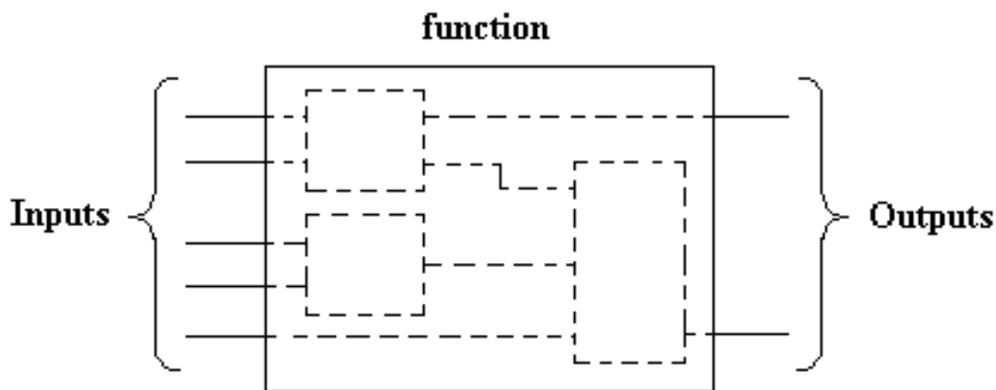


Рисунок 22 -Функциональный блок языка FBD.

Выход функционального блока может быть соединен с другими блоками. Логика программы выводится из соединений между этими блоками. Иными словами, выход одной команды (например, блока AND) может использоваться для того, чтобы разрешить другой команде (например, таймеру) создать необходимую логику управления. На рисунке 23 показан элемент программы, созданной при помощи языка функционального плана.

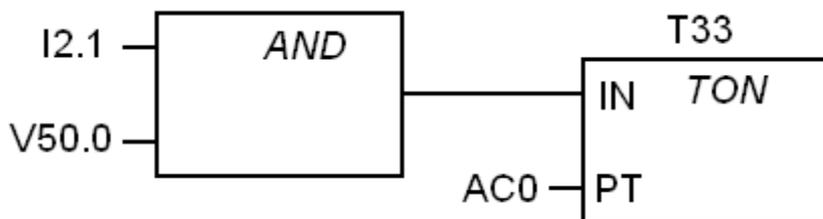


Рисунок 23 - Вид программы на языке FBD

В качестве библиотечных блоков используются не только элементарные функции, но и алгоритмы П-, ПИ-, ПИД-регулирования, фильтрация сигналов, стабилизация заданных параметров. Набор из математических и статистических функциональных блоков позволяет просто организовать необходимые вычисления и обработку сигналов.

Приведем основные формальные правила языка FBD:

- функциональные блоки могут располагаться в поле программы произвольно;
- не может быть свободных (несоединенных) входов и выходов функционального блока;
- любая связь (NET) может иметь имя переменной;
- входы и выходы функциональных блоков, присоединенные к связям, имеющим одинаковые имена, считаются соединенными;
- очередность выполнения блоков в программе: слева направо, сверху вниз.

Переменные FBD-программ присоединяются к входным/выходным точкам функциональных блоков.

На входе FBD-блока может быть

- константное выражение;
- любая внутренняя или входная переменная;
- выходная переменная.

На выходе FBD-блока может быть любая внутренняя или выходная переменная. В таблице 3 приведены примеры базовых функций языка FBD. Программы на языке FBD напоминают электрические принципиальные схемы логических устройств и формально соблюдают алгоритмы их работы. Однако,

несмотря на всю схожесть с электрическими схемами, язык FBD содержит метки, операторы условного и безусловного переходов, которые свойственны традиционным процедурным языкам программирования. Графический стиль представления программы в форме логических элементов удобно использовать для последующего составления ее блок-схемы.

Таблица 3-Примеры базовых функций языка FBD

Функции двоичного типа	NOT, AND, OR, XOR, SET, RESET
Функции управления программой	RETURN, GOTO, CALL, TSTART, TSTOP, GSTART, GSTOP
Арифметические функции	ADD, SUB, DIV, MUL
Функции сравнения	=, <>, >, <, =>, <=
Математические функции	ABS, EXPT, LOG, SQRT
Тригонометрические функции	ACOS, ASIN, ATAN, COS, SIN, TAN

Язык структурированного текста STL представляет собой язык программирования, в котором каждая команда программы содержит операцию, мнемоника которой определяет ту или иную функцию контроллера. Эти операции связываются в одну программу так, чтобы создать систему управления приложением. По способу представления программа, созданная в списке команд, очень похожа на обычный ассемблер. Например, программа, представленная на рисунке 28, в этой форме будет выглядеть следующим образом:

```
LD I0.1
I I0.2
O I0.3
= Q0.1
```

Программа, представленная на рисунке 26, в этой форме будет выглядеть так:

LD I0.1

+ AC1, AC2

= AC0

MUL AC1, VD102

= VD100

DEL VW202, VW10

= VD200

Пакеты по созданию программ для ПЛК, как правило, предоставляют возможность выбора языка представления. Например, Вы можете предпочесть создавать программы в среде, более ориентированной на графическое представление, тогда как кто-то другой может предпочесть стиль языка ассемблера, основанный на текстовом представлении. Более того, обычно предоставляется возможность переходить от одной к другой форме представления уже созданной программы, т.е. работать с ней в той форме представления, которая является более привычной именно для Вас.

2.1 Стандарт МЭК 61131

В 1979 году в рамках Международной электротехнической комиссии (МЭК) была создана специальная группа технических экспертов по проблемам ПЛК, включая аппаратные средства, монтаж, тестирование, документацию и связь.

Каждый производитель, создавая контроллер, создавал и язык программирования к нему, что вынуждало покупателей искать программистов под конкретную модель. Но со временем стала очевидной необходимость стандартизации контроллеров – от интерфейсов до языка программирования.

Первый вариант стандарта был опубликован в 1982 году. Ввиду сложности получившегося документа было решено разбить его на несколько частей. В настоящее время стандарт включает следующие части.

Таблица 4 – Области действия стандартов серии МЭК 61131

Номер стандарта	Наименование	Год издания	Требования	
			АПК	ПО
МЭК 61131-1	Общая информация	1992	+	+
МЭК 61131-2	Требования к оборудованию и испытаниям	1992	+	–
МЭК 61131-3	Языки программирования	1993	–	+
МЭК 61131-4	Руководства для пользователя	1994	+	+
МЭК 61131-5	Разработка сообщений	–	–	+
МЭК 61131-6	Соединения полевыми шинами	–	–	+
МЭК 61131-7	Язык программирования нечёткой логики	–	–	+
МЭК 61131-8	Директивы по применению и реализации языков программирования контроллеров	1999	–	+

Первоначально стандарт имел номер 1137, с 1997 года МЭК перешел на 5-цифровые обозначения. Теперь правильное наименование международной версии стандарта – МЭК 61131.

2.2 Структура стандарта МЭК 61131

Обзор отдельных глав стандарта

Первая часть этой серии стандартов – стандарт МЭК 61131-1 – посвящен общему описанию программируемого контроллера (ПЛК), включая терминологию по ПЛК, описание его структуры и функциональных характеристик – функций обработки сигналов, интерфейсов ввода/вывода, связи, человеко-машинного взаимодействия, программирования, надежности и эргономики.

Основным стандартом, относящимся к аппаратуре программируемых контроллеров, является МЭК 61131-2, который достаточно широко используется. В нем приведены:

1. требования к контроллерам по условиям их эксплуатации, транспортировки и хранения, включая температуру, влажность, давление, механические воздействия и пр.;

2. электрические требования, включая требования по аналоговому и дискретному вводу/выводу, питающим напряжениям, связным интерфейсам, процессорам, памяти, периферии, устойчивости к внешним помехам и излучаемым помехам, изоляции, самотестированию и диагностике;

3. требования безопасности включая защиту от электрических поражений, разряды через зазоры, огнестойкость, ограждения, термостойкость механических соединений, заземления, батареи и др.;

4. тестирование и испытания по проверке выполнения;

5. информация, которую должен прикладывать к продукции производитель контроллеров.

МЭК 61131-4 – это руководство пользователя, работающего по этим стандартам, в котором собраны полезные данные для всех этапов жизненного цикла контроллеров.

Важнейшим расширением стандарта МЭК 61131-3 является стандарт МЭК 61131-5, посвященный программированию связей между контроллерами и другими программируемыми системами. Этот стандарт опирается на стандарт спецификации производственного сообщения – Manufacturing Message

Specification (MMS, ISO/IEC 9506). В стандарте определены сервисы прикладного уровня, такие как сетевая проверка (сертификация) приборов, сбор данных (по вызову и программируемый), управление по сети (параметрическое и защитное), сообщения о тревогах, вызовы исполнения программ, передача прикладных программ, управление связью. Сервисы представлены на языках программирования, определенных в МЭК 61131-3. Даны диаграммы состояний и переходов для каждого сервиса.

Шестая часть стандарта сначала была зарезервирована для будущего использования, и недавно было принято решение об отведении её для описания полевых шин.

Седьмая часть серии стандартов – МЭК 61131-7 – посвящена языку программирования нечеткого контроля и управления. Здесь речь идет о введении стандартных описаний элементов нечеткого управления на языках программирования контроллеров, определенных в МЭК 61131-3 (главным образом, на языке FBD). Определены модель и функциональные элементы нечеткого управления, таких как размытие (fuzzification), обратное уплотнение (defuzzification), производящие правила по алгоритму последовательных операций соединения (aggregation), активации (activation) и накопления (accumulation). В результате в стандарте определен родовой язык нечеткого контроля и управления (Fuzzy Control Language – FCL). В стандарте приводятся многочисленные примеры FCL. Рассмотрены уровни соответствия систем контроля и управления, использующих FCL.

МЭК 61131-8 содержит руководящие принципы для разработчиков программного обеспечения для языков программирования, определенных в части 3.

Третья часть стандарта МЭК 61131

Основным стандартом, относящимся к программному обеспечению контроллеров, является стандарт по языкам программирования контроллеров – МЭК 61131-3. В этой части определены два текстовых языка программирования (IL – Instruction List и ST – Structured Text) и два графических языка (LD – Ladder Diagram и FBD – Function Block Diagram) и конкретизирован для контроллеров графический язык последовательных функциональных диаграмм (SFC – Sequential Function Chart), введенный ранее как стандарт МЭК 848.

Все эти языки программирования контроллеров взаимосвязаны. Для них всех стандарт определяет единые модели программного обеспечения, связанных функциональных блоков и собственно модель программирования. Стандартизованы общие элементы этих языков и, прежде всего, используемые символы, типы данных и переменные. Определены функции и функциональные блоки, их объявление, наборы стандартных функций и функциональных блоков, понятия программ на этих языках. Стандарт определяет и такие общие элементы, как конфигурации, ресурсы, пути доступа, задачи. Все это дает возможность проведения программирования на любом из этих языков или их совместного использования с обеспечением генерации кодов единой программы. И, наконец, в стандарте рассмотрена специфика каждого из этих языков.

В приложениях к стандарту даны формальные спецификации элементов текстовых языков, а в качестве информационного материала – множество примеров конкретных функций, блоков функций и программ, написанных на этих языках. Они вместе со стандартными функциями и функциональными блоками становятся основой для создания богатых библиотек стандартных элементов программ для систем контроля и управления.

Разработка языков стандарта

Стандарт МЭК 61131-3 определяет языки для программируемых контроллеров таким образом, что части прикладной программы могут быть запрограммированы на любом языке и скомпонованы в единую исполняемую программу. При разработке стандарта было найдено очень много вариаций языков для программируемых контроллеров, что было невозможно выбрать одну из них в качестве общего языка. Стандарт включает структурное программирование, абстрактные типы данных, выделение данных и процедур в блок в сочетании с сохранением тесной связи с классическими языками для программируемых контроллеров.

Стандарт МЭК 61131-3 описывает два графических языка: «Релейные диаграммы» (LD) и «Диаграммы функциональных блоков» (FBD). В этих языках графические символы обеспечивают прямое соответствие между графическим представлением решения задачи и программой, решающей эту задачу.

LD использует стандартизированный набор символов для ступенчатого программирования. По существу эти диаграммы являются представлением релейной логики. Люди, понимающие релейную логику, могут с легкостью программировать на языке релейных диаграмм.

FBD – это графический язык, который повсеместно используется в Европе. Элементы этого языка выглядят как блоки, соединенные проводами в электрическую цепь, делая язык удобным для множества прикладных программ, содержащих передачу информации или данных между различными компонентами. Функциональные блоки – это программные объекты, которые представляют специализированные функции управления, используемые в управляющих системах. В терминах объектно-ориентированного программирования функциональные блоки – это объекты, но не все объекты являются функциональными блоками.

В дополнение к графическим языкам стандарт МЭК 61131-3 определяет элементы языка «Последовательные функциональные схемы» (SFC) – «шаги», «переходы» и «блоки операций», которые могут быть использованы для организации «операций», написанных на любом языке, и для получения алгоритмов последовательного управления. Основой элементов SFC служит стандарт МЭК 848, который является международной версией известного французского стандарта «Grafcet».

В МЭК 61131-3 определяется также два стандартных текстовых языка: «Список инструкций» (IL) и «Структурированный текст» (ST). IL – это язык низкого уровня, в то время как ST – это язык высокого уровня, разработанный для структурного программирования. Язык ST предоставляет булевы и арифметические операторы, а также конструкции структурного программирования – развилки и циклы.

Одной из задач, появившейся у подразделения экспертов, было гармоничное использование функциональных блоков в среде четырех стандартных языков. Традиционным использованием функциональных блоков в языке LD было программное замещение простых электромеханических элементов, таких как счетчики и таймеры, в то время как их использование в FBD оказалось более сложным, напоминающим разводку кросс-плат в стационарных управляющих системах. Использование функциональных блоков в языке IL и LD примерно одинаково, в то время как не было никакого механизма их использования в традиционных структурированных языках, таких как Паскаль или Си.

В итоге МЭК 61131-3 содержит:

1. богатый набор стандартных функций:

- функции с битовыми строками (AND, OR, XOR, NOT, SHL, SHR, ROL, ROR);

- числовые функции (ADD, SUB, MUL, DIV, MOD, EXPT, MOVE, ABS, SQRT, LN, LOG, EXP, SIN, COS, TAN, ASIN, ACOS, ATAN);

- преобразования типов;
- функции выбора (GT, GE, EQ, LT, LE, NE);
- функции сравнения (LEN, LEFT, RIGHT, MID, CONCAT, INSERT, DELETE, REPLACE, FIND);
- функции, определяемые производителем и пользователем.

2. функциональные блоки:

- блоки синхронизации состояний (SR, RS, SEMA);
- дифференцирование переднего и заднего фронта (R.TRIG, F.TRIG);
- счетчики (CTU, CTD, CTUD);
- таймеры (TP, TON, TOF, RTC);
- функциональные блоки, определяемые пользователем и производителем.

С помощью этих функций и блоков выполняется множество операций над большим количеством стандартных типов данных:

- битовые строки (BOOL, TYPE, WORD, DWORD, LWORD);
- целые (SINT, INT, DINT, LINT);
- беззнаковые целые (USINT, DINT, IDINT, ULINT);
- вещественные (REAL, LREAL);
- временные (TIME, DATE, TIME OF DAY, DATE AND TIME);
- строки символов (STRING);
- типы данных, определяемые производителем и пользователем:
 - а) единичный элемент, подмножество, численное значение;
 - б) массив, структура.

Эксперты понимали, что не смогут определить все типы данных и операции для решения всех возможных проблем управления. Поэтому в МЭК 61131-3 детально описываются механизмы, посредством которых производители и пользователи могут определить новые типы данных, функции и функциональные блоки. Таким образом, данный стандарт является

«саморасширяющимся», и можно ожидать, что он будет в состоянии обслуживать много поколений новых технологий управления.

В МЭК 61131-3 детально описываются механизмы инкапсуляции данных и операций. Например, если пользователь хочет снова и снова применять одну и ту же последовательность функций управления, он может выделить ее в функциональный блок, поместить этот функциональный блок в «библиотеку», а затем установить копии этого функционального блока столько раз, сколько потребуется для управляющих программ. Причем, внутренняя реализация может быть выполнена на любом языке МЭК 61131-3 или даже на каком-либо другом стандартном языке (таком, как Си или Паскаль).

Инкапсуляция дает возможность упрятать детали реализации блока от его пользователя, которому необходимо иметь представление только о его применении. Такой подход ведет к созданию высококачественного, общедоступного и экономичного программного обеспечения для управляющих систем.

2.3 Комплексы программирования МЭК 61131-3

Контроллеры, программирование которых осуществляется со встроенного или выносного пульта, встречаются сегодня достаточно редко. Как правило, это простые специализированные ПЛК, предназначенные для управления освещением по расписанию, регулировки температуры и т. д. Все программирование таких контроллеров сводится обычно к заданию набора констант. Для программирования ПЛК универсального назначения применяются ПК. Процесс разработки и отладки программного обеспечения происходит при помощи специализированных комплексов программ, обеспечивающих комфортную среду для работы программиста.

Традиционно все ведущие изготовители программируемых ПЛК имеют собственные фирменные наработки в области инструментального

программного обеспечения. Большинство из них представляют удобные инструменты, оптимизированные под конкретную аппаратуру. Понятно, что в разработке универсальных систем программирования, приемлемых для своих ПЛК и для ПЛК конкурентов, изготовители не заинтересованы. Кроме того, это достаточно сложная задача. Системы программирования ПЛК небольших фирм в лучшем случае реализуют один из языков МЭК с некоторыми расширениями, призванные сохранить совместимость со своими же более ранними системами. Крупнейшие же лидеры рынка ПЛК предлагают сегодня очень мощные комплексы с поддержкой МЭК-языков.

Рынок средств и систем автоматизации наводнен промышленными контроллерами самых различных изготовителей. Что, казалось бы, общего между изделиями таких известных фирм, как АВВ, Allen Bradley, Honeywell, Omron, Moore Products, PEP Modular Computers и др. Контроллеры этих фирм роднит “язык общения”: ПО для них разрабатывается в соответствии со стандартом на языковые конструкции (языки программирования PLC). По разным оценкам, до 80% PLC-рынка обслуживается программными продуктами, реализующими в той или иной мере этот стандарт. В списке инструментальных программных систем, реализующих стандарт IEC 61131-3, свыше двух десятков наименований (таблица 5).

Таблица 5 - Примеры инструментальных систем, реализующих МЭК 61131-3.

МЭК-система	ФИРМА
CoDeSys	Smart Software Solution (www.3S-software.com), Германия
ACCON-ProSys	Deltalogic (www.deltalogic.de), Германия
OpenPCS	Infoteam Software (www.infoteam.de), Германия
PUMA	KEBA (www.keba.com), Австрия
SUCOsoftS340	Klokner-Moeller (www.moeller.net), Германия
NAIS CONTROL	Matsushita Electric Works AG, Германия

PDS7	Philips Industrial Electronics Services, Нидерланды
SELECONTROL	Selectron Lyss, Швейцария
4Control	Softing (www.softing.com), Германия
ISaGRAF	CJ International (www.isagraf.com), Франция

Открытость МЭК-стандарта – с одной стороны, и сложность реализации высококлассных комплексов программирования – с другой, привели к появлению специализированных фирм, занятых исключительно инструментами программирования ПЛК.

2.4 Выбор контроллеров

Выбор рационального для каждой конкретной задачи ПЛК является важным для любого заказчика. В то же время определение наилучшего ПЛК из всей их совокупности, имеющейся на рынке, является далеко не однозначной и не простой задачей, поскольку оно должно учитывать все свойства автоматизируемого объекта, удовлетворять поставленным требованиям к системе контроля и управления, находить некий рациональный компромисс между различными противоречивыми критериями (мощность, надежность, открытость, стоимость и т.д.).

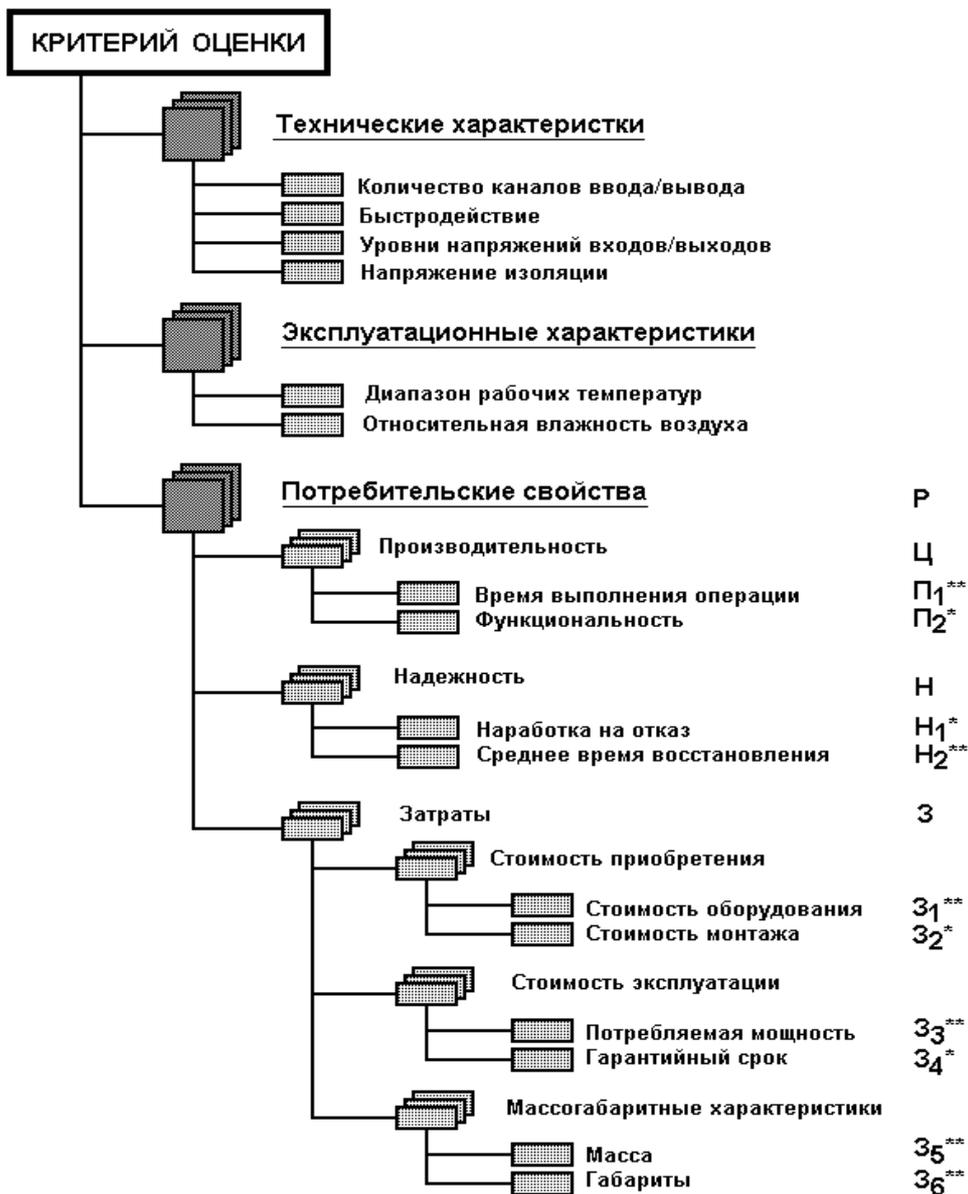


Рисунок 24 - Классификация критериев выбора ПЛК

Учитывая специфику устройств, критерии оценки можно разделить на три группы (рисунок 24):

- технические характеристики;
- эксплуатационные характеристики;
- потребительские свойства.

При этом критериями выбора считать потребительские свойства, т.е. соотношение показателей затраты/производительность/надежность, а технические и эксплуатационные характеристики ограничениями для процедуры выбора.

Кроме того, необходимо разделить характеристики на прямые – для которых положительным результатом является её увеличение (на рисунке обозначены – "*") и обратные – для которых положительным результатом является её уменьшение (на рисунке обозначены – "**").

Так как характеристики между собой конфликтны, т.е. улучшение одной характеристики почти всегда приводит к ухудшению другой, необходимо для каждой характеристики K_i определить весовой коэффициент a_i , учитывающий степень влияния данной характеристики на полезность устройства.

Терминология и состав критериев оценки ПЛК приведены в соответствии с основными положениями квалиметрии и стандартами качества (ГОСТ 15467-79). Выбор аппаратуры производится в четыре этапа:

- определение соответствия технических характеристик предъявленным требованиям;
- определение соответствия эксплуатационных характеристик предъявленным требованиям;
- оценка потребительских свойств выбираемой аппаратуры;
- ранжирование изделий.

На первом этапе каждая техническая характеристика анализируемого изделия сравнивается с предъявленными к проектируемой системе требованиями, и если данная характеристика не удовлетворяет этим требованиям, изделие снимается с рассмотрения.

Такой же анализ проводится на втором этапе с эксплуатационными характеристиками, и только если технические и эксплуатационные характеристики соответствуют поставленной задаче и предъявленным требованиям, проводится оценка потребительских свойств ПЛК.

Для этого используется аддитивный метод оценки, когда суммарная оценка каждого свойства вычисляется по формуле:

$$K = \sum_{i=1}^n \frac{K_i}{K_i^{\wedge}} \alpha_i + \sum_{j=1}^m \frac{K_j}{K_j^{\wedge}} \beta_j$$

где K_i, K_j – прямая и обратная характеристики выбираемого изделия;
 $K_i^{\wedge}, K_j^{\wedge}$ – соответствующие характеристики аналога; α_i, β_j – весовые коэффициенты характеристик;

n, m – количество прямых и обратных характеристик.

Даже если не пользоваться рассмотренной методикой для формирования количественных оценок критериев выбора, она полезна предложенной структуризацией и классификацией параметров ПЛК участвующих в их выборе.

2.5 Современный рынок контроллерных средств

Основные производители данной продукции: АВВ (распространяющая также контроллерные средства фирм Baily Controls и Gartman & Braun), Beckhoff, Foxboro, Groupe Schneider, Emerson, General Electric Fanuc Automation, Honeywell, Koyo Electronics, Tornado, Triconex, PEP, Trey, Control Microsystems, GF Power Controls Metso Automation, Moore Products, Omron, Rockwell Automation, Siemens, Yokogawa, VIPA, ICP DAS, Schneider Electric, Mitsubishi и др.

С зарубежными производителями в разных классах контроллерных средств конкурирует большое число российских предприятий: "Автоматика", ДЭП, "ВЕГА", "Волмаг", "ДЭП", "Завод электроники и механики", "ЗЭИМ Инжиниринг", "Интеравтоматика", "НВТ Автоматика", "ПИК ЗЕБРА", "РИУС", "Реалтайм", "Системотехника", "Трей", "Эмикон", "Импульс", "Инсист Автоматика", "Интеравтоматика", "Квантор", НИИтеплоприбор, "НВТ-Автоматика", ПИК "Прогресс", "Саргон", "Системотехника", ТЕКОН, "Электромеханика", ЭМИКОН и др.).

Контроллеры, выпускаемые различными фирмами, в подавляющем большинстве случаев весьма похожи друг на друга. Отчасти это, конечно, может быть связано и с тем, что каждый производитель старается перенять любое удачное решение фирмы-конкурента. Но главной и основной причиной такой схожести является понимание производителей, что работать они должны в рамках общепринятых стандартов и соглашений. Только при соблюдении этого условия их продукция будет максимально широко использоваться и будет конкурентоспособной. Стандарты и соглашения касаются самых разнообразных сторон этой области – речь идет и о языках программирования контроллеров, драйверов их сетевого обмена, электрической и конструктивной совместимости и т. д.

Для того чтобы сложилось достаточно полное представление обо всем классе этих устройств, достаточно детально познакомимся с отдельными его представителями.

2.6 Проектирование систем логического управления на базе промышленных контроллеров

Логическим управлением называют управление с помощью дискретных сигналов, принимающих конечное число фиксированных значений, и

осуществляемое в соответствии с заданным алгоритмом управления. Обычно используют двухуровневые сигналы, обозначаемые условно «0» и «1».

Логическое управление используется для решения следующих задач автоматизации:

- 1) пуск и остановка технологических объектов с помощью некоторой последовательности действий;
- 2) переход с одного режима на другой;
- 3) управление периодическими процессами, когда необходимо циклическое выполнение некоторой последовательности действий;
- 4) защита технологических объектов в аварийных и предаварийных ситуациях.

Автоматизация логического управления осуществляется с помощью аппаратных и программных средств, получивших название устройств или систем логического управления (УЛУ или СЛУ).

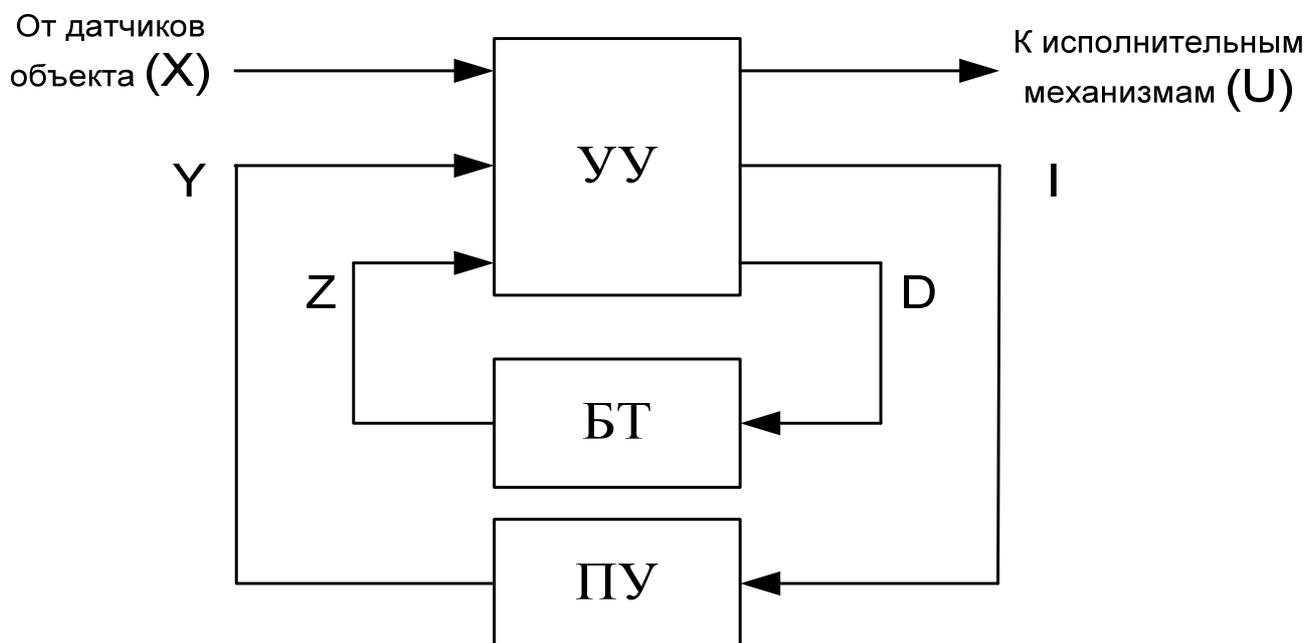


Рисунок 25 - Структурная схема устройства логического управления

На схеме приняты следующие обозначения для блоков и сигналов:

УУ - управляющее устройство или автомат;

БТ – блок таймеров, предназначенный для организации всех задержек времени, требуемых в алгоритме управления;

ПУ – пульт управления, предназначенный для взаимодействия с человеком-оператором.

X – множество сигналов от автоматических датчиков объекта;

U – множество управляющих сигналов, поступающих на двухпозиционные и трехпозиционные исполнительные механизмы объекта (например, клапаны, задвижки, двигатели и т.п.);

Y – множество сигналов от кнопок и ключей ручного управления, находящихся на пульте управления;

I – множество сигналов, поступающих на пульт управления с целью включения/выключения элементов световой и звуковой сигнализации;

D – множество сигналов, управляющих таймерами в блоке БТ, т.е. включающих таймеры для отсчета требуемых выдержек времени;

Z – множество сигналов из блока таймеров, свидетельствующих об окончании требуемых выдержек времени.

При построении устройств логического управления в блоке БТ и ПУ обычно используются стандартные элементы автоматики, выпускаемые промышленностью, такие как реле времени, таймеры, светодиодные индикаторы и т.п.

Наиболее сложным при проектировании является блок управляющего устройства, т.к. его схема или программа работы существенно зависит от требований конкретного объекта и алгоритма управления.

2.7 Основные этапы проектирования управляющего устройства

Процесс проектирования управляющего устройства включает следующие этапы:

- 1) системное проектирование;
- 2) логический синтез;
- 3) техническое проектирование и реализация (при создании УУ на жесткой логике) или программирование контроллера (при создании УУ на гибкой логике).

На этапе системного проектирования получают все необходимые сведения об объекте управления и решают следующие задачи:

1. Постановка и обоснование задачи автоматизации логического управления заданным объектом. Словесная формулировка алгоритма управления объектом.
2. Выбор необходимых датчиков и исполнительных механизмов.
3. Введение всех необходимых входных и выходных сигналов, их обозначение и интерпретация:

$$X_1 = \begin{cases} 0, & \text{если } P_1 < P_{дон} \\ 1, & \text{если } P_1 \geq P_{дон} \end{cases} \text{ - это полная запись}$$

$$X_2 = \begin{cases} 0, & \text{если } H < H_{вы} \\ 1, & \text{если } H \geq H_{вы} \end{cases} \text{ - это полная запись}$$

Более короткая запись:

$$X_1 - \text{давление } P_1 \geq P_{дон}$$

$$X_2 - \text{уровень } H \geq H_{вы}$$

Читается как превышение давления в первом случае или превышение уровня во втором случае.

$$\text{Управляющие сигналы: } U_1 = \begin{cases} 0, & \text{выключить двигатель} \\ 1, & \text{включить двигатель} \end{cases} \text{ - полная запись}$$

Упрощенная запись:

$$U_1 - \text{управление двигателем:}$$

$$U_1^0 - \text{выключить двигатель;}$$

U_1^1 – включить двигатель.

Также следует учитывать особенности датчиков и исполнительных механизмов конкретного объекта.

Например, для задвижки возможен вариант «внешнего управления», когда конечные положения задвижки контролируются системой управления и вариант «внутреннего управления», когда достижение конечного положения контролируются самой задвижкой.

В первом случае нам необходимо ввести 2 входных сигнала (X_{30} – задвижка полностью открылась и X_{33} – задвижка полностью закрылась) и 2 управляющих сигнала (U_{30} – задвижку открыть и U_{33} – задвижку закрыть). При этом при открытии/закрытии задвижек УУ должно контролировать появление сигналов о конечном положении задвижек.

Во втором случае достаточно одного управляющего сигнала (U_3), единичное значение которого будет соответствовать открытому положению задвижки, а нулевое – закрытому. При этом конечные положения обрабатываются самой задвижкой.

Далее осуществляется формализованное представление алгоритма управления. Это выполняется с целью возможного уточнения алгоритма управления и упрощения последующего проектирования управляющего устройства. Формализованные способы представления алгоритмов должны удовлетворять следующим требованиям:

- 1) однозначности, т.е. каждый шаг алгоритма должен быть однозначно определен;
- 2) полнота, т.е. необходимость учета всех возможных ситуаций на объекте;
- 3) не противоречивость, т.е. в одной и той же ситуации в алгоритме управления должны формироваться одни и те же управляющие воздействия.

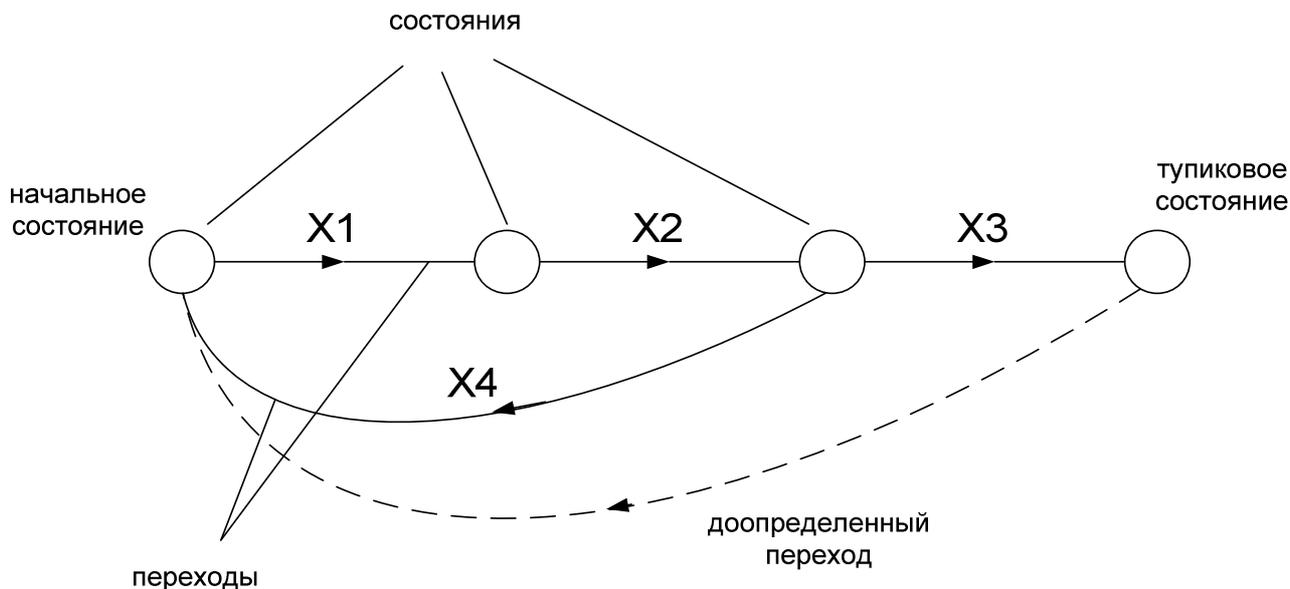


Рисунок 26 - Основные компоненты графа переходов

Существуют разные способы формализации алгоритмов управления, например, автоматные таблицы, блок схемы алгоритмов и другие. Одним из эффективных способов формализации является графы переходов или состояний.

В общем виде граф переходов или состояний представляет собой следующее изображение (рисунок 26).

Основным понятием графа переходов является понятие состояние. Состояния рассматриваются как некоторые абстракции, вводимые в процессе алгоритмизации, путем однозначного сопоставления каждого из них с одним из физических состояний управляемого объекта, так как обычно функционирование САУ проявляется через изменение их состояний. При этом каждое состояние в алгоритме поддерживает объект в соответствующем состоянии, а переход в новое состояние в алгоритме приводит к переходу объекта в новое соответствующее состояние, что обеспечивает процесс логического управления объектом.

Например, объект "задвижка" может находиться в одном из четырех рабочих состояний ("закрота", "открывается", "открота", "закрывается"),

каждое из которых может поддерживаться соответствующим состоянием в алгоритме управления (рисунок 27 и 28).

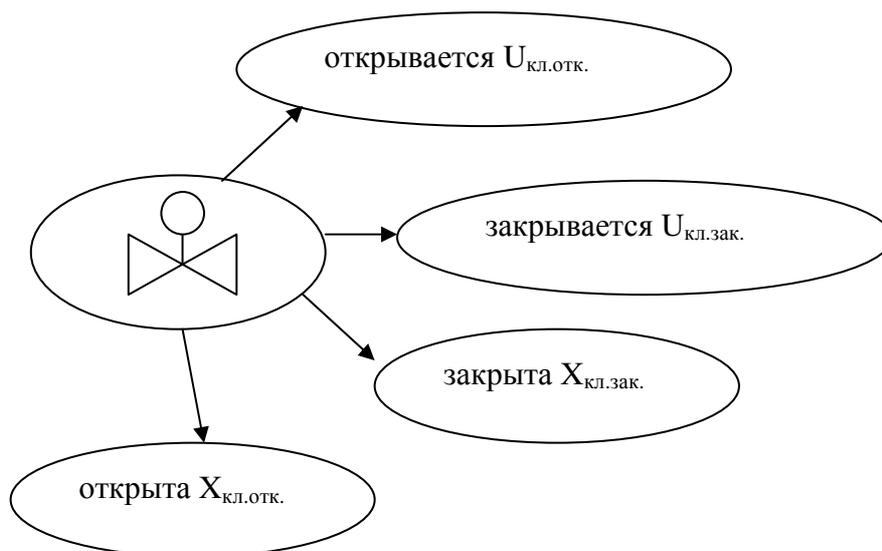


Рисунок 27 - Возможные состояния задвижки

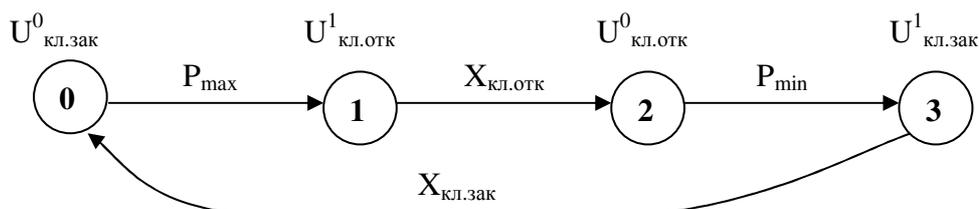


Рисунок 28 - Граф переходов переключения задвижки

В каждый момент времени граф переходов может находиться только в одном состоянии.

Другим понятием графа является переход. Над переходом указывается условие перехода, состоящее из обозначений входных сигналов УЛУ. На каждом переходе указываются не все входные воздействия УЛУ, а только те из них, которые обеспечивают переход по этой дуге. На каждом переходе входные воздействия могут объединяться в булевы формулы. Появление входного

воздействия, записанного на переходе, переводит граф переходов в следующее состояние.

После составления графа переходов для решаемой задачи, по нему формальными методами либо строится функциональная схема из библиотечных элементов с помощью системы булевых формул, либо составляется программа.

Описываемая технология позволяет участникам разработки проекта (заказчик, технолог, разработчик и программист) общаться не традиционным путем, в терминах технологического процесса, а на полностью формализованном языке, что не вызывает разночтений, возникающих из-за неоднозначности понимания естественного языка.

Граф переходов составляется на основании словесной формулировки алгоритмов управления, а также введения обозначения сигналов в следующей последовательности:

1. Определяется начальное состояние объекта, из которого начинается процесс управления, и для него вводится начальное состояние на графе переходов, обозначаемое кружком "О". Для этого состояния определяются значения всех выходных сигналов УЛУ, а также соответствующие им состояния ИМ объекта и элементов на пульте управления (ПУ). Все начальные состояния выходных сигналов приписываются в начальном состоянии графа переходов с помощью введенных ранее обозначений.

2. Определяется очередность состояний объекта и необходимое для этого изменение состояний ИМ. Каждый переход из начального состояния в последующие изображается направленной линией, связывающей изображение этих состояний. Стрелка, указывающая направление перехода,

обычно изображается в средней части линии. Над стрелкой указывается логическое условие, при выполнении которого осуществляется данный переход, т.е. переход разрешается, когда логическое условие перехода принимает единичное значение. Если из данного состояния возможны несколько переходов, то все они изображаются направленными переходами с соответствующими условиями переходов, при этом все условия должны быть взаимоисключающими, т.е. не должно выполняться более одного условия в данный момент времени (иначе это будет противоречить требованию однозначности алгоритма).

3. Чтобы избежать избыточного числа состояний на графе переходов, каждое новое состояние вводится только тогда, когда аналогичного состояния на графе не вводилось. Аналогичными или идентичными состояниями графа переходов называют состояния, в которых формируются одинаковые выходные сигналы для объекта и ПУ.

4. Построение графа переходов продолжается до тех пор, пока все последовательности состояний не образуют замкнутые циклы или подграфы. Наличие или появление тупиковых состояний, из которых нет переходов в другие состояния, свидетельствует, как правило, либо об ошибках построения графа переходов, либо о неполноте или ошибочности исходных данных, приведенных в словесной формулировке алгоритма. В этом случае необходимо доопределить и замкнуть граф соответствующим переходом.

2.8 Логический синтез управляющих логических устройств с позиционной структурой

УЛУ с позиционной структурой состоит из следующих элементов (рисунок 29):

- БЛУ – блок логических условий переключения элементов памяти в БП, формирующий включающие S и выключающие R сигналы;
- БП – блок памяти, содержащий RS-триггеры, формирующие на своих прямых выходах сигналы "а";
- БВ – блок выходов, формирующий выходные сигналы управляющего устройства и поступающие на объект U, блок таймеров D и пульта управления I.

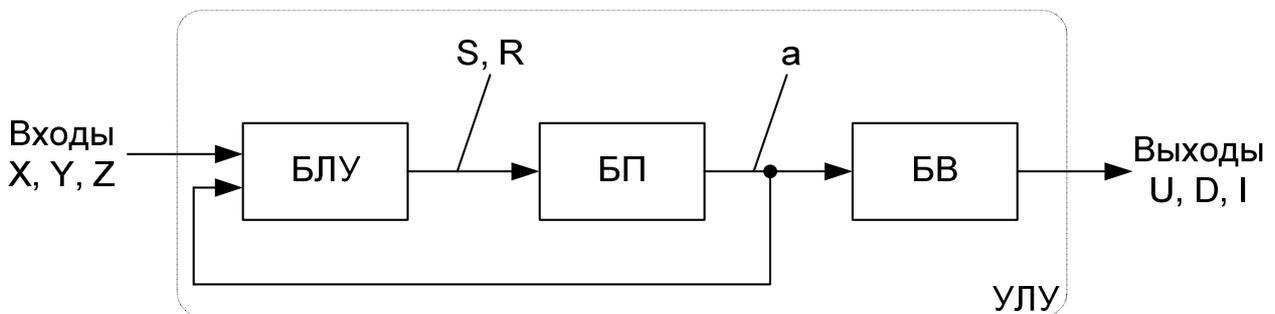


Рисунок 29 - Структурная схема УЛУ с позиционной структурой

Синтез выполняется на основе составленного графа переходов в следующей последовательности:

1. Синтез блока БП. Количество элементов памяти БП выбирается равным числу состояний на графе переходов

$$N_{ЭП} = N,$$

где N – число состояний на графе.

2. Синтез блока БЛУ. Под синтезом данного блока понимается составление структурных формул для включающих S_i и выключающих R_i сигналов каждого элемента памяти в БП. При этом под включающим сигналом понимается дискретный сигнал, приводящий триггеры в единичное состояние

на прямом выходе. Каждый включающий сигнал S_i формируется при одновременном выполнении следующих условий:

- БП находится в правильном предшествующем состоянии, чему соответствует включённое состояние только одного из триггеров БП;
- выключены все триггеры возможных предшествующих состояний данного состояния;
- выполнены условия перехода в последующие состояния.

Все три условия объединяются логической операцией “И”:

$$S_i = \text{условие1} \& \text{условие2} \& \text{условие3}.$$

Если включающих сигналов несколько, то все они объединяются логической операцией “ИЛИ”.

Выключающий сигнал R_i формируется при следующем условии: при переходе БП в каждое последующее состояние все триггеры предшествующих состояний выключаются. Если выключающих сигналов несколько, то все они объединяются логической операцией “ИЛИ”.

3. Синтез блока выходов.

БВ может быть построен двумя способами:

- на базе элементов “ИЛИ”;
- на базе RS-триггеров.

В первом случае на графе переходов находят все состояния, в которых данный выходной сигнал принимает единичные значения. Тогда структурная формула для этого сигнала принимает вид

$$U_i = a_{i1} + a_{i2} + \dots + a_{in}.$$

Во втором случае на графе переходов находят состояния, в которых данный выходной сигнал совершает переход из 0 в 1. Выходами элементов памяти этих состояний формируют включающий сигнал S_i . Аналогично записывают выключающий сигнал при переходе 1 в 0. Если таких состояний несколько, то все они объединяются логической операцией «ИЛИ».

4. Составление функциональной схемы УЛУ или программы в виде совокупности всех трёх блоков.

Рассмотрим конкретный пример – сравнительно простую и вместе с тем достаточно типичную схему смесительной установки (рисунок 35). Алгоритм работы установки очень прост. После её запуска в работу открывается клапан Y1, и ёмкость начинает заполняться компонентом № 1. При достижении уровня 2 срабатывает датчик SL2, закрывается клапан Y1 и открывается клапан Y2. В ёмкость начинает поступать компонент № 2. После заполнения ёмкости до уровня 1 по сигналу от датчика SL1 закрывается клапан Y2 и включается привод мешалки M. Через 15 минут он выключается, смесь готова. Для её выгрузки открывается клапан Y3. Окончание процесса фиксируется датчиком SL3 (уровень 3). После закрытия клапана Y3 установка готова к новому циклу приготовления смеси.

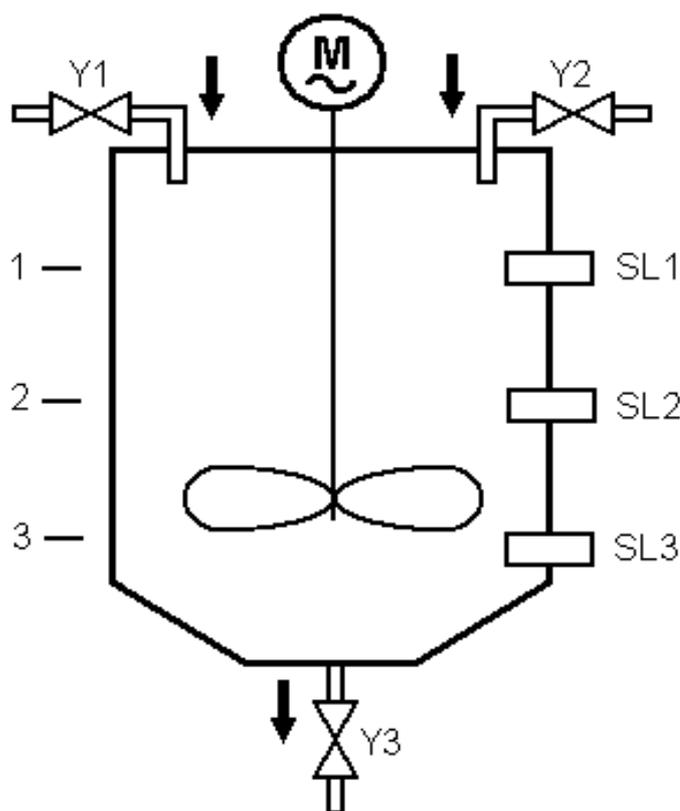


Рисунок 30 - Схема смесительной установки

Как поступил бы разработчик системы управления, скажем, лет 10-15 назад? Он использовал бы схему, подобную той, что приведена на рисунке 31.

Все просто, но усложним немного задачу и добавим условие возможности приготовления на этой же установке смеси по другой рецептуре. Это означает, что в ёмкость должно загружаться иное число компонентов с индивидуальными для каждого из них дозой и порядком загрузки, а также изменено время работы мешалки. Нет необходимости говорить о том, что такое условие уже не может быть реализовано с помощью релейной схемы. Но такая задача по плечу контроллеру, в котором алгоритм управления реализуется программно, а аппаратных средств имеется ровно столько, сколько необходимо для решения подобных по сложности задач.

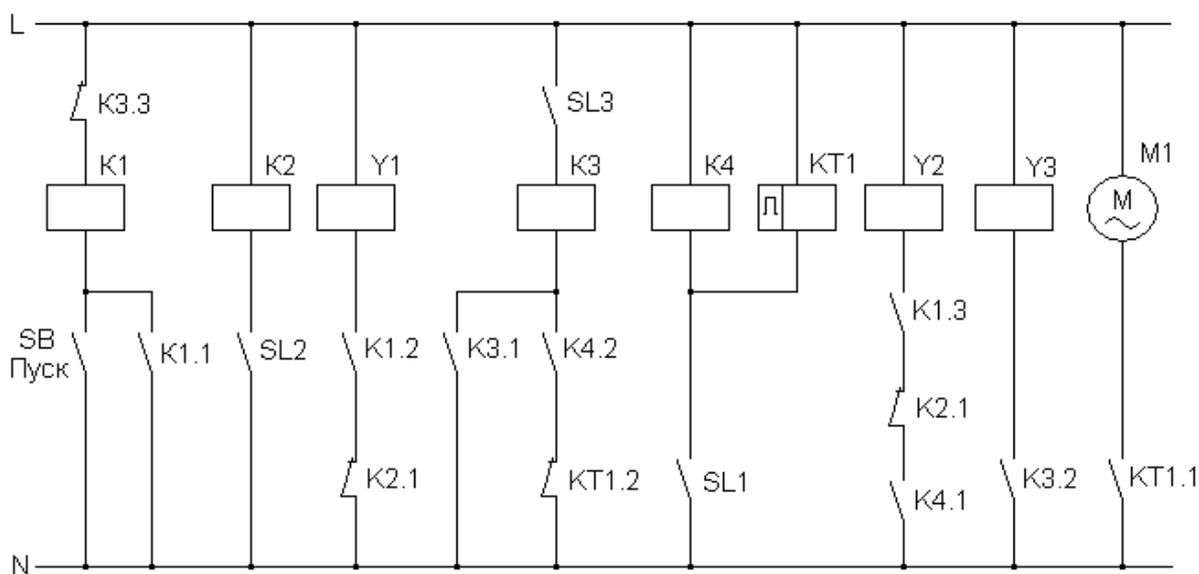


Рисунок 31 - Электрическая схема системы управления на реле

Рассмотрим реализацию данной системы управления смесительной установкой с использованием контроллера LOGO. На рисунке 44 приведена схема подключения контроллера, а на рисунке 45 блок-схема программы.

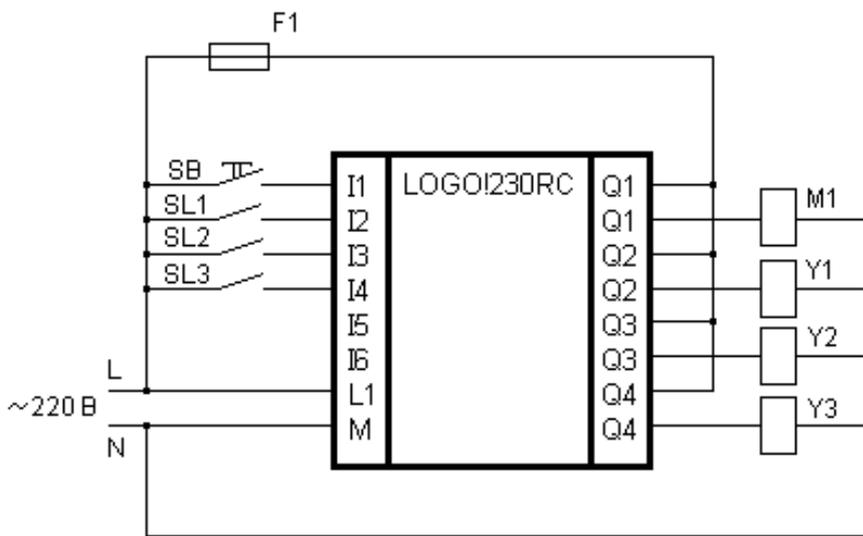


Рисунок 32 - Схема подключения

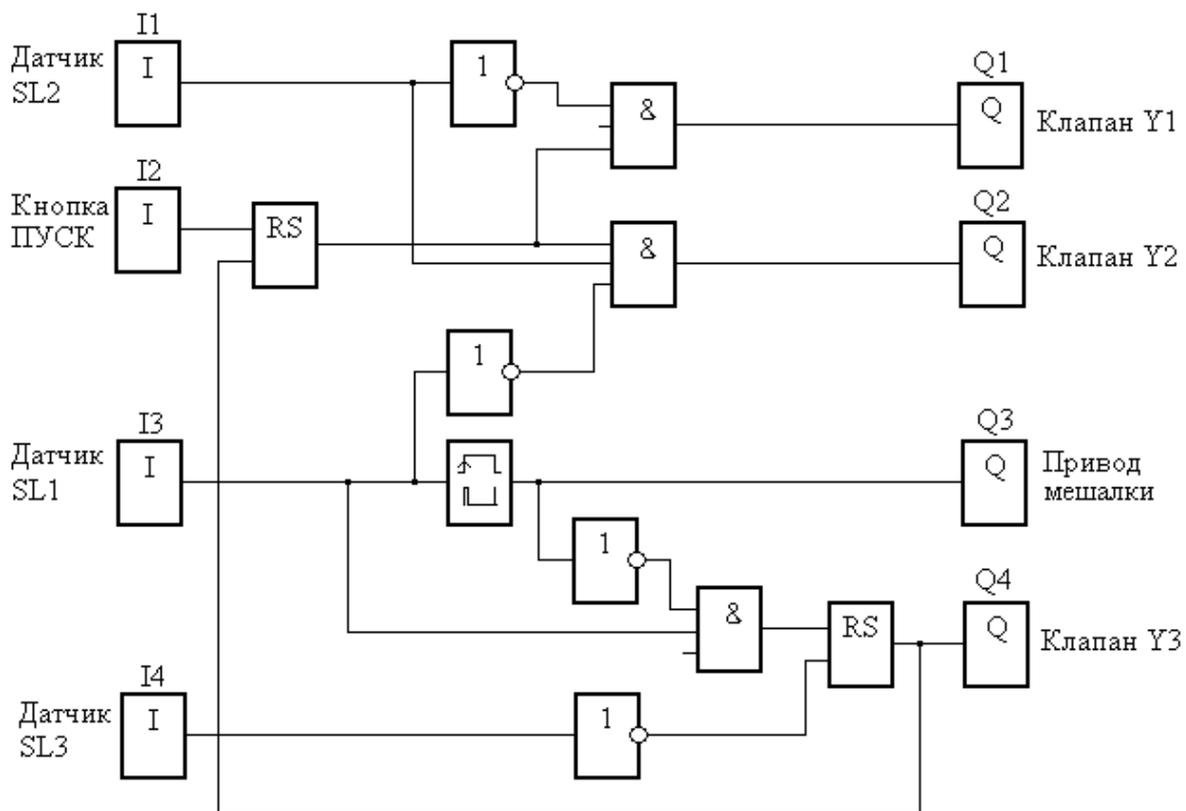


Рисунок 33 - Блок-схема программы

Рассмотрим ещё один пример, с более формализованным подходом к синтезу логического устройства. Необходимо автоматизировать управление

данным объектом, т.е. насосом и задвижкой в соответствии с требуемым алгоритмом управления.

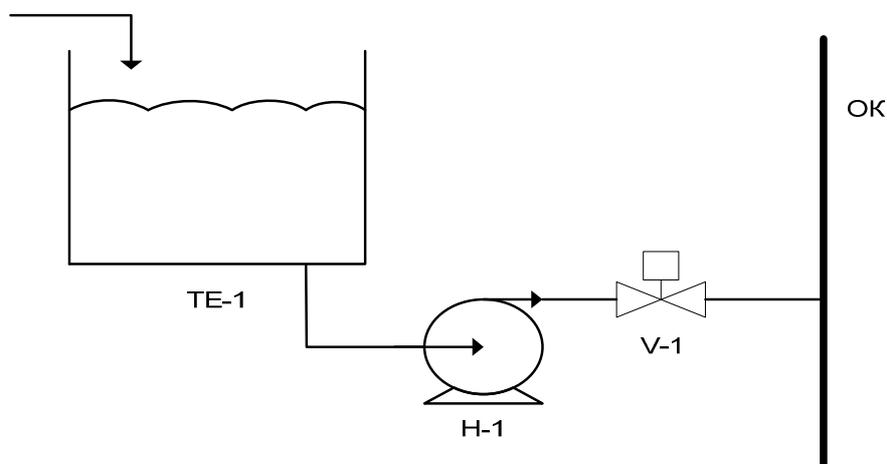


Рисунок 34 – Исходная структура объекта управления

«TE-1» – технологическая емкость; «Н-1» – насосный агрегат с электроприводом; «V-1» – задвижка с электроприводом; «ОК» – общий коллектор.

Словесная формулировка требуемого алгоритма управления.

При достижении жидкостью в технологической емкости верхнего уровня необходимо включить двигатель насоса. При достижении на выходе насоса заданного давления открыть задвижку. При достижении жидкостью нижнего уровня необходимо закрыть задвижку и выключить двигатель насоса.

Если в процессе откачки произойдет перегрев подшипников насоса или давление в общем коллекторе упадет ниже заданного значения, необходимо отключить насос, закрыть задвижку и включить сигнализацию.

Выбор датчиков и исполнительных механизмов, обозначения сигналов.

Датчики:

датчики уровня;

датчик давления на выходе насоса в общем коллекторе;

датчик температуры в подшипниках насоса;

датчик состояние задвижки (открыта или закрыта).

ИМ:

двигатель насоса;

электропривод задвижки;

аварийная сигнализация сигнализации.

Входные сигналы:

ВУ, НУ – сигнал о достижении верхнего, нижнего уровня;

Р_{ОК} – сигнал падения давления в общем коллекторе ниже допустимого;

Р – сигнал достижения заданного давления на выходе насоса;

ЗО, ЗЗ – сигнал состояния задвижки (открыта, закрыта);

У_{сб} – сигнал сброса аварийной сигнализации;

Т – сигнал превышения температуры в подшипниках $T_{\text{допуст}}$.

Выходные сигналы:

U_Н – включение двигателя насосного агрегата;

U_{ЗО} – включение привода ЗД на открытие;

U_{ЗЗ} – включение привода ЗД на закрытие;

I_С – включение сигнализации на ПУ.



Рисунок 35 – информационные потоки управляющего устройства

Формализация алгоритма управления с помощью графа переходов.

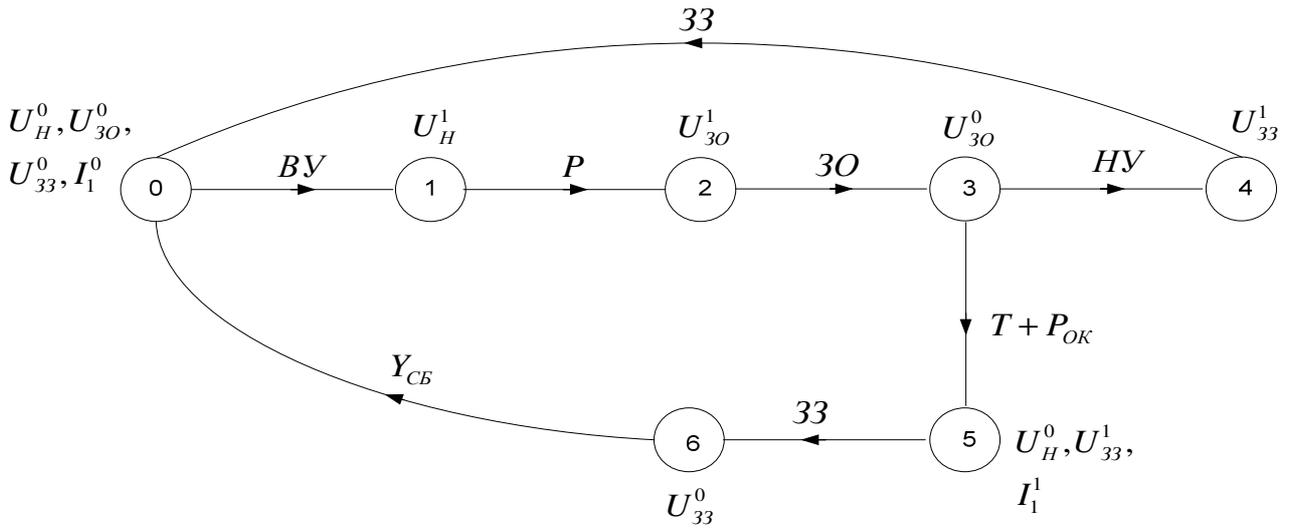


Рисунок 36 – граф состояний управляющего устройства

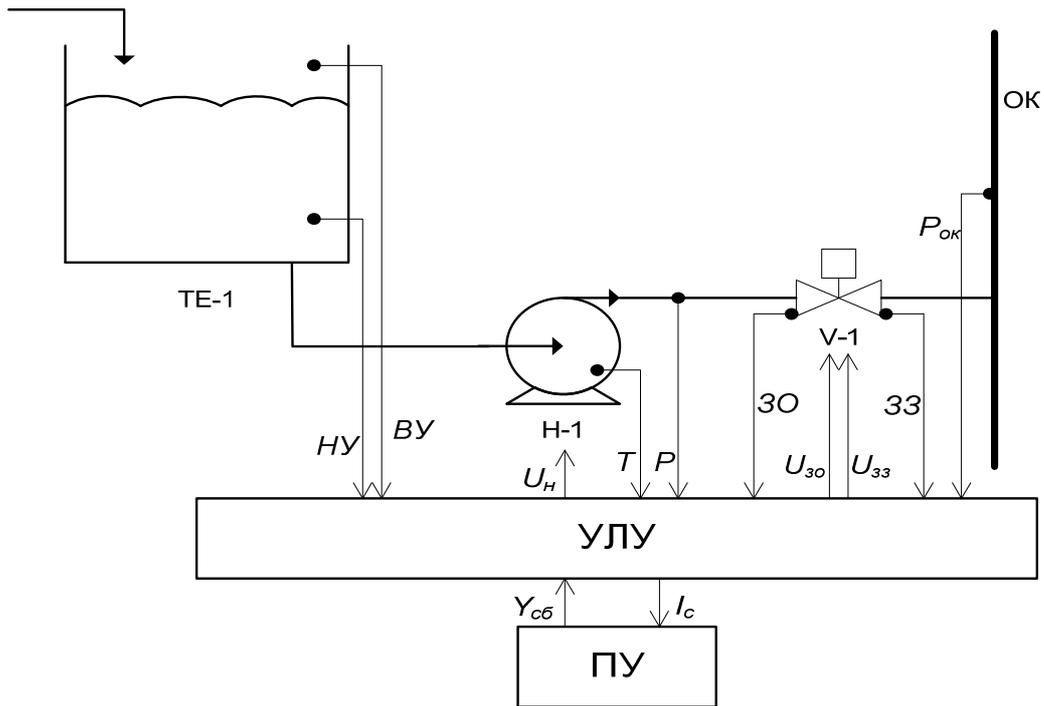


Рисунок 37 – Структурная схема информационных потоков системы управления

1. Синтез блока памяти (БП)

$$N_{ЭП} = 7.$$

2. Синтез блока логических условий (БЛУ)

Структурные формулы включающих и выключающих сигналов.

<p>триггер T_0</p> $S_0 = a_4 \cdot \overline{a_3} \cdot 33 + a_6 \cdot \overline{a_5} \cdot Y_{CB},$ $R_0 = a_1.$	<p>триггер T_1</p> $S_1 = a_0 \cdot (\overline{a_6} + \overline{a_4}) \cdot BY,$ $R_1 = a_2.$	<p>триггер T_2</p> $S_2 = a_1 \cdot \overline{a_0} \cdot P,$ $R_2 = a_3.$
<p>триггер T_3</p> $S_3 = a_2 \cdot \overline{a_1} \cdot 30,$ $R_3 = a_4 + a_5.$	<p>триггер T_4</p> $S_4 = a_3 \cdot \overline{a_2} \cdot HY,$ $R_4 = a_0.$	<p>триггер T_5</p> $S_5 = a_3 \cdot \overline{a_2} \cdot (T + P_{OK}),$ $R_5 = a_6.$
<p>триггер T_6</p> $S_6 = a_5 \cdot \overline{a_3} \cdot 33,$ $R_6 = a_0.$		

3. Синтез блока выходов БВ:

а) на элементах “ИЛИ”

$$U_H = a_1 + a_2 + a_3 + a_4,$$

$$U_{30} = a_2,$$

$$U_{33} = a_4 + a_5,$$

$$I_c = a_5 + a_6.$$

б) на RS-триггерах

для U_H :

$$S_H = a_1,$$

$$R_H = a_0 + a_5.$$

для U_{30} :

$$U_{30} = a_2.$$

для U_{33} :

$$S_{33} = a_4 + a_5,$$

$$R_{33} = a_0 + a_6.$$

для I_c :
 $S_c = a_5$,
 $R_c = a_0$.

4. Программа на языке FBD.

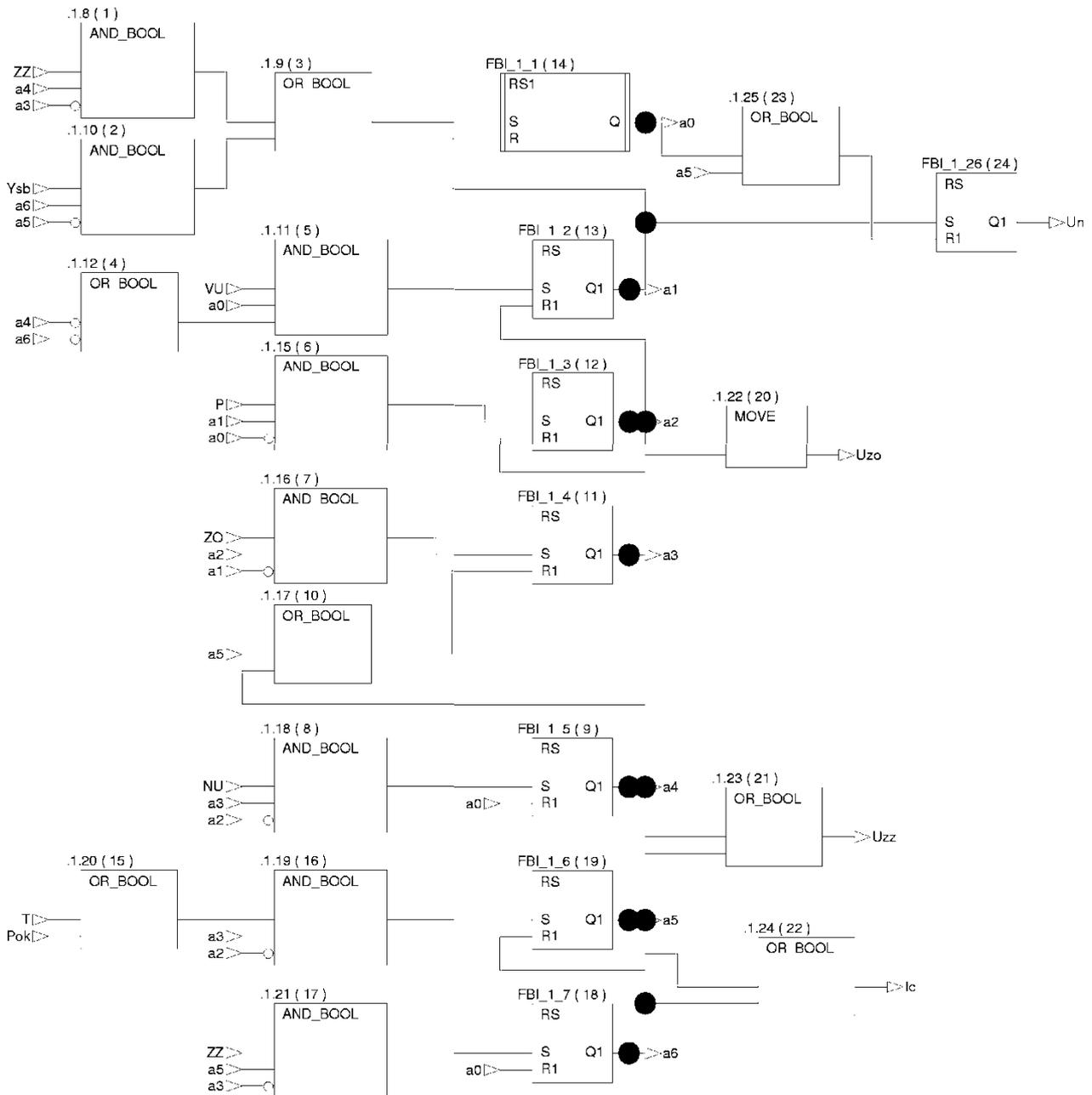


Рисунок 38 – Программа на языке FBD

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Автоматическое управление в химической промышленности: Учебник для ВУЗов./Под ред. Е.Г. Дудникова. - М.: Химия, 1987. 168 с., ил.
2. Беспалов А.В., Харитонов Н.И. Системы управления химико-технологическими процессами. Учебник для вузов. М.: ИКЦ "АКАДЕМКНИГА", 2007. – 696 с
3. Емельянов А.И. Проектирование систем автоматизации технологических процессов. М., Энергоатомиздат, 1983. - 399 с.
4. Ключев А.С. и др. Проектирование систем автоматизации технологических процессов. М., Энергоатомиздат, 1990, - 464 с.
5. Комаровская Н.М., Медведева Л.И., Трушников М.А. Технические средства автоматизации и измерения в химическом производстве. Часть IV. Автоматические регуляторы. Общие сведения о микропроцессорных системах управления: учебное пособие. Волгоград: ИУНЛ ВолгГТУ, 2005. 80 с.
6. Маларев В.И. Проектирование и расчет систем автоматики (учебное пособие). СПб, Изд. СПГГИ, 2003.- 88 с.
7. Нестеров А.Л. Проектирование АСУТП. Методическое пособие. СПб, Изд. ДЕАН, 2009 – 944 с.
8. Родионов В.Д., Терехов В.А., Яковлев В.Б. Технические средства АСУ ТП / Под ред. В.Б. Яковлева. М. : Высшая школа, 1989. 263 с.
9. Стефани Е.П. Основы построения АСУ ТП: - М.: Энергия, 1982. -352 с, ил.
10. Трушников М.А., Медведева Л.И., Гольцов А.С., Носенко В.А. Автоматизированные системы управления в промышленности: учебное пособие, Волгоград: ИУНЛ ВолгГТУ, 2010. 100с.
11. Шишов О.В. Современные технологии промышленной автоматизации: учебник / Саранск : Изд-во Мордов. ун-та, 2007. 250 с.

Электронное учебное издание

Максим Алексеевич **Трушников**

ДИСКРЕТНО-ЛОГИЧЕСКИЕ СИСТЕМЫ УПРАВЛЕНИЯ
Учебное пособие

Электронное издание сетевого распространения

Редактор Матвеева Н.И.

Темплан 2017 г., поз. № 47

Подписано к использованию 26.12.2017. Формат 60x84 1/16.

Гарнитура Times. Усл. печ. л. 5,0

Волгоградский государственный технический университет
400005, г. Волгоград, пр. им. В. И. Ленина, 28. корп. 1

ВПИ (филиал) ВолгГТУ.
404121, г.Волжский. ул. Энгельса, 42а.