



ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

И.А. ДАНИЛУШКИН

АППАРАТНЫЕ СРЕДСТВА И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМ ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ

Учебное пособие

*Рекомендовано УМО по образованию в области радиотехники,
электроники, биомедицинской техники и автоматизации
в качестве учебного пособия
для студентов высших учебных заведений,
обучающихся по специальности 210100
«Управление и информатика в технических системах»*

Самара 2007

УДК 004.(384+7) + 681.58

Аппаратные средства и программное обеспечение систем промышленной автоматизации: Учеб. пособ. / *И.А. Данилушкин*; Самар. гос. техн. ун-т. Самара, 2005. 168 с.

Посвящено вопросам построения систем автоматизированного управления техническими системами и технологическими процессами на базе современных промышленных контроллеров.

Рассматриваются принципы применения вычислительной техники в системах автоматизированного управления. В первой главе рассмотрены основные компоненты систем управления, выделены вопросы защиты измерительной системы от влияния электрических помех. Во второй главе рассмотрены устройство и принципы функционирования программируемых логических контроллеров с магистрально-модульной архитектурой соответствующих стандарту МЭК 61131. Приведены типовые электрические схемы модулей ввода-вывода информации, рассмотрены основные характеристики типовых модулей контроллера. Третья глава посвящена общим принципам функционирования вычислительных сетей и использующихся для их построения последовательных интерфейсов. В четвёртой главе приведены принципы функционирования и основные технические характеристики популярных протоколов промышленных сетей.

Предназначено для студентов старших курсов обучающихся по специальности «Управление и информатика в технических системах».

ISBN

Ил.: 86. Табл.: 28. Библиогр.: 62 назв.

Печатается по решению редакционно-издательского совета Самарского государственного технического университета

Рецензенты: д-р техн. наук, проф. С.Я. Галицков;
д-р техн. наук, проф. М.Ю. Лившиц.

ISBN

© И.А. Данилушкин, 2007

© Самарский государственный технический университет, 2007

ПРЕДИСЛОВИЕ

Компьютерные технологии проникли практически во все сферы человеческой жизни. И, если первые компьютеры использовались исключительно для выполнения различных вычислений, то сейчас компьютер воспринимается в первую очередь как средство для организации отдыха, развлечения и общения. Настоящее учебное пособие посвящено применению вычислительной техники в ещё одной области человеческой жизнедеятельности – области автоматизации технических систем и технологических процессов. Рассмотрен круг вопросов связанных с принципами построения современных автоматизированных систем управления на базе промышленных контроллеров и систем распределённого сбора данных.

В первой главе рассматриваются принципы и особенности применения вычислительной техники в системах управления технологическими процессами. На базе типовой структуры системы управления рассматриваются основные компоненты системы и их характеристики. Особое внимание посвящено вопросам объединения отдельных компонентов системы с помощью электрических проводников. Рассмотрены вопросы согласования сигналов и подбора входных и выходных сопротивлений подключаемых устройств; защиты сигнальных проводов от воздействия электрических помех; выбора тока или напряжения в качестве носителя информационного сигнала.

Вторая глава посвящена вопросам аппаратной организации управляющих ЭВМ. Наиболее подробно рассмотрен такой класс управляющих ЭВМ, как магистрально-модульные программируемые логические контроллеры (ПЛК). Приведены сведения о возможных конструктивных исполнениях контроллеров на базе открытых и закрытых системных интерфейсов. Рассмотрены принципы функционирования ПЛК в соответствии с требованиями стандарта МЭК 61131: функциональная структура, принципы исполнения рабочего цикла, режимы работы, приведены основные функциональные характеристики модулей центрального процессора. Для модулей дискретного и аналогового ввода-вывода приведены типовые функциональные и электрические схемы, варианты подключения датчиков и исполнительных устройств, их основные характеристики.

В третьей главе рассматриваются общие принципы построения промышленных сетей: основные типы сетевых топологий, способы управления доступом к среде передачи, организация сетей на базе модели взаимодействия открытых систем. Достаточно часто на физическом уровне промышленных сетей используются последовательные интерфейсы. В учебном пособии приведены их технические характеристики, принципы организации асинхронной и синхронной передачи данных.

В четвёртой главе рассмотрены наиболее часто применяемые стандарты промышленных сетей. Приводятся сведения о технических и электрических характеристиках сетей, используемых методах кодирования и передачи данных; рассматривается реализация протокола доступа к среде передачи и форматы передаваемых сообщений. Каждый из рассмотренных стандартов имеет свою «изюминку» – оригинальное решение в одном из аспектов функционирования промышленных сетей. Например, протокол Modbus абстрагировался от реализации физического уровня, ограничившись только самыми общими требованиями, а протокол CAN использует для обмена данными исключительно широкополосные передачи. Таким образом, знакомство с приведёнными в учебном пособии стандартами сетей позволит читателю в дальнейшем быстро разобраться с любым вновь встретившимся сетевым протоколом.

Автор выражает благодарность своей жене и родителям, оказавшим неоценимую поддержку в процессе написания пособия; коллегам Колпащикову С.А. и Серенкову В.Е., чей практический опыт разработки и внедрения АСУ ТП позволил автору глубже разобраться во многих изложенных в пособии вопросах; заведующему кафедрой автоматики и управления в технических системах СамГТУ – проф. Рапопорту Э.Я., неустанно напоминавшему о необходимости довести начатое пособие до конца.

Автор надеется, что данное учебное пособие окажется полезным при изучении дисциплин, рассматривающих современные средства промышленной автоматизации, а также на этапе становления специалиста по автоматизации при разработке его первой системы.

1. ПРИНЦИПЫ ПОСТРОЕНИЯ СИСТЕМ ЦИФРОВОГО УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ

1.1. РОЛЬ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ В УПРАВЛЕНИИ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ

Применение вычислительной техники в автоматическом управлении – важнейшая черта технической инфраструктуры современного общества. Промышленность, транспорт, системы связи и защита окружающей среды существенно зависят от компьютерных систем управления. Практически ни одна техническая система – от железной дороги до ядерного реактора – не работает без той или иной формы управления. ЭВМ играют здесь ключевую роль; во многих случаях не существует реальной альтернативы компьютерному управлению процессами.

Что же подразумевается под термином «процесс»? Физический процесс – это последовательная смена состояний объектов физического мира. Процессами в этом смысле являются движение, химические реакции или теплообмен. Примеры процессов – промышленное или химическое производство, кондиционирование воздуха в помещении (изменение физических параметров – температуры и влажности), движение транспортного средства (изменение его скорости и положения). Технический или технологический процесс – это процесс, физические переменные которого можно измерить и изменить техническими средствами [1].

Разница между физическим и технологический процессом заключена в том, что физический процесс не обязательно должен управляться извне, а технологический процесс включает обработку информации для достижения заданной целевой функции.

Любой физический процесс характеризуется входом и выходом в виде:

- материальных компонентов;
- энергии;
- информации.

Некоторые примеры приведены в табл. 1.1.

В общем случае материальные компоненты (энергию и информацию) можно рассматривать как входные и выходные потоки, которые изменяются в ходе физических/технических процессов.

Материалы и энергия являются основными составляющими физического процесса. Информация – тоже неотъемлемая часть всякого процесса, однако, осознание этого факта произошло не так давно.

Примеры технологических процессов

Вход/выход	Химический реактор	Кондиционирование воздуха
Вход материальных ресурсов	Потоки исходных реагентов	
Выход материальных компонентов (продукции)	Один или несколько новых продуктов	
Вход энергии	Нагревание или охлаждение	Нагревание или охлаждение
Выход энергии	Получение тепла от реакции	Излучение или поглощение тепла
Ввод информации	Управление входными потоками реагентов и дополнительным нагревом	Управление температурой и интенсивностью поступления нагревающей/ охлаждающей жидкости
Вывод информации	Измерение температуры, давления, интенсивности потоков, концентрации	Измерение температуры

Процесс производства (рис. 1.1) заключается в выпуске продукции из сырья с соответствующими затратами (вводом) энергии. Входной информацией являются технологические инструкции, выраженные в виде набора параметров, которые можно явно контролировать. Выходная информация есть набор измеряемых переменных и параметров, которые описывают текущее состояние процесса и его изменение. Большое количество информации заключено в самом конечном продукте. Информация, следовательно, есть не только данные для наблюдения и управления, но и технологические и организационные процедуры вплоть до циркулирующих по кабинетам служебных документов и заявок на поставку. Этот вид информации также важен, как и любой другой элемент, обеспечивающий нормальный ход и оптимизацию производственного процесса.

Информация – важнейший компонент управления физическими процессами, поскольку она позволяет лучше использовать два других слагаемых процесса – материю и энергию.

Компьютеры, собственно, и предназначены для обработки информации, в том числе и относящейся к техническим процессам. В большинстве случаев компьютеры выполняют две основные функции: во-первых, контролируют, находятся ли параметры технического процесса в заданных пределах, и, во-вторых, инициируют соответствующие управляющие воздействия, чтобы параметры оставались в этих пределах даже при наличии внешних возмущений.

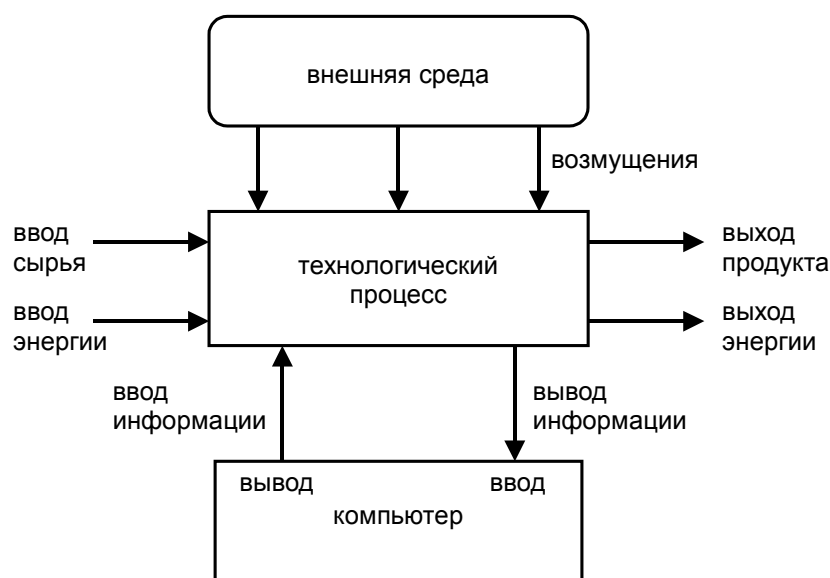


Рис. 1.1. Применение компьютера в управлении процессом

В качестве внешних возмущений выступают посторонние по отношению к цели процесса факторы, которыми нельзя управлять, но которые оказывают влияние на процесс. Эти факторы отклоняют процесс от штатного рабочего режима. Возмущения сами по себе не являются физическими величинами, а проявляются в виде случайных флуктуаций в потоках материалов, энергии и информации.

Управление технологическим процессом отличается от обычной обработки данных. В таких приложениях, как бухгалтерский счёт или редактирование текста, и вход, и выход представляют собой данные в чистом виде, т.е. их можно хранить или передавать с помощью любого носителя информации. В этом случае время обработки зависит только от производительности компьютера, а результат будет всегда один и тот же.

Ситуация меняется в случае управляющих компьютеров. Здесь обработка данных не должна зависеть от компьютера и его производительности, а, напротив, должна следовать за событиями во внешнем мире, т.е. за процессом. Компьютерная система управления должна достаточно быстро реагировать на внешние события и постоянно обрабатывать поток входных данных, чаще всего не имея возможности изменить их количество или скорость поступления. Одновременно может потребоваться и выполнение других операций, например обмен информацией с оператором, вывод данных на экран и реакция на определённые сигналы. Этот режим обработки данных оказался настолько важным, что получил специальное название – **режим реального времени** (real-time mode).

1.2. ОСОБЕННОСТИ ЦИФРОВОГО УПРАВЛЕНИЯ

Под «особенностями» систем цифрового управления обычно понимают преимущества, которые даёт использование ЭВМ в автоматическом управлении, и

недостатки, о которых необходимо знать и которые должны учитываться при разработке автоматических систем.

Первые шаги в использовании ЭВМ для реализации автоматических систем управления были сделаны применительно к задачам управления последовательностью событий. Например, приготовление смеси: залить компонент А до уровня h_1 ; залить компонент Б до уровня h_2 ; перемешивать 5 минут; слить готовую смесь. Такие задачи раньше решались с помощью реализации логики переключений на релейных схемах. Поскольку внесение изменений в электрическую схему более трудоёмкая задача, чем перепрограммирование, то здесь ЭВМ сразу же оказались на своём месте.

Однако, помимо задач управления последовательностью событий, есть и другие задачи, требующие автоматического управления – это задачи регулирования аналоговых параметров: расходов, давлений, температур и т.п. И в этой области использование ЭВМ наталкивается на ряд ограничений, связанных с дискретным представлением информации и особенностью её обработки. Изучением вопросов дискретного управления занимается теория дискретных систем автоматического управления [2–4]. Попытаемся на примере рассмотреть те подводные камни, которые могут ожидать при использовании ЭВМ в задачах управления.

Рассмотрим систему круиз-контроля современного автомобиля. Задача, которую решает система – поддержание постоянной скорости движения. Когда машина движется по горизонтальной поверхности – достаточно держать на заданном уровне расход топлива. Но, движение под гору, в гору, встречный и попутный ветер, качество дорожного покрытия приводят к изменению скорости автомобиля. Следовательно, необходимо постоянно сравнивать текущую скорость движения с заданной скоростью и, если текущая скорость стала меньше, необходимо увеличить подачу топлива, если больше – снизить (см. рис. 1.2). В идеале выполнять сравнение нужно непрерывно, однако, ЭВМ требуется некоторое время для преобразования сигнала с датчика текущей скорости в цифровой сигнал, выполнения расчёта управляющего воздействия и формирования сигнала для управления подачей топлива. ЭВМ цикл за циклом последовательно выполняет измерение – расчёт – формирование, поэтому измерение скорости и изменение расхода топлива происходит только в дискретные моменты времени.



Рис. 1.2. Возмущения, вносимые внешней средой в скорость движения автомобиля

Процедура преобразования аналогового сигнала в цифровой, приводит к появлению погрешности измерения аналоговых сигналов (дискретизация по уровню), что, в общем, должно учитываться при разработке алгоритмов управления.

Дискретизация (квантование) по времени, связанная с преобразованием входных/выходных сигналов и выполнением программы приводит к «отставанию» ЭВМ от реального процесса. Если отставание окажется достаточно большим, то система не сможет правильно реагировать в условиях быстро меняющейся ситуации, что приведёт к выходу из строя объекта управления.

Например, в момент t_1 (рис. 1.3), по результатам измерения скорости автомобиля, движущегося в гору, принимается решение увеличить подачу топлива, но из-за долгого времени выполнения алгоритма, управляющее воздействие будет выдано в момент t_2 , из-за чего произойдёт увеличение скорости при движении под гору, что может привести к аварии.

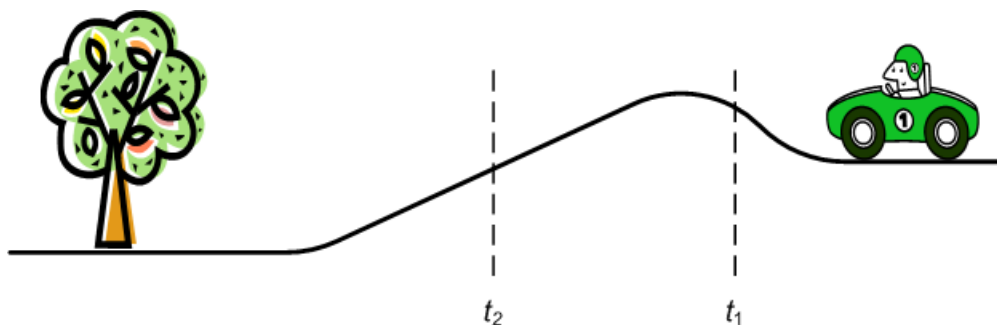


Рис. 1.3. Иллюстрация к режиму реального времени

Таким образом, от управляющей ЭВМ требуется достаточная скорость реакции, определяемая динамикой объекта управления.

Ещё один недостаток цифровой системы управления – последовательная обработка информации, что входит в противоречие с «параллельностью» процессов окружающего нас реального мира. Все события вокруг нас, мы сами и фактически любые физические процессы можно представить в виде множества «подпроцессов», которые протекают параллельно.

Эта проблема успешно решается за счёт специального способа построения программного обеспечения управляющей ЭВМ.

1.3. ТИПОВАЯ СТРУКТУРА СИСТЕМЫ УПРАВЛЕНИЯ

В общем случае система цифрового управления физическим/техническим процессом состоит из следующих компонентов (рис. 1.4) [1]:

- управляющая ЭВМ (УВМ);
- каналы обмена информацией;

- аналого-цифровые и цифро-аналоговые преобразователи (АЦП и ЦАП);
- датчики и исполнительные механизмы;
- сам физический/технический процесс.

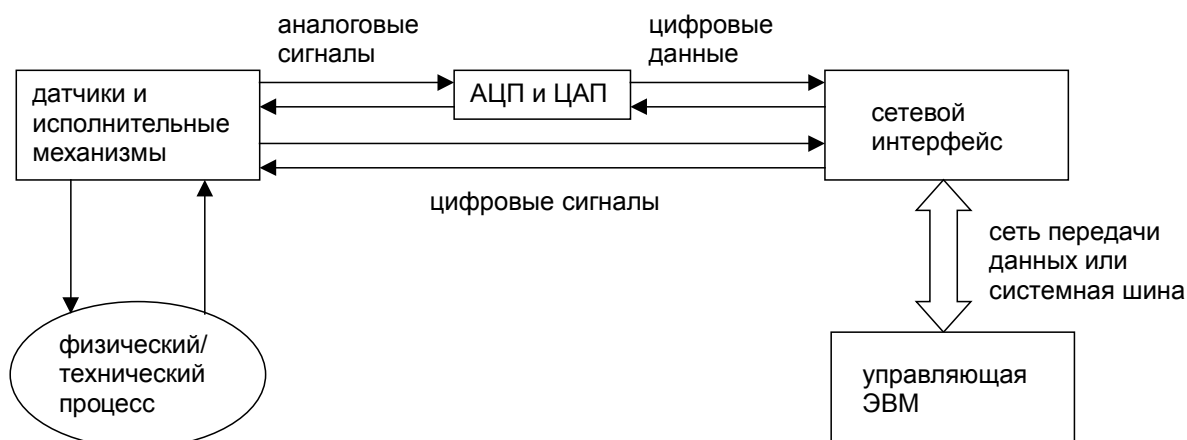


Рис. 1.4. Основная структура системы цифрового управления процессом

Физический процесс контролируется с помощью датчиков, т.е. устройств, преобразующих физические параметры процесса (температуру, давление или координаты) в электрические величины, которые можно непосредственно измерить (сопротивление, ток или разность потенциалов). Непосредственное влияние на процесс осуществляется с помощью исполнительных механизмов. Они преобразуют электрические сигналы в физические воздействия, главным образом движение – перемещение или вращение, которые можно использовать для разных целей, например, для открытия или закрытия заслонки.

Цифровые системы управления работают только с информацией, представленной в цифровой форме, поэтому полученные в результате измерений электрические величины необходимо обработать с помощью АЦП. Обратная операция – управление исполнительными механизмами – несколько проще, поскольку компьютер может непосредственно вырабатывать электрические сигналы.

Информация от удалённых объектов через каналы связи поступает к управляющему компьютеру, который:

- интерпретирует все поступающие от физического процесса данные;
- принимает решения в соответствии с алгоритмами программ обработки;
- посылает управляющие сигналы;
- обменивается информацией с человеком-оператором и реагирует на его команды.

1.4. КОМПОНЕНТЫ ИНТЕРФЕЙСА МЕЖДУ ПРОЦЕССОМ И УПРАВЛЯЮЩИМ КОМПЬЮТЕРОМ

Общая структура ввода/вывода между процессом и управляющим компьютером показана на рис. 1.5. Хотя на практике используются разнообразные датчики, исполнительные механизмы и согласующие устройства, основная структура интерфейса всегда одна и та же.

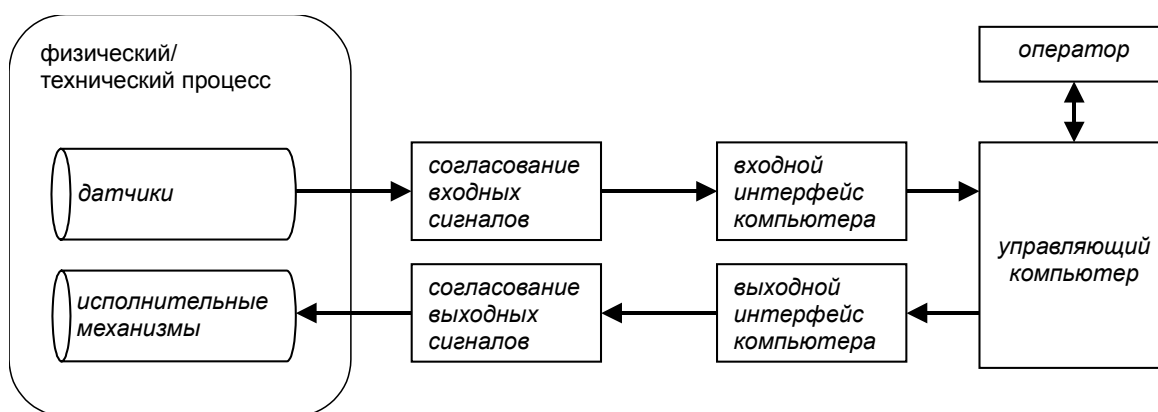


Рис. 1.5. Общая структура ввода/вывода между процессом и управляющим компьютером

Датчики (первичные измерительные преобразователи)

Для большинства физических величин существует множество измерительных технологий, характеризующихся зависимостью между вырабатываемым сигналом и измеряемой величиной. Датчик или первичный преобразователь состоит из двух частей: чувствительного элемента и измерительного преобразователя.

Результат измерения – это «реакция чувствительного элемента», которая на выходе датчика представляет собой электрическую величину, распространяющуюся дальше по проводнику. Иногда в качестве выходного сигнала используется пневматический сигнал.

Главное достоинство электрических датчиков – это гибкость и разнообразие способов обработки сигнала. Следует отметить, что электрический сигнал можно передавать на большие расстояния с очень малыми затратами энергии. Пневматические датчики, по сравнению с электрическими, обычно дешевле, меньше по размерам, проще и нечувствительны к возмущениям. Более того, в условиях взрыво- и пожароопасной среды пневматические датчики более безопасны, чем электрические.

Различают три класса датчиков:

- аналоговые датчики, т.е. датчики, вырабатывающие аналоговый сигнал;
- бинарные (двоичные, дискретные) датчики, которые вырабатывают сигнал только двух уровней «включено/выключено» (иначе говоря, «0» или «1»);

- цифровые датчики, генерирующие последовательность импульсов или двоичное слово.

Исполнительные устройства и исполнительные механизмы

Исполнительные устройства и механизмы преобразуют электрическую энергию в механическую или физическую величину для воздействия на управляемый процесс. Как уже говорилось, исполнительные механизмы служат для преобразования электрической энергии в движение. В качестве исполнительных механизмов чаще всего выступают электродвигатели, которые используются в роли приводов насосов и вентиляторов, применяются для перемещения манипуляторов роботов и т.п. К исполнительным механизмам относят и электромагнитные клапаны, где в качестве привода используется соленоид.

Исполнительные устройства используются для преобразования электрической энергии в другие виды энергии (индукционные печи, печи сопротивления, микроволновые печи, лазеры). Для работы исполнительных механизмов обычно требуется электроэнергия большой мощности, подачей которой управляет также исполнительное устройство, поэтому понятие «исполнительное устройство» является обобщающим.

Следует отметить, что исполнительные устройства лишь опосредованно влияют на параметры физических процессов, измеряемые датчиками. Например, датчики измеряют температуру, а исполнительные устройства управляют подводом тепла. И уже от динамических характеристик физической системы зависит, как измеряемые величины изменятся из-за управляющих воздействий исполнительных устройств.

Требования к исполнительным устройствам – потребляемая мощность, разрешающая способность, повторяемость результата, рабочий диапазон и т.д. – могут существенно различаться в зависимости от конкретного технологического процесса. Для успешного управления процессом правильный выбор исполнительных устройств так же важен, как и правильный выбор датчиков.

Для перемещения клапанов часто применяется сжатый воздух. Если необходимо развивать значительные усилия, обычно используют гидропривод. Электрический сигнал компьютера должен быть преобразован в давление или расход воздуха либо масла. Бинарное управление обеспечивается электромеханическими реле или электронными переключателями.

Характеристики аналоговых датчиков

Дискретные датчики используются для *качественной* оценки измеряемого параметра: жидкость достигла заданного уровня, температура поднялась выше допустимого значения и т.п. С помощью аналоговых датчиков можно получить

количественную оценку величин измеряемых параметров. Кроме того, непрерывное измерение аналогового параметра позволяет учитывать не только текущее значение параметра, но и динамику его поведения (быстро увеличивается, медленно уменьшается и т.п.), что, в свою очередь, позволяет реализовать более точную систему стабилизации параметра. Именно поэтому аналоговые датчики занимают особое место в системах автоматического управления.

Аналоговый датчик должен воспроизводить физическую величину максимально быстро и точно. Хотя чаще всего датчик выбирают исходя из надёжности и удобства обслуживания, его точность, стабильность и повторяемость результатов остаются важнейшими факторами. Основой работы управляющего компьютера является входная информация, поэтому точные и надёжные измерения – это необходимое условие качества управления.

Различают статические и динамические характеристики датчиков.

Статические характеристики датчиков показывают, насколько корректно выход датчика отражает измеряемую величину спустя некоторое время после её изменения, когда выходной сигнал установился на новое значение. Важными статическими параметрами являются: чувствительность, разрешающая способность или разрешение, линейность или нелинейность, дрейф нуля и полный дрейф, рабочий диапазон, повторяемость и воспроизводимость результата [1, 5].

Чувствительность датчика (коэффициент усиления) определяется как отношение величины выходного сигнала к единичной входной величине.

Разрешение – это наименьшее измерение измеряемой величины, которое может быть зафиксировано и точно показано датчиком.

Линейная зависимость выхода датчика y от входа x – это свойство датчика, при котором коэффициент усиления датчика постоянен и не зависит от значения измеряемой величины:

$$y(x) = k \cdot x + b, \quad k = const, \quad b = const. \quad (1.1)$$

Большинство датчиков имеют линейную характеристику в рабочем диапазоне. Нелинейная статическая характеристика датчика чаще всего связана со способом измерения физической величины, например, при измерении расхода по перепаду давлений до и после сужающего устройства, расход оценивается исходя из уравнения Бернулли по формуле:

$$G = \sqrt{\Delta P}. \quad (1.2)$$

Другой пример нелинейности – зависимость выхода термометра сопротивления от значения измеряемой температуры [6]

$$R = R_0 \cdot \exp(\alpha \cdot (T - T_0)). \quad (1.3)$$

Дрейф определяется как отклонение показаний датчика, когда измеряемая величина остаётся постоянной в течение длительного времени. Величина дрейфа

может определяться при нулевом, максимальном или некотором промежуточном значении входного сигнала. Дрейф датчика вызывается нестабильностью усилителя, изменением окружающих условий (например, температуры, давления, влажности или уровня вибраций), параметров электроснабжения или самого датчика (старение, выработка ресурса и т.д.).

Рабочий диапазон датчика определяется допустимыми верхним и нижним пределами значения входной величины или уровня выходного сигнала.

Повторяемость характеризуется как отклонение между несколькими последовательными измерениями при заданном значении измеряемой величины в одинаковых условиях, в частности приближение к заданному значению должно происходить всегда и либо как нарастание, либо как убывание. Измерения должны быть выполнены за такой промежуток времени, чтобы не проявлялось влияние дрейфа. Повторяемость обычно выражается в процентах от рабочего диапазона.

Воспроизводимость аналогична повторяемости, но требует большего интервала между измерениями. Между проверками на воспроизводимость датчик должен использоваться по назначению и, более того, может быть подвергнут калибровке. Воспроизводимость задаётся в виде процентов от рабочего диапазона, отнесённых к единице времени (например, к месяцу).

Свойства, отражающие работу датчика в условиях изменяющихся входных воздействий, называются **динамическими характеристиками**. Они существенно влияют на работу системы управления. Идеальный датчик мгновенно реагирует на изменение измеряемой физической величины. На практике любому датчику необходимо некоторое время на отработку нового входного сигнала. Очевидно, что для адекватного отображения реальных изменений наблюдаемой величины время реакции датчика должно быть как можно меньше. Это тот же самый принцип, который применяется ко всей системе управления процессом реального времени в целом: временные характеристики физического процесса определяют быстродействие системы. Однако чаще требуется компромисс между скоростью реакции датчика и его чувствительностью к шуму.

Динамические свойства датчика характеризуются целым рядом параметров, которые, однако, довольно редко приводятся в технических описаниях производителей. Динамическую характеристику датчика (его передаточную функцию) можно экспериментально получить как реакцию на скачок измеряемой входной величины (переходная функция). Параметры, описывающие реакцию датчика, дают представление о его скорости, инерционных свойствах и точности.

Выделяют следующие динамические характеристики датчика [1, 5].

Запаздывание – время между началом изменения физической величины и моментом реакции датчика, т.е. моментом начала изменения выходного сигнала. В литературе встречаются и другие определения запаздывания.

Время нарастания – время, за которое выходной сигнал увеличивается от 10 до 90% установившегося значения. Малое время нарастания всегда указывает на быструю реакцию.

Время переходного процесса, время установления – время, начиная с которого отклонение выхода датчика от установившегося значения становится меньше заданной величины (например, $\pm 5\%$).

Время достижения первого максимума – время достижения максимума выходного сигнала (максимум перерегулирования).

Относительное перерегулирование – разность между максимальным и установившимся значениями, отнесённая к установившемуся значению (в процентах).

Последние две характеристики встречаются довольно редко, поскольку большинство датчиков всё же имеют апериодический переходный процесс, без перерегулирования. Ведь наличие перерегулирования говорит о колебательном характере переходного процесса, а значит и о наличии резонансной частоты, вблизи которой поведение датчика будет резко ухудшаться.

Полоса пропускания и шум

При обработке измерительных сигналов **полоса пропускания** определяется как диапазон рабочих частот датчика. Только те физические величины, рабочие частоты которых лежат в полосе пропускания, можно надлежащим образом измерить или изменить. Это означает, что скорость реакции датчика достаточна для правильного отображения изменений исходной физической величины, при этом сигнал не искажается из-за несоответствия динамики датчика и процесса.

Аналогично, исполнительный механизм должен иметь соответствующую полосу пропускания, чтобы реализовать нужное управляющее воздействие. Чем шире полоса пропускания, тем быстрее будет реакция датчика или исполнительного механизма. Последнее не всегда является положительным фактором, поскольку в этом случае устройство более восприимчиво к нежелательным высокочастотным возмущениям.

Любой измерительный сигнал искажается **возмущениями** и **шумом** как в процессе формирования, так и передачи. Одна из основных проблем передачи сигнала – уменьшение влияния шума. Источники шума должны быть изолированы, или, в крайнем случае, их влияние должно быть снижено до минимально возможного уровня. Искажение сигналов или сообщений шумом является не только проблемой организации интерфейса «процесс – компьютер», но и проявляется

при любых типах передачи информации. Регуляторы обычно проектируются в расчёте на наличие возмущений и шумов.

1.5. СОГЛАСОВАНИЕ И ПЕРЕДАЧА СИГНАЛОВ

Передача информации между различными частями системы управления является одним из неотъемлемых и критически важных элементов. Электрические сигналы, вырабатываемые датчиками, обычно имеют весьма низкий уровень, поэтому для дальнейшей передачи их необходимо обработать и усилить. Уровни сигнала и импедансы выхода датчика, кабеля и входа компьютера должны соответствовать друг другу. Обработка сигнала для достижения указанного соответствия называется **согласованием сигнала**.

Другой очень важной практической проблемой являются наводки. Любое электронное устройство способствует возникновению электрических возмущений. Если две электрические цепи по тем или иным причинам расположены рядом друг с другом, то изменение тока или напряжения в одной цепи вызывает также изменения тока и напряжения в другой. В частности, соединительные провода и кабели выступают в качестве антенны для шумов и возмущений. Многие проблемы, связанные с электрическими наводками, можно решить с помощью экранирования цепей заземления. Выбор способа передачи сигнала (напряжение, ток или свет) зависит от нескольких факторов, главным из которых является устойчивость к наводкам и шумам.

Согласование сигналов

Последовательное соединение одного или нескольких электронных устройств преобразования измерительного сигнала может существенно изменить поведение каждого из них по сравнению с автономным использованием. Например, подключение вольтметра может изменить токи и напряжения в цепи. Такой эффект называется **эффектом нагрузки**. Ошибки, вызванные эффектом нагрузки, могут превосходить все другие типы ошибок измерения; их следует предусматривать при соединении различных датчиков и устройств передачи/преобразования информации.

Процедура уменьшения влияния эффекта нагрузки называется согласованием сигналов. Она выполняется путём подбора значений входных и выходных **импедансов** (полных сопротивлений) соединяемых устройств. Для согласования импедансов при подключении датчика к системе сбора данных может потребоваться включение одного или нескольких усилителей между датчиком и входом системы.

Подбор входных и выходных импедансов

Рассмотрим два усилителя (рис. 1.6), соединённых последовательно. Каждый усилитель представлен в виде четырёхполюсника, ЭДС источника которого определяется формулой

$$e = K \cdot U_{\text{вх}} . \quad (1.4)$$

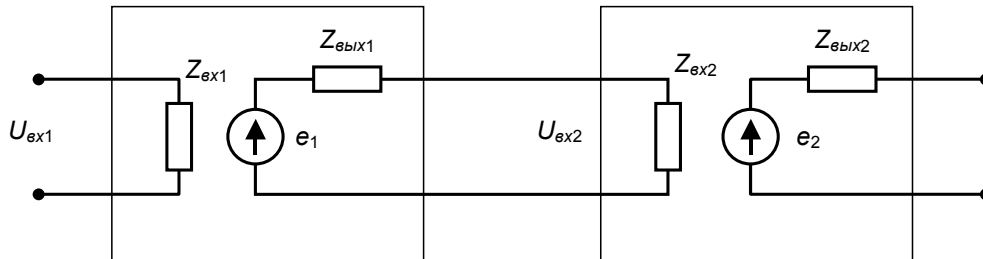


Рис. 1.6. Согласование импедансов последовательно соединённых четырёхполюсников

Задав коэффициенты усиления K_1 и K_2 , связывающие входные напряжения и ЭДС можно получить следующие выражения.

$$U_{\text{вх}2} = \frac{e_1}{Z_{\text{вых}1} + Z_{\text{вх}2}} \cdot Z_{\text{вх}2} , \quad (1.5)$$

$$e_2 = \frac{K_1 \cdot U_{\text{вх}1}}{Z_{\text{вых}1} + Z_{\text{вх}2}} \cdot Z_{\text{вх}2} \cdot K_2 = K_1 \cdot K_2 \cdot \frac{Z_{\text{вх}2}}{Z_{\text{вых}1} + Z_{\text{вх}2}} \cdot U_{\text{вх}1} . \quad (1.6)$$

Из выражения (1.6) видно, что значение ЭДС e_2 будет стремиться к произведению $K_1 \cdot K_2 \cdot U_{\text{вх}1}$, если выходной импеданс первого четырёхполюсника $Z_{\text{вых}1}$ будет намного меньше, чем входной импеданс второго $Z_{\text{вх}2}$, т.е. амплитудные искажения сигнала будут стремиться к нулю. Кроме того, частотная характеристика устройства с двумя последовательно соединёнными усилителями не будет иметь значительных искажений, если их импедансы согласованы.

Вообще говоря, устройство с высоким **входным импедансом** потребляет меньший ток при заданном напряжении и, соответственно, меньшую мощность. Устройство с низким входным импедансом потребляет больший ток при данном напряжении. Поскольку оно отбирает большую мощность у предшествующего в цепи устройства, это может вызвать ошибки нагрузки. Напряжение, генерируемое устройством с высоким **выходным импедансом**, чрезвычайно чувствительно к эффекту нагрузки. Напротив, низкий выходной импеданс позволяет уменьшить зависимость выходного напряжения от тока нагрузки. Для выходных сигналов в виде силы тока картина прямо противоположная: высокий выходной импеданс делает выходной сигнал менее подверженным влиянию токов нагрузки. Поэтому во многих случаях требуется применение специальных согласующих электронных

устройств для усиления сигналов и приведения в соответствие импедансов. В случае сигналов в виде напряжений применяются усилители с высоким входным импедансом и низким выходным импедансом. Процедура согласования импедансов должна тщательно выполняться на каждом этапе.

Электрические проводники

Электрические проводники являются средой, по которой электрические сигналы распространяются в пространстве. Передаваемый по проводнику сигнал по мере удаления от источника подвергается изменениям, определяемым физическими характеристиками линии передачи. Поэтому проводники должны рассматриваться как отдельный элемент системы «датчик – согласующие цепи – управляющий компьютер». Передача постоянного тока – простая задача в тех случаях, когда импеданс проводника мал по сравнению импедансом приёмника на конце линии.

Передача импульсов или других быстро меняющихся сигналов порождает, как правило, некоторые проблемы. Если длина линии такова, что время распространения сигнала соизмеримо с величиной периода сигнала или временем нарастания, возникают побочные эффекты (скорость света 300 м/мкс). Одним из таких эффектов является отражение сигналов.

Передача электрического сигнала может осуществляться изменением уровня напряжения относительно «земли» – провода с нулевым потенциалом. Такая цепь называется **несбалансированной** (рис. 1.7, а).

Другой способ передачи – использование **сбалансированной** цепи (рис. 1.7, б). В этом случае по обоим проводам двухпроводной цепи текут токи, имеющие одинаковые амплитуды и противоположную полярность.

Преимущество сбалансированной цепи в том, что она менее чувствительна к внешним возмущениям; наоборот, эта цепь излучает меньшую энергию, поскольку электромагнитные поля отдельных проводников компенсируют друг друга. С другой стороны, сбалансированная линия является таковой только тогда, когда она правильно смонтирована и обслуживается. Несбалансированная цепь значительно более чувствительна к возмущениям, чем сбалансированная, но не требует почти никаких усилий для установки и эксплуатации.

В качестве соединительных проводов в основном используются три типа электрических проводников – простая пара проводов, витая пара и коаксиальный кабель. Соединительные провода измерительной схемы обычно называют сигнальными, в отличие от проводов, использующихся для подвода напряжения питания – силовых проводов.

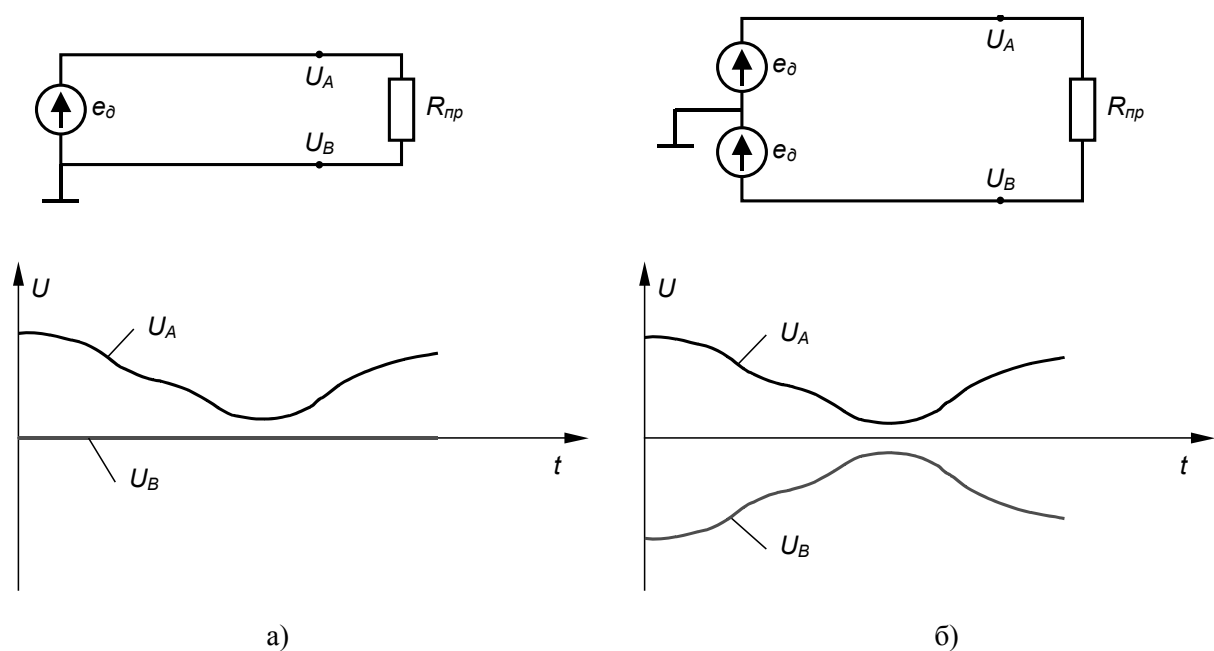


Рис. 1.7. Изменение сигналов в проводниках несбалансированной (а) и сбалансированной (б) цепи

Для организации несбалансированной цепи используется простая пара проводов. Обычно измерительные сигналы от нескольких датчиков, расположенных рядом друг с другом передаются каждый по своей паре проводов, объединённых в один кабель.

Для систем, требующих более высокую точность измерений и защиту от помех, используется витая пара. Витая пара состоит из двух изолированных медных проводов, свитых вместе. Электрическая цепь на основе витой пары может быть сбалансированной и несбалансированной.

Коаксиальные кабели содержат электрический проводник, изолированный пластиковой оболочкой, которая окружена проводящим плетёным экраном. Благодаря своей геометрии коаксиальный кабель существенно несбалансирован. Заземлённый экран предотвращает рассеивание энергии с центрального проводника, которое на высоких частотах является весьма заметным возмущающим воздействием. И наоборот, экран предохраняет центральный проводник от влияния внешних возмущений.

Естественно, коаксиальный кабель не может использоваться для организации сбалансированной цепи. Область его применения – передача быстроизменяющихся сигналов.

Электрические помехи

Каждый провод в соединительных цепях датчика с обрабатывающим электронным устройством является потенциальным приёмником электрических помех. Помехи бывают двух видов – **дифференциальные**, когда ток помехи в сиг-

нальных проводах течёт в разные стороны, и **синфазные**, когда ток помехи в проводах течёт в одну сторону. Также можно сказать, что дифференциальная помеха – это помеха между двумя сигнальными проводами, а синфазная – между сигнальными проводами и землей.

Помехи, возникающие в электрическом проводнике, могут иметь различную природу. Обычно они вызваны одним из следующих типов связи между источником помех и проводником:

- гальванической (резистивной, кондуктивной);
- емкостной;
- индуктивной (магнитной).

Гальваническая связь между проводником и источником помех не зависит от частоты возмущающего сигнала. При емкостной или индуктивной связи степень влияния, напротив, зависит от частоты помех – чем выше частота, тем больше энергии получается от источника возмущений. На практике это означает, что электрические цепи, в которых происходят быстрые изменения тока и/или напряжения, могут быть более серьёзными источниками помех, чем низкочастотные. Вообще говоря, взаимодействие с источником возмущений редко относится к одному типу, обычно – это комбинация всех трёх вышеперечисленных типов. Серьёзные проблемы возникают с маломощными сигналами расположенными вблизи силовых кабелей. Каждый провод в соединительных цепях датчика с обрабатывающим электронным устройством является потенциальным приёмником электрических помех.

Чтобы создать для электронного оборудования среду, максимально свободную от наводок, постоянно проводится множество исследований и разработок. Целью является достижение электромагнитной совместимости в рамках электрических цепей, а также между различными цепями и системами. Электрический прибор должен, с одной стороны, быть нечувствительным к внешним помехам и, с другой стороны, не должен генерировать помех, которые могут оказать влияние на другую аппаратуру.

Гальваническая связь

Когда несколько электронных устройств одновременно имеют общий источник питания и/или общую землю, могут возникать взаимодействия резистивного характера. Довольно часто встречающиеся источники помех – плохо заземлённые электродвигатели и преобразователи частоты с полупроводниковыми вентилями.

Один из способов избежать такого типа взаимодействия – обеспечить для чувствительного электронного оборудования выделенный источник питания. Другая возможность – это гальваническая развязка источников питания и аппара-

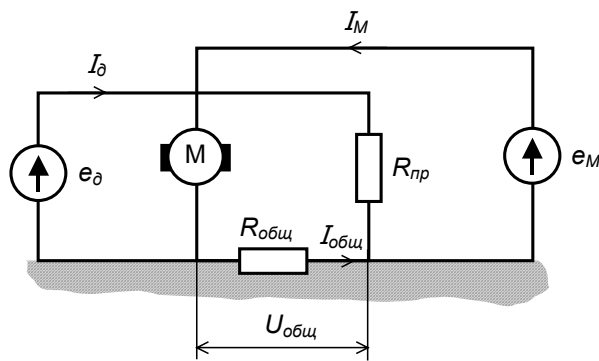
туры. В этом случае прямая электрическая связь между различными источниками питания и электрооборудованием отсутствует.

Источником помех при передаче измерительного сигнала может быть и падение напряжения $U_{общ}$ на участке провода, общем для сигнала и некоторой нагрузки (рис. 1.8, а).

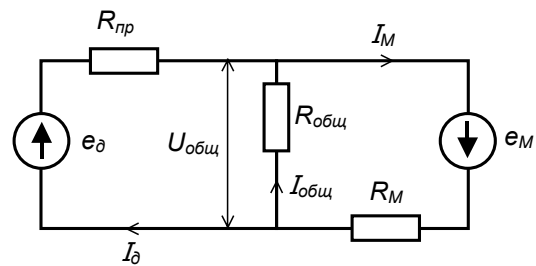
В самом деле, если рассмотреть эквивалентную электрическую схему (рис. 1.8, б), то по второму закону Кирхгофа [7], падение напряжения на измерительном приборе U_{np} определяется выражением

$$U_{np} = I_{\partial} R_{np} = e_{\partial} + U_{общ} = e_{\partial} + (e_M - I_M R_M). \quad (1.7)$$

Т.е. помимо напряжения, генерируемого датчиком e_{∂} , измерительный прибор будет измерять ещё и падение напряжения на общем участке. Вносимая погрешность будет тем сильнее, чем больше сопротивление общего участка провода. Решением описанной проблемы является подсоединение «земли» источника сигнала к приёмнику отдельным проводом, который не используется ни для каких иных целей [8].



а) Общий участок провода



б) Эквивалентная электрическая схема

Рис. 1.8. Паразитный ток $I_{общ}$, протекающий по общему участку провода «земля», создаёт падение напряжения $U_{общ}$.

Ёмкостная связь

Любые два проводника образуют конденсатор, ёмкость которого зависит от расстояния между ними, диэлектрической проницаемости среды и геометрии проводников. Предположим, что рядом с сигнальным проводом проходит силовой провод с переменным напряжением (рис. 1.9). Тогда напряжение U_{np} , измеряемое прибором будет определяться выражением [8]

$$U_{np} = e_{\partial} + \frac{\omega R_{\text{экв}} C_{\text{экв}}}{\sqrt{1 + (\omega R_{\text{экв}} C_{\text{экв}})^2}} \cdot e_c, \quad (1.8)$$

где $R_{\text{экв}} = R_{\partial} R_{\text{пр}} / (R_{\partial} + R_{\text{пр}})$, $C_{\text{экв}} = C_1 C_2 / (C_1 + C_2)$, ω – частота источника напряжения e_c .

Поскольку сопротивление приёмника должно стремиться к бесконечности по правилам согласования сигнала, ёмкостная наводка через паразитную ёмкость между проводниками определяется величиной внутреннего сопротивления датчика R_{δ} . Из выражения (1.8) видно, что при $R_{\delta}=0$ ёмкостная помеха полностью отсутствует. В действительности сигнальный проводник имеет некоторое активное и индуктивное сопротивление, падение напряжения помехи на котором не позволяет полностью устранить ёмкостную наводку с помощью источника сигнала с низким внутренним сопротивлением.

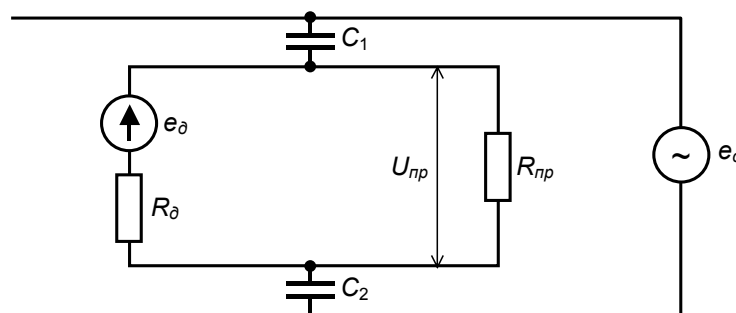


Рис. 1.9. Путь прохождения ёмкостной помехи от источника e_c

Ёмкостные связи должны быть сведены до минимума. Во-первых, они уменьшаются с увеличением расстояния между проводниками. Во-вторых, для снижения ёмкостной связи может быть использован защитный электростатический экран. Экран должен быть заземлён, чтобы его потенциал равнялся нулю. Заземление выполняется в одной точке, лучше всего со стороны источника сигнала. Незаземлённый экран под воздействием внешних возмущений станет одной из пластин конденсатора, что вновь приведёт к появлению ёмкостной связи. Нельзя соединять экран с землёй источника и приёмника одновременно, поскольку при этом через экран течет ток, обусловленный неравенством потенциалов этих земель (до нескольких ампер в цеховых условиях). Ток, протекающий по экрану, является источником индуктивных наводок на соседних проводах и проводах внутри экрана. В общем случае рекомендуется использовать популярную схему гибридного заземления (рис. 1.10). В данной схеме ёмкость C_3 позволяет ослабить высокочастотную составляющую помехи.

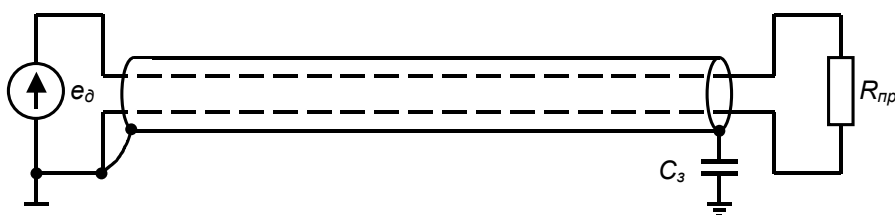


Рис. 1.10. Пример правильного заземления экрана при передаче сигнала от удалённого источника с высоким сопротивлением

Индуктивная (магнитная) связь

Если рядом с сигнальным проводом проходит силовой провод, по которому протекает переменный ток I_c (рис. 1.11), то из-за эффекта электромагнитной индукции в сигнальном проводе будет наводиться напряжение помехи ΔU :

$$\Delta U = \frac{\omega M (R_d + R_{np})}{\sqrt{(R_d + R_{np})^2 + \omega^2 L^2}} \cdot I_c. \quad (1.9)$$

Здесь M – величина взаимной индукции между проводами; L – индуктивность сигнального провода; $\omega = 2\pi f$, f – частота тока помехи.

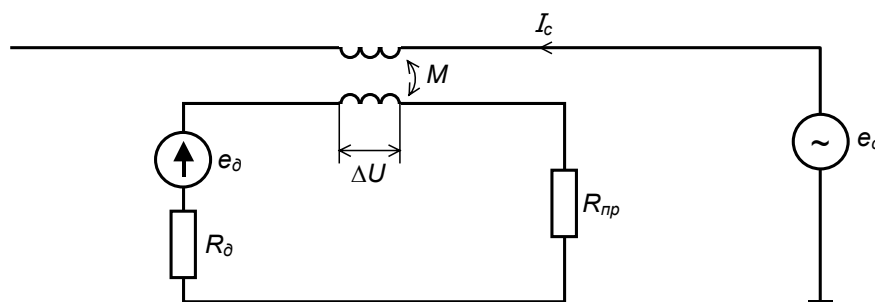


Рис. 1.11. Путь прохождения индуктивной помехи от источника e_c .

Величина взаимной индукции пропорциональна площади контура, который пересекается магнитным полем, созданным током I_c . На рис. 1.11 этот контур образован сигнальными проводами. Для уменьшения взаимной индукции площадь данного контура должна быть минимальной, т.е. сигнальные провода должны быть проложены максимально близко друг к другу. Эффективную площадь контура можно уменьшить, если расположить его в плоскости перпендикулярной плоскости контура с током, наводящим помехи. Площадь контура можно уменьшить, используя витые провода. Более того, при скрутке «изменяется знак» потокоцепления на каждом витке, так что результирующее потокоцепление становится незначительным. Собственно поэтому и применяется кабель на основе витой пары.

Из формулы (1.5) следует, что индуктивная наводка увеличивается с ростом частоты и отсутствует на постоянном токе. Напряжение помехи на рис. 1.11 включено последовательно с источником сигнала, то есть вносит аддитивную погрешность в результат измерения. При бесконечно большом сопротивлении R_{np} напряжение на входе приёмника имеет вид:

$$U_{np} = e_1 + \omega M I_c \quad (1.10)$$

и не зависит от сопротивления источника сигнала.

Проводник, по которому передаётся измерительная информация, должен быть расположен как можно дальше от источников помех. В частности, чувствительные электронные приборы не должны размещаться вблизи трансформаторов

и индукторов. Кабели должны располагаться таким образом, чтобы возможные поля помех распространялись вдоль них. Необходимо следовать двум простым правилам: во-первых, низковольтные сигнальные кабели и высоковольтные силовые кабели не должны прокладываться вблизи друг друга в одних и тех же каналах и, во-вторых, сигнальные и силовые кабели должны пересекаться, если это неизбежно, только под прямым углом.

Магнитное поле можно ослабить экранированием. Медный или алюминиевый экран имеет очень высокую проводимость, и, благодаря возбуждению магнитным полем вихревых токов в экране, магнитный поток ослабляется. Экран можно выполнить из материала с высокой магнитной проницаемостью, например из железа. Магнитный экран часто бывает довольно объёмным, поскольку для демпфирования магнитного потока требуется достаточная толщина стенок. Поэтому экранирование используется в основном для аппаратуры, генерирующей сильные магнитные поля.

Выбор носителя сигнала: напряжение или ток

Выбор носителя сигнала для передачи измерительных данных от датчика к компьютеру зависит от нескольких факторов. Наиболее существенное соображение, которое следует принимать во внимание, – сигнал должен быть по возможности малочувствителен к электрическим возмущениям.

Передача сигнала напряжением

Каждый проводник обладает определённым погонным сопротивлением. Если входной импеданс последнего элемента в цепи – устройства обработки сигнала – не бесконечность, то по кабелю будет протекать ток и в результате произойдёт падение напряжения, что приведёт к искажению измеряемого сигнала. Вдобавок сопротивление проводов будет изменяться в зависимости от температуры окружающей среды.

Поскольку измеряемый сигнал изменяется во времени, то необходимо учитывать искажения, которые могут быть внесены в передаваемый сигнал из-за динамических характеристик сигнальных проводов. Любая пара проводов представляет собой распределённую ёмкость, поэтому эквивалентная схема замещения электрической цепи передачи сигнала выглядит следующим образом (рис. 1.12, а). Изменение напряжения U , измеряемое приёмником сигнала равно падению напряжения на распределённой ёмкости проводов $C_{пр}$:

$$U(p) = I(p) \cdot \frac{1}{pC_{пр}}. \quad (1.11)$$

Ток $I(p)$ в цепи определяется по закону Ома как ЭДС датчика отнесённый к полному сопротивлению проводов,

$$I(p) = \frac{e_d(p)}{Z_{\text{общ}}(p)} = \frac{e_d(p)}{R_{\text{пр}} + \frac{1}{pC_{\text{пр}}}}, \quad (1.12)$$

следовательно, подставив (1.12) в (1.11) для $U(p)$ можем записать

$$U(p) = e_d(p) \cdot \frac{1}{R_{\text{пр}} C_{\text{пр}} p + 1}. \quad (1.13)$$

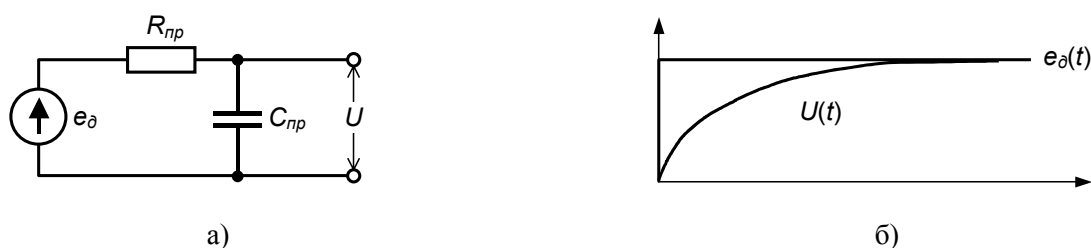


Рис. 1.12. Эквивалентная схема замещения электрической цепи передачи сигнала (а) и переходный процесс в цепи (б)

Выражение (1.13) описывает динамику апериодического звена первого порядка, которое является фильтром низких частот (рис. 1.12, б). Т.е. искажения, вносимые в передаваемый сигнал линией связи, будут тем больше, чем выше частота его изменения.

Требование, чтобы устройства обработки имели высокий входной импеданс, приводит к тому, что они очень чувствительны к помехам, поскольку даже незначительное изменение тока в измерительной цепи приводит заметному изменению падения напряжения на входном сопротивлении приёмника. Следовательно, напряжение не слишком пригодно для передачи данных в случаях, когда могут быть заметные помехи.

Главная причина популярности напряжения для передачи сигналов – это, с одной стороны, присущая этому методу простота, а с другой – широкая доступность устройств для усиления, фильтрации и других видов обработки. Например, если необходимо, чтобы один и тот же сигнал поступил на вход нескольких схем, достаточно соединить эти схемы параллельно (с учётом входного импеданса). Несмотря на это, напряжение не очень часто используется в промышленных системах, поскольку сигналы в них должны передаваться на большие расстояния и влияние источников помех может стать значительным.

Наиболее важные уровни сигналов напряжения стандартизированные Международной Электротехнической Комиссией (МЭК, International Electrotechnical

Commission – IEC) в стандарте МЭК 381: +1 ... +5 В; 0 ... +5 В; 0 ... +10 В; –10 ... +10 В [1].

Передача сигнала током

Для передачи сигнала на значительные расстояния лучше использовать не напряжение, а ток, потому что он остаётся постоянным по длине кабеля, а напряжение падает из-за сопротивления кабеля. На конце кабеля токовый сигнал можно преобразовать в напряжение с помощью высокоточного шунтирующего резистора.

При передаче токовых сигналов выходное напряжение датчика преобразуется операционным усилителем в ток. Приёмник в идеале должен иметь нулевой входной импеданс. В действительности, импеданс определяется шунтом и обычно имеет порядок нескольких сотен Ом. Для тока 20 мА при сопротивлении шунта 250 Ом падение напряжения будет составлять 5 Вольт. Если источник сигнала, т.е. преобразователь напряжения в ток, имеет высокий выходной импеданс, тогда любая помеха при передаче приведёт к небольшому, обычно допустимому падению напряжения на шунте.

Токовые сигналы, как правило, используются на низких частотах до 10 Гц. При постоянном токе и идеальной изоляции сопротивление кабеля не влияет на сигнал, т.е. величина тока на входе приёмника такая же, как на выходе источника сигнала. При переменном токе влияние ёмкостного эффекта становится заметным, и часть тока будет теряться по длине кабеля, уходя либо в обратный провод, либо в заземленный экран. Стандарт МЭК 381 рекомендует для передачи сигналов диапазон токов 4–20 мА. Минимальный уровень сигнала определён как 4 мА, чтобы можно было обнаружить разрыв цепи (0 мА).

Подводя итоги, можно сказать, что измерительная система, использующая ток для передачи сигнала имеет несколько преимуществ:

- удовлетворительно работает на протяжённых коммуникациях;
- допускает простую процедуру проверки, поскольку величина тока 0 мА означает, что датчик отключён или линия разомкнута;
- обеспечивает хорошую защиту от помех.

Вопросы для самоконтроля

1. В чём отличие между физическим и техническим/технологическим процессом?
2. Что такое «режим реального времени»?
3. Перечислите основные компоненты системы управления.

4. На какие классы можно разделить датчики по типу вырабатываемых ими сигналов?
5. Перечислите основные статические и динамические характеристики аналоговых датчиков.
6. Дайте определение термину «эффект нагрузки». В каком случае он может возникнуть при согласовании сигналов?
7. Как должны подбираться импедансы согласуемых устройств при передаче сигнала напряжением? током?
8. Укажите преимущества и недостатки сбалансированной цепи, по сравнению с несбалансированной.
9. Что такое дифференциальная помеха? синфазная?
10. Какой тип связи между источником помех и сигнальными проводами возникает при расположении сигнальных проводов рядом с мощными силовыми установками (электродвигателями)? рядом с высоковольтными проводами?
11. Укажите способы борьбы с гальванической, емкостной и индуктивной связями между источником помех и сигнальным проводником.
12. Перечислите преимущества и недостатки передачи измерительного сигнала током по сравнению с передачей напряжением.

2. АППАРАТНАЯ ОРГАНИЗАЦИЯ УПРАВЛЯЮЩИХ ЭВМ

2.1. ОСНОВНЫЕ ТРЕБОВАНИЯ К АППАРАТНОЙ ОРГАНИЗАЦИИ УПРАВЛЯЮЩИХ ЭВМ

Управляющие компьютеры, применяющиеся для построения систем цифрового управления могут значительно отличаться друг от друга как по конструктивному исполнению, так и по вычислительной мощности. Обычно выбор компьютера обусловлен требованиями технического процесса по количеству подключаемых датчиков и исполнительных механизмов, сложностью управляющего алгоритма, а также требованиями по быстродействию системы. Основные свойства и тенденции развития современных управляющих ЭВМ (УВМ) можно сформулировать следующим образом [9].

1) *Магистрально-модульное исполнение. Модульность* – свойство УВМ, означающее конструктивное выполнение всех узлов УВМ в виде набора функционально законченных автономных модулей, имеющих одинаковый интерфейс с общей *магистралью* или *шиной*, таким образом, подразумевается, что все модули связаны между собой через одинаковые разъёмы с параллельно соединёнными контактами. Каждый контакт и соединяющая их линия имеют определённое назначение и наименование. Временные и электрические параметры сигналов также определены правилами, созданными для конкретного типа шины.

Модульность позволяет легко заменять вышедшие из строя модули, т.е. повышается *ремонтпригодность* системы.

2) *Гибкость* – свойство модульных УВМ легко изменять свою структуру за счёт различного сочетания модулей. Кроме того, под *гибкостью* системы понимается возможность осуществления модернизации УВМ за счёт независимой модернизации каждого из модулей. Облегчается разработка новых систем и увеличивается срок морального старения за счёт применения комбинации новых и уже существующих модулей, повышаются регулярность структуры и, как следствие, контрольно-диагностические качества, обеспечивается простота переориентации УВМ на другой объект управления.

3) *Наращиваемость (масштабируемость)* – свойство, заключающееся в возможности простого увеличения или уменьшения количества модулей УВМ.

4) *Многофункциональность* – свойство УВМ, означающее, что она не имеет жёсткой специализации и привязки к объекту управления и, соответственно, может выполнять разные функции.

5) *«Интеллектуальность»* – свойство большинства модулей современных УВМ, связанное с наличием в них микроконтроллеров или микропроцессоров, работающих по своей индивидуальной программе, уровень интеллектуальности

можно легко увеличивать в уже работающей УВМ путём модификации программного обеспечения модулей.

б) *Конструктивное единство* – понятие, означающее конструктивное единоеобразное исполнение всех модулей УВМ.

Развитие магистрально-модульных УВМ предполагает обеспечение принципа *максимальной универсализации*, состоящего в том, что каждый вводимый в состав УВМ модуль должен обеспечивать построение системы, удовлетворяющей требованиям максимального числа применений.

Кроме того, очевидно выполнение принципа *асинхронности и минимальности количества сигналов управления*, упрощающего протокол обмена между модулями, повышающего надёжность связи и быстродействие обмена.

2.2. ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ

В большинстве случаев, при построении автоматизированных систем управления технологическими процессами, используются *программируемые логические контроллеры* (ПЛК, PLC – Programmable Logic Controller).

Первоначально термином ПЛК обозначали специальные УВМ, предназначенные для выполнения операций переключения в промышленных условиях. В настоящее время это название не соответствует действительности, т.к. ПЛК сегодня могут гораздо больше, чем просто выполнять логические операции. Однако эта аббревиатура сохранена, чтобы избежать путаницы с более общими терминами (программируемый контроллер – ПК, персональный компьютер – ПК).

Конструктивно ПЛК обычно приспособлены для работы в типовых промышленных условиях, с учётом уровней сигналов, термо- и влагостойкости, ненадёжности источников питания, механических ударов и вибраций. Кроме того, термин ПЛК отображает ещё одну особенность конструктивного исполнения такой УВМ – это отсутствие интерфейса с оператором. Т.е. в состав типового ПЛК не входят ни клавиатура, ни мышь, ни монитор – ничего из тех устройств, которые используются оператором для взаимодействия с обычным компьютером. Конечно, в случае необходимости ПЛК может быть оснащён средствами отображения информации, однако его основная задача – обмен информацией с техническим процессом, а не с человеком.

Типичный программируемый логический контроллер состоит из следующих частей:

- промышленное шасси или крейт (от англ. crate – корзина);
- модуль центрального процессора;
- модули ввода-вывода данных.

Промышленное шасси представляет собой сборный конструктив (рис. 2.1), включающий защитный кожух, пассивную объединительную плату (кросс-плата, backplane), вентиляторы, фильтры, источник питания.

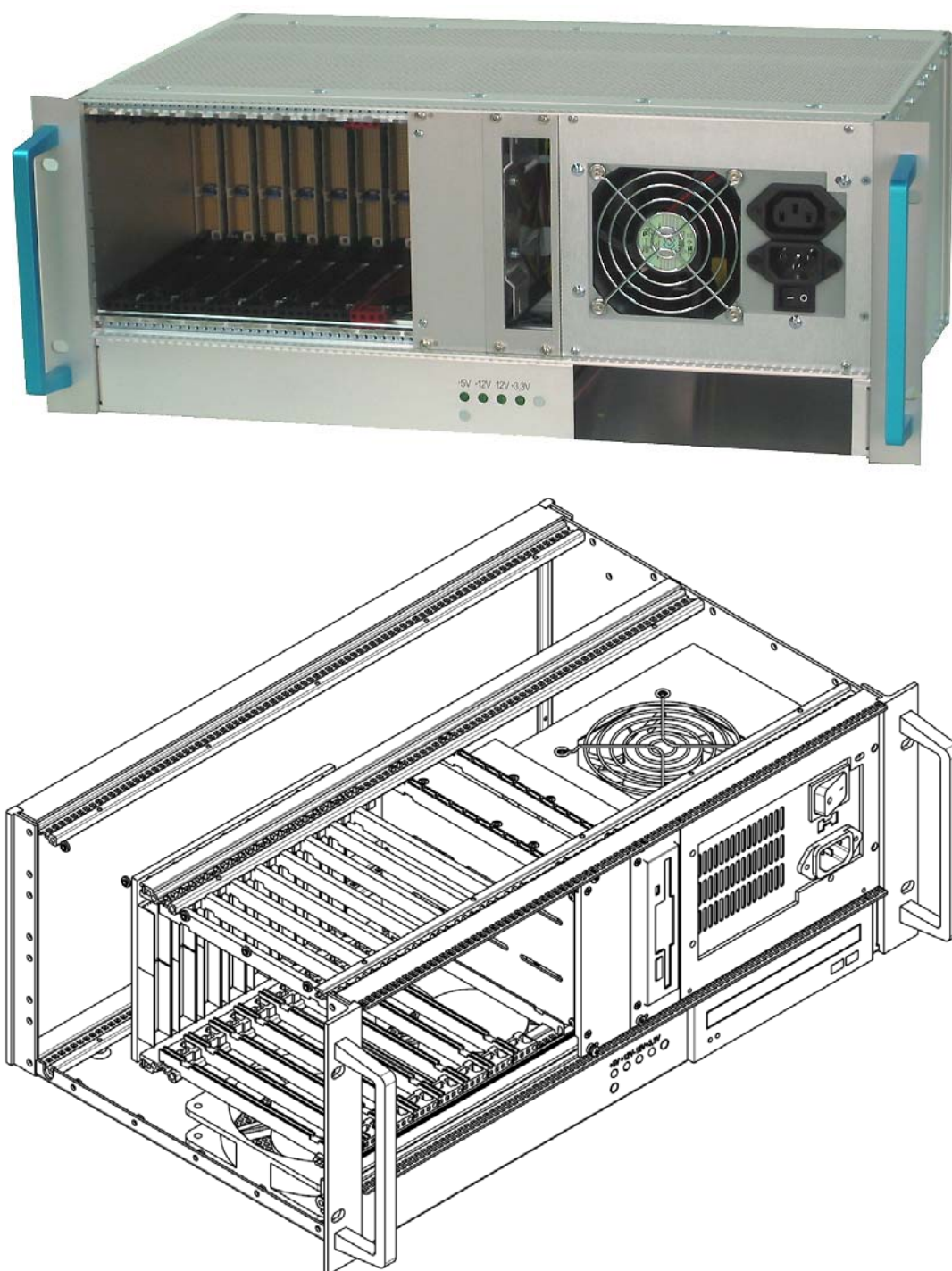


Рис. 2.1. Промышленное шасси стандарта CompactPCI

Защитный кожух изготавливается из прочной стали, возможно, оцинкованной, и алюминиевых элементов. Он обеспечивает защиту компьютера от пыли, вибрации, влаги, ударов, перегрева, электромагнитных помех и позволяет размещать компьютер в сложных цеховых условиях. В шасси устанавливаются и фикс-

сируются электронные компоненты компьютера. Пассивная объединительная плата, которая содержит несколько разъёмов служит для электрического объединения электронных компонентов. Тип разъёмов определяется стандартом шины.

Модуль центрального процессора (одноплатный компьютер), содержит процессор и оперативную память, контроллер шины и контроллер интерфейса для программирования. В некоторых случаях здесь же может размещаться и контроллер сети.

Модуль центрального процессора устанавливается в один из разъёмов объединительной платы, а остальные заполняются *модулями ввода-вывода данных*.

Пользователь может выбрать наиболее подходящее шасси, отвечающее лучшим образом требованиям по количеству посадочных мест, мощности источника питания и т.д.

В том случае, когда каждый модуль выполнен в корпусе с соответствующей защитой, промышленное шасси может вырождаться до рейки или стойки (rack – стойка), иногда выполняющей функции объединительной платы. В этом случае в состав контроллера должен входить и модуль источника питания.

В качестве системного интерфейса ПЛК используются либо открытые стандарты шин (VME, ISA, CompactPCI), либо собственные стандарты фирм-изготовителей контроллеров (закрытые стандарты). Естественно, собственные стандарты могут позволить себе только крупные фирмы, которые в состоянии поставлять на рынок весь спектр модулей расширения для своей шины, а также средства программирования таких контроллеров. К таким компаниям относятся Siemens, Schneider Electric, ABB, Mitsubishi.

2.3. КОНСТРУКТИВ «ЕВРОМЕХАНИКА»

Любой стандарт системной шины должен описывать не только электрические характеристики и назначения сигналов, не только протокол обмена по шине, но и конструктивное исполнение устройств, подключаемых к шине. Сюда относятся типы разъёмов, размеры подключаемых устройств, расстояние или шаг между разъёмами.

В настоящее время два наиболее популярных открытых стандарта системных магистралей VME и CompactPCI используют для описания конструктивного исполнения модулей стандарт «Евромеханика» (Eurocard) [10]. Это международный стандарт на типоразмеры и конструктивы печатных плат, модулей, субблоков, блоков и 19-дюймовых (19”) шкафов и стоек (рис. 2.2).

В начале 1960-х годов стандарт Евромеханики или 19” стандарт МЭК 60297 разрабатывался на базе стандарта США ASA C 83.9 как нормативный документ, определявший размеры передних панелей блоков (прежде всего источников пита-

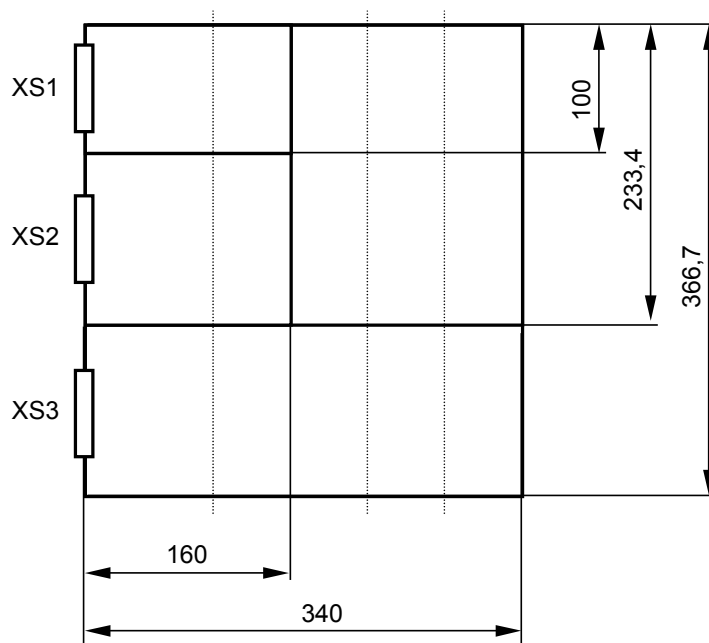
ния) для установки в шкафы [11, 12]. В 1980 году стандарт был дополнен спецификациями, определившими типоразмеры модулей и печатных плат для установки в монтажные субблоки (крейты), а в 1983 году принял законченный вид, определив и габаритные размеры шкафов. Естественно, в следующие 20 лет стандарт не оставался статичным, а постоянно совершенствовался, дополняясь спецификациями, призванными обеспечить оптимальное использование определяемых стандартом конструктивов с шинами, созданными на базе новых стандартов и находившими применение в промышленной, военной сфере, на транспорте и в области телекоммуникаций. Но основные параметры конструкции были определены ещё тогда.



Рис. 2.2. Стойки и шкафы для размещения оборудования в стандарте 19"

Базовым параметром в евроконструктивах является уже упоминавшийся размер 19 дюймов, который равен 482,6 мм (один дюйм равен 25,4 мм) – это ширина передней панели субблока. Высота субблоков определяется в единицах U (Unit, модуль), каждая из которых равна 1,75 дюйма, или 44,45 мм. Субблоки могут быть разной высоты, но рекомендуются к применению субблоки под модули высотой 3U и 6U. Соответственно, стандартные размеры модуля для установки в субблок высотой 3U составляют 100 мм в высоту и 160 мм в глубину. Глубина модуля может быть и больше, увеличиваясь с шагом 60 мм (рис. 2.3). Высота модуля может изменяться с шагом 1U, и для типоразмера 6U её габариты составят

233,35×160 мм или 233,35×340 мм. Возможно использование модулей для типоразмера 9U, с габаритами 366,7×340 мм.



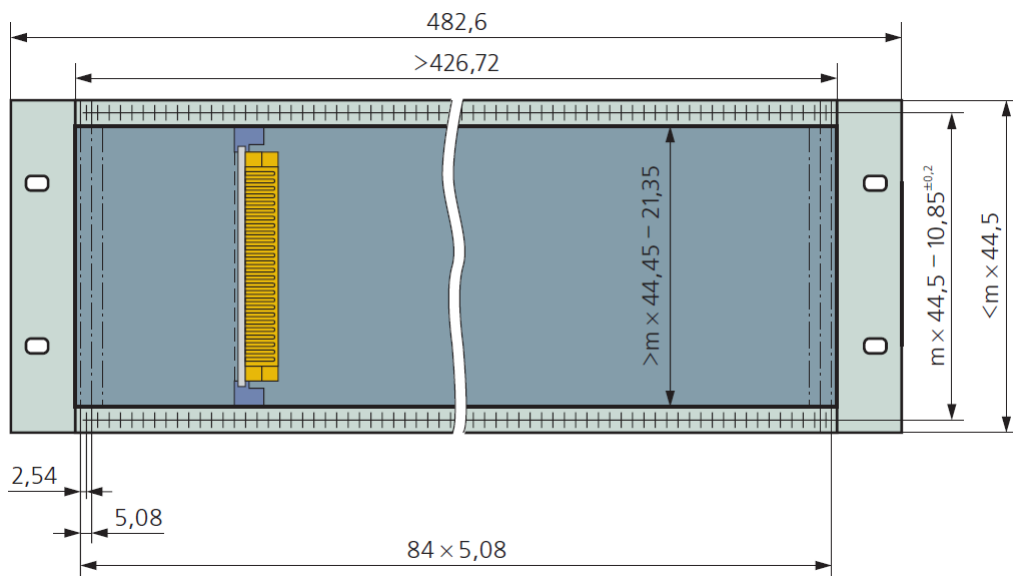
XS1, XS2, XS3 – разъёмы для подключения к кросс-плате

Рис. 2.3. Типоразмеры модулей, устанавливаемых в субблоки

Субблоки представляют из себя монтажные каркасы, состоящие из двух боковых панелей с монтажными фланцами и минимум четырёх поперечных несущих монтажных рельсов, к которым с шагом 5,08 мм (0,2 дюйма, или 1НР) крепятся направляющие (рис. 2.1, 2.4). Всего предусмотрено до 84 посадочных мест для направляющих, которые, в принципе, могут устанавливаться в произвольном порядке, однако минимальное расстояние между двумя соседними направляющими составляет 3НР, а в большинстве современных шинных стандартов – и вообще 4НР, или 20,32 мм. Таким образом, в стандартный 19” субблок можно установить до 21 модуля. Во встраиваемых системах часто используются субблоки меньшей ширины; обычно в этих случаях выбираются 28НР, 42НР или 63НР, хотя встречаются и другие размеры.

Модули, устанавливаемые в субблок представляют собой печатные платы с некоторым внешним оформлением – передней панелью и, возможно, кожухом (рис. 2.5). Модули устанавливаются в каркасы субблоков и, с помощью разъёмов, установленных на задней стороне их плат (рис. 2.3), соединяются с кросс-платой [10]. Поскольку разъёмы XS2 и XS3 присутствуют только в больших форматах, магистральная шина, объединяющая модули, использует разъём XS1. Шина может использовать не все контакты разъёма, часть контактов выделяется под про-

извольное использование конкретным модулем в конкретном устройстве. Таким образом, к модулю могут подводиться внешние цепи.



m – натуральное число из допустимого ряда

Рис. 2.4. Размеры субблоков

Для подключения внешних цепей может использоваться и лицевая панель модуля, тогда на ней устанавливают разъёмы и/или клеммные колодки.



Рис. 2.5. Модуль сетевого адаптера для установки в субблок высотой 3U

С точки зрения охлаждения идеальной является композиция, в которой и модули, и задняя панель (кросс-плата) располагаются в вертикальных плоскостях.

При небольшом числе модулей применяют и их горизонтальное расположение. Часто конструкция каркаса допускает установку модулей разной высоты, что делает конструктив более гибким.

Все модули и платы устанавливаются в направляющие полозья и имеют надёжную систему фиксации с обеих сторон – и сверху, и снизу (см. рис. 2.1). Модули вынимаются с помощью специальных механических экстракторов (рис. 2.5).

В качестве коннекторов, соединяющих модули с кросс-платой, обычно применяются разъёмы со штырьковыми контактами, у которых все параметры значительно лучше, чем у печатных ламельных разъёмов карт ПК.

Основательность системы крепления и присоединения модулей и позволяет использовать данный конструктив «Евромеханика» в жестких условиях эксплуатации. Степень защищенности блока с модулями можно варьировать в зависимости от условий, помещая его в пыле-, влаго-, брызго- (и т.п.) защищённый корпус, применяя, при необходимости, и кондиционирование воздуха.

2.4. ОТКРЫТЫЕ СТАНДАРТЫ СИСТЕМНЫХ МАГИСТРАЛЕЙ

Стандарт VME

Магистрально-модульный стандарт VME (VERSAmodule Eurocard) разработан и поддерживается международной ассоциацией VITA (VME International Trade Association) [12, 13].

VME – асинхронная 32-хразрядная шина с отдельными 32 линиями адреса и 32 линиями данных на кросс-плате [14]. Шина VME имеет 64-разрядное расширение (VME64).

Конструктивно в основу шины VME положен стандарт Евромеханика. Конечная система комплектуется из функциональных модулей VME, устанавливаемых в крейты, число которых не ограничено. В каждый крейт можно поместить до 21 модуля VME. Модули соединяются через объединительную плату с нормированным волновым сопротивлением и терминаторами на каждой сигнальной линии. В качестве соединителей используются 96-штырьковые разъёмы DIN602-3, причём восьми- и 16-тиразрядные модули имеют один разъём, 32/64-разрядные – два. Высота платы VME с одним разъёмом составляет 3U (100х160 мм), с двумя разъёмами – 6U (233х160 мм) (рис. 2.6).

В отличие от других 32-разрядных шин стандарт VME не предусматривает мультиплексирования линий адресов и данных и, следовательно, требует применения относительно большего числа информационных сигналов интерфейсных формирователей/приёмников и наличия большего количества контактов в используемых разъёмах. Для реализации 16-разрядной шины VME достаточно «старше-

го» разъёма XS1, показанного в верхней части рис. 2.6, а для организации 32-разрядной шины необходимо использовать оба показанных на рисунке разъёма XS1 и XS2. При работе с 16-ти разрядной шиной «младший» разъём XS2 может служить для организации 32-разрядной шины локального расширения VMX или для ввода-вывода. Одним из преимуществ немultipлексируемых шин является возможность конвейеризации операций чтения и записи в одном цикле; при использовании multipлексируемой шины для этого затрачивается два шинных цикла.

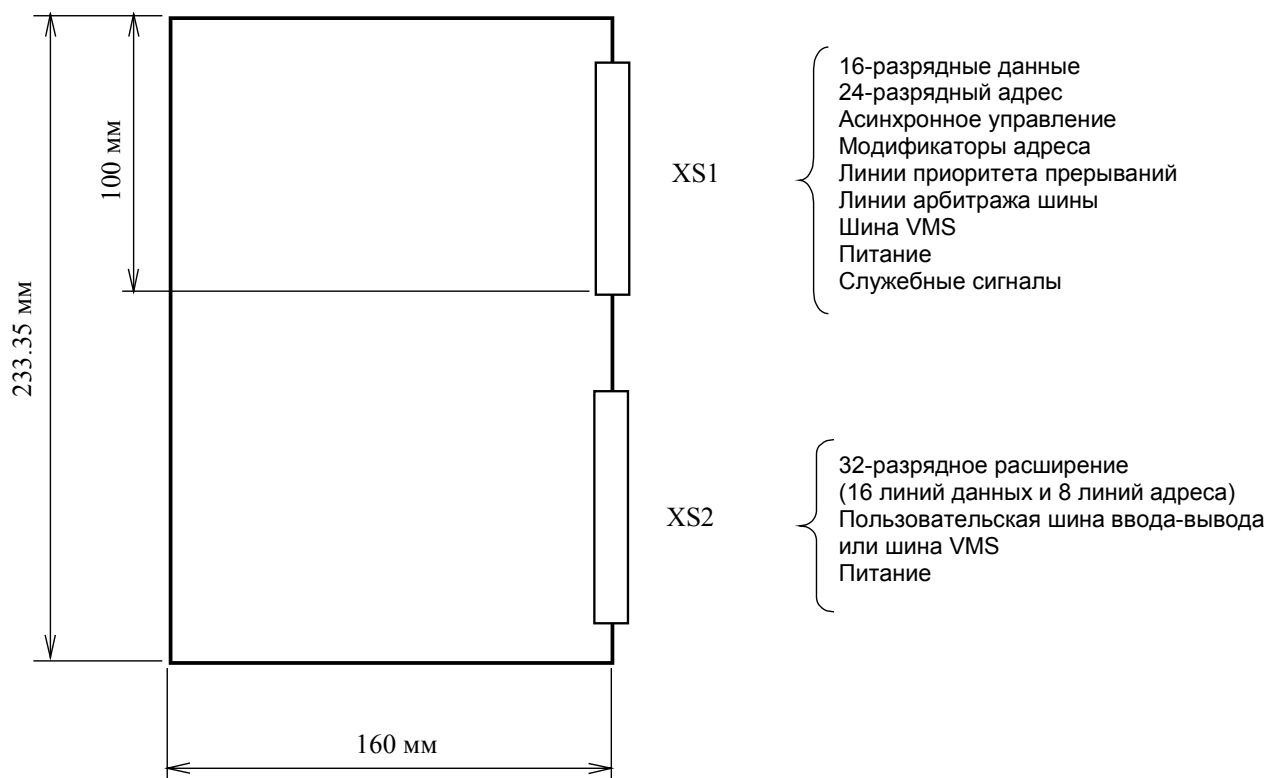


Рис. 2.6. Назначение разъёмов платы в соответствии со стандартом VME

В шине VME64 разъёмы расширили до 160 контактов – по обеим сторонам коннектора XS1 и XS2 ввели ряды дополнительных контактов для сигнальной «земли» и произвольного использования [10]. Конструкция этих дополнений позволяет устанавливать модули VME (96 контактов) в шину VME64 (160 контактов) и наоборот, правда при этом дополнительные ряды контактов не используются. Кроме того, между разъёмами XS1 и XS2 определён необязательный 95-тиконтактный разъём XS0. В этой шине возможно применение географической адресации модулей (по позиции в крейте). Введены дополнительные шины питания +3,3 В и +48 В с допустимым диапазоном от 38 В до 75 В, так что питание +48 и +60 В, популярное в телекоммуникациях, укладывается в этот диапазон. Шина обеспечивает возможность горячей замены модулей – имеются контакты с опережающим соединением и запаздывающим разъединением. Пропускная спо-

способность шины составляет до 160 Мбайт/с. Прорабатывается и вариант шины VME320 с пропускной способностью 320 Мбайт/с.

Системная шина VME реализует асинхронный обмен информации. В асинхронных системах для задания моментов пересылок не используются какие бы то ни было периодические сигналы, поэтому при передаче информации по асинхронным шинам скорость передачи не обязательно должна быть кратна частоте тактовых сигналов, как это имеет место при использовании синхронных шин. Кроме того, асинхронные шины допускают применение дешёвых плат, на которых размещены устройства с большими временами ответа, совместно с платами, содержащими быстродействующие устройства.

Системная шина VME содержит четыре основных элемента:

- тракт пересылки данных (8, 16 или 32 разряда параллельно);
- линии арбитража (4 уровня);
- линии обработки прерываний (7 уровней);
- сервисные линии.

Схема арбитража, применяемая в системной шине VME, является централизованной в той степени, что одна плата действует как глобальный арбитр, использующий четыре линии запроса шины. Плата может быть организована таким образом, чтобы реализовывать процедуру «освобождение по завершении – запросу» или процедуру «освобождение по запросу».

В процессорных платах обычно используется процедура «освобождение по запросу», позволяющая процессору крепко удерживать управление шиной и передавать его, только когда запрос на использование шины поступает от другого ведущего процессора. В контроллерах прямого доступа к памяти, как правило, применяется процедура «освобождение по завершении», потому что они обычно пользуются шиной лишь в течение коротких интервалов времени, после чего перестают управлять шиной.

Стандарт CompactPCI

Магистрально-модульный стандарт CompactPCI разработан и поддерживается международной Ассоциацией PICMG (PCI Industrial Computers Manufacture Group) [15].

CompactPCI (cPCI) – это стандарт, описывающий современную высокопроизводительную магистрально-модульную систему (конструктив, электрические и логические параметры) для промышленных, телекоммуникационных и графических встраиваемых компьютерных приложений повышенной надёжности [10]. В основе электрических и логических параметров стандарта лежат спецификации

самого популярного стандарта «офисных» компьютеров – PCI [16], механическая основа – Еврокарты формата 3U (100x160 мм) и 6U (233x160 мм).

Шина CompactPCI разрабатывалась на основе спецификации PCI 2.1. От обычной PCI эта шина отличается большим количеством поддерживаемых слотов для одной шины: восемь против четырёх. В связи с этим появились новые 4 пары сигналов запросов и предоставления управления шиной. Шина поддерживает 32-битные и 64-битные обмены (с индивидуальным разрешением байт). В отличие от обычных PCI модулей для настольных офисных компьютеров, в CompactPCI-модулях в качестве шинных разъёмов используются высоконадёжные пятирядные 235-контактные разъёмы с шагом 2 мм (рис. 2.7), соответствующие стандартам МЭК и Belcore (МЭК 1076-4-101).

CompactPCI-контроллеры на базе микропроцессоров Pentium могут управляться любой операционной системой, работающей на обычном персональном компьютере: MS-DOS, Windows 3.xx, Windows 98, Windows 2000, Windows XP, OS/2, BSD UNIX, LINUX, Solaris... CompactPCI-контроллеры на базе микропроцессоров PowerPC могут работать под управлением операционных систем AIX, Solaris, MacOS и самого широкого спектра операционных систем реального времени от ведущих мировых поставщиков. То же самое можно сказать о CompactPCI процессорах Alpha. Любая операционная система, работающая на этих процессорах для настольных офисных компьютеров будет без модернизации работать в CompactPCI.

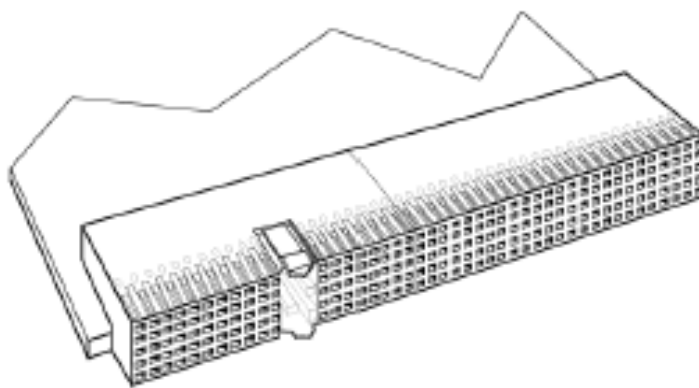


Рис. 2.7. Разъём CompactPCI

CompactPCI – 32/64 разрядная шина с мультиплексируемыми линиями адрес/данные на объединительной панели. Шина CompactPCI синхронная, с тактовой частотой 33 МГц. Арбитраж шины – централизованный, одноуровневый полностью унаследован от стандарта PCI 2.1 [17].

Сравнительные характеристики стандартов VME и CompactPCI

В табл. 2.1 приведены для сравнения основные характеристики обеих шин. Среди главных достоинств стандарта VME – асинхронный обмен информацией и большее количество модулей, устанавливаемых в один крэйт. CompactPCI позволяет установить лишь восемь модулей при использовании пассивной объединительной панели. Использование моста PCI-to-PCI позволяет увеличить количество модулей в крейте до 16-ти, однако приводит к усложнению конструкции кросс-платы, которая перестаёт быть пассивным устройством, поскольку именно на ней размещается мост.

К достоинствам CompactPCI следует отнести полную совместимость по программному обеспечению с офисными компьютерами, что значительно снижает время на переподготовку обслуживающего персонала.

Таблица 2.1.

Основные характеристики стандартов VME и CompactPCI

	VME	CompactPCI
Разрядность шины	32/64	32/64
Адрес/Данные	VME32 – раздельные VME64 –мультиплексируемые	мультиплексируемые
Тип шины	асинхронная	синхронная, 33 МГц
Конструктив	Евромеханика 3U, 6U, 9U	Евромеханика 3U, 6U
Расстояние между модулями в крейте	4HP 0,8” (20,32 мм)	4HP 0,8” (20,32 мм)
Максимальное количество модулей в крейте	21	8/16
Пропускная способность	VME32 – 40 Мбайт/с VME64 – 80 Мбайт/с (до 320 Мбайт/с в режиме пакетных передач)	базовый вариант – 132 Мбайт/с 64-разрядные передачи – 264 Мбайт/с

2.5. ОСНОВНЫЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ РАБОТЫ ПЛК

Разработка программного обеспечения ПЛК

Программирование ПЛК осуществляется на этапе разработки и внедрения системы АСУ ТП. Тем не менее, необходимость изменения программы в ПЛК возникает регулярно иногда и непредвиденно. Поэтому, написана она должна быть так, чтобы любой человек, а не только её автор мог в ней быстро разобраться и оперативно внести необходимые доработки.

Говорить о том, что программы написаны для ПЛК, не вполне корректно [18]. Все программы написаны человеком и предназначены для чтения человеком. Любые инструменты программирования дают в конечном итоге микропроцессору инструкции в его машинных кодах. Для него нет разницы, на каком языке написана программа. То есть, говоря о различных характеристиках языков программирования, имеют в виду их способность эффективно выражать мысли программиста.

В семидесятых годах прошлого века появилось множество языков для программирования ПЛК, ориентированных в первую очередь не на программистов, а на специалистов-технологов. В конце семидесятых годов сложилась крайне сложная ситуация. Каждый изготовитель ПЛК разрабатывал собственный язык программирования, поэтому ПЛК разных производителей были программно несовместимы, кроме того, существовала проблема аппаратной несовместимости. Замена ПЛК на продукт другого изготовителя превратилась в огромную проблему. Покупатель ПЛК был вынужден использовать изделия только одной фирмы либо тратить силы на изучение разных языков и средства на приобретение соответствующих инструментов.

В итоге в 1979 году в рамках Международной Электротехнической Комиссии (МЭК) была создана специальная группа технических экспертов по проблемам ПЛК. Ей была поставлена задача – выработать стандартные требования к аппаратным средствам, программному обеспечению, правилам монтажа, тестирования, документирования и средствам связи ПЛК. В 1982 году был опубликован первый черновой вариант стандарта, который получил наименование МЭК 1131. Ввиду сложности получившегося документа, было решено разбить его на несколько частей. После перехода в 1997 году на пяти цифровые обозначения, в настоящее время выделяют следующие части стандарта МЭК 61131 [19]:

- МЭК 61131-1. Программируемые контроллеры. Часть первая. Общая информация. В Российской Федерации на базе этого стандарта разработан ГОСТ Р 51840–2001.
- МЭК 61131-2. Программируемые контроллеры. Часть вторая. Требования к оборудованию и испытаниям (ГОСТ Р 51841–2001).
- МЭК 61131-3. Программируемые контроллеры. Часть третья. Языки программирования.
- МЭК 61131-4. Программируемые контроллеры. Часть четвёртая. Руководства для пользователя.
- МЭК 61131-5. Программируемые контроллеры. Часть пятая. Технические характеристики средств обмена сообщениями.

- МЭК 61131-7. Программируемые контроллеры. Часть седьмая. Программирование нечёткого управления.
- МЭК 61131-8. Программируемые контроллеры. Часть восьмая. Руководства по применению и реализации языков программирования.

Шестая часть, МЭК 61131-6, зарезервирована для возможного использования в будущем.

Вопросам программирования посвящена третья часть стандарта «Языки программирования ПЛК». Рабочей группой МЭК было принято достаточно оригинальное решение. Из всего многообразия существовавших на момент разработки стандарта языков программирования ПЛК были выделены пять языков, получивших наибольшее распространение. Спецификации языков были доработаны, так что стало возможным использовать в программах написанных на любом из этих языков стандартизованный набор элементов и типов данных. Такой подход МЭК не раз подвергался критике, но время доказало правильность этого решения. Реализация подобного подхода позволила привлечь к программированию одного и того же ПЛК специалистов различных областей знаний (и что особенно важно – различной квалификации): специалистов по релейной автоматике (и даже электриков), программирующих в LD, специалистов в области полупроводниковой схемотехники и автоматического регулирования для которых привычен язык FBD, программистов, имеющих опыт написания программ для компьютеров на языке ассемблера (ему соответствует язык IL для ПЛК), на языках высокого уровня (язык ST), даже далекие от программирования специалисты-технологи получили свой инструмент программирования – язык SFC. Хотя внедрение МЭК систем программирования и не позволило полностью отказаться от услуг профессиональных программистов (впрочем такая цель и не ставилась), но зато позволило снизить требования к квалификации и соответственно затраты на оплату труда программистов ПЛК.

Стандартизация языков позволила (по крайней мере, частично) решить проблему зависимости пользователя ПЛК от конкретного изготовителя. Все современные ПЛК оснащаются средствами МЭК 61131-3 программирования, что упрощает работу пользователям контроллеров (можно использовать ПЛК различных фирм без затрат на переучивание) и одновременно снимает ряд проблем для изготовителей ПЛК (можно использовать компоненты ПЛК других изготовителей). Следует отметить, что некоторые старейшие изготовители ПЛК до сих пор вынуждены поддерживать свои собственные языки (системы программирования), однако все они, в той ли иной форме, стремятся обеспечить поддержку МЭК 61131-3. Стандарт существенно расширил возможности на рынке труда специалиста, занимающегося программированием ПЛК. Подобно тому, как автомеханик, имею-

щий стандартный набор инструментов, может браться за ремонт любого узла (кроме нестандартных) машины любой фирмы, так и специалист, изучивший языки МЭК 61131-3 сможет разобраться с программой любого современного ПЛК. Это позволило уменьшить как зависимость фирмы от специалиста по программированию ПЛК, так и специалиста от фирмы.

Еще одним важным положительным результатом стандартизации языков явилась возможность специализации изготовителей ПЛК как на производстве аппаратных средств ПЛК, так и на производстве средств программирования ПЛК. Результаты такой специализации хорошо видны на примере индустрии персональных компьютеров: существуют компании выпускающие высококлассные аппаратные средства, они умеют делать это лучше других и не испытывают необходимости выпускать программное обеспечение, в то же время на рынке программных средств есть свои лидеры, вооруженные опытом и имеющимися у них технологиями. Благодаря стандартизации обеспечивается совместимость и в результате пользователь может свободно выбрать лучшие продукты как из аппаратных, так и из программных средств.

Принципы функционирования ПЛК

Естественно, при программировании контроллеров программист должен опираться на определённую функциональную структуру ПЛК (с учётом конкретной конфигурации) и принципы организации его работы. Требования к функциональности ПЛК определены в первой части стандарта МЭК 61131 [20].

Далее рассматриваются основные свойства ПЛК, которые должны учитываться пользователем при разработке программного обеспечения ПЛК.

Области памяти ПЛК

В большинстве ПЛК выделяют три типа памяти:

- **загрузочная память** используется для хранения программы и начальных значений данных. Загрузочная память чаще всего выполняется энергонезависимой (РПЗУ, EEPROM, FLASH). Часто загрузочная память выполняется в виде отдельной карты памяти, объём которой зависит от конкретных потребностей пользователя;
- **рабочая память** используется для размещения рабочей программы и данных во время исполнения программы;
- **системная память** содержит такие элементы памяти как области отображения входов и выходов технологического процесса (образа технологического процесса), внутренняя память, доступная программисту, в т.ч. память с побитовым доступом (флаги), таймеры и счётчики.

Если проводить аналогию с обычной ЭВМ, то загрузочная память соответствует жёсткому диску, рабочая память – оперативной памяти ЭВМ. А вот системная память – это особенность программируемых контроллеров. Её появление неразрывно связано с **рабочим циклом ПЛК**.

Рабочий цикл ПЛК

Обмен данными между ПЛК и управляемым технологическим процессом (датчиками и исполнительными механизмами) является неотъемлемой частью АСУ ТП. Поэтому в большинстве ПЛК операция опроса состояния входов и установки нужных значений на выходах реализуется системным программным обеспечением ПЛК и не требует никаких усилий со стороны программиста. Такие программируемые контроллеры работают по методу *периодического опроса* входных данных. Именно на такую модель опирается стандарт МЭК 61131-3.

Рабочий цикл ПЛК включает четыре фазы [18]:

- опрос входов;
- выполнение пользовательской программы;
- установку значений выходов;
- некоторые вспомогательные операции (диагностика, подготовка данных для отладчика, сетевой обмен и т.д).

Прикладная программа имеет дело с одномоментной копией значений входов. Внутри одного цикла выполнения программы, значения входов можно считать константами. Такая модель упрощает анализ и программирование сложных алгоритмов. Очевидно, что время реакции ПЛК на изменение значения на входе будет зависеть от времени выполнения одной итерации прикладной программы – одного цикла ПЛК.

Такая организация работы ПЛК обеспечивает:

- неизменность входных переменных в течение рабочего цикла;
- гарантированное время постоянства значений на выходных каналах контроллера (изменение не чаще чем один раз в рабочий цикл);
- возможность оценивать и управлять длительностью рабочего цикла всего приложения.

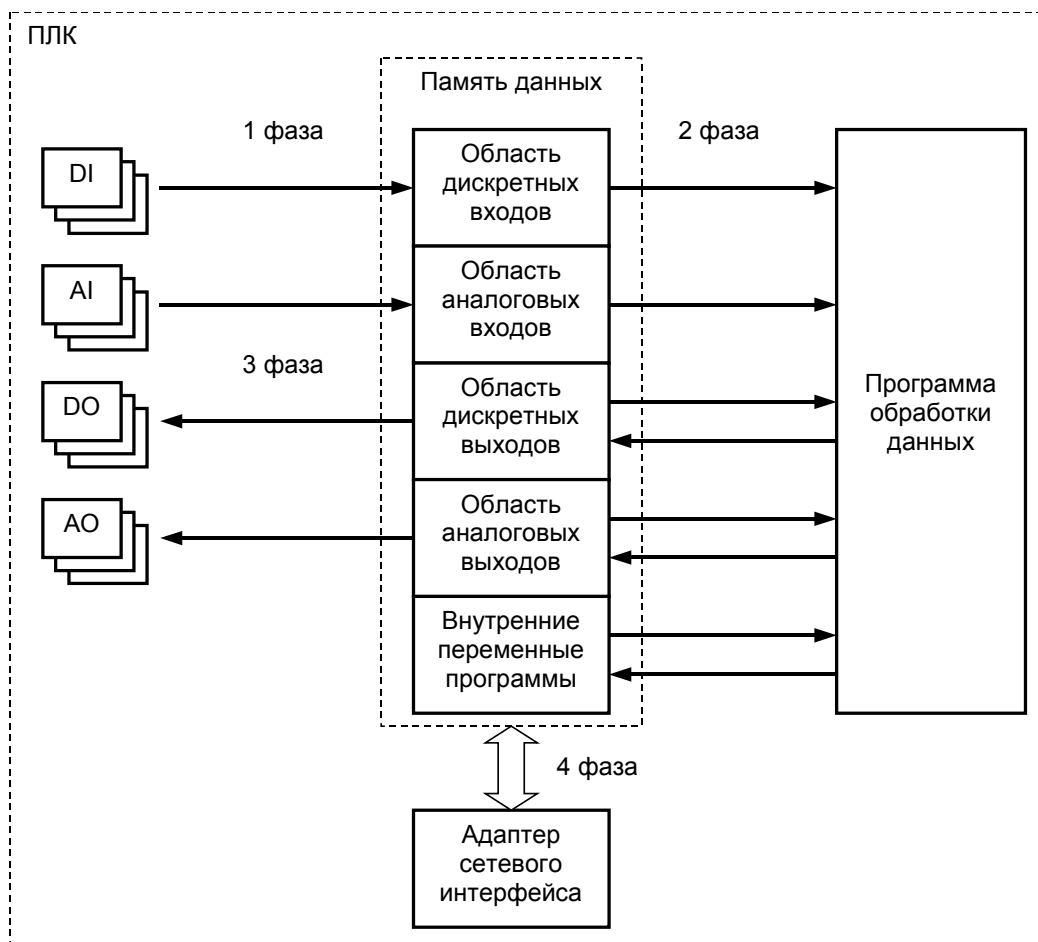
Для доступа программиста к значениям на входах и выходах ПЛК выделяются специальные области памяти. Чаще всего, это либо две области: область памяти входов и область памяти выходов, либо четыре: области памяти дискретных и аналоговых входов и области памяти дискретных и аналоговых выходов. Тогда рабочий цикл ПЛК можно представить следующим образом (рис. 2.8).

Входные дискретные и аналоговые сигналы обрабатываются модулями дискретного ввода (Discrete Input, DI) и аналогового ввода (Analog Input, AI) и в виде двоичного кода записываются в области памяти входов (первая фаза).

Во время исполнения программы (вторая фаза) используются данные из всех областей памяти. В результате исполнения программы рассчитываются новые значения внутренних переменных, а также значения, которые должны быть установлены на выходах контроллера. Рассчитанные значения размещаются в соответствующих областях памяти (см. рис. 2.8).

Во время третьей фазы цифровые значения из областей выходов пересылаются в модули дискретного вывода (Discrete Output, DO) и аналогового вывода (Analog Output, AO) соответственно, модули преобразуют их в соответствующие значения тока или напряжения на своих выходах.

Четвёртая фаза используется, например, для обмена данными по сети.



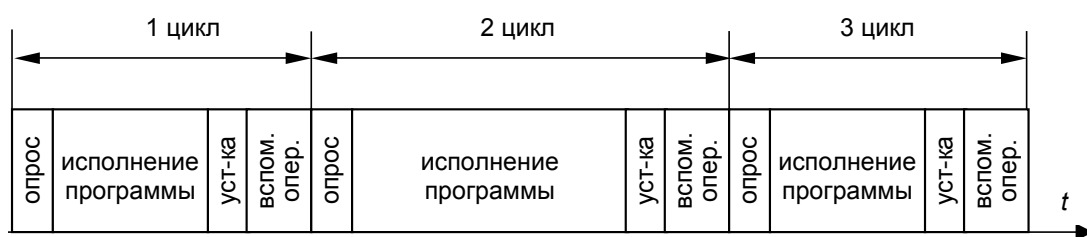
DI – модули дискретного ввода; AI – модули аналогового ввода;
DO – модули дискретного вывода; AO – модули аналогового вывода

Рис. 2.8. Информационные потоки в структуре ПЛК и фазы рабочего цикла

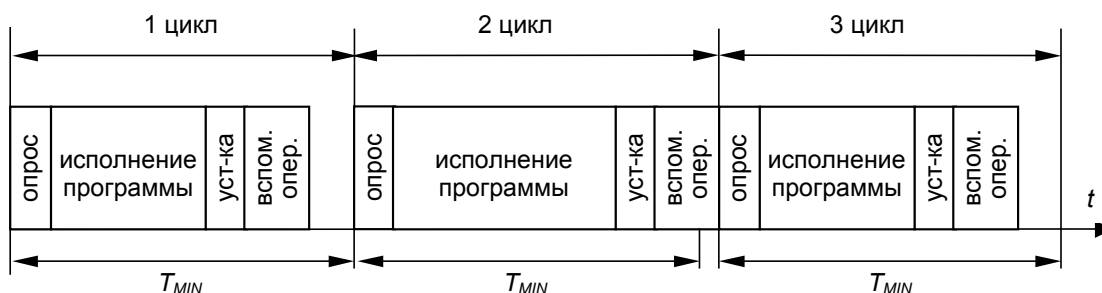
Способы исполнения рабочего цикла ПЛК

Некоторые ПЛК позволяют настраивать способ исполнения рабочего цикла:

- **циклический** – новый цикл начинается сразу по окончании последнего цикла (рис. 2.9, а);
- **периодический** – программист задаёт минимальную длительность цикла ПЛК. Если последний цикл завершился раньше заданного времени, то начало нового цикла задерживается до тех пор, пока не пройдёт установленная длительность цикла (рис. 2.9, б).



а) циклическое исполнение



б) периодическое исполнение

Рис. 2.9. Способы исполнения рабочего цикла ПЛК

Некоторые программируемые контроллеры поддерживают «параллельное» выполнение нескольких программ (многозадачный режим). При этом, поскольку одновременное выполнение нескольких программ (задач) на одном процессоре невозможно, имеется однозначно определённая структура приоритетов этих программ. Чаще всего такой многозадачный режим реализуется следующим образом.

Основная программа с низшим приоритетом выполняется циклически или периодически. Периодическая программа с более высоким приоритетом выполняется периодически, прерывая основную. При этом длительность исполнения периодической программы должна быть в несколько раз меньше чем длительность периода. Кроме того, имеется ряд программ с более высокими приоритетами, которые вызываются при возникновении какого-то события (прерывания) (рис. 2.10). Событиями могут быть как аппаратные сигналы (например, неисправ-

ность модуля ввода-вывода), так и программные (срабатывание сторожевого таймера при превышении максимальной длительности рабочего цикла контроллера).

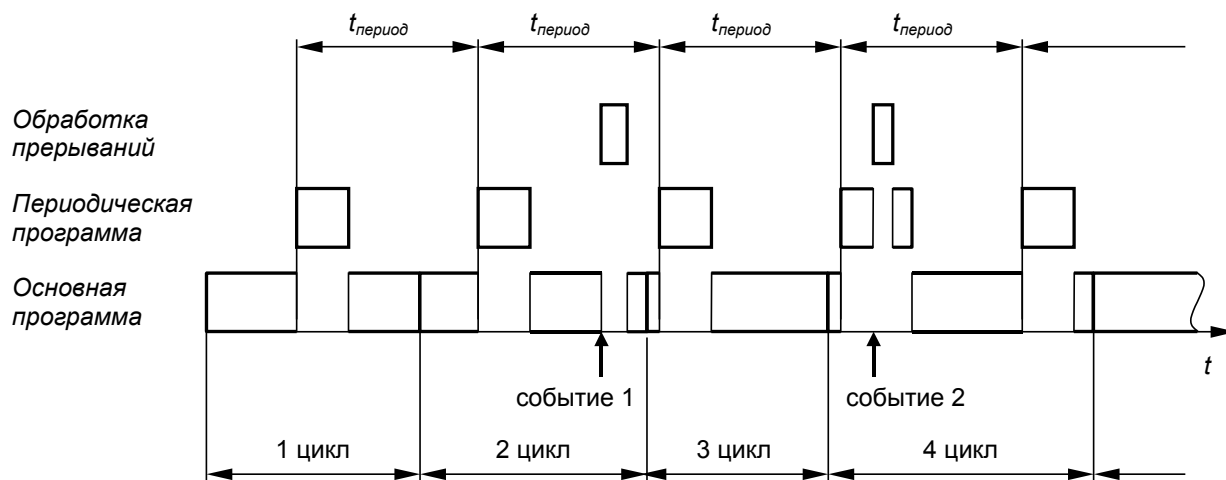


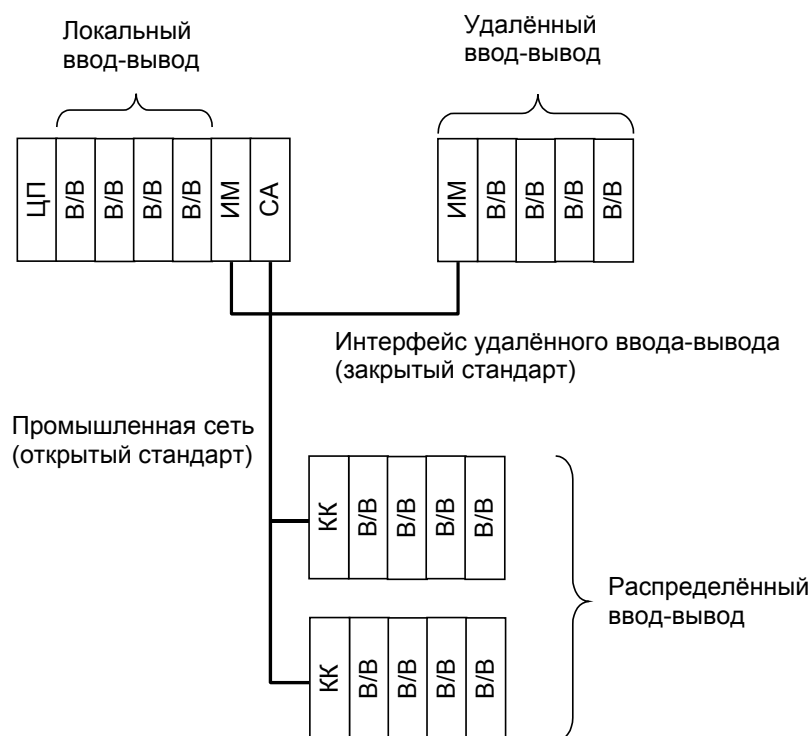
Рис. 2.10. Многозадачный режим исполнения рабочего цикла ПЛК

Распределённый ввод/вывод

Количество модулей расширения, подключаемых на одну шину с модулем центрального процессора, ограничено электрическими характеристиками системной шины. Для того чтобы подключить большее количество модулей используются технологии удалённого и распределённого ввода/вывода (такую терминологию использует компания Schneider Electric). Дополнительные модули ввода-вывода подключаются к специальному модулю – станции расширения, которая опрашивает подключенные к ней модули ввода-вывода и обменивается информацией с модулем центрального процессора по последовательному интерфейсу (рис. 2.11). При этом модули, подключаемые на одну шину с модулем центрального процессора, называются локальным вводом-выводом.

Главная особенность удалённого ввода-вывода – использование закрытого стандарта для организации интерфейса между ПЛК и станцией расширения. Поэтому его характеристики могут весьма сильно отличаться для разных фирм-производителей. Например, контроллеры Siemens серии S7-300 используют пару интерфейсных модулей (ИМ, рис. 2.11) со встроенным кабелем связи длиной до четырёх метров. Платформа Quantum компании Schneider Electric предлагает использование специальных модулей для организации удалённого ввода-вывода на расстоянии до 800 метров (Remote I/O). Платформа Premium компании Schneider Electric вообще не использует интерфейсные модули. Для того чтобы организовать удалённый ввод-вывод предлагается использовать специальные расширяемые рейки с разъёмами расширения шины, кабель (до 12 метров) и терминаторы для нормализации волновых сопротивлений на концах шины.

Для организации удалённого ввода-вывода некоторые производители предлагают использовать интерфейс программирования ПЛК, встроенный в модуль центрального процессора.



ИМ – интерфейсный модуль, СА – сетевой адаптер,
КК – коммуникационный контроллер

Рис. 2.11. Варианты организации распределённого ввода-вывода

При распределённом вводе-выводе для организации связи используется стандартная промышленная сеть. В этом случае в качестве станции расширения выступает коммуникационный контроллер (КК, рис. 2.11), который обеспечивает поддержку протокола промышленной сети.

Обмен данными между модулем центрального процессора и модулями ввода-вывода

При организации обмена данными между узлами ПЛК используются те же принципы, что и в ЭВМ с шинной организацией [21]. Центральный процессор под управлением операционной системы опрашивает входы ПЛК, размещает полученные данные в специальных областях, исполняет программу и передаёт новые значения на выходы ПЛК.

Каждый модуль ввода-вывода, входящий в состав ПЛК, есть не что иное, как периферийное устройство, конкретные характеристики и способы работы с которым задаются на этапе конфигурирования контроллера. Обмен данными с периферийными устройствами осуществляется с помощью портов ввода-вывода.

Каждому модулю ввода-вывода назначается свой уникальный диапазон портов ввода-вывода. Таким образом, центральный процессор ПЛК может обратиться к любому модулю ввода-вывода с запросом на чтение (модули ввода) или записи данных (модули вывода).

В состав каждого модуля входит один или несколько регистров данных – локальная память модуля. При обращении к модулю ввода-вывода центральный процессор передаёт по шине адреса адрес соответствующего порта ввода-вывода, а по шине управления – тип операции (чтение из порта или запись в порт). Данные передаются по шине данных.

Таким образом, центральный процессор и модули ввода-вывода работают асинхронно: модули ввода непрерывно опрашивают свои входы, постоянно поддерживая в актуальном состоянии локальную память модуля, а центральный процессор периодически считывает значения из локальной памяти каждого модуля ввода во время фазы опроса входов (первая фаза рабочего цикла ПЛК). Модули вывода постоянно удерживают на выходах значения из своей локальной памяти. Поэтому, как только значение локальной памяти модуля вывода изменяется центральным процессором во время фазы установки выходов (третья фаза рабочего цикла ПЛК), на выходах модуля вывода устанавливаются новые значения.

Режимы работы ПЛК

В каждый момент времени ПЛК находится в одном из состояний [22]:

- выключен (OFF);
- останов (STOP);
- запуск или рестарт (STARTUP);
- выполнение (RUN);
- неисправность (ERROR).

В состоянии «выключен» ПЛК находится при отсутствии напряжения питания. После включения питания контроллер переходит в состояние «останов».

В состоянии «останов» ПЛК проверяет, существуют ли фактически все сконфигурированные модули, и устанавливает выходы в предварительно определённое начальное состояние. В состоянии «останов» программа пользователя не выполняется.

В состоянии «запуск» различают типы запуска «холодный рестарт», «теплый рестарт» и «горячий рестарт».

При **теплом рестарте** рабочий цикл ПЛК начинается с опроса входов контроллера при текущих значениях всех внутренних переменных программы. Теплый рестарт – это повторный запуск, который выполняется вручную после перехода в состояние «останов» в случае появления критического события (неисправ-

ности) обработка которого не описана в программе контроллера (например, сбрасывание сторожевого таймера).

При **холодном рестарте** рабочий цикл ПЛК начинается с начала (также при теплом рестарте). Все внутренние переменные принимают предварительно заданные значения из загрузочной памяти. Холодный рестарт может быть автоматическим (например, после пропадания питания, потери информации, хранящейся в оперативной памяти) или ручным (например, с помощью кнопки перезапуска).

При **горячем рестарте** выполнение программы возобновляется в точке приостанова программы. ГОСТ Р 51840-2001 [20] определяет горячий рестарт как повторный запуск после пропадания питания в течение максимально допустимого промежутка времени при котором происходит восстановление работоспособности ПЛК, как если бы пропадания питания и не было. Возможность горячего рестарта требует наличия автономно питаемого таймера для определения времени отсутствия питания, а также наличия доступных пользователю средств для задания максимального времени отсутствия питания для обслуживаемого технологического процесса. Горячий рестарт поддерживается далеко не всеми контроллерами.

Первый запуск после загрузки в контроллер конфигурации или программы выполняется в режиме холодного рестарта. После успешного рестарта контроллер переходит в состояние «выполнение».

В состоянии «выполнение» ПЛК выполняет программу пользователя, обновляет входы и выходы, обслуживает прерывания и обрабатывает сообщения об ошибках.

При возникновении отказа, либо какого-то иного события, обработка которого не реализована в программе контроллера, ПЛК переходит в состояние «неисправность». Состояние «неисправность» фактически соответствует состоянию «останов», за одним исключением – на модуле центрального процессора загорается индикатор неисправности, а в диагностический буфер ПЛК заносится запись о времени и типе возникшей неисправности.

На рис. 2.12 представлена карта переходов ПЛК из состояния в состояние. При включении контроллер попадает в состояние «останов». В состояние «выключен» контроллер может перейти из любого другого состояния при отключении питания. Из состояния «останов», контроллер переходит в состояние «запуск». В зависимости от ситуации, при которой контроллер был переведён в состояние «останов», это может быть холодный, теплый или горячий рестарт. В случае неуспешного запуска контроллер переходит в состояние «неисправность». При успешном запуске, ПЛК переходит в состояние «выполнение», в котором может находиться сколь угодно долго. Из состояния «выполнение» контроллер может перейти в состояние «Останов» по команде оператора или программной

инструкции, либо в состояние «неисправность» в случае возникновения исключительной ситуации. Из состояния «неисправность» по команде оператора через состояние «останов» контроллер может быть переведён в состояние «запуск».

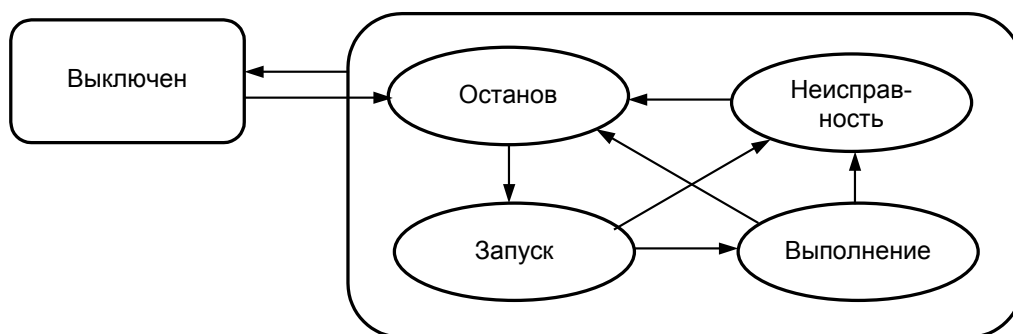


Рис. 2.12. Переход ПЛК из одного режима работы в другой

Загрузка конфигурации ПЛК выполняется в состоянии «останов». Загрузка программы может производиться как в состоянии «останов», так и в состоянии «выполнение». Нужно отметить, что загрузка программы в состоянии «выполнение» поддерживается далеко не всеми контроллерами и обычно является очень удобным свойством ПЛК, например, при поэтапной разработке программы ПЛК, управляющего непрерывным технологическим процессом.

Для защиты данных программы на случай выключения питания ПЛК применяется буферная батарея, которая позволяет сохранять состояние рабочей памяти. Если в состав ПЛК входит буферная батарея, то в таком контроллере может быть реализован запуск в режиме теплого и горячего рестарта.

2.6. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ МОДУЛЕЙ ЦЕНТРАЛЬНЫХ ПРОЦЕССОРОВ

Одним из этапов работы по проектированию АСУ ТП является выбор типа и конфигурации промышленного контроллера.

Исходя из требований к быстродействию системы, общего количества используемых входов/выходов контроллера, сложности реализуемых алгоритмов управления выбирается компания-производитель и платформа контроллера. Далее, уже в рамках выбранной платформы, выбирается конфигурация контроллера.

Конфигурация контроллера (количество и тип модулей ввода-вывода) определяется исходя из количества и типов датчиков и исполнительных механизмов. Для каждого модуля, входящего в состав ПЛК, производитель обычно указывает до двух десятков характеристик, которые должны учитываться при выборе модулей во время проектирования системы.

Основные характеристики модулей центральных процессоров во многом определяются принципами функционирования ПЛК:

- память. Измеряется в байтах, словах или инструкциях. Для памяти указывают следующие характеристики:

- ✓ объём рабочей, загрузочной и системной памяти;
- ✓ возможность использования карт расширения памяти, как загрузочной, так и рабочей;
- ✓ возможность защиты данных рабочей памяти – наличие буферной батареи;
- ✓ объём побитно адресуемой памяти (флаговой памяти);

- быстродействие. Указывается либо время выполнения различных операций (логических, операций со словами, операций с вещественными числами и т.д.) как это делает Siemens [23], либо характеристики используемого процессора (например, Intel 80186, 20 МГц), как это делают Schneider Electric, PEP Modular Computers [24].

- средства программирования и особенности реализации пользовательской программы:

- ✓ пакет или среда программирования;
- ✓ используемые языки программирования;
- ✓ максимальный размер и количество программных блоков (секций), блоков данных;
- ✓ максимальная глубина вложения вызова процедур и функций;
- ✓ способы исполнения программы (циклический, периодический, по событию);
- ✓ максимальное количество счётчиков, таймеров;
- ✓ наличие сторожевого таймера;
- ✓ наличие часов реального времени;

- система ввода-вывода:

- ✓ адресное пространство ввода-вывода (в байтах или словах). Размер адресного пространства позволяет определить максимальное количество обслуживаемых контроллером входов и выходов. Так для подключения одного аналогового сигнала требуется два байта или одно слово. Эти же два байта могут использоваться для подключения 16-ти дискретных сигналов;
- ✓ максимальное количество модулей ввода-вывода, которые можно подключить локально;
- ✓ максимальное количество модулей ввода-вывода, которые можно подключить к станции расширения;

- ✓ максимальное количество станций расширения, подключаемых удалённо или распределённо;
- ✓ максимальное количество сетей удалённого и распределённого ввода-вывода;
- реализация коммуникационных функций – встроенная поддержка интерфейсов промышленных сетей;
- технические характеристики
 - ✓ напряжение питания: номинал, допустимый диапазон;
 - ✓ потребляемый ток: при включении, во время работы;
 - ✓ потребляемая мощность;
 - ✓ габариты;
 - ✓ масса;
 - ✓ диапазон рабочих температур;
 - ✓ относительная влажность.

2.7. ПРИНЦИПЫ ФУНКЦИОНИРОВАНИЯ И ОСНОВНЫЕ ХАРАКТЕРИСТИКИ МОДУЛЕЙ ВВОДА-ВЫВОДА ПЛК

Для всех модулей ввода-вывода указываются общие характеристики:

- потребляемая мощность или потребляемый ток по шине;
- масса;
- габариты;
- рабочий температурный диапазон, [°C];
- степень защиты корпуса от пыли и влаги (IP) [25]

и т.д.

Потребляемая мощность позволяет определиться с необходимой мощностью источника питания. Габариты учитываются, например, при выборе глубины шкафа контроллера. Рабочий температурный диапазон и степень защиты корпуса позволяют определить, возможна ли эксплуатация выбранного контроллера без дополнительных мер защиты или нет, поскольку контроллеры могут эксплуатироваться как внутри помещения, так и на открытом воздухе.

Помимо общих характеристик указываются параметры, имеющие смысл для конкретного типа модуля ввода-вывода.

Модуль дискретного ввода

Принципиальная электрическая схема типового модуля дискретного ввода приведена на рис. 2.13. Дискретные датчики (Д0 ... Д7) представляют собой ключи (электронные или механические – неважно). Замыкание соответствующего ключа замыкает электрическую цепь, подключая светодиод соответствующей оп-

топары к источнику питания датчиков. Ток, протекающий через светодиод оптопары, переводит транзистор оптопары в проводящее состояние и потенциал на его коллекторе становится равен нулю (относительно внутренней земли модуля), что соответствует логической единице в ТТЛ-логике. Если ток через светодиод не протекает (ключ разомкнут), то транзистор закрывается и на коллекторе устанавливается напряжение +5В (логический ноль). Текущие значения дискретных входов (логические нули и единицы) сохраняются в регистре данных (локальная память модуля), откуда в любой момент могут быть переданы по системной шине ПЛК по запросу модуля центрального процессора.

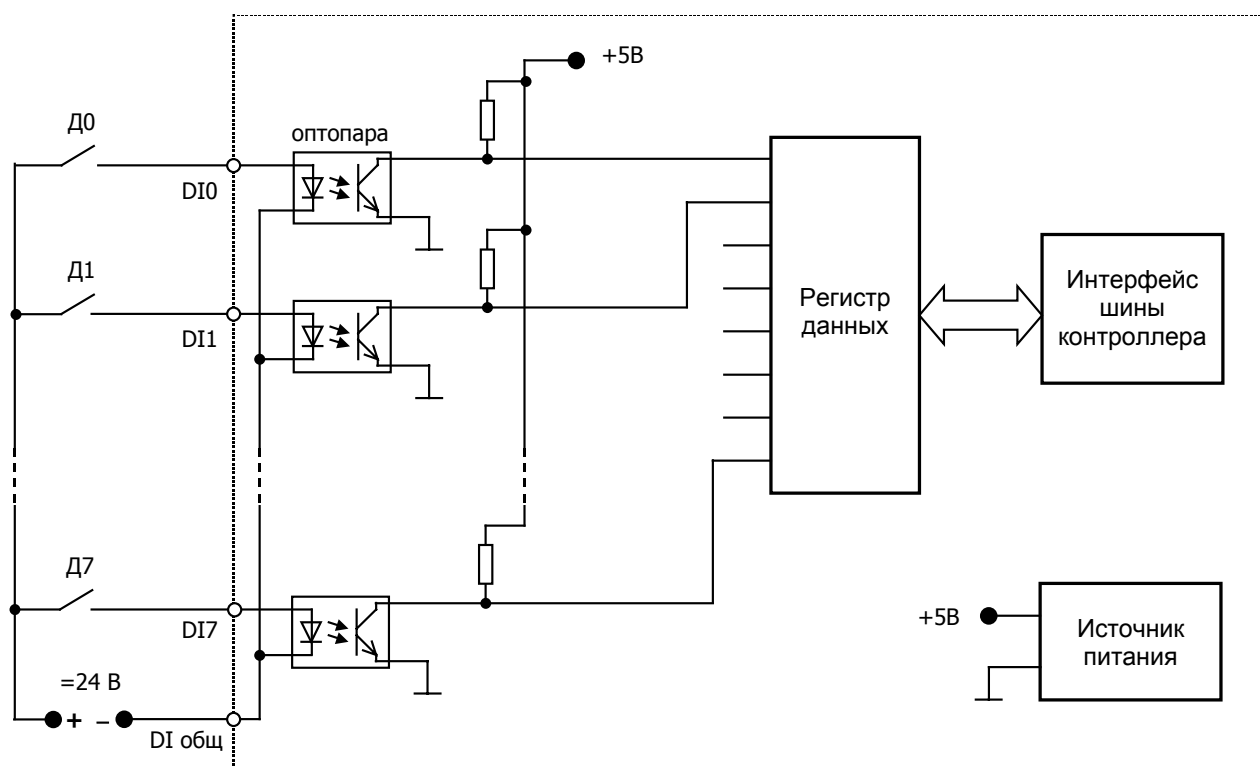


Рис. 2.13. Электрическая схема модуля дискретного ввода

Возможен вариант, при котором питание датчиков осуществляется переменным напряжением, электрическая схема подключения датчиков к модулю дискретного ввода выглядит следующим образом (рис. 2.14). Два противоположно включенных параллельных светодиода оптопары обеспечивают открытие транзисторного ключа на каждом полупериоде сигнала.

В обеих схемах (рис. 2.13, 2.14) оптопары обеспечивают гальваническую развязку входов модуля с внутренними цепями.

К основным характеристикам модуля дискретного ввода относятся следующие:

- количество входов (каналов) и способ подключения сигнальных линий. Количество каналов обычно кратно восьми. По способу подключения модули мо-

гут быть с дифференциальными входами (каждый вход – отдельная пара проводов) и с общей точкой (передача сигнала относительно общего провода – земли, см. рис. 2.13);

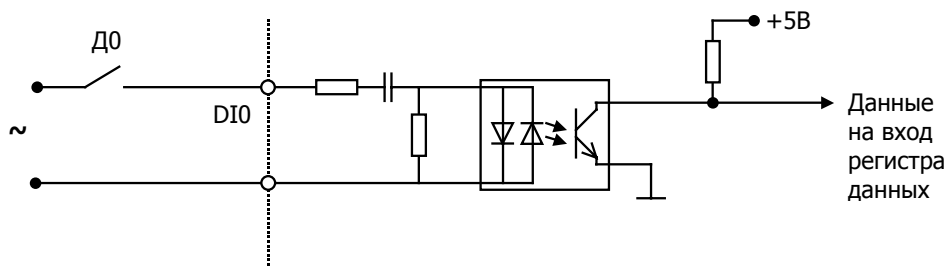


Рис. 2.14. Электрическая схема преобразования входного сигнала модуля дискретного ввода переменного тока

- количество входов в группах. Если подключение сигнальных линий осуществляется с общей точкой, то несколько входов объединяются в одну группу с общей землей;
- тип сигнала (постоянное/переменное напряжение). Для переменного напряжения дополнительно указывается допустимый диапазон частот сигнала, [Гц];
- уровень логического нуля, [В];
- уровень логической единицы, [В];
- входное сопротивление, [кОм];
- максимальная частота переключения входного сигнала, [Гц] или задержка распространения входного сигнала, [с];
- наличие гальванической развязки (гальваническая изоляция, $U_{\text{ПРОБОЯ}}$). Отсутствие изоляции снижает стоимость модуля, но повышает риск выхода контроллера из строя при попадании на сигнальные линии датчика силового напряжения.

Модуль дискретного вывода

Принципиальная электрическая схема модуля дискретного вывода представлена на рис. 2.15. Программа ПЛК сохраняет нужное состояние дискретных выходов контроллера в области памяти дискретных выходов, которая затем, во время установки выходов записывается в регистр данных (локальную память модуля дискретного вывода). Модуль дискретного вывода устанавливает на своих выходах соответствующие значения. Логическая единица на любом из выходов регистра данных приводит к появлению разности потенциалов между выходами светодиода оптопары и, как следствие, протеканию тока через светодиод, что, в свою очередь, приводит открытию транзисторного ключа и подачи напряжения на со-

ответствующий исполнительный механизм (ИМ, рис. 2.15). На рис. 2.15 приведена схема модуля дискретного вывода с транзисторными ключами с восемью выходами, объединёнными в одну группу с общей точкой (DO общ) и внешним источником питания постоянного тока. Для защиты транзисторных ключей от сгорания параллельно им подключены стабилитроны, которые ограничивают максимальный ток через транзисторный ключ.

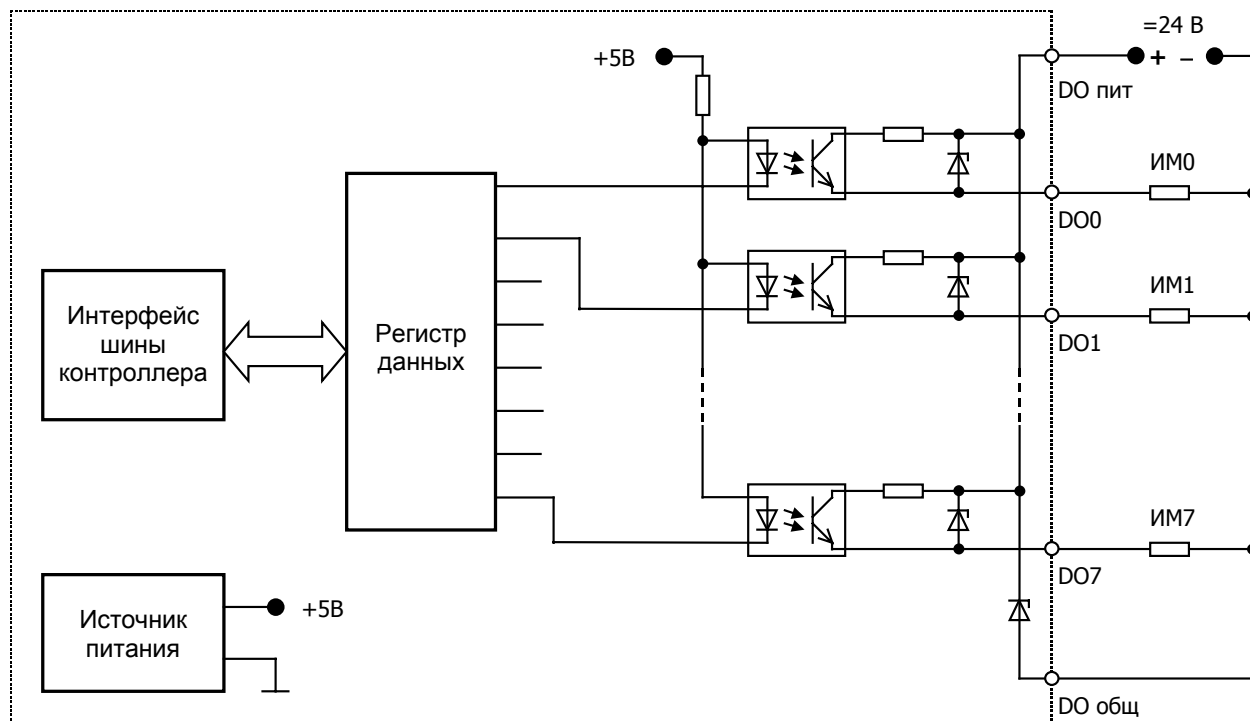


Рис. 2.15. Электрическая схема модуля дискретного вывода с транзисторными ключами

Использование транзисторного ключа для управления дискретным выходом обеспечивает высокое быстродействие выходов; отсутствие механических контактов – высокую износостойкость. Однако транзисторный ключ не позволяет коммутировать большие токи, а также требует учитывать полярность коммутируемого напряжения, а значит, может работать только с постоянным напряжением.

Перечисленных недостатков лишены модули дискретного вывода с релейным выходом (рис. 2.16). Вход «Упр.» подключается к одному из выходов регистра данных. При появлении на входе «Упр.» логической единицы, т.е. потенциал на эмиттере становится равен нулю, транзистор открывается и через катушку реле протекает ток, что приводит к срабатыванию реле и замыканию контакта реле. Контакты реле могут коммутировать напряжение любого вида (постоянное, переменное, любое другое). При появлении на входе «Упр.» логического нуля (потенциал +5 В) ток по цепи «база–эмиттер» не протекает и транзистор переходит в не-

проводящее состояние. Катушка реле обесточивается, реле отключается, и электрическая цепь исполнительного механизма разрывается. Поскольку катушка реле является индуктивной нагрузкой, то ток через неё моментально измениться не может (первое правило коммутации) [7]. Поэтому, чтобы защитить транзисторный ключ параллельно катушке реле включается диод. Когда через катушку протекает ток, через диод, включенный встречно, ток не течёт. При закрытии транзистора через диод протекает ток, индуцированный катушкой реле не вызывая увеличения потенциала на коллекторе транзистора.

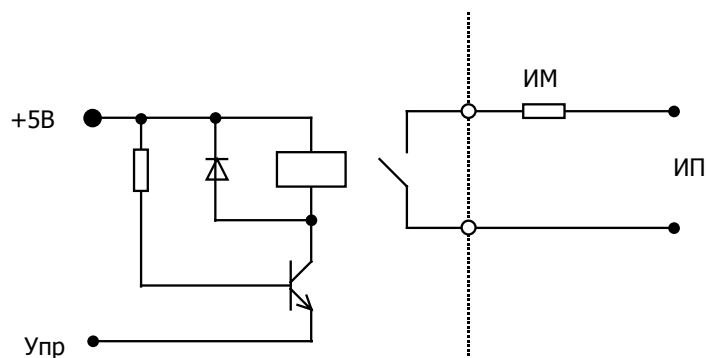


Рис. 2.16. Электрическая схема управления релейным выходом

На рис. 2.16 для управления включением реле используется обычный транзистор, а не оптопара, поскольку само реле выступает в качестве гальванической развязки.

Можно сказать, что реле является полной противоположностью транзисторного ключа: недостатки одного являются достоинствами другого. Реле позволяет коммутировать напряжение независимо от полярности. Реле, встраиваемые в модули дискретного вывода, обеспечивают коммутацию достаточно больших токов – до 8 Ампер. Для сравнения: транзисторные ключи обычно коммутируют токи не более 0,5 Ампер. Однако реле обладают сравнительно невысоким быстродействием и ограниченным ресурсом из-за искрения и обгорания контактов при размыкании цепи [26].

На рис. 2.16 показано реле с нормально разомкнутым (нормально открытым) контактом. Т.е. при отсутствии питания катушки реле цепь разомкнута. Возможно использование в составе модуля дискретного вывода реле с нормально замкнутыми (нормально закрытыми) контактами. Такие реле реализуют инверсную логику и используются в системах управления в тех случаях, когда по требованиям безопасности при пропадании питания контроллера цепи дискретного выхода должны оставаться замкнутыми. Иногда релейный выход называют выходом типа «сухой контакт».

Промежуточное положение между транзисторными и релейными дискретными выходами занимают модули с симисторовыми выходами (рис. 2.17). Симисторы довольно часто называют «твердотельными реле», термин появился как перевод соответствующего английского термина –solid state relay. Они имеют высокое быстродействие, в них отсутствуют механические контакты и одновременно их конструкция не накладывает никаких ограничений на характер питающего напряжения.

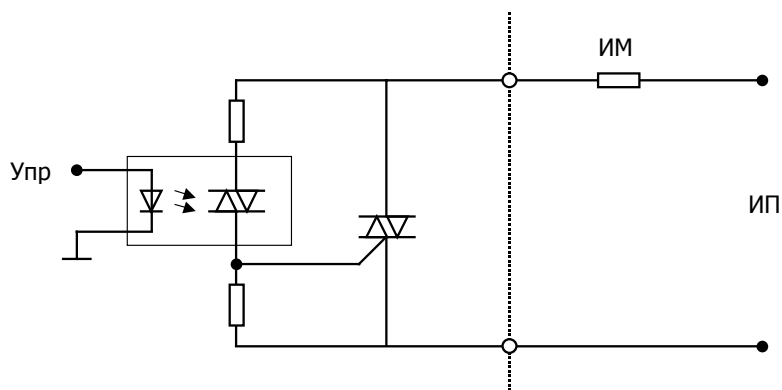


Рис. 2.17. Электрическая схема управления выходом на симисторе

На рис. 2.17 представлен один из вариантов реализации схемы управления симисторовым выходом. С помощью маломощного симистора, входящего в состав оптопары осуществляется управление обычными силовыми симисторами, которые замыкают электрическую цепь, подключая исполнительный механизм к источнику питания [26, 27].

К основным характеристикам модуля дискретного вывода относятся следующие:

- количество выходов (каналов) и способ подключения сигнальных линий (с общим проводом, изолированные);
- количество выходов в группах;
- тип сигнала. Тип сигнала зависит в первую очередь от того, как реализован коммутирующий элемент. Соответственно различают модули с транзисторными ключами, реле и симисторами. Иногда встречаются модули с транзисторными ключами и внутренним источником напряжения. В этом случае уже говорят не о замыкании или размыкании электрической цепи с помощью дискретного выхода, а о подаче/снятии напряжения на дискретном выходе;
- для транзисторных ключей указываются:
 - ✓ напряжение логического нуля, [В];
 - ✓ напряжение логической единицы, [В];
 - ✓ максимальный выходной ток логической единицы, [А];

- для релейных выходов указываются:
 - ✓ логика – прямая или инверсная (нормально открытые или нормально закрытые контакты реле);
 - ✓ номинальное коммутируемое напряжение, [В];
 - ✓ коммутационная способность контактов (максимальный коммутируемый ток), [А];
 - ✓ количество циклов срабатывания контактов (обычно – до нескольких миллионов);
- для симисторовых выходов указываются:
 - ✓ номинальное коммутируемое напряжение, [В];
 - ✓ максимальный коммутируемый ток, [А];
- частота переключений выходов при разных типах нагрузки (активная, индуктивная, емкостная), [Гц]. При этом указывается предельно допустимая индуктивность [Гн] и ёмкость [мкФ] нагрузки;
- наличие гальванической развязки.

Модули счёта импульсов

Для обработки быстро переключающихся дискретных сигналов, использование простых модулей дискретного ввода может привести к потере информации, в тех случаях, когда длительность цикла контроллера дольше, чем длительность импульса и/или частота его появления. Для обработки сигналов с датчиков, генерирующих последовательность импульсов, используются модули быстрого счёта.

Модуль быстрого счёта в своей основе имеет модуль дискретного ввода (рис. 2.18). Импульсы, которые нужно считать приходят на вход Pulse (импульс). Вход Gate (ворота) используется для аппаратного разрешения/запрещения счёта импульсов. Иногда в модуле присутствует вход Direction (направление). В этом случае, в зависимости от состояния входа Direction происходит увеличение или уменьшение переменной «Значение счётчика». При достижении заданных пороговых значений переменная «Порог достигнут» принимает значение логической единицы, а счёт импульсов прекращается. Значение переменной «Порог достигнут» проверяется программой ПЛК, и, после обработки, по специальной команде в переменную «Значение счётчика» пересылается «Начальное значение». После чего счёт импульсов возобновляется.

Диаграмма работы модуля счёта импульсов представлена на рис. 2.19. Логическая единица на входе Gate запрещает счёт импульсов, логический нуль на входе Direction соответствует увеличению счётчика, а логическая единица – уменьшению.

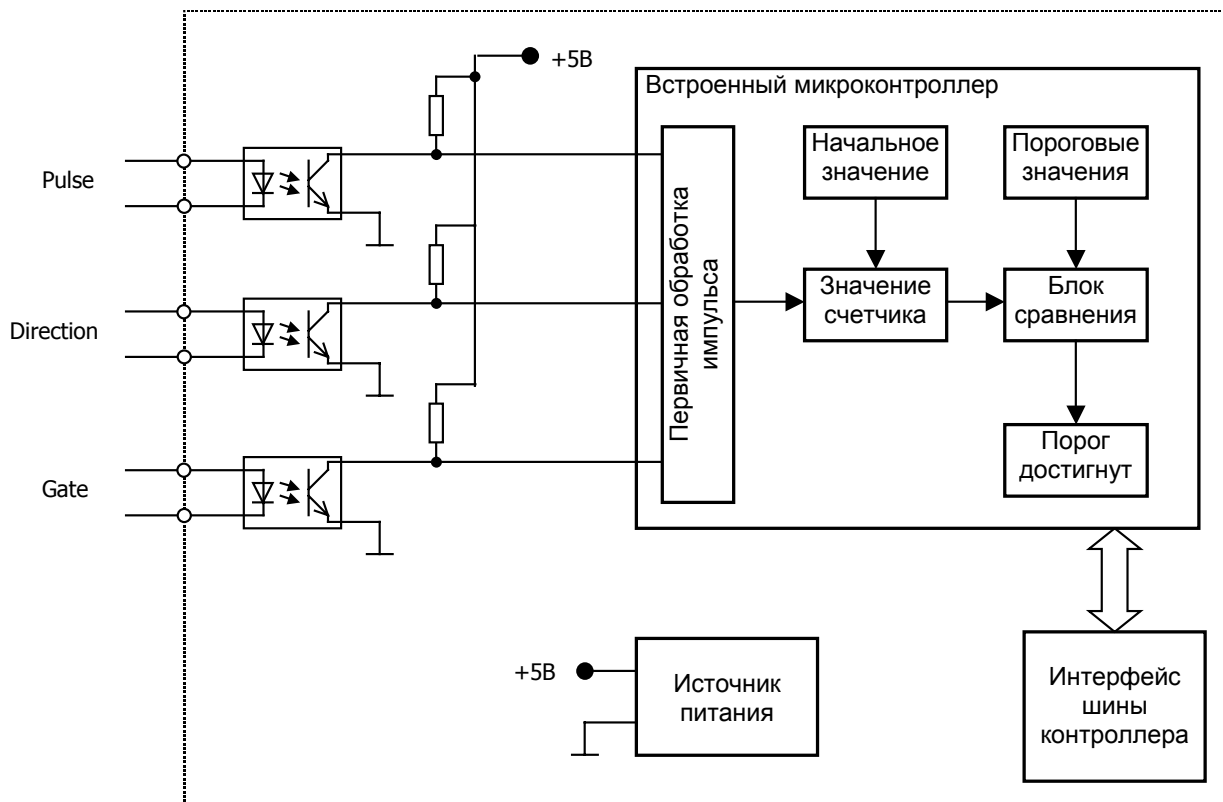


Рис. 2.18. Функциональная схема модуля быстрого счёта

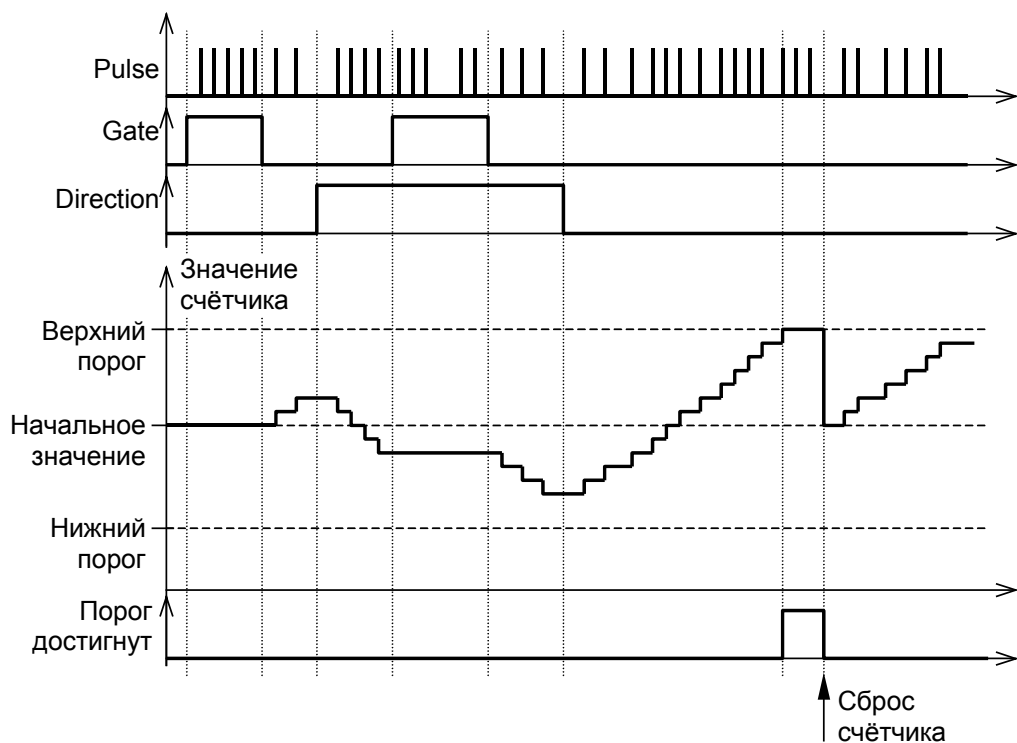


Рис. 2.19. Диаграмма работы модуля счёта импульсов

Поскольку алгоритм обработки импульсов реализуется на встроенном контроллере модуля быстрого счёта, то обычно модуль быстрого счёта может выступать и в роли модуля измерения частоты. Фактически, модуль измерения частоты

– это тот же модуль быстрого счёта, но с другим алгоритмом обработки сигнала. Частота определяется как количество импульсов в единицу времени. Результат работы модуля – формирование логической единицы в случае выхода измеряемой частоты за границы заданного диапазона.

Функциональная схема модуля измерения частоты представлена на рис. 2.20. Переменная «Значение счётчика» инкрементируется всякий раз, когда приходит очередной импульс. Частота поступления импульсов рассчитывается как отношение количества импульсов, пришедших за период усреднения, к величине периода усреднения. Если рассчитанное значение частоты выходит за пределы пороговых значений, то переменной «Порог достигнут» присваивается значение логической единицы. Которая удерживается до тех пор, пока не будет сброшена специальной командой из программы ПЛК.

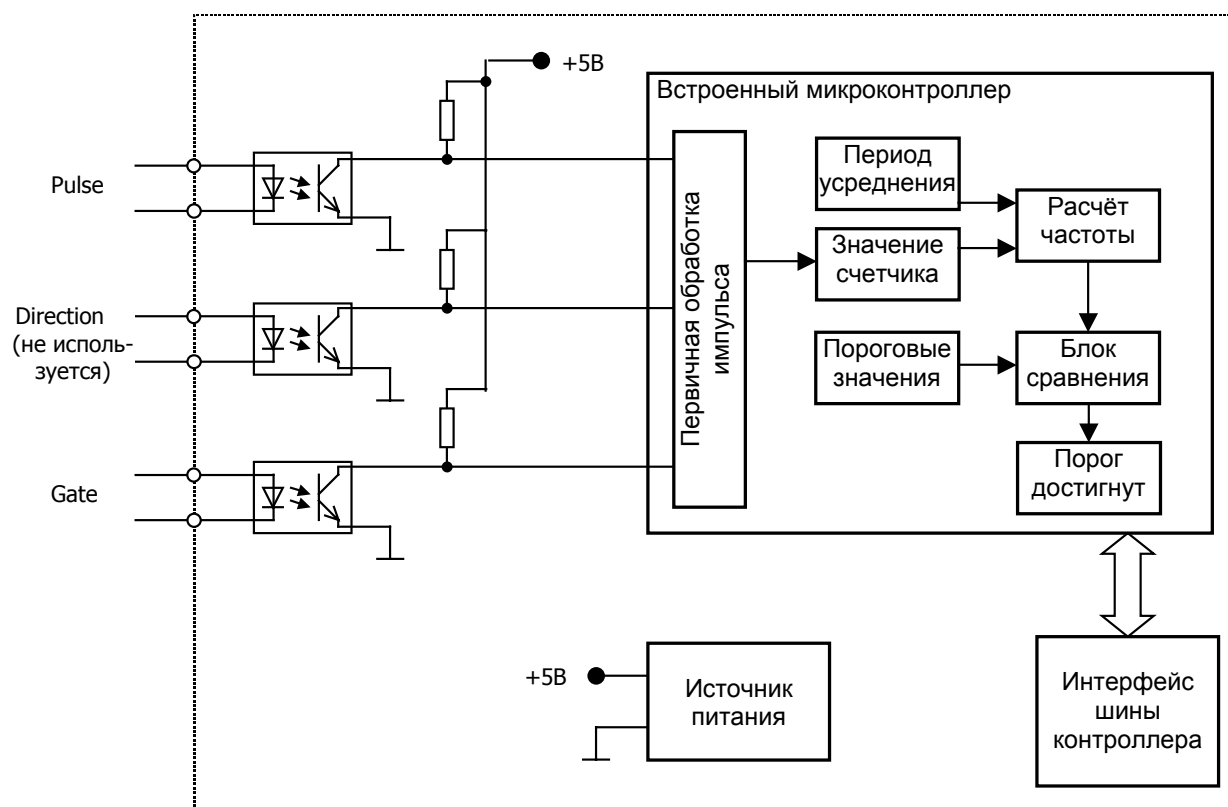


Рис. 2.20. Функциональная схема модуля измерения частоты

Чаще всего модуль измерения частоты объединяется с модулем быстрого счёта, а режим работы модуля задаётся во время конфигурации модуля.

Модуль аналогового ввода

Обработка аналоговых сигналов начинается с фильтрации всех высокочастотных помех, наводимых в сигнальных проводах при передаче сигнала, с помощью фильтра низких частот (ФНЧ, рис. 2.21). Отфильтрованные измерительные

сигналы собираются в мультиплексоре. Это устройство, которое имеет адресные входы, несколько информационных входов и один выход. В зависимости от адреса, один из входов коммутируется на выход. Основное назначение мультиплексора – уменьшить общую стоимость модуля аналогового ввода за счёт применения только одного устройства обработки сигналов (в данном случае – аналогово-цифровой преобразователь, АЦП), которое обычно существенно дороже мультиплексора для всех входных сигналов. В АЦП происходит преобразование аналогового сигнала в цифровой. К выходам АЦП подключаются параллельно все регистры данных (РД) буфера данных [1].

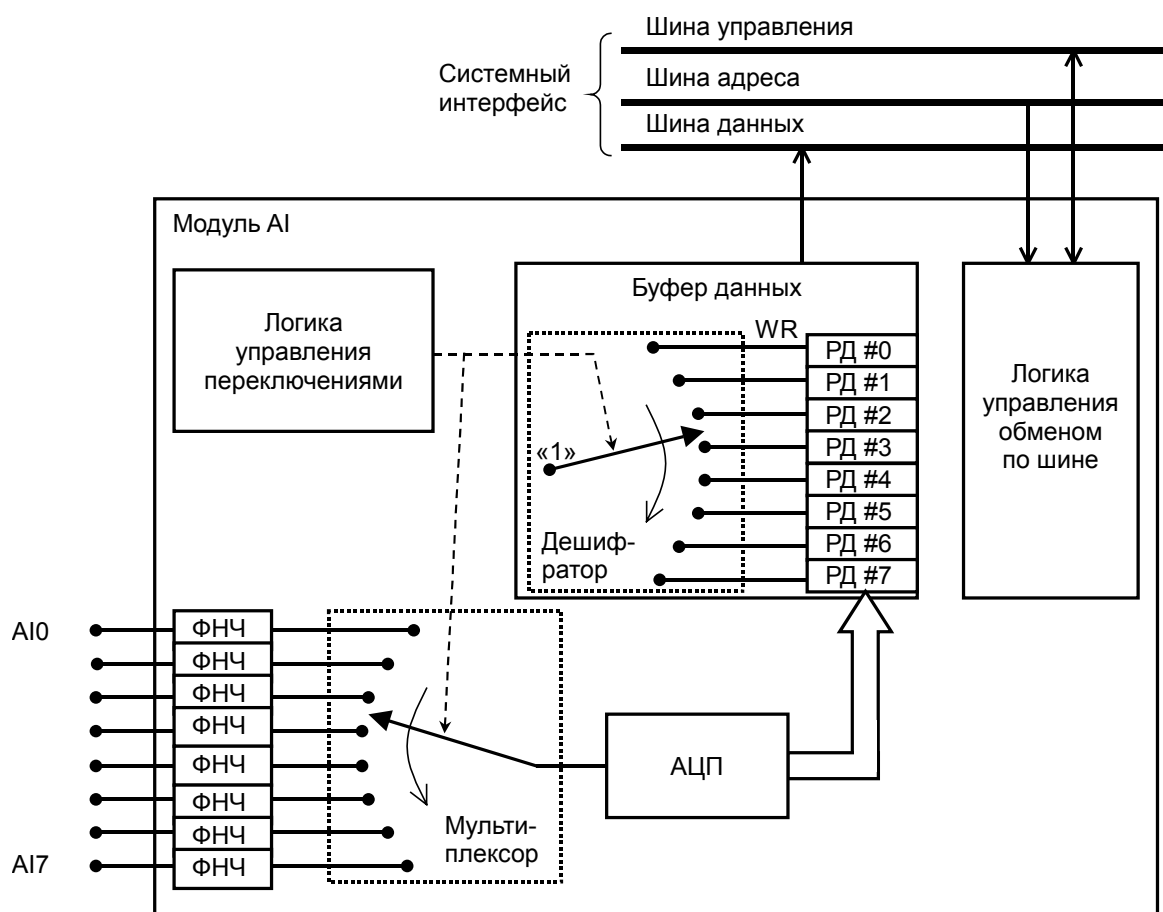


Рис. 2.21. Функциональная схема модуля ввода аналоговых сигналов

С помощью мультиплексора входы модуля поочерёдно подключаются ко входу АЦП. Одновременно с переключениями мультиплексора дешифратор буфера данных переключает подачу разрешающего сигнала «запись» (WR) на управляющий вход очередного регистра данных. Таким образом, цифровой сигнал на выходе АЦП, соответствующий i -тому входу запоминается в i -том регистре данных локальной памяти модуля. Управление переключениями мультиплексора и дешифратора осуществляет функциональный узел «Логика управления переключениями», который обычно реализуется микроконтроллером модуля. Час-

тота переключений мультиплексора определяется временем АЦ-преобразования, что, в свою очередь, влияет на стоимость АЦП и модуля аналогового ввода в целом: чем меньше время преобразования, тем выше стоимость. От времени АЦ-преобразования зависит и частота обновления значений в регистрах данных. Так, например, для модуля аналогового ввода со временем АЦ-преобразования 500 мс и восемью каналами, значение в регистре данных для каждого входа будет обновляться один раз в четыре секунды. Другими словами – частота сканирования или частота выборки в этом случае составляет 0,25 Гц.

Таким образом, модуль аналогового ввода представляет собой экстраполятор нулевого порядка с определённым периодом квантования. Экстраполятор нулевого порядка может рассматриваться как фильтр низких частот (теорема Котельникова-Шеннона, [2]), в связи с этим некоторые производители модулей позволяют настраивать время АЦ-преобразования. Иногда от настроенного времени АЦ-преобразования зависит точность преобразования.

Буфер данных (локальная память модуля) хранит результаты последнего измерения значений сигналов для каждого входа. По запросу центрального процессора эти данные передаются по системной шине для размещения в соответствующей области памяти.

Электрическая схема модуля аналогового ввода представлена на рис. 2.22. В составе модуля используются два мультиплексора (АМХ) – по одному на каждый провод измерительной цепи. Использование одного мультиплексора можно встретить в модулях с общей точкой подключения каналов, однако обычно такие модули редко используются, поскольку такая конструкция приводит к возникновению резистивной связи между измерительными каналами. Иногда два аналоговых мультиплексора конструктивно объединяются в одну микросхему.

Обратите внимание, гальваническая развязка расположена после АЦП. Хотя АЦП – дорогое устройство, однако, изготовление гальванической развязки аналогового сигнала с линейной характеристикой ещё более дорогостоящая и трудноосуществимая задача, поэтому гальванически развязывают линии передачи дискретных сигналов после АЦП.

При подключении токовых информационных сигналов сила тока оценивается по падению напряжения на шунтирующем сопротивлении с известным номиналом. Многие производители встраивают прецизионные шунтирующие сопротивления в модули аналогового ввода, поэтому при измерении токового сигнала достаточно подключить сопротивление параллельно входу модуля аналогового ввода (рис. 2.23).

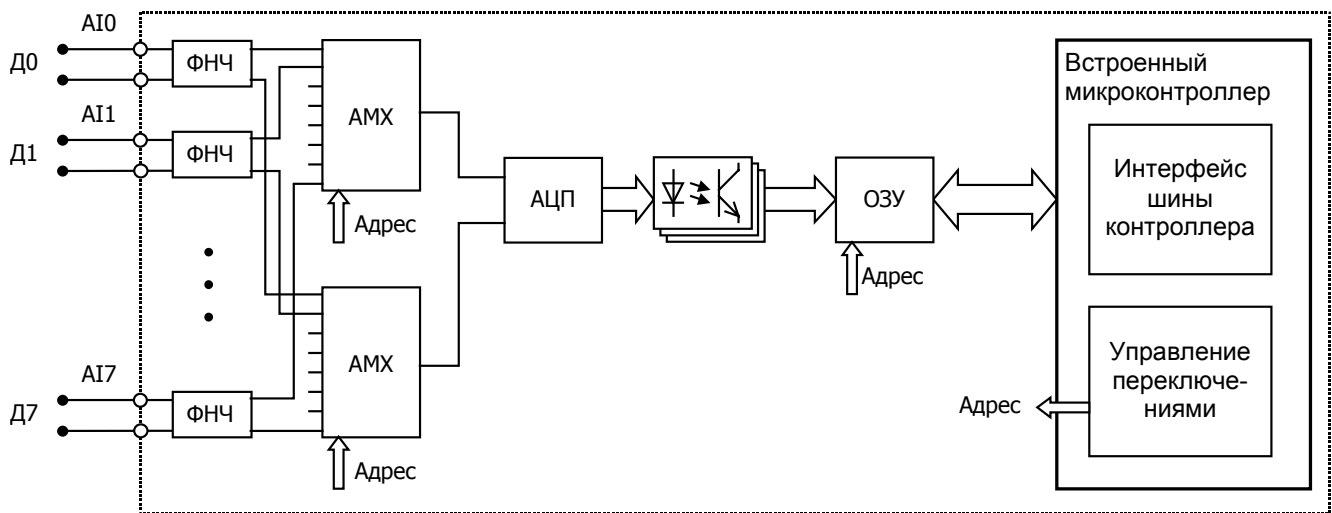


Рис. 2.22. Электрическая схема модуля аналогового ввода

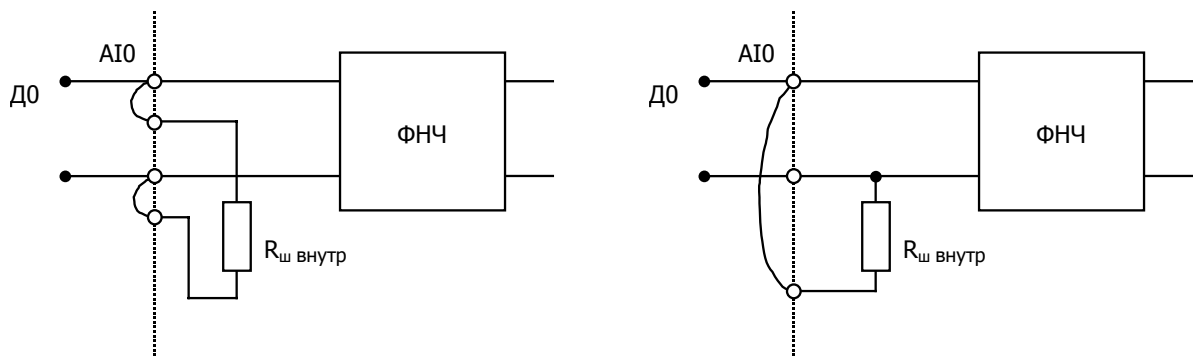


Рис. 2.23. Варианты электрических схем подключения токовых сигналов с использованием встроенных шунтирующих резисторов

Основные характеристики модуля аналогового ввода:

- количество входов (каналов) и способ подключения сигнальных линий (с общим проводом, изолированные). Для каналов с общим проводом указывается количество входов в группах;
- типы подключаемых датчиков (унифицированный вход напряжения или тока, термопара, терморезистор, тензодатчик);
- диапазон изменения входного сигнала (в зависимости от типа датчика), [В, мА, тип термопары, тип терморезистор, Ом];
- способ передачи аналогового сигнала (дифференциальный, с относительной землёй);
- максимально допустимая разность потенциалов между входами модуля. Этот параметр определяется характеристиками мультиплексора. При превышении установленного значения мультиплексор может сгореть;
- разрешающая способность АЦП, [бит];
- полоса пропускания, [Гц];

- частота выборки или сканирования, [Гц], или время АЦ-преобразования, [мс];
- для входного сигнала напряжения указывается входное сопротивление, [МОм];
- погрешность АЦП, [%];
- температурный дрейф нуля, [мкВ/°С];
- наличие гальванической развязки.

При обработке данных полученных от модуля аналогового ввода, необходимо учитывать, что диапазон измеряемого сигнала обычно немного меньше диапазона изменения значений на выходе АЦП. Такая организация работы АЦП позволяет предоставить программисту дополнительные возможности по обработке цифровых данных. Например, для модуля аналогового ввода контроллера Quantum 140 AVI 030 00 компании Schneider Electric с шестнадцатиразрядным АЦП, имеет место следующий способ преобразования (табл. 2.2) [28]. Благодаря этому, из программы можно протестировать режим работы датчика и исправность сигнальных линий.

Таблица 2.2.

**Диапазон изменения цифрового сигнала на выходе АЦП
модуля аналогового ввода 140 AVI 030 00**

Диапазон изменения входного сигнала	Разряды АЦП		
	Меньше нижней границы диапазона	Норма	Больше верхней границы диапазона
–10... +10 В	<768	768...64 768	>64 768
–5...+5 В, –20...20 мА	<16 768	16 768...48 768	>48 768
0... +10 В	–	0...64 000	>64 000
0... +5 В, 0...20 мА	–	0...32 000	>32 000
+1...+5 В, 4...20 мА	<6 400	6 400...32 000	>32 000

Модуль аналогового вывода

Функциональная схема модуля аналогового вывода представлена на рис. 2.24. Значения из области аналоговых выводов пересылаются в локальную память модуля. К информационным выходам каждого регистра данных (РД) через гальваническую развязку (ГР) подключен цифроаналоговый преобразователь (ЦАП), который преобразует цифровой код на входе в уровень напряжения на выходе. Аналоговый сигнал с выхода ЦАП поступает на фильтр низких частот и далее, через выходной усилительный каскад (ВК) – на входы исполнительных устройств и механизмов. Здесь уже не используется мультиплексор. Во-первых, стоимость ЦАП не так высока, а во-вторых, и это главное, при использовании

мультиплексора пришлось бы включать в состав модуля функциональное устройство способное хранить значение сигнала не в виде цифрового кода, а в аналоговой форме в виде уровня напряжения, что труднореализуемо.

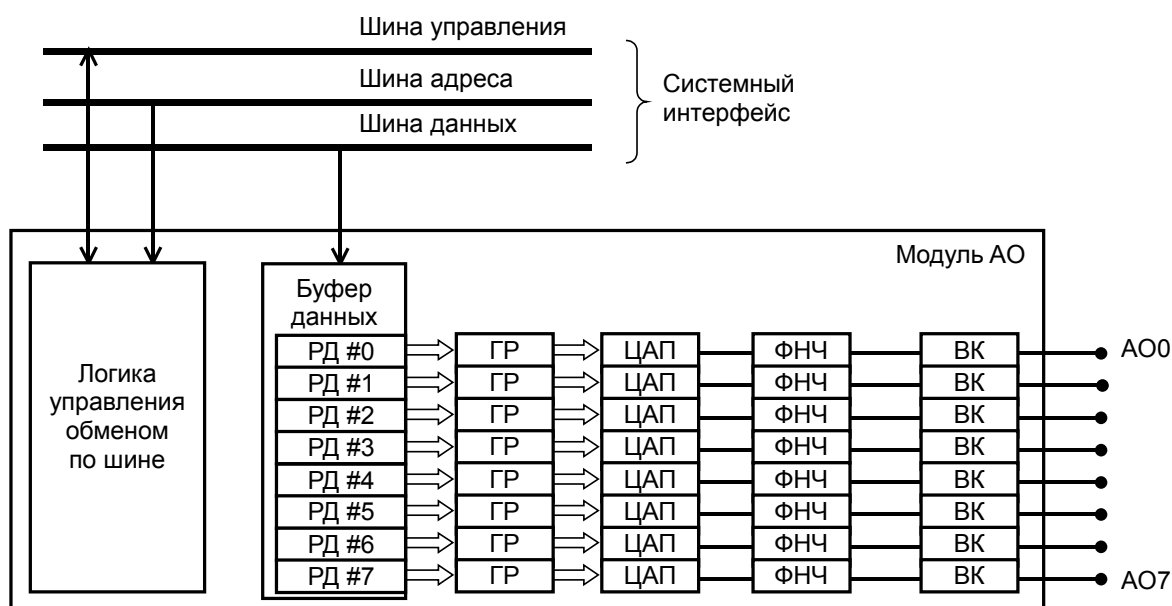


Рис. 2.24. Функциональная схема модуля вывода аналоговых параметров

Реализация выходного усилительного каскада может быть выполнена различными способами. Например, компания ICPCon (Тайвань) [29] при реализации модуля аналогового вывода I-8024 использует следующую схему (рис. 2.25). Аналоговый сигнал может передаваться напряжением, либо током. В случае использования аналогового сигнала напряжения, исполнительное устройство подключается к выходам U и GND. Фактически, на выходе модуля поддерживается заданное напряжение, при этом реальное напряжение на исполнительном механизме никак не контролируется. В случае использования токового сигнала, на клемму выхода U дополнительно подключается выход I, который используется для организации обратной связи по току.

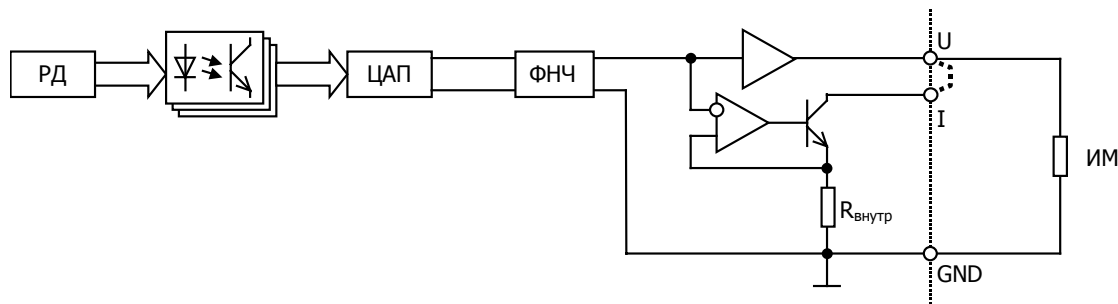


Рис. 2.25. Электрическая схема реализации аналогового вывода на примере модуля ICPCon I-8024

Компания Siemens использует более сложную, но и более точную схему реализации аналогового вывода с обратными связями, как по току, так и по напряжению (рис. 2.26) [30]. В случае передачи аналогового сигнала токовым сигналом, исполнительный механизм подключается к клеммам модуля I+ и GND. Сила тока в цепи при этом контролируется встроенным микроконтроллером по падению напряжения на внутреннем шунте $R_{\text{внутр}}$ с помощью АЦ-преобразования. При передаче аналогового сигнала напряжением, для того, чтобы избежать погрешности, связанной с падением напряжения на соединительных проводах, непосредственно к исполнительному механизму подключаются измерительные провода, которые позволяют измерить падение напряжения на исполнительном механизме. С другой стороны эти провода подключаются на клеммы модуля $U_{\text{изм}+}$ и $U_{\text{изм}-}$ и с помощью АЦ-преобразования оцениваются встроенным микроконтроллером, который корректирует значение на входе ЦАП, чтобы обеспечить заданное напряжение на исполнительном механизме. Благодаря большому входному сопротивлению АЦП, ток в измерительной цепи, а значит и падение напряжения на измерительных проводах очень мало.

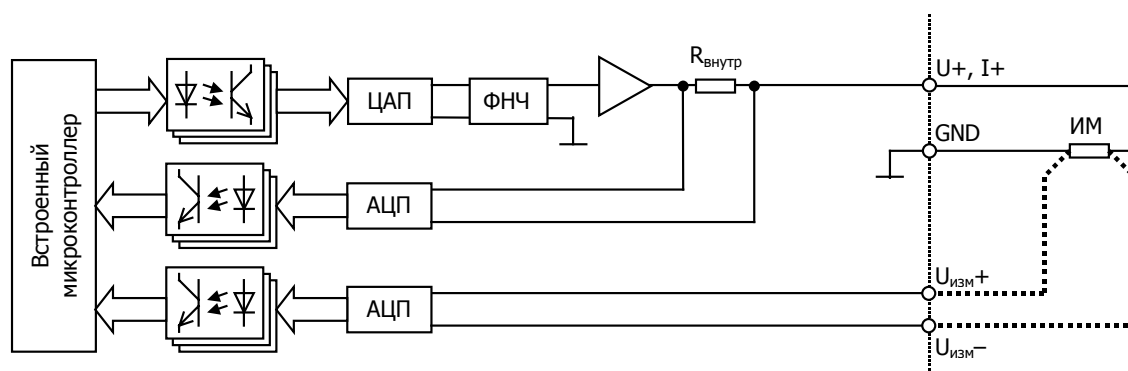


Рис. 2.26. Электрическая схема реализации аналогового вывода на примере модуля Simatic S7-332

Основные характеристики модуля аналогового вывода:

- количество и тип выходов (каналов) и способ подключения сигнальных линий (с общим проводом, изолированные). Для каналов с общим проводом указывается количество входов в группах;
- диапазон изменения выходного сигнала, [В, мА];
- разрядность ЦАП, [бит];
- скорость нарастания сигнала, [В/с, мА/с]. Из-за конструктивных особенностей ЦАП на его выходе невозможно получить скачкообразное изменение сигнала. При появлении скачка на входе ЦАП, напряжение на его выходе будет линейно увеличиваться. Скорость этого увеличения и указывается в качестве характеристики модуля;

- входное сопротивление нагрузки, [кОм];
- максимальная ёмкость нагрузки, [мкФ];
- максимальная индуктивность нагрузки, [Гн];
- базисная и рабочая погрешности преобразования ЦАП, [%];
- температурный дрейф нуля, [мкВ/°С, мкА/°С];
- наличие гальванической развязки.

Вопросы для самоконтроля

1. Перечислите основные требования к аппаратной организации управляющих ЭВМ.
2. Назовите основные конструктивные элементы ПЛК.
3. Какой стандарт конструктивного исполнения модулей используется при построении ПЛК на базе шин VME и CompactPCI?
4. Модули какого размера используются при построении ПЛК на базе шин VME и CompactPCI?
5. Какие принципиальные различия в организации обмена данными по шине между стандартами VME и CompactPCI?
6. Перечислите основные типы памяти ПЛК.
7. Что такое рабочий цикл ПЛК? Из каких фаз он состоит? В чём преимущество организации исполнения программы ПЛК в виде рабочего цикла?
8. Чем различаются временные диаграммы циклического и периодического исполнения рабочего цикла ПЛК?
9. В чём различия между локальным, удалённым и распределённым способами подключения модулей ввода-вывода к ПЛК?
10. Перечислите и опишите основные режимы работы ПЛК. Какой тип ресурта обязательно присутствует у любого ПЛК?
11. Перечислите характеристики модулей центральных процессоров, модулей ввода и модулей вывода аналоговых и дискретных сигналов.
12. Приведите типовые электрические схемы дискретных и аналоговых модулей ввода и модулей вывода.
13. Для чего в состав модулей ввода-вывода включается блок гальванической развязки?
14. Какими преимуществами и недостатками обладают модули дискретного вывода на базе транзисторных ключей, реле, симисторов?
15. Почему блок гальванической развязки в модуле аналогового ввода устанавливается не до АЦП, а после, на цифровой шине?

16. Обязательно ли вхождение в состав модуля аналогового ввода мультимплексора входных сигналов?

17. Какой тип модуля должен использоваться при подключении к ПЛК термопары? порогового датчика уровня? энкодера? лампочки-индикатора?

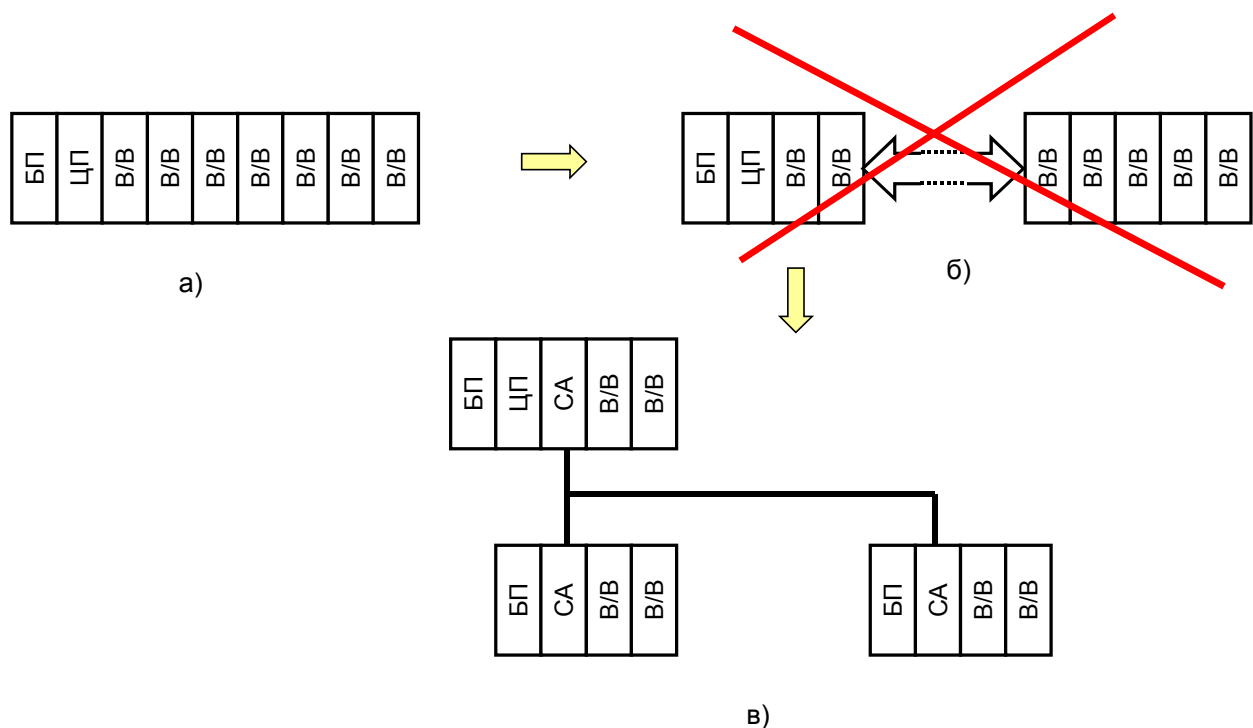
18. Как подобрать величину шунтирующего сопротивления для подключения датчика с токовым выходом 4...20 мА к модулю аналогового входа, предназначенному для измерения напряжения в диапазоне 1...5 В?

3. ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ПРОМЫШЛЕННЫХ СЕТЕЙ

3.1. ПРЕДПОСЫЛКИ ПРИМЕНЕНИЯ ПРОМЫШЛЕННЫХ СЕТЕЙ

До сих пор мы явно не говорили об аппаратной структуре системы управления, для удобства полагая, что в системе присутствует одна управляющая ЭВМ, к которой через модули ввода-вывода подключаются датчики и исполнительные механизмы. Такая структура автоматизированных систем управления действительно применялась на заре появления ЭВМ. Однако, при построении системы автоматизированного управления оборудованием, рассредоточенным на большой площади, с большим количеством контролируемых параметров приходится выполнять дорогостоящие монтажные работы для сбора всех сигнальных линий в одну точку – место установки контроллера. И если раньше такое построение системы оправдывалось высокой стоимостью цифровой техники, то в настоящее время в большинстве случаев становится экономически целесообразной установка на площади цеха или участка нескольких локальных контроллеров или интеллектуальных устройств связи с объектом, объединённых в единую сеть [31].

В целом идея выглядит так: необходимо вынести часть модулей ввода-вывода поближе к местам установки датчиков и исполнительных механизмов (рис. 3.1). К сожалению, этого нельзя сделать, просто удлив системную шину ПЛК (рис. 3.1, б). Системная шина реализуется с помощью группы параллельных проводников и, для нормальной работы шины, их физические и геометрические характеристики должны быть *одинаковыми*, что довольно трудно обеспечить на значительном расстоянии. Кроме того, нельзя забывать о влиянии различных источников помех и о взаимном влиянии проводников друг на друга. И то и другое влияние увеличивается с ростом длины системной шины. Выход из этой ситуации состоит в применении последовательного интерфейса для организации связи ПЛК с распределённым вводом-выводом. При этом в состав ПЛК и каждого узла распределённого ввода-вывода вводится дополнительный модуль – сетевой адаптер или коммуникационный контроллер (рис. 3.1, в), который обеспечивает передачу данных по сети. Сетевой адаптер играет роль своеобразного моста, осуществляющего преобразование данных передающихся по параллельному интерфейсу системной шины в последовательный интерфейс промышленной сети. Другими словами, он позволяет осуществить переход от параллельной передачи данных к последовательной и обратно.



БП – блок питания, ЦП – модуль центрального процессора,
В/В – модуль ввода-вывода, СА – сетевой адаптер.

Рис. 3.1. Этапы «замены» системной шины промышленной сетью

Помимо сокращения суммарной длины сигнальных проводов автоматизированной системы управления, использование промышленных сетей обладают и другими преимуществами. Стоимость работ по установке, тестированию, вводу в эксплуатацию и сопровождению централизованной системы гораздо выше, чем у распределённой. Использование промышленной сети позволяет легко подключать дополнительные узлы распределённого ввода-вывода, т.е. обеспечивает масштабируемость АСУ ТП. Кроме того, применение промышленных сетей позволяет включать в состав АСУ ТП интеллектуальные датчики и исполнительные механизмы, реализующие локальные САУ и «разгружающие» управляющую ЭВМ.

3.2. СТАНДАРТИЗАЦИЯ В ПРОМЫШЛЕННОСТИ

Применение распределённого ввода-вывода поставило перед разработчиками проблему разработки общих принципов подключения устройств распределённого ввода-вывода. Естественно, эти принципы были оформлены в виде стандартов, описывающих правила взаимодействия подключаемых устройств – стандарты промышленных сетей. Промышленные сети также называют полевыми сетями (дословный перевод термина *fieldbus* из иностранной литературы). Их выделяют в отдельную группу, чтобы не путать с локальными вычислительными сетями, которые также могут использоваться в составе системы промышленной автоматизации, но для решения совсем других задач.

Вообще, в системах промышленной автоматизации выделяют пять уровней (рис. 3.2):

I. Ввод/вывод (В/В) (Input/Output – I/O). На данном уровне «работают» устройства связи с объектом (УСО). К ним относятся датчики, исполнительные устройства и устройства удалённого сбора данных (УУСД).

II. Управление вводом/выводом. Управление вводом/выводом чаще всего осуществляется программируемыми логическими контроллерами (ПЛК). Кроме того, имеется целый класс устройств – интеллектуальные датчики и интеллектуальные исполнительные механизмы, которые реализуют в себе функции первого и второго уровней.

III. Диспетчерское управление и сбор данных (Supervisory Control and Data Acquisition – SCADA). Этот и следующие уровни реализуются с помощью программного обеспечения. Обычно программное обеспечение SCADA-систем выполняется на компьютерах, стоящих непосредственно в цехах. Часто это промышленные и панельные рабочие станции с повышенной надёжностью.

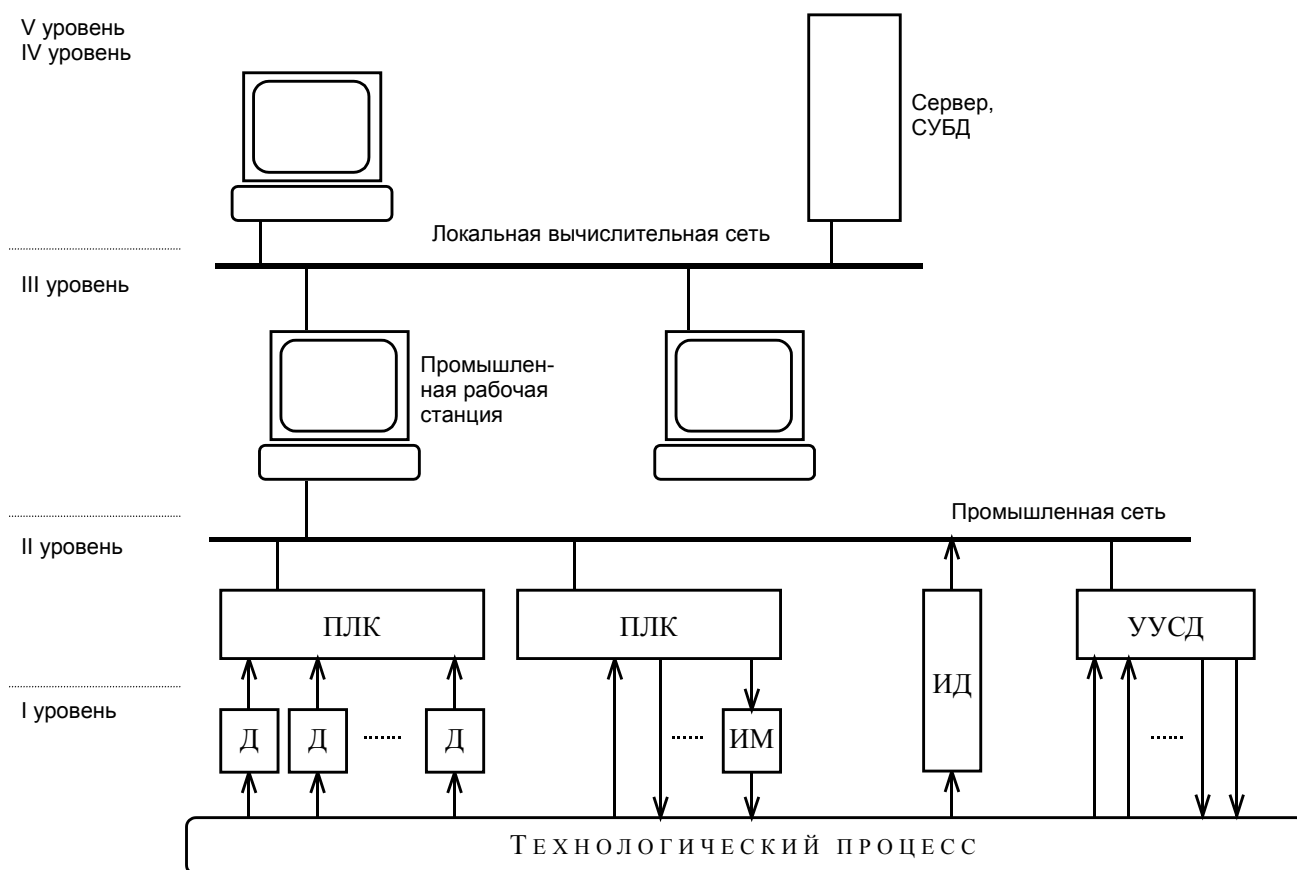
IV. Управление процессом производства. На этом уровне работают системы автоматизации управленческой и финансово-хозяйственной деятельности.

V. Планирование ресурсов предприятия (уровень стратегического управления).

Первые три уровня относят к автоматизированным системам управления технологическими процессами (АСУ ТП), последние два – к автоматизированным системам управления предприятием (АСУП).

В настоящее время вместо понятия АСУП используется более точное понятие «интегрированные системы планирования ресурсов предприятия» (Enterprise Resource Planning Systems – ERP-системы). Под ними понимают системы, в которых функционально объединяются существовавшие ранее как автономные системы для решения задач автоматизации учёта и управления производством, финансами, снабжением и сбытом, кадрами и информационными ресурсами. Техническую базу современных ERP-систем, использующих преимущественно распределённую архитектуру клиент-сервер, составляют серверы и рабочие места пользователей, объединённые локальными вычислительными сетями [32]. На российском рынке наиболее известными ERP-системами являются системы «1С», «Парус».

При разработке систем промышленной автоматизации решаются и аппаратные, и программные задачи: первый и частично второй уровень составляют аппаратную базу для программного обеспечения верхних слоев.



Д – датчик; ИМ – исполнительный механизм; ИД – интеллектуальный датчик; УУСД – устройство удалённого сбора данных; ПЛК – программируемый логический контроллер; СУБД – система управления базой данных

Рис. 3.2. Типовая структура системы управления предприятием

Чтобы повысить скорость устранения неполадок и монтажа оборудования в целом, каждое из устройств использующихся для построения АСУ ТП, относят к тому или иному уровню, для которого определены стандарты сопряжения с другими уровнями. Так для сопряжения I и II уровней определены стандартные уровни напряжения для дискретных сигналов и стандартные диапазоны изменения напряжения и тока для аналоговых. Для взаимодействия устройств II уровня между собой, а также с устройствами III уровня используется промышленная сеть, реализующая тот или иной стандарт организации сети. Для взаимодействия III, IV и V уровней используются стандарты, определяющие протоколы обмена между разными программами, а также способы доступа к информации, хранящейся в базе данных.

3.3. ОСНОВНЫЕ СЕТЕВЫЕ ТОПОЛОГИИ

Объединение нескольких устройств в одну сеть подразумевает, что все устройства имеют доступ к *среде передачи* и руководствуются общими для всех узлов данной сети правилами обмена информацией – *протоколом*. В качестве среды

передачи могут использоваться различные конструкции электрических кабелей – витая пара и коаксиальный кабель, оптические кабели со стеклянными или пластиковыми волокнами, радиоканал.

В зависимости от используемой среды передачи, могут применяться различные сетевые *топологии* – способы сетевого объединения устройств. Помимо топологии *физического* соединения узлов с помощью среды передачи необходимо отдельно рассматривать способ управления доступом к среде передачи. В зависимости от того, какая топология используется на физическом уровне, возможны различные способы организации управления доступом [33].

Топология «звезда»

В данной топологии вся информация передаётся через некоторый центральный узел. Каждое устройство имеет свою собственную среду соединения с центральным узлом. Все периферийные узлы могут обмениваться друг с другом только через центральный узел. Преимущество этой структуры в том, что никто другой не может влиять на среду передачи. С другой стороны, центральный узел должен быть исключительно надёжным устройством.

Топология «кольцо»

В кольцевой структуре информация передаётся от узла к узлу по физическому кольцу. Приёмник получает данные и передаёт их вместе со своей квитанцией подтверждения следующему устройству в сети. Когда начальный передатчик получает свою собственную квитанцию, это означает, что его информация была корректно получена адресатом.

Отказ в работе хотя бы одного узла приводит к нарушению работы кольца, а, следовательно, и к остановке всех передач. Чтобы этого избежать, необходимо включать в сеть автоматические переключатели, которые берут на себя инициативу, если узел перестал работать. То есть, они позволяют включать/выключать отдельные узлы без прерывания нормальной работы сети.

Топология «шина»

В шинной структуре все устройства подсоединены к общей среде передачи данных или шине. По концам шины устанавливаются согласующие сопротивления – терминаторы, предотвращающие отражение сигнала от конца шины. В отличие от «кольца» и «звезды» адресат получает свой информационный пакет без посредников.

Преимущества шинной топологии заключаются в простоте организации сети и низкой стоимости. Недостатком является низкая устойчивость к повреждениям – при любом обрыве кабеля вся сеть перестает работать, а поиск повреждения весьма затруднителен.

Управление доступом к среде передачи

Чаще всего, при организации промышленных сетей используется шинная топология, которая требует жесткой регламентации доступа к среде передачи (протокол доступа). Существует два метода регулирования такого доступа, известного ещё под термином «шинный арбитраж»:

- «фиксированный мастер» (централизованный контроль шины): доступ к шине контролируется центральным узлом – мастер-узлом;
- «плавающий мастер» (децентрализованный контроль шины): благодаря собственному интеллекту каждое устройство само определяет регламент доступа к шине.

В случае централизованного контроля за доступом к шине выделяется узел с правами *мастера* (master, ведущее устройство, шинный контроллер). Право инициировать циклы чтения/записи на шине имеет только мастер-узел. Он адресует каждого пассивного участника (slave, ведомое или подчинённое устройство), обеспечивает их данными и запрашивает у них данные. Если вдруг мастер «сломался», то и циклы обмена по шине останавливаются.

Именно по этой причине децентрализованный контроль с переходящими функциями мастера от одного узла сети к другому получил наибольшее внимание и развитие. Здесь права мастера назначаются группе устройств сети. Широко приняты и используются две модели децентрализованного доступа:

- модель случайного доступа к шине (CSMA/CD);
- модель с передачей маркера (Token Passing Model).

Случайный метод доступа к шине (CSMA/CD)

Известным механизмом управления сетью шинной конфигурации является метод множественного доступа с контролем несущей и обнаружением конфликтов (Carrier Sense Multiple Access with Collision Detection, CSMA/CD). Широко известна реализация этого метода – спецификация Ethernet.

Все узлы шины имеют право передавать данные. Каждый из них постоянно прослушивает шину. Если шина свободна, любой из участников сети может занять шину под свой цикл передач. В том случае, когда несколько узлов претендуют на шину одновременно, это приводит к так называемому конфликту (коллизии), и тогда все «претенденты снимают свою заявку».

Затем каждый из участников включает генератор случайных чисел, который задаёт случайный интервал ожидания до следующего момента запроса шины.

Метод CSMA/CD получил широкое распространение в офисных системах и наиболее эффективен в условиях относительно низкой общей загрузки канала пе-

редачи данных (менее 30%). В условиях большей загрузки канала выгоднее использовать сети, реализующие модель с передачей маркера.

Метод передачи маркера (The Token Passing Method)

В этом методе маркер (token), специальный пакет данных, владелец которого имеет право инициировать циклы обмена по сети, передаётся от узла к узлу *по логическому кольцу*.

Этот метод предлагает каждому участнику сети «справедливое» распределение сетевых ресурсов в соответствии с их запросами. Принцип передачи маркера используется в системах, где реакция на события, возникающие в распределённой системе, должна проявляться за определённое время (системы реального времени).

В кольцевой топологии невозможно применение централизованного контроля. Единственно возможный в данной топологии метод управления доступом – метод передачи маркера.

В топологии «звезда», в зависимости от способа физического подключения устройств к центральному узлу возможно применение как централизованного, так и децентрализованного метода управления доступом к среде передачи. Если центральный узел выступает в роли фиксированного мастера сети, то каждый канал связи «периферийный узел – центральный узел» можно рассматривать как независимое соединение двух устройств. Такое соединение называют «точка – точка» (point-to-point, PtP). Возможность и необходимость обмена данными между периферийными узлами, подключёнными к центральному узлу, определяется исключительно программным обеспечением центрального узла.

Если же центральный узел выступает в роли пассивного объединительного устройства (концентратор, hub), служащего для электрического подключения узлов сети к единой среде передачи, то в этом случае «звезда» фактически становится «шиной», и для неё тогда применимы все способы шинного арбитража.

Основные виды топологий можно сравнить по трём основным критериям [33]:

- режим доступа к сети;
- надёжность сети;
- возможность изменения числа узлов сети.

Сравнение основных топологий представлено в табл. 3.1.

Сравнительные характеристики основных топологий

Сравнительные характеристики	Звезда	Кольцо	Шина
1. Режим доступа	Доступ через центральный узел.	Децентрализованный доступ. Доступ от узла к узлу.	Возможен централизованный и децентрализованный доступ.
2. Надёжность	Сбой центрального узла – сбой всей сети.	Сбой узла – сбой всей сети.	Сбой любого узла не приводит к сбою всей сети. ¹
3. Расширяемость	Ограничено числом физических портов на центральном узле	Возможно увеличение числа узлов, но время ответа увеличивается	Возможно увеличение числа узлов.

¹Кроме мастер-узла при централизованном контроле за доступом к шине

3.4. МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ

Объединение в одну цифровую сеть нескольких устройств – это только начальный шаг к эффективной и надёжной работе системы связи между ними. В дополнение к аппаратным требованиям предъявляется также ряд программных требований. Там, где системы связи, или сети, гомогенные (однородные), то есть объединяют устройства одного производителя, эти проблемы, как правило, решены. Но когда речь идет о построении сети из устройств различных производителей – эти проблемы обретают множественный характер [33].

Системы, являющиеся уникальными (т.е. их делает и поддерживает только один производитель), работающие по уникальным протоколам связи, получили название «закрытых систем» (closed/proprietary systems), большинство таких систем зародилось во времена, когда проблема интеграции изделий других производителей не считалась актуальной.

«Открытые системы» (open systems) приводят специфические требования в соответствие интересам всех. Только при использовании принципов открытых систем интеграция изделий разных производителей в одну сеть может быть решена без особых проблем.

В 1978 году Международной организацией по стандартизации (ISO, International Standards Organization) в противовес закрытым сетевым системам и с целью разрешения проблемы взаимодействия открытых систем с различными видами вычислительного оборудования и различающимися стандартами протоколов была предложена «Описательная модель взаимодействия открытых систем» (модель BOC, OSI-модель, Open System Interconnection model). Модель BOC предлагает

структуру для идентификации и разграничения различных составляющих коммуникационного процесса.

Модель ВОС не связана с конкретными реализациями и описывает коммуникационный процесс в абстрактных понятиях. Модель ВОС – это концептуальная модель процесса коммуникации, основанная на разбиении этого процесса на несколько функциональных уровней, каждый из которых взаимодействует только со своими непосредственными соседями. Такой подход позволяет предоставлять услуги, скрывая при этом механизм реализации, а значит обеспечить определённую степень совместимости и взаимозаменяемости.

Ниже представлены все уровни и функции этой модели [1, 33, 34].

1) Физический уровень (Physical Link Layer)

Представляет собой физическую среду передачи – электрическую или оптическую – с соответствующими интерфейсами к сопрягаемым объектам, которые называются **станциями** (stations) или **узлами** (nodes). Все вопросы, касающиеся среды передачи, уровней сигналов и частот, кодирования данных, методов передачи, формы и типов разъёмов и т.п. рассматриваются на этом уровне. Физический уровень является единственной материальной связью между узлами.

2) Уровень канала данных (Data Link Layer)

Реализует функции, связанные с формированием и передачей **кадров** (frames) от одного узла к другому, обнаружением и исправлением ошибок, возникающих на физическом уровне. При появлении ошибки, например, из-за помех на линии, на этом уровне запрашивается повторная передача повреждённого кадра. В результате канальный уровень обеспечивает верхние уровни услугами по безошибочной передаче данных между узлами. Если несколько устройств используют общую среду передачи, то на этом уровне также осуществляется управление доступом к среде. Обычно функции этого уровня реализованы в сетевом адаптере.

3) Сетевой уровень (Network Layer)

Устанавливает маршрут и контролирует прохождение сообщений от источника к узлу назначения. Маршрут может состоять из нескольких физических сегментов сетей, не связанных непосредственно. На данном уровне работают маршрутизаторы.

4) Транспортный уровень (Transport Layer)

Управляет доставкой сообщений от источника к приёмнику. Этот уровень представляет собой интерфейс между прикладным программным обеспечением, запрашивающим передачу данных, и физической сетью, представленной первыми тремя уровнями. Одна из главных задач транспортного уровня – обеспечить независимость верхних уровней от физической структуры сети, в частности от мар-

шрута доставки сообщений. Транспортный уровень несёт ответственность за проверку правильности передачи данных от источника к приёмнику и доставку данных к прикладным программам.

5) Сеансовый уровень (Session Layer)

Отвечает за установку, поддержку синхронизации и управление соединением (сеансом связи, диалогом) между объектами уровня представления данных. На этом уровне, в частности происходит удалённая регистрация в сети.

6) Уровень представления данных (Presentation Layer)

Обеспечивает синтаксическую модель данных, т.е. кодирование и преобразование неструктурированного потока бит с предыдущего уровня в формат, понятный приложению-получателю на следующем уровне или, иначе говоря, восстановление исходного формата данных – сообщение, текст, рисунок и т.п.

Задачей уровня является трансляция данных из одного формата в другие, сжатие/распаковка данных и их шифровка/расшифровка при необходимости. Этот уровень включает функции дисковой и сетевой операционных систем.

7) Уровень приложений (Application Layer).

Самый верхний уровень, на котором решаются, собственно, прикладные задачи. На этом уровне работают приложения, с которыми имеет дело пользователь. Этот уровень является наивысшим и, в то же время единственным не полностью скрытым от пользователя.

В процессе передачи сообщение проходит от уровня 7 до уровня 1 передающей системы, причем каждый уровень добавляет к нему свой заголовок или подвергает его какой-либо обработке. Кадры, составляющие сообщение, передаются через среду связи в принимающую систему, где, проходя через уровни от первого до седьмого, они лишаются заголовков и вновь собираются в сообщение.

Физический уровень – единственный имеющий материальное воплощение (аппаратную реализацию). Остальные уровни представляют собой наборы правил или описания вызовов функций, реализованные программными средствами. Три нижних уровня называются сетевыми или коммуникационными уровнями, так как они отвечают за доставку сообщений. Три верхних уровня относятся к прикладному программному обеспечению и связаны с содержательной стороной сообщений. Четвертый, транспортный уровень, осуществляет связь между коммуникационно-ориентированными и проблемно-ориентированными уровнями.

Все, что находится выше седьмого уровня модели, это задачи, решаемые в прикладных программах. Идея семиуровневого открытого соединения состоит не в попытке создания универсального множества протоколов связи, а в обеспечении

«модели», в рамках которой могут быть использованы уже существующие различные протоколы.

Основная идея модели ВОС довольно проста [1]. Два объекта, находящиеся на одном уровне, соединяются виртуальной (логической) связью. Для объектов виртуальная связь представляется реальным каналом связи, хотя виртуальное и физическое соединения совпадают только на первом уровне. Объекты обмениваются данными в соответствии с протоколом, определённым для их уровня. На самом деле объекты запрашивают услуги непосредственно у нижележащего уровня с помощью вызова процедур (рис. 3.3), при этом внутренние механизмы недоступны запрашивающему объекту и могут измениться в любой момент без его уведомления. Объект каждого уровня может взаимодействовать только с объектами своего уровня или с соседями.

Набор правил, определяющих начало, проведение и окончание процесса связи между одноранговыми объектами называется протоколом. Сообщения, которыми обмениваются одноранговые объекты, содержат либо пользовательские данные, либо являются протокольными (управляющими) сообщениями. Перед передачей на следующий, нижележащий, уровень к сообщению добавляется управляющая информация – заголовок уровня – в соответствии с протоколом, принятом на данном уровне. Результат напоминает матрёшек, которые вкладываются друг в друга (рис. 3.4). Самая маленькая матрёшка соответствует исходному сообщению, т.е. прикладным данным, самая большая – тому, что в действительности передаётся по физическому тракту. Протоколы и вызовы процедур описаны в документах модели ВОС и соответствующих стандартах с указанием конкретного синтаксиса каждой функции и её параметров.

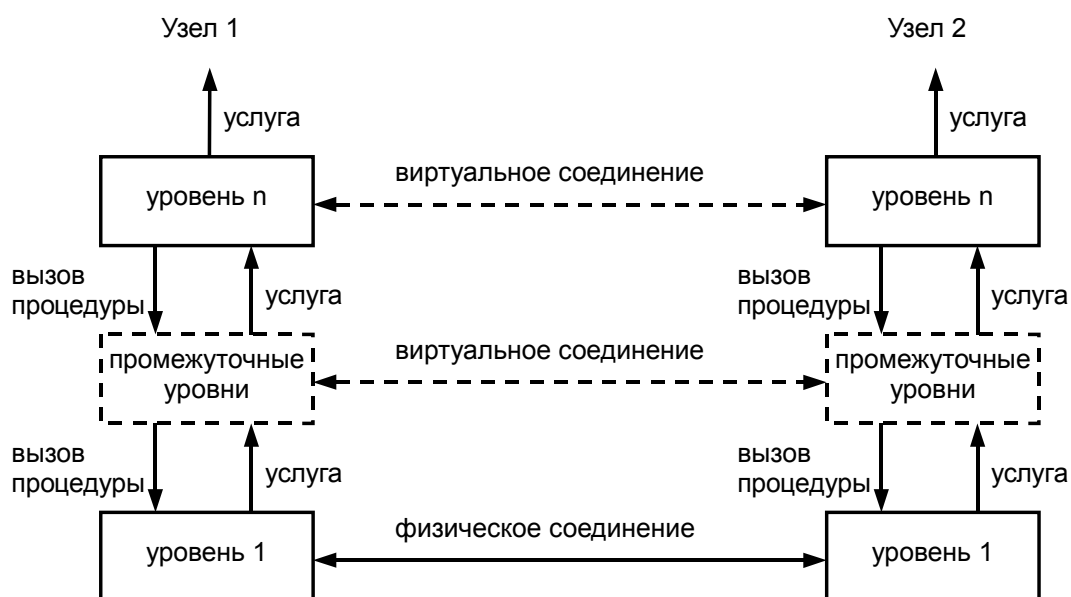


Рис. 3.3. Виртуальное соединение одноранговых объектов в модели ВОС



CRC – циклический избыточный код или контрольная сумма

Рис. 3.4. Общая схема передачи информации между уровнями

Большинство промышленных сетей поддерживают 1-ый, 2-ой и 7-ой уровни модели ВОС: физический уровень, уровень передачи данных и уровень приложений. Все другие уровни, как правило, избыточны.

3.5. УРОВЕНЬ ПРИЛОЖЕНИЙ В ПРОМЫШЛЕННЫХ СЕТЯХ

Уровень приложений обычно реализуется процедурами и функциями, доступными программисту при разработке программы ПЛК. Это в первую очередь функции приёма и передачи данных, а также специальные функции диагностики, как самой сети, так и подключаемых к ней устройств. Вообще же функциональность уровня приложений промышленных сетей довольно часто зависит от функциональности объединяемых сетью устройств. При узкой специализации промышленной сети набор команд может учитывать смысловую нагрузку передаваемых данных.

Например, возможен вариант, когда каждый узел сети рассматривается как набор доступных для чтения и записи адресуемых ячеек памяти (регистров). В этом случае обмен информацией выполняется примерно такими командами: «узлу номер N записать в регистр с номером A значение X », «считать значение из регистра с номером A узла номер N ». Такой подход можно считать достаточно универсальным. Он применяется в семействе протоколов Modbus.

Возможно также использование менее универсальных команд, команд учитывающих специфику работы адресуемого узла, например, «считать из узла с номером N значение главной измеряемой величины», «записать в узел N новое значение уставки регулируемой величины». Такие протоколы могут быть универ-

сальными для определённого класса устройств (например, HART-протокол), а могут быть ориентированы только на конкретные устройства отдельного взятого производителя. В этом случае, производитель фактически разрабатывает свой собственный протокол обмена.

3.6. ПРОТОКОЛЫ КАНАЛЬНОГО УРОВНЯ

Построение сети начинается с организации физического тракта передачи данных. Однако, создание работоспособного физического тракта передачи – это первый шаг в построении надёжной коммуникационной системы. Второй шаг – гарантировать доставку данных без искажений. Помимо этого при подключении более чем одного передатчика к одному и тому же физическому тракту должен быть упорядочен их доступ к общему каналу. Все эти задачи рассматриваются на канальном уровне модели ВОС.

Физический уровень не обеспечивает защиту данных от помех и восстановление повреждённых данных. Контроль правильности полученных данных осуществляется на уровнях выше физического и, в первую очередь, на канальном. Для этого применяются специальные протоколы и дополнительная информация в пакетах данных.

Канальный уровень обычно делится на два подуровня: MAC (Media Access Control – управление доступом к среде) и LLC (Logical Link Control – управление логическим звеном данных). Подуровень MAC отвечает за организацию доступа нескольких передатчиков к одному физическому тракту передачи. Подуровень LLC обеспечивает установку и поддержку соединения для передачи кадров данных от одного устройства к другому.

Последовательная передача данных

Первым этапом появления промышленных сетей можно считать момент, когда для передачи измерительной информации от датчиков к УВМ стали использоваться цифровые каналы связи. Набор правил, описывающих взаимодействие двух электронных устройств, вычислительных систем или программ называется *интерфейсом (interface)*. Интерфейс описывает элементы соединения и вспомогательные схемы управления, используемые для соединения устройств [16].

Для интерфейса, соединяющего (физически или логически) два устройства, различают три возможных режима обмена: дуплексный, полудуплексный и симплексный. *Дуплексный* режим позволяет по одному каналу связи одновременно передавать информацию в обоих направлениях. Он может быть асимметричным, если пропускная способность в направлениях «туда» и «обратно» имеет существенно различающиеся значения, или симметричным. *Полудуплексный* режим по-

зволяет передавать информацию «туда» и «обратно» поочередно, при этом интерфейс имеет средства переключения направления канала. *Симплексный* (одно-сторонний) режим предусматривает только одно направление передачи информации (во встречном направлении передаются только вспомогательные сигналы интерфейса).

По способу передачи информации интерфейсы подразделяются на параллельные и последовательные. В *параллельном* интерфейсе все биты передаваемого слова (обычно байта) выставляются и передаются по соответствующим параллельно идущим проводам одновременно. Последовательный интерфейс для передачи данных использует одну сигнальную линию, по которой информационные биты передаются друг за другом последовательно, отсюда и название интерфейса.

При рассмотрении интерфейсов важным параметром является *пропускная способность*. Технический прогресс приводит к неуклонному росту объёмов передаваемой информации. Вполне очевидно, что при одинаковом быстродействии приёмопередающих цепей и пропускной способности соединительных линий по скорости передачи параллельный интерфейс должен превосходить последовательный. Однако повышение производительности за счёт увеличения тактовой частоты передачи данных упирается в волновые свойства соединительных кабелей. В случае параллельного интерфейса начинают сказываться задержки сигналов при их прохождении по линиям кабеля, и, что самое неприятное, задержки в разных линиях интерфейса могут быть различными вследствие неидентичности проводов и контактов разъёмов. Для надёжной передачи данных временные диаграммы обмена строятся с учетом возможного разброса времени прохождения сигналов, что является одним из факторов, сдерживающих рост пропускной способности параллельных интерфейсов. В последовательных интерфейсах, конечно же, есть свои проблемы повышения производительности, но, поскольку в них используется меньшее число линий (в пределе – одна), то повышение пропускной способности линий связи обходится дешевле.

Другим немаловажным параметром интерфейса является *допустимое удаление* соединяемых устройств. Оно ограничивается как частотными свойствами кабелей, так и помехозащищённостью интерфейсов. Часть помех возникает от соседних линий интерфейса – это перекрёстные помехи, защитой от которых может быть применение витых пар проводов для каждой линии. Другая часть помех вызывается искажением уровней сигналов.

Последовательная передача позволяет сократить количество сигнальных линий и увеличить дальность связи, что является определяющим фактором для применения последовательных интерфейсов на физическом уровне промышленных сетей.

Кодирование бит

Существуют два основных способа передачи битовой последовательности по физическому каналу [1]:

- посылка бит в линию в непосредственном или закодированном виде при сохранении цифрового характера данных;
- модуляция несущей по амплитуде/частоте/фазе и передача модулированного сигнала.

Передача данных непосредственным способом

Непосредственный способ передачи цифровых данных является наиболее простым. При непосредственном кодировании, например, уровень напряжения 0 В представляет логический «0», а +10 В – логическую «1» (рис. 3.5). Возможно использование обратной кодировки – при «0» линия находится под высоким напряжением, а «1» соответствует низкий уровень напряжения. Широко используется полярное кодирование – сигналы, соответствующие «0» и «1» имеют противоположные знаки по отношению к общей базе (например, логическому нулю соответствует +10 В, а логической единице соответствует –10 В). Прямое, обратное и полярное кодирования называются кодированием без возвращения к нулю (Non-Return to Zero – NRZ), так как в нём отсутствует обязательный переход к нулевому уровню. Последовательность единиц будет поддерживать линию при постоянном высоком или низком потенциале в соответствии с принятой схемой кодирования.

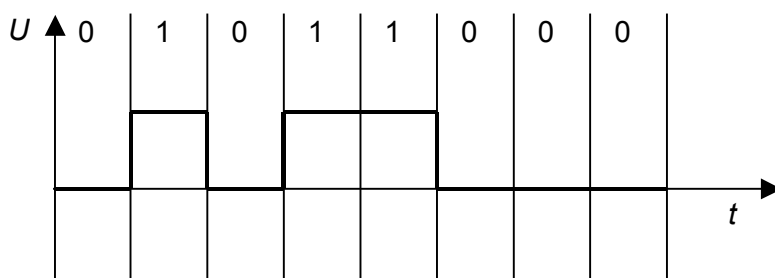


Рис. 3.5. Прямое двоичное кодирование без возвращения к нулю

Метод кодирования без возвращения к нулю прост, но чувствителен к помехам и искажениям. Вдобавок, при непосредственном применении NRZ-кодирования возникает ещё одна существенная проблема. Приёмник не может различить, где начинается и кончается каждый отдельный бит. Иными словами, чистое NRZ-кодирование не позволяет отличить отсутствие сообщения от последовательности несущих информацию единиц. Кроме того, непосредственно по принимаемому сигналу приёмник не может определить ширину битового интервала и, соответственно, скорость передачи. Возможное решение – предварять ка-

ждое сообщение специальной преамбулой – последовательностью чередующихся нулей и единиц, обеспечивающих синхронизацию передатчика и приёмника. Однако при этом остается риск потери синхронизации в процессе передачи данных и их неправильной интерпретации. Наконец, если все импульсы имеют одинаковую полярность, распределённая ёмкость линии ведёт к накоплению постоянного электрического потенциала.

Все указанные проблемы решаются при кодировании с возвращением к нулю (Return to Zero – RZ) (рис. 3.6). В этом случае исходные данные комбинируются с сигналом синхронизации. Как и при прямом кодировании, здесь также определены два уровня потенциала, один из которых соответствует логическому нулю, а второй – логической единице. Каждый бит начинается с определённого для его значения уровня потенциала, а в середине каждого битового интервала осуществляется переход на нулевой уровень; фронт перехода используется для синхронизации приёмника. При другом типе RZ-кодирования применяются сигналы разной полярности (рис. 3.7). RZ-кодирование требует в два раза более широкой полосы, чем NRZ-кодирование, и соответствующая электроника интерфейсных устройств гораздо сложнее, однако достоинства метода отодвигают эти недостатки на задний план.

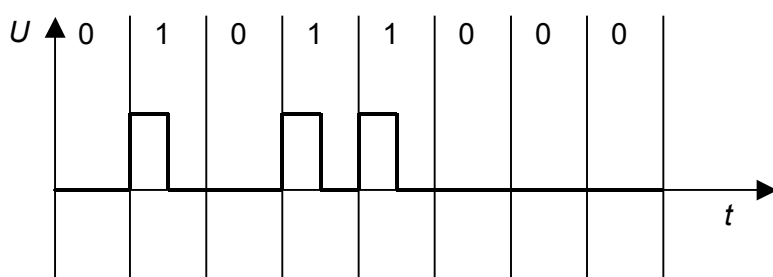


Рис. 3.6. Прямое двоичное кодирование с возвращением к нулю

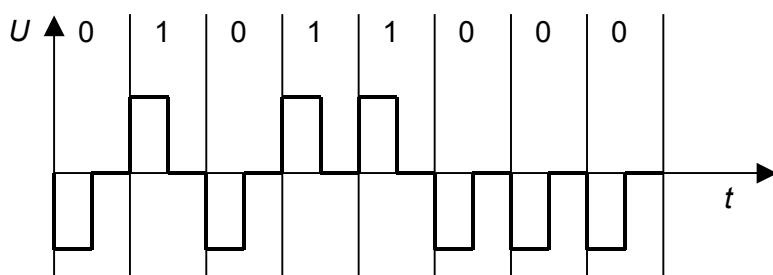


Рис. 3.7. Трехуровневое кодирование с возвращением к нулю

Другой широко применяемый способ – это манчестерское или двухфазное кодирование (Bi-phase Level – BiФ-L). При манчестерском кодировании каждый бит кодируется двумя уровнями напряжения с переходом в середине каждого би-

тового интервала. При прямом манчестерском кодировании бит «0» представляется переходом от уровня низкого напряжения к высокому, а бит «1» – переходом от высокого уровня к низкому (рис. 3.8). Похожая схема используется при дифференциальном манчестерском кодировании, при котором бит «0» представляется переходом уровня в начале каждого нового битового интервала, а бит «1» – отсутствием такого перехода (рис. 3.9). В середине битового интервала переход осуществляется всегда.

Манчестерское и дифференциальное манчестерское кодирования являются самосинхронизирующимися (т.е. позволяют приёмнику настроиться на скорость передатчика без специальных синхросигналов) и имеют более высокую защиту от шума по сравнению с RZ-кодированием. Как и для кодирования с возвращением к нулю манчестерские схемы требуют в два раза более широкой полосы пропускания, чем NRZ-кодирование. Преимущество манчестерского кодирования в том, что оно использует два уровня напряжения вместо трёх и соответствующая аппаратура проще, чем при RZ-кодировании.

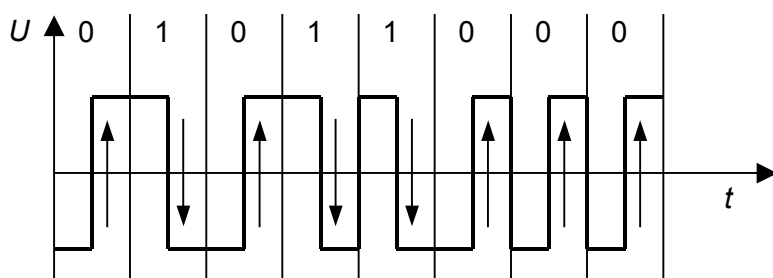


Рис. 3.8. Прямое манчестерское кодирование

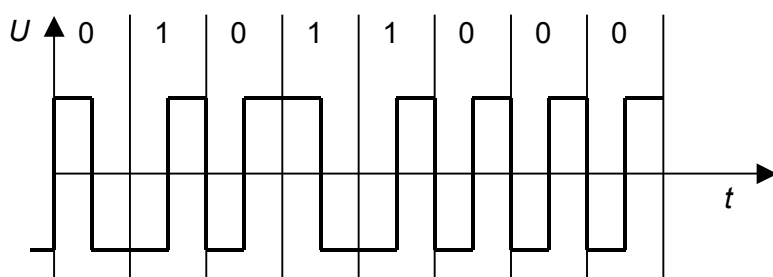


Рис. 3.9. Дифференциальное манчестерское кодирование

Важная характеристика манчестерского кода — отсутствие у сигнала постоянной составляющей при передаче длинной последовательности единиц или нулей. Благодаря этому передатчики и приёмники можно «развязать» гальванически с помощью импульсных трансформаторов. Спектр сигнала при манчестерском кодировании содержит только две частотные составляющие. Например, для передачи на скорости 10 Мбит/с, это 10 МГц при передаче последовательности одних

нулей или единиц и 5 МГц при их чередовании. Поэтому все другие частоты можно удалить с помощью полосовых фильтров.

Манчестерское кодирование широко применяется в локальных сетях, например в Ethernet.

Передача данных методом модуляции несущей

Модуляция несущей (carrier modulation) используется для согласования информационного сигнала с линией, по которой он передаётся. Модуляция представляет собой изменение некоторых параметров (амплитуды, частоты, фазы) высокочастотной несущей в зависимости от исходного информационного сигнала; в качестве несущей используется высокочастотный синусоидальный или импульсный сигнал. Приёмник выделяет исходный сигнал из модулированного.

Амплитудная модуляция редко используется при передаче цифровых данных. Более распространены частотная и фазовая модуляция. При применении каждого из видов модуляции соответствующий параметр несущей изменяется как функция входного сигнала. При амплитудной модуляции каждый уровень напряжения исходного сигнала соответствует определённой амплитуде несущей, при частотной модуляции – частоте в определённом диапазоне, а при фазовой – определённому фазовому сдвигу несущего сигнала.

Синхронизация

Для правильной обработки сообщений необходимо, чтобы передатчик и приёмник были синхронизированы, т.е. приёмник, принимая сигнал из линии, должен «знать» где располагаются границы битовых интервалов, чтобы правильно декодировать информацию. Синхронизирующий сигнал передаётся либо по специальному проводу, либо вместе с передаваемой цифровой информацией, как, например, при RZ- или манчестерском кодировании, в которых изменение сигнала в середине каждого битового интервала представляет собой синхронизирующий сигнал. В первом случае требуется дополнительный провод, во втором – дополнительная полоса пропускания.

Передача может осуществляться и без сигналов синхронизации. Такая передача называется асинхронной. При асинхронной передаче приёмник должен заранее знать все параметры связи – в первую очередь, скорость – для того, чтобы правильно идентифицировать поступающие сигналы. Более того, приёмник должен различать границы между отдельными битами и соответственно настраивать свой отсчёт времени.

Асинхронная передача данных

При **асинхронной** передаче каждому байту предшествует **старт-бит**, сигнализирующий приёмнику о начале посылки, за которым следуют **биты данных** и, возможно, **бит паритета** (бит чётности, parity – чётность). Завершает посылку **стоп-бит**, гарантирующий паузу между посылками (рис. 3.10). Старт-бит следующего байта посылается в любой момент после стоп-бита, то есть между передачами возможны паузы произвольной длительности. Старт-бит, имеющий всегда строго определённое значение (логический 0), обеспечивает простой механизм синхронизации приёмника по сигналу от передатчика. Подразумевается, что приёмник и передатчик работают на одной скорости обмена. Внутренний генератор синхронизации приёмника использует счётчик-делитель опорной частоты, обнуляемый в момент приёма начала старт-бита. Этот счётчик генерирует внутренние стробы, по которым приёмник фиксирует последующие принимаемые биты. В идеале стробы располагаются в середине битовых интервалов, что позволяет принимать данные и при незначительном рассогласовании скоростей приёмника и передатчика [16].



Рис. 3.10. Формат асинхронной передачи

Очевидно, что при передаче восьми бит данных, одного контрольного и одного стоп-бита предельно допустимое рассогласование скоростей, при котором данные будут распознаны верно, не может превышать 5%. С учетом фазовых искажений и дискретности работы внутреннего счётчика синхронизации реально допустимо меньшее отклонение частот. Чем меньше коэффициент деления опорной частоты внутреннего генератора (чем выше частота передачи), тем больше погрешность привязки стробов к середине битового интервала, и требования к согласованности частот становятся более строгими. Чем выше частота передачи, тем больше влияние искажений фронтов на фазу принимаемого сигнала. Взаимодействие этих факторов приводит к повышению требований к согласованности частот приёмника и передатчика с ростом частоты обмена.

Для асинхронного режима принят ряд **стандартных скоростей обмена**: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19 200, 38 400, 57 600 и 115 200 бит/с. Иногда вместо единицы измерения "бит/с" используют "бод" (baud).

Количество бит данных может составлять 5, 6, 7 или 8 (5- и 6-битные форматы распространены незначительно).

Количество стоп-бит может быть 1, 1,5 или 2 («полтора бита» означает только длительность стопового интервала).

Бит чётности дополняет количество единичных бит данных до чётного или до нечётного в зависимости от принятого соглашения.

Количество бит данных, длительность стоп-бита, наличие и способ обработки бита чётности – настраиваемые параметры асинхронной передачи.

Посимвольную передачу просто реализовать благодаря существованию специальной микросхемы – универсального асинхронного приёмопередатчика (Universal Asynchronous Receiver Transmitter – UART). Микросхемы UART выполняют преобразование параллельного кода в последовательный при передаче и обратное преобразование при приёме данных.

Метод асинхронной последовательной передачи байтов используется в протоколах передачи символов – байт-ориентированных протоколов [1]. В этом случае по промышленной сети передаётся строка байт с некоторыми ограничивающими управляющими символами. К таким протоколам относятся протоколы HART, Modbus, PROFIBUS.

Возможно также использование бит-ориентированных протоколов, когда содержание каждого передаваемого сообщения представляет собой строку бит определённой длины, которая может изменяться в зависимости от типа сообщения или длины передаваемых данных. Интерпретация значений битов зависит от стандарта протокола промышленной сети. В качестве примера бит-ориентированных протоколов можно привести протоколы AS-interface, CAN.

3.7. СТАНДАРТЫ ЭЛЕКТРИЧЕСКИХ ИНТЕРФЕЙСОВ ПРИМЕНЯЕМЫХ В ПРОМЫШЛЕННЫХ СЕТЯХ

Последовательный интерфейс имеет различные реализации, различающиеся способом передачи электрических сигналов. Существует ряд родственных международных стандартов: RS-232C, RS-423A, RS-422A и RS-485. На рис. 3.11 приведены схемы соединения приёмников и передатчиков, а также показаны ограничения на длину линии L и максимальную скорость передачи данных V [16].

Несимметричные линии интерфейсов RS-232C и RS-423A имеют самую низкую защищённость от синфазной помехи, хотя дифференциальный вход приёмника RS-423A несколько смягчает ситуацию. Лучшие параметры имеет двухто-

чечный интерфейс RS-422A и его магистральный (шинный) аналог RS-485, работающие на симметричных линиях связи. В них для передачи каждого сигнала используются дифференциальные сигналы с отдельной (витой) парой проводов.

В перечисленных стандартах сигнал представляется потенциалом. Существуют последовательные интерфейсы, где информативен ток, протекающий по общей цепи передатчик-приёмник – «токовая петля» и MIDI. Для связи на короткие расстояния приняты стандарты беспроводной инфракрасной связи. Наибольшее распространение в персональных компьютерах получил простейший из перечисленных – стандарт RS-232C, реализуемый COM-портами. В промышленной автоматике широко применяется RS-485, а также RS-422A. Существуют преобразователи сигналов для согласования этих родственных интерфейсов.

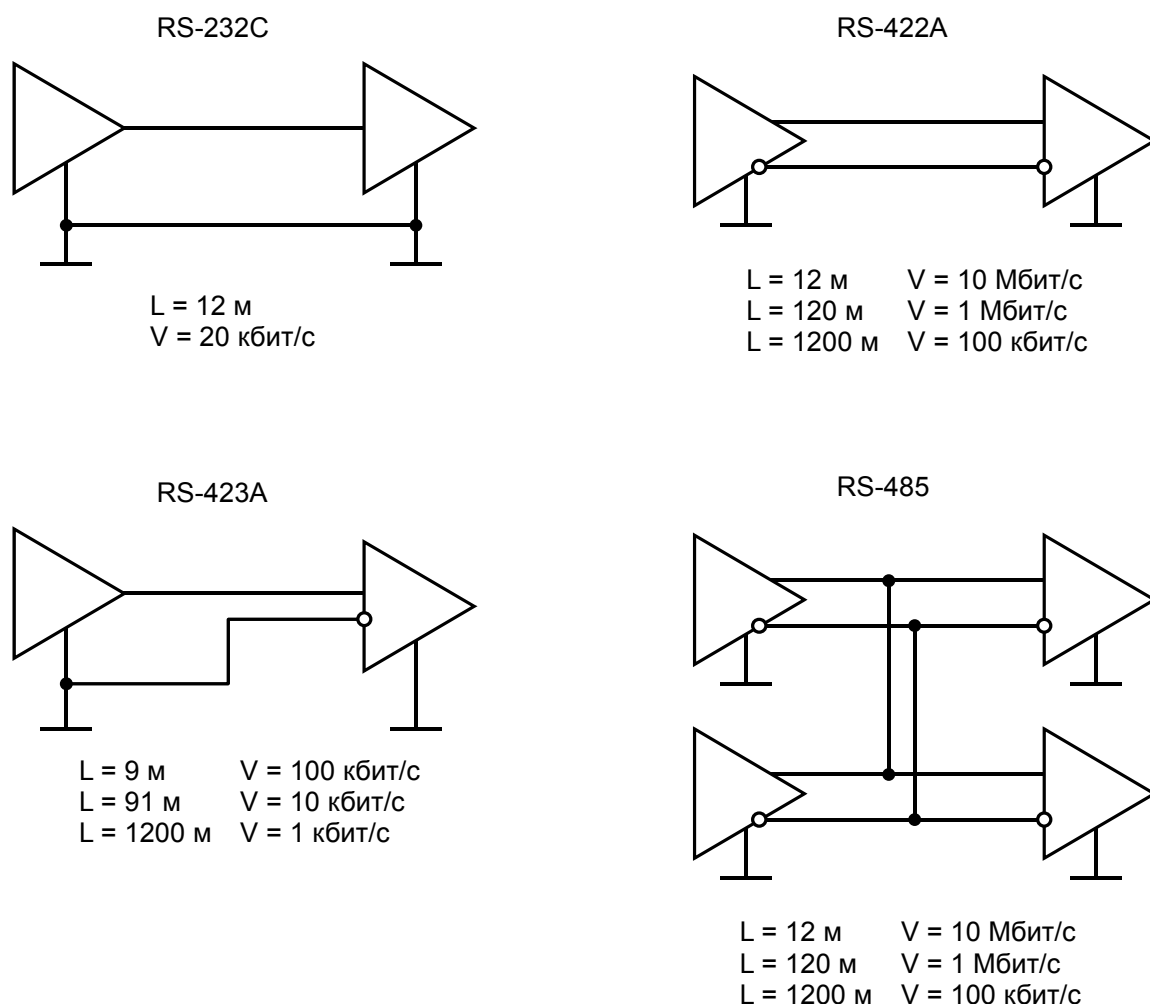


Рис. 3.11. Стандарты последовательного интерфейса

Интерфейс RS-232C (EIA-232-D)

Стандарт интерфейса RS-232C (RS от Recommended Standard – рекомендуемый стандарт) является наиболее известным и распространённым среди стандар-

тов для последовательной передачи данных. Он был введён в 1969 году Ассоциацией электронной промышленности (Electronic Industries Association, EIA) для описания требований к интерфейсу между ЭВМ и модемами. Его последняя редакция, при которой название было изменено с RS-232 на EIA-232-D, датируется декабрем 1987 года.

Интерфейс предназначен для подключения аппаратуры, передающей или принимающей данные (АПД) к оконечной аппаратуре каналов данных (АКД) [16]. В роли АПД может выступать компьютер, принтер, плоттер и другое периферийное оборудование. В роли АКД обычно выступает модем. Конечной целью подключения является соединение двух устройств АПД. Полная схема соединения приведена на рис. 3.12. Интерфейс позволяет исключить канал удалённой связи вместе с парой устройств АПД, соединив устройства непосредственно с помощью нуль-модемного кабеля (рис. 3.13).



Рис. 3.12. Полная схема соединения по RS-232

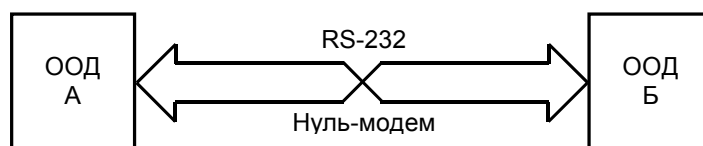


Рис. 3.13. Соединение по RS-232 нуль-модемным кабелем.

Стандарт RS-232 описывает управляющие сигналы интерфейса, пересылку данных, электрический интерфейс и типы разъёмов. В стандарте предусмотрены асинхронный и синхронный режимы обмена, но COM-порты поддерживают только асинхронный режим.

Электрический интерфейс

Стандарт RS-232 использует несимметричные передатчики и приёмники – сигнал передаётся относительно общего провода – схемной земли [16]. Интерфейс не обеспечивает гальванической развязки устройств. Логической единице соответствует напряжение на входе приёмника в диапазоне $-12...-3$ В. Логическому нулю соответствует диапазон $+3...+12$ В. Диапазон $-3...+3$ В – зона нечувствительности, обуславливающая гистерезис приёмника: состояние линии будет считаться изменённым только после пересечения порога (рис. 3.14). Уровни сигналов на выходах передатчиков должны быть в диапазонах $-12...-5$ В и $+5...+12$ В для

представления единицы и нуля соответственно. Разность потенциалов между схемными землями соединяемых устройств должна быть менее 2 В, при более высокой разности потенциалов возможно неверное восприятие сигналов.

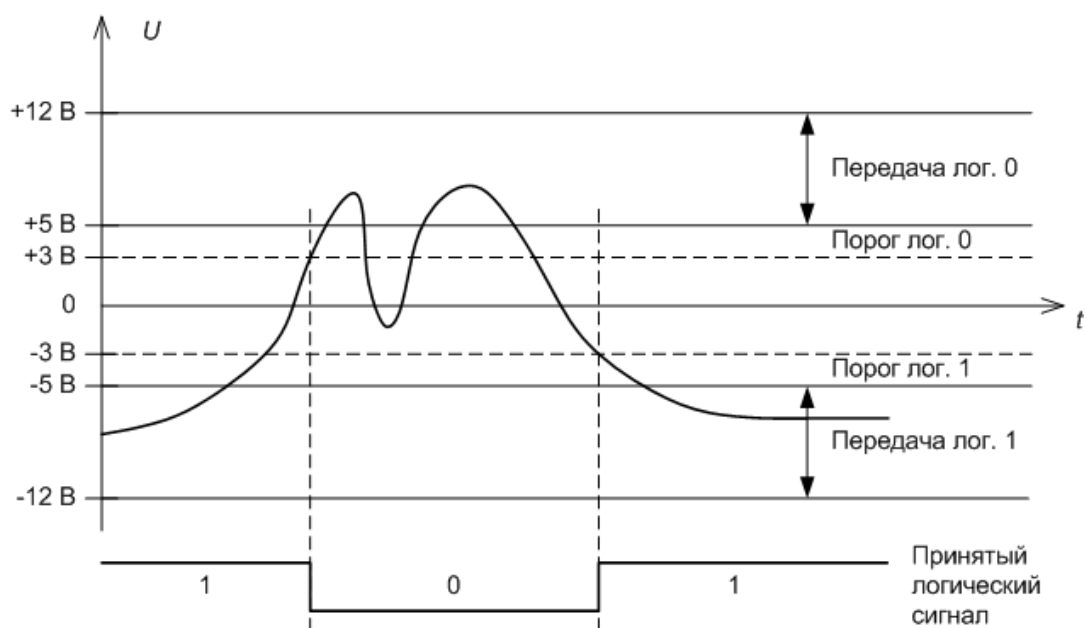


Рис. 3.14. Приём сигналов RS-232

Подключение и отключение интерфейсных кабелей устройств с автономным питанием должно производиться при отключенном питании. Иначе разность невыровненных потенциалов устройств в момент коммутации может оказаться приложенной к выходным или входным (что опаснее) цепям интерфейса и вывести из строя микросхемы.

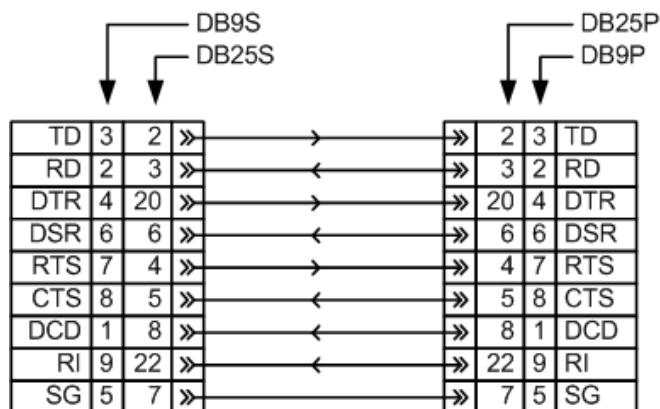
Стандарт RS-232 регламентирует типы применяемых разъёмов. На аппаратуре АПД (в том числе на COM-портах) принято устанавливать вилки (male – «папа») DB-25P или более компактный вариант – DB-9P. Девятиштырьковые разъёмы не имеют контактов для дополнительных сигналов, необходимых для синхронного режима (в большинстве 25-тиштырьковых разъёмов эти контакты не используются).

На аппаратуре АКД (модемах) устанавливают розетки (female – «мама») DB-25S или DB-9S.

Это правило предполагает, что разъёмы АКД могут подключаться к разъёмам АПД непосредственно или через переходные «прямые» кабели с розеткой и вилкой, у которых контакты соединены «один в один». Переходные кабели могут являться и переходниками с 9-ти на 25-тиштырьковые разъёмы (рис. 3.15).

Если аппаратура АПД соединяется без модемов, то разъёмы устройств (вилки) соединяются между собой нуль-модемным кабелем (Zero-modem или Z-modem), имеющим на обоих концах розетки, контакты которых соединяются пе-

рекрёстно по одной из схем, приведённых на рис. 3.16. Чаще всего в системах промышленной автоматизации используется минимальный нуль-модемный кабель (рис. 3.16, а), позволяющий реализовать двунаправленный асинхронный обмен. Остальные сигналы, служащие для управления потоком данных, диагностики состояний модема и передатчика не используются.



SG (Signal Ground) – сигнальная земля; TD (Transmit Data) – передаваемые данные (выход передатчика); RD (Receive Data) – принимаемые данные (вход приёмника); RTS (Request To Send) – запрос от АПД на передачу данных; CTS (Clear To Send) – разрешение АПД передавать данные; DSR (Data Set Ready) – сигнал готовности от АКД; DTR (Data Terminal Ready) – сигнал готовности АПД к обмену данными; DCD (Data Clear Detected) – сигнал обнаружения несущей от АКД; RI (Ring Indicator) – индикатор вызова (звонка) от АКД

Рис. 3.15. Кабели подключения модемов

Если на каком-либо устройстве АПД установлена розетка – это почти стопроцентный признак того, что к другому устройству оно должно подключаться прямым кабелем, аналогичным кабелю подключения модема. Розетка устанавливается обычно на тех устройствах, у которых удалённое подключение через модем не предусмотрено.

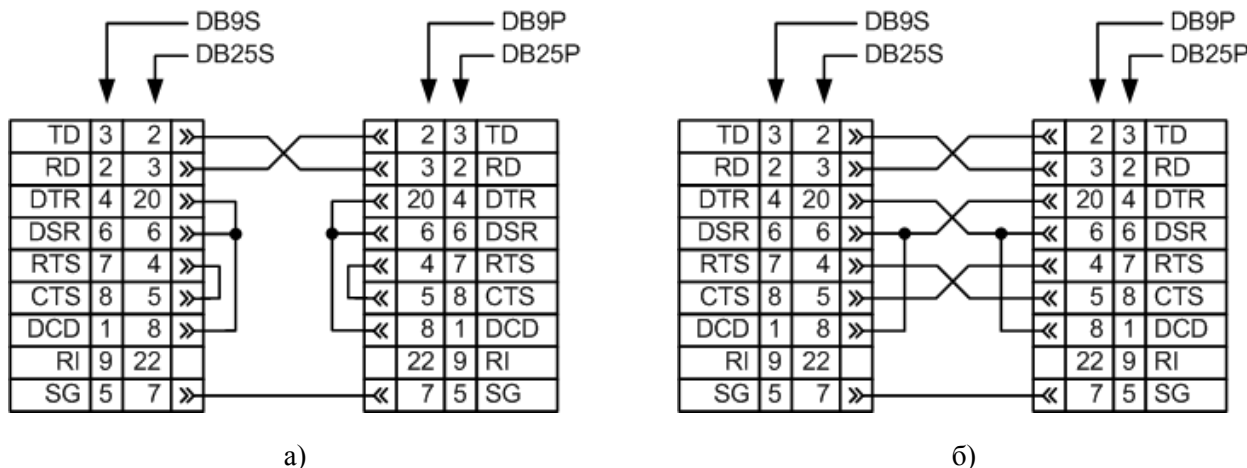


Рис. 3.16. Нуль-модемный кабель: а – минимальный, б – полный

Интерфейсы RS-422A и RS-485 (EIA/TIA-422A и EIA/TIA-485)

Стандарт RS-485 был совместно разработан двумя ассоциациями производителей: Ассоциацией электронной промышленности (Electronic Industries Association, EIA) и Ассоциацией промышленности средств связи (Telecommunications Industry Association, TIA). Многие инженеры продолжают использовать обозначение RS, однако ассоциация официально заменила «RS» на «EIA/TIA» с целью облегчить идентификацию происхождения своих стандартов.

Стандарты RS-485 и RS-422A имеют много общего, и поэтому их часто путают [35]. В табл. 3.2 приведены их основные характеристики. RS-485 определяет двунаправленную полудуплексную передачу данных и является единственным стандартом EIA/TIA, допускающим объединение нескольких приёмников и передатчиков в единую сеть с помощью шины. RS-422, с другой стороны, определяет единственный однонаправленный передатчик с несколькими приёмниками. Элементы RS-485 обратно совместимы и взаимозаменяемы со своими двойниками из RS-422A, однако передатчики RS-422A не должны использоваться в системах на основе RS-485, поскольку они не могут отказаться от управления шиной.

Таблица 3.2.

Основные характеристики интерфейсов RS-422A и RS-485

	RS-422A	RS-485
Макс. число передатчиков/приёмников	1/10	32/32
Мин. выходной диапазон передатчика	± 2 В	± 1.5 В
Макс. выходной диапазон передатчика	± 5 В	± 5 В
Макс. ток короткого замыкания передатчика	± 150 мА	± 250 мА
Макс. выходное сопротивление передатчика	100 Ом	54 Ом
Чувствительность по входу приёмника	± 200 мВ	± 200 мВ
Мин. входное сопротивление приёмника	4 кОм	12 кОм
Диапазон напряжений входного сигнала приёмника	± 7 В	от -7 В до $+12$ В
Уровень логической единицы приёмника	$> +200$ мВ	$> +200$ мВ
Уровень логического нуля приёмника	< -200 мВ	< -200 мВ

Разное количество приёмников, подключаемых по интерфейсам RS-422A и RS-485, объясняется более жёсткими требованиями к минимальному входному сопротивлению приёмника со стороны RS-485, и одновременно большей мощностью передатчика. Допустимая нагрузка передатчика RS-485/RS-422A количественно определяется в терминах единичной нагрузки, которая, в свою очередь, определяется как входной импеданс одного стандартного приёмника (для RS-485 – 12кОм) [35]. Таким образом, стандартный передатчик RS-485 может управлять 32

единичными нагрузками (32 параллельных 12-килоомных нагрузки). Однако для некоторых приёмников RS-485 входное сопротивление является более высоким – 48 кОм (1/4 единичной нагрузки) или даже 96 кОм (1/8 единичной нагрузки) – и, соответственно, к одной сигнальной линии могут быть подключены сразу 128 или 256 таких приёмников. В общем случае можно подключить любую комбинацию типов приёмников, если их параллельный импеданс не превышает 32 единичных нагрузки (т.е. общее сопротивление параллельно включенных приёмников не меньше 375 Ом).

Для передачи информации оба интерфейса используют принцип дифференциальной (балансной) передачи данных. Суть его заключается в передаче одного сигнала по двум проводам (рис. 3.17) [36]. Причём по одному проводу (условно А) идет оригинальный сигнал, а по другому (условно В) – его инверсная копия. Другими словами, если на одном проводе «1», то на другом «0» и наоборот. Таким образом, между двумя проводами витой пары всегда есть разность потенциалов: при «1» она положительна, при «0» – отрицательна.

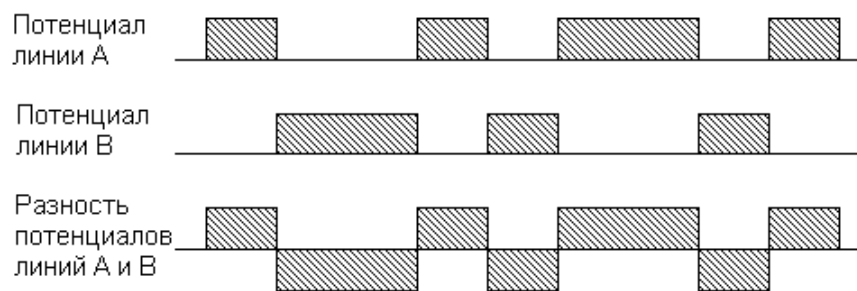


Рис. 3.17. Дифференциальный метод передачи данных

Именно этой разностью потенциалов и передаётся сигнал. Такой способ передачи обеспечивает высокую устойчивость к синфазной помехе. К примеру, электромагнитная волна, проходя через участок линии связи, наводит в обоих проводах потенциал. Если сигнал передаётся потенциалом в одном проводе относительно общего провода, как в RS-232, то наводка на этот провод может исказить сигнал относительно хорошо поглощающей наводки схемной земли. А при дифференциальной передаче искажения не происходит. В самом деле, если два провода пролегают близко друг к другу, да ещё перевиты, то наводка на оба провода одинакова. Потенциал в обоих одинаково нагруженных проводах изменяется одинаково, при этом информативная разность потенциалов остается без изменений.

Приёмник, получая на дифференциальных входах (АВ) разность потенциалов (U_{AB}) переводит их в цифровой сигнал на выходе. Когда $U_{AB} > +200$ мВ – приёмник определяет «1», когда $U_{AB} < -200$ мВ – приёмник определяет «0». Если раз-

ность потенциалов в линии настолько мала, что не выходит за пороговые значения – правильное распознавание сигнала не гарантируется. Кроме того, в линии могут быть и не синфазные помехи, которые исказят столь слабый сигнал.

Как уже говорилось, стандарт RS-422A разрешает подключение к одной паре проводников одного передатчика и до десяти приёмников (рис. 3.18) [35]. При этом реализуется симплексный режим передачи информации.

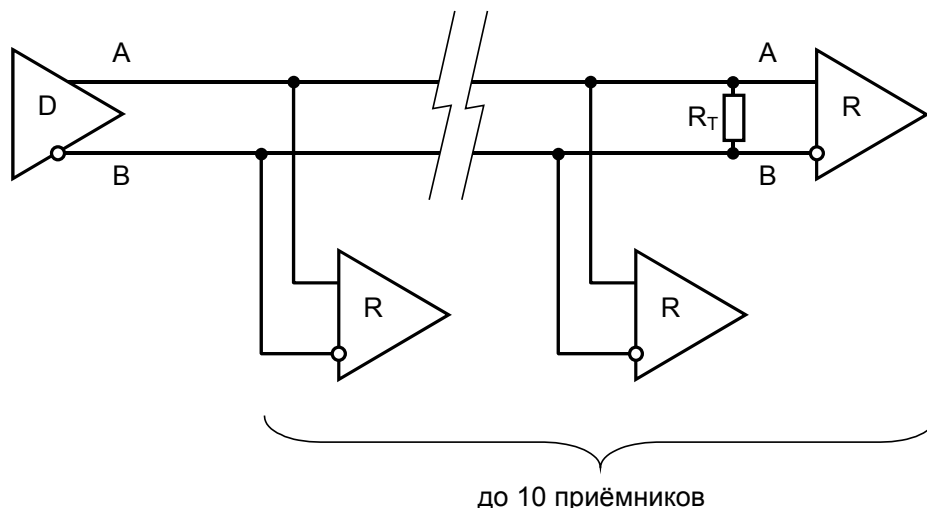


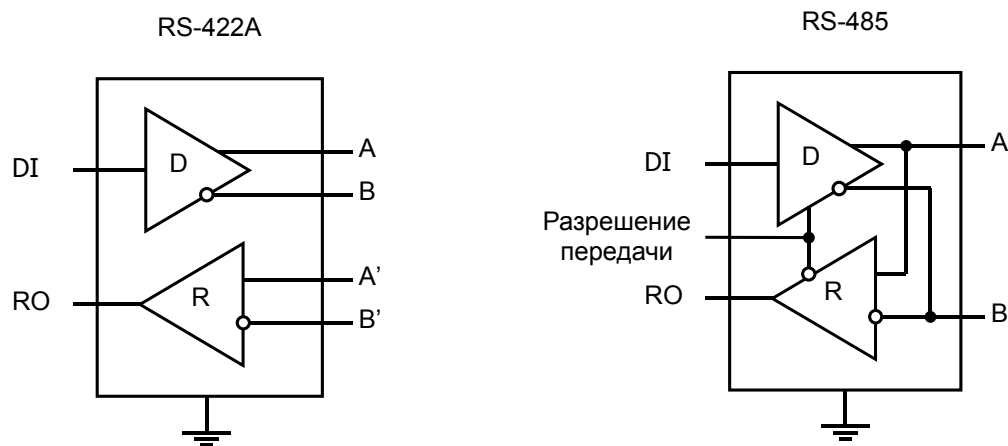
Рис. 3.18. Двухпроводное соединение устройств с помощью интерфейса RS-422A

Для реализации двунаправленного обмена (дуплексный режим) используются устройства, объединяющие в себе приёмник и передатчик (рис. 3.19), и четырёхпроводная схема подключения (рис. 3.20). Приём и передача идут по двум отдельным парам проводов. При подобном подключении с помощью RS-422A возможно реализовать соединение «точка-точка». Терминальные сопротивления $R_T=100$ Ом устанавливаются на конце сигнальной линии на стороне приёмника.

RS-485 – полудуплексный интерфейс [37, 38]. Приём и передача идут по одной паре проводов с разделением по времени. В сети может быть много передатчиков, так как они «умеют» отключаться в режиме приёма (рис. 3.21). Все устройства подключаются к одной витой паре одинаково: прямые выходы (A) к одному проводу, инверсные (B) – к другому. На обоих концах витой пары устанавливаются терминаторы – $R_T=100$ Ом.

Интерфейс RS-485 может использоваться в четырёхпроводном режиме для организации шины с централизованным арбитражем (рис. 3.22) [37]. В этом случае только один ведущий узел (Master) имеет право инициировать обмен данными, а все остальные ведомые узлы (Slave) только отвечают на его запросы. Приёмники всех ведомых узлов подключаются к передатчику ведущего узла, а передатчики всех ведомых – к приёмнику ведущего.

Специальный тип разъёмов для RS-422A/RS-485 не определён.



D (Driver) – передатчик; R (Receiver) – приёмник; DI (Driver Input) – цифровой вход передатчика; RO (Receiver Output) – цифровой выход приёмника; A, A' – прямой дифференциальный вход/выход; B, B' – инверсный дифференциальный вход/выход

Рис. 3.19. Сигналы приёмопередатчиков RS-422A и RS-485

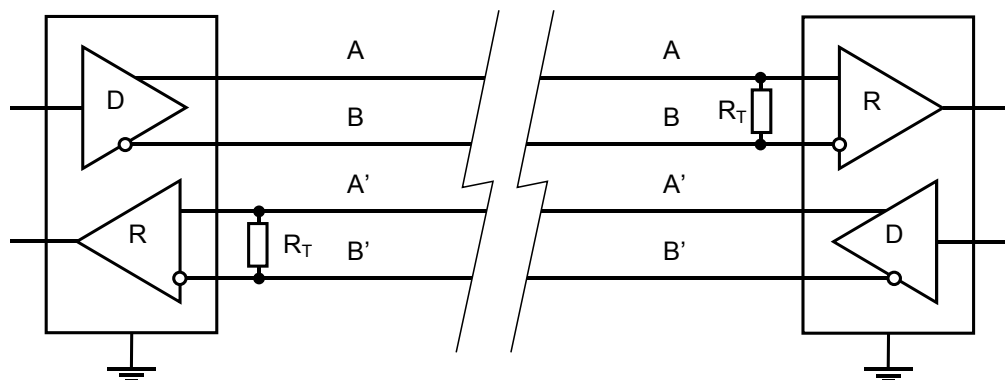


Рис. 3.20. Четырёхпроводное соединение устройств с помощью интерфейса RS-422A

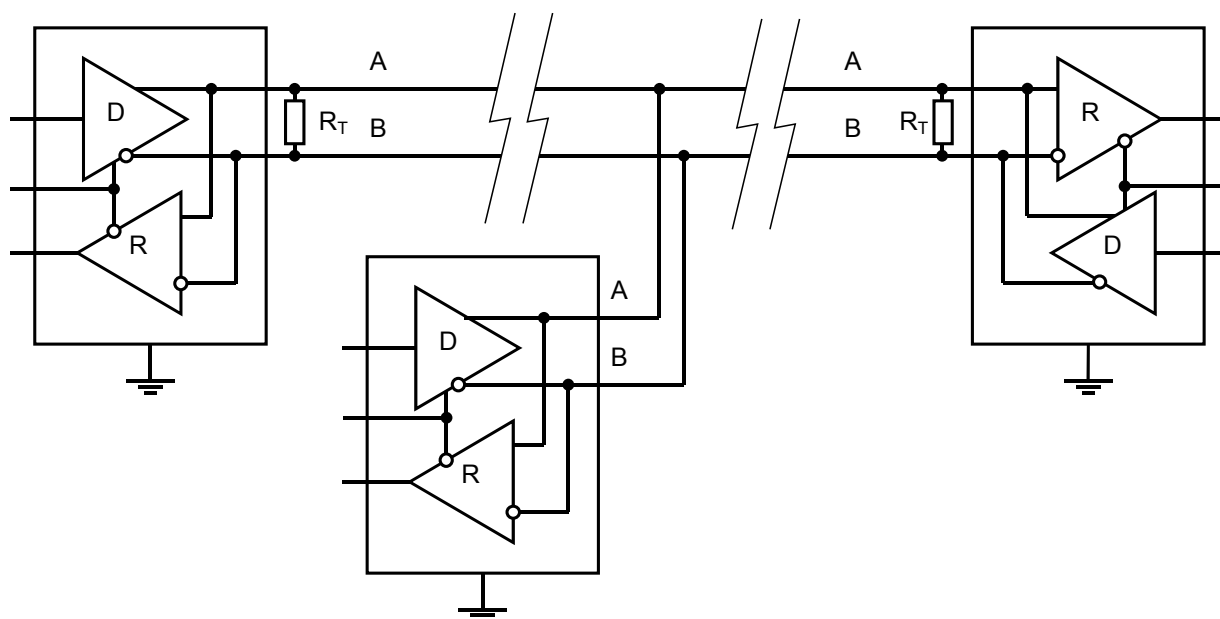


Рис. 3.21. Двухпроводное соединение нескольких устройств с помощью интерфейса RS-485

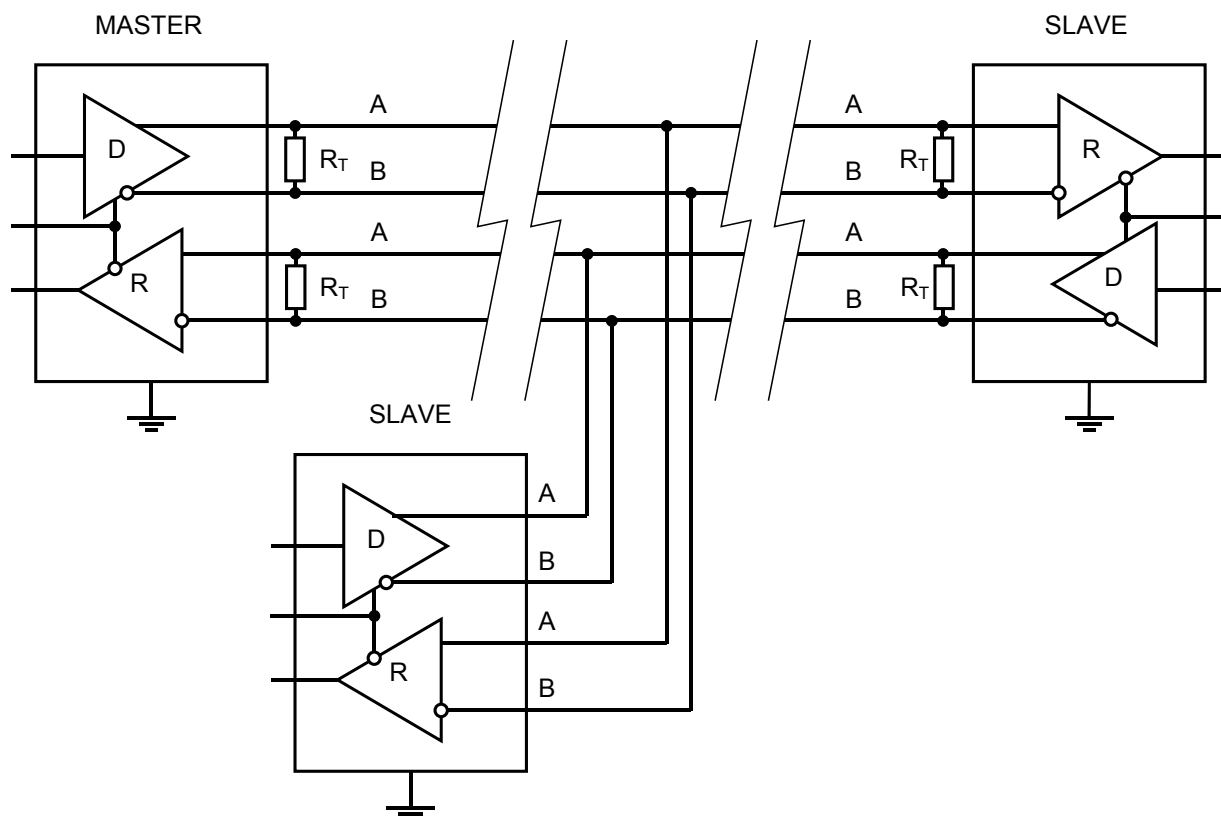


Рис. 3.22. Четырёхпроводное соединение нескольких устройств с помощью интерфейса RS-485

Интерфейс «токовая петля»

Распространённым вариантом последовательного интерфейса является токовая петля. В ней электрическим сигналом является не уровень напряжения относительно общего провода, а ток в двухпроводной линии, соединяющей приёмник и передатчик. Логической единице (состоянию «включено») соответствует протекание тока 20 мА, а логическому нулю отсутствие тока. Такое представление сигналов для описанного формата асинхронной посылки позволяет обнаружить обрыв линии – приёмник заметит отсутствие стоп-бита (обрыв линии действует как постоянный логический нуль). Скорость передачи – 19200 кбит/с.

Токовая петля обычно предполагает гальваническую развязку входных цепей приёмника от схемы устройства. При этом источником тока в петле является передатчик (этот вариант называют активным передатчиком). Возможно и питание от приёмника (активный приёмник), при этом выходной ключ передатчика может быть также гальванически развязан с остальной схемой передатчика. Существуют упрощенные варианты без гальванической развязки, но это уже вырожденный случай интерфейса.

Токовая петля с гальванической развязкой позволяет передавать сигналы на расстояния до нескольких километров. Расстояние определяется сопротивлением пары проводов и уровнем помех.

В табл. 3.3 приводятся сравнительные характеристики рассмотренных физических интерфейсов.

Таблица 3.3.

Сравнительные характеристики стандартных физических интерфейсов

Характеристика	RS-232C	RS-422	RS-485	Токовая петля
Макс. число передатчиков/приёмников на линии	1/1	1/10	32/32	1/1
Макс. длина линии	15 м	1220 м	1220 м	6400 м
Макс. скорость передачи	115200 кбит/с	12 м – 10 Мбит/с 1200 м – 100 кбит/с		19200 кбит/с

Вопросы для самоконтроля

1. К какому уровню типовой структуры АСУ ТП относятся датчики? исполнительные механизмы? ПЛК? SCADA-системы?
2. Перечислите и опишите основные сетевые топологии. Какие способы управления доступом к среде передачи используются в каждой из них?
3. Какие уровни взаимодействия описывает модель взаимодействия открытых систем? Какие уровни чаще всего реализуются в промышленных сетях?
4. Чем различаются симплексный, дуплексный и полудуплексный режимы передачи данных?
5. Какие способы непосредственного кодирования данных при передаче являются самосинхронизирующимися?
6. Можно ли по временной диаграмме изменения сигнала в линии передачи отличить манчестерское кодирование от дифференциального манчестерского кодирования?
7. Назовите обязательные поля кадра асинхронной передачи.
8. Можно ли по временной диаграмме изменения сигнала в линии передачи определить частоту асинхронной передачи?
9. Какие последовательные электрические интерфейсы обладают наибольшей помехоустойчивостью?
10. Какой интерфейс поддерживает самую большую длину линии связи?
11. Какие характеристики интерфейса зависят от длины линии связи?
12. Чем определяется максимальное количество приёмопередатчиков объединяемых последовательным интерфейсом?
13. Сколько приёмников можно подключить к передатчику RS-422?
14. Сколько приёмопередатчиков можно объединить с помощью интерфейса RS-232? RS-422? RS-485? токовая петля?

4. СОВРЕМЕННЫЕ СТАНДАРТЫ ПРОМЫШЛЕННЫХ СЕТЕЙ

Термин «промышленная сеть» (fieldbus, полевая шина) подчёркивает, что сеть применяется для решения задач передачи информации в области, связанной непосредственно с производственной зоной, где работают контроллеры, датчики и исполнительные устройства. Задача промышленной сети состоит в организации физической и логической связи датчиков и исполнительных устройств с системным интеллектом, роль которого выполняют ПЛК или промышленные компьютеры таким образом, чтобы информация с этого уровня была доступна общезаводской информационной системе.

Промышленная сеть должна отвечать множеству разнообразных, а зачастую противоречивых требований. От выбора сетевой архитектуры зависят не только затраты на создание системы, но и срок её жизни, её способность к развитию, то есть интегральная стоимость системы. Вот как могут быть сформулированы некоторые основные требования, которые предъявляются к «идеальной» промышленной сети [31].

- 1) производительность;
- 2) предсказуемость времени доставки информации;
- 3) помехоустойчивость;
- 4) доступность и простота организации физического канала передачи данных;
- 5) максимальный сервис для приложений верхнего уровня;
- 6) минимальная стоимость устройств аппаратной реализации, особенно на уровне контроллеров;
- 7) возможность получения «распределённого интеллекта», путём предоставления максимального доступа к каналу нескольким ведущим узлам;
- 8) управляемость и самовосстановление в случае возникновения нештатных ситуаций.

Как видно, в получившемся списке первое требование противоречит второму, третье — четвёртому и так далее. Более того, подобные противоречия приходится обходить постоянно и на всех уровнях проектирования, начиная с того, какой формат пакета передачи данных выбрать: тот, который позволит осуществлять расширенное управление сетью и удалённую загрузку, или тот, который обеспечит максимально быструю работу с большим числом дискретных сигналов, заканчивая решением философской проблемы, что лучше: применить не самое современное, но проверенное годами решение, или применить кажущееся блестящим и современным решение, которое почему-то оказывается дороже и до сих пор ещё не применяется на предприятии-конкуренте.

Таким образом, можно полагать, что промышленная сеть – суть один большой компромисс. И от того, как расставлены акценты в этом компромиссе, зависит успешность решения задач, стоящих перед сетевой архитектурой.

Промышленная сеть должна полностью удовлетворять запросам потребителя по модульности, надёжности, защите от внешних помех, простоте в построении, монтаже и программировании логики работы. Предпочтительность того или иного сетевого решения как средства транспортировки данных можно оценить по следующей группе критериев [33]:

- объём передаваемых полезных данных;
- время передачи фиксированного объёма данных;
- удовлетворение требованиям задач реального времени;
- максимальная длина шины;
- допустимое число узлов на шине;
- помехозащищённость;
- денежные затраты в расчёте на узел.

Поскольку часто улучшение по одному параметру может привести к снижению качества по другому, то при выборе того или иного протокольного решения необходимо следовать принципу разумной достаточности.

Области применения промышленных сетей

В зависимости от области применения весь спектр промышленных сетей можно разделить на два уровня:

- системный уровень (Field level, полевой уровень). Промышленные сети этого уровня решают задачи по управлению процессом производства, сбором и обработкой данных на уровне промышленных контроллеров;
- уровень датчиков и исполнительных механизмов (Sensor/actuator level). Задачи сетей этого уровня сводятся к опросу датчиков и управлению работой разнообразных исполнительных механизмов.

Другими словами, необходимо различать промышленные сети для системного уровня (field busses) и уровня датчиков (sensor/actuator busses). Сравнение этих двух классов в самом общем виде можно получить по критериям из таблицы 4.1.

На сегодняшний день спектр протоколов для каждого класса промышленных сетей довольно широк. К тому же можно выделить целый ряд протоколов, способных успешно работать на обоих уровнях.

- Типичные представители промышленных сетей системного уровня:
 - ✓ Fieldbus Foundation HSE (High Speed Ethernet);
 - ✓ Modbus/TCP;
 - ✓ Modbus Plus;

- ✓ PROFIBUS FMS (Process Field Bus for Fieldbus Message Specification);
- ✓ Bitbus.
- Типичные сенсорные (датчиковые) сети:
 - ✓ Fieldbus Foundation H1;
 - ✓ PROFIBUS PA (PROFIBUS Process Automation);
 - ✓ HART;
 - ✓ AS-interface (Actuator/Sensor interface);
 - ✓ Interbus-S;
 - ✓ SERCOS interface.
- Типичные сети для обоих уровней применения:
 - ✓ Modbus;
 - ✓ CAN (Controller Area Network);
 - ✓ PROFIBUS DP (PROFIBUS for Distributed Peripheral);
 - ✓ FIP (Factory Instrumentation Protocol);
 - ✓ LON (Local Operating Network).

Необходимо также отметить появление комплексных сетевых решений, представленных на каждом из уровней и позволяющих организовать единое информационное пространство с помощью протоколов изначально ориентированных на совместное использование. Примерами таких сетей являются семейство протоколов PROFIBUS (FMS, DP, PA) и протокол Fieldbus Foundation (HSE, H1).

Таблица 4.1.

Сравнительные критерии промышленных сетей

Основные критерии	Fieldbus	Sensorbus
1. Предельное расстояние	от 100 м до 1 км	до 100 м
2. Время цикла сети	от 10 мс до 10 с	от 1 мс до 1 с
3. Объём передаваемых данных в одном пакете	от 8 байт до нескольких сотен байт	от 1 до 8 байт
4. Управление доступом к шине	централизованное/ децентрализованное	централизованное

4.1. ПРОТОКОЛЫ УРОВНЯ ДАТЧИКОВ

HART-протокол

Стандарт для передачи аналоговых сигналов значениями тока в диапазоне 4–20 мА известен уже несколько десятков лет и широко используется при создании систем АСУ ТП, в химической промышленности, теплоэнергетике, в пищевой и многих других отраслях промышленности. Традиционно для измерения различных физи-

ческих величин (давления, объёма, температуры и т.д.) предлагается множество приборов с токовым выходом 4–20 мА. Достоинством данного стандарта является простота его реализации, массовое использование в приборах и возможность помехоустойчивой передачи аналогового сигнала на относительно большие расстояния. Однако при создании нового поколения интеллектуальных приборов и датчиков потребовалось наряду с передачей аналоговой информации передавать и цифровые данные, соответствующие их новым расширенным функциональным возможностям.

Перевод существующих АСУ ТП на новые датчики, использующие цифровой обмен данными связан с закупками дорогостоящего оборудования и программного обеспечения. Кроме того, использование даже одного цифрового датчика в системе, потребует отдельного цифрового контроллера и изменения всей архитектуры АСУ ТП. Преимущество HART-протокола в том, что HART-устройство можно ставить на место аналогового датчика и, не меняя всей системы, использовать цифровой обмен.

HART-протокол (Highway Addressable Remote Transducer – удалённый адресуемый приёмник) был разработан в середине 80-х годов американской компанией Rosemount. В начале 90-х годов протокол был дополнен и стал открытым коммуникационным стандартом. Вначале он был нормирован только для применения в режиме соединения «точка-точка», затем появилась возможность применять протокол в режиме многоточечного соединения («multidrop»). Основные технические параметры, определяемые стандартом на HART-протокол, представлены в табл. 4.2. [39, 33].

Метод передачи данных

HART-протокол основан на методе передачи данных с помощью частотной модуляции (Frequency Shift Keying, FSK – частотная манипуляция), в соответствии с широко распространенным коммуникационным стандартом Bell 202. Цифровая информация передаётся частотами 1200 Гц – логическая единица и 2200 Гц – логический ноль, которые накладываются на аналоговый токовый сигнал (рис. 4.1). Частотно-модулированный сигнал является двухполярным ($\pm 0,5$ мА) и при применении соответствующей фильтрации не влияет на основной аналоговый сигнал 4–20 мА. Скорость передачи данных для HART составляет 1,2 кбит/с. Каждое HART-устройство требует для цифровой передачи соответствующего модема.

Благодаря наличию двух ведущих устройств каждое из них может быть готово к передаче через 270 мс (время ожидания). Цикл обновления данных повторяется 2-3 раза в секунду в режиме запрос/ответ и 3-4 раза в секунду в пакетном режиме. Несмотря на относительно большую длительность цикла, в большинстве

случаев он является достаточным для управления непрерывными технологическими процессами.

Таблица 4.2.

Технические параметры, определяемые стандартом HART-протокола

Топология	«Точка-точка» (стандартная) или шина
Максимальное количество устройств	Одно подчинённое устройство и два ведущих устройства (стандартный режим); 15 подчинённых устройств, 2 ведущих устройства (многоточечный режим с удалённым питанием)
Максимальная протяжённость линии связи	3 км (стандарт) 100м (многоточечный режим)
Тип линии	Экранированная витая пара
Интерфейс	4–20 мА, токовая петля (аналоговый)
Скорость передачи	1200 бит/с
Метод обращения	Polling (механизм опроса с уникальной адресацией каждого устройства)
Максимальная длина пакета данных	25 байт
Время цикла обновления данных	Около 500 мс (в пакетном режиме – 330 мс)
Надёжность передачи данных	1 ошибка на 10^5 бит, контроль по четности каждого байта, байт контрольной суммы для каждого пакета
Возможность использования во взрывоопасной зоне	Да

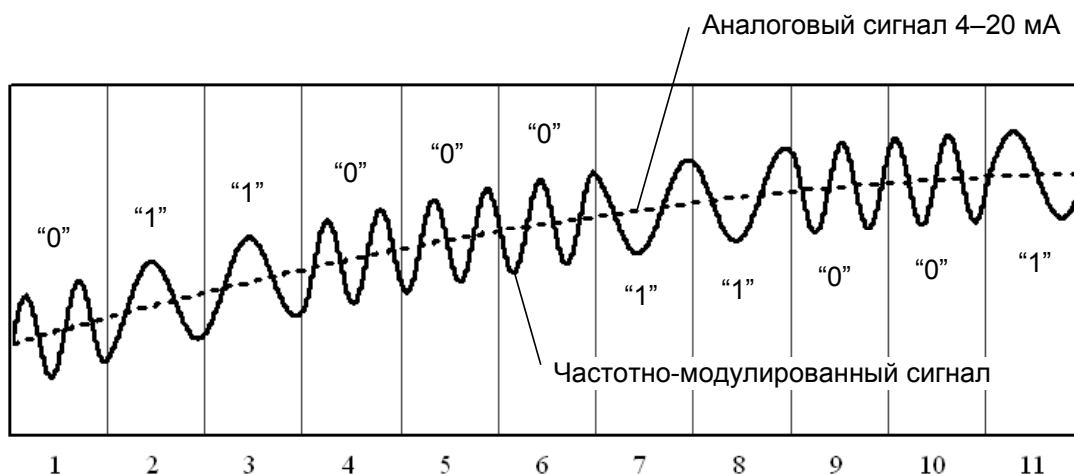


Рис. 4.1. Кодирование HART-сигнала

Подключение устройств и топология

HART-протокол используется в двух режимах подключения. В большинстве случаев применяется соединение «точка-точка» (рис. 4.2, а), то есть непосред-

венное соединение прибора низовой автоматики (преобразователя информации, датчика, исполнительного устройства и т.п.) и не более чем двух ведущих устройств. В качестве первичного ведущего устройства, как правило, используется ПЛК, а в качестве вторичного — портативный HART-терминал или отладочный ПК с HART-модемом. При этом аналоговый токовый сигнал передаётся от ведомого устройства к соответствующему ведущему устройству. Цифровые сигналы могут приниматься или передаваться как от ведущего, так и от ведомого устройства. Так как цифровой сигнал наложен на аналоговый, процесс передачи аналогового сигнала происходит без прерывания.

В многоточечном режиме (рис. 4.2, б) до 15 ведомых устройств могут соединяться параллельно двухпроводной линией с теми же двумя ведущими устройствами. При этом по линии осуществляется только цифровая связь. Сигнал постоянного тока 4 мА обеспечивает вспомогательное питание ведомых приборов по сигнальным линиям.

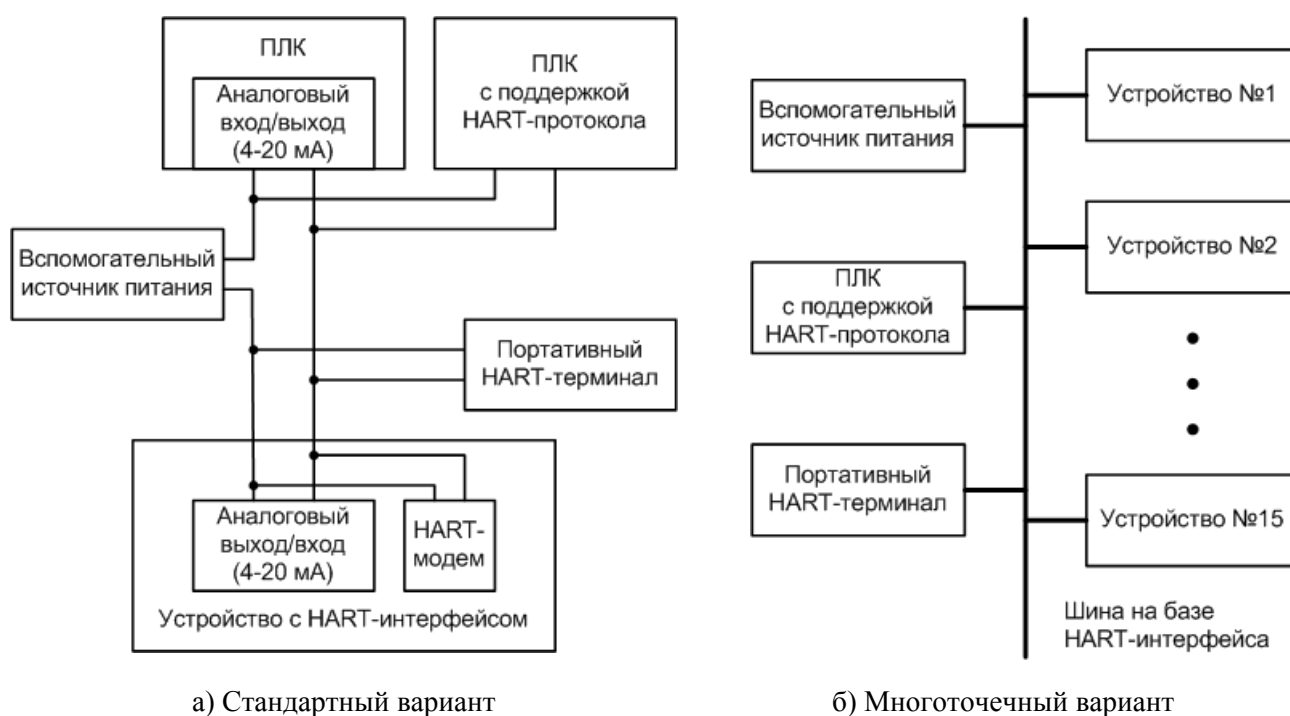


Рис. 4.2. Структурная схема подключения HART-устройств:

а — цифровой канал «точка-точка» с аналоговым сигналом; б — цифровой канал (топология — шина) без передачи аналогового сигнала, но с удалённым питанием по цепям связи

Реализация HART-протокола

HART-протокол реализует уровни 1, 2 и 7 модели взаимодействия открытых систем. Дополнительно протокол предусматривает надстройку к уровню 7 в форме HART Device Description Language.

При реализации физического уровня HART-протокол опирается на стандарт Bell 202. Таким образом, аппаратно он ориентирован на так называемые Bell- или HART-модемы. На уровне канала данных реализуется протокол передачи данных, который использует принцип «ведущий-ведомый» (master-slave). Активное ведущее устройство передаёт соответствующую HART-команду на ведомое HART-устройство (как правило, приборы низовой автоматики). Запрашиваемое HART-устройство интерпретирует соответствующую команду и отвечает. Оба ведущих имеют различные адреса, что и гарантирует однозначность при обмене командами и ответами. Передача данных происходит асинхронно в полудуплексном режиме. Структура пакетов во всех режимах работы одинаковая, что создаёт однозначное соответствие между HART-командами и ответами устройств в многоточечном режиме. Различия существуют только между структурой запроса ведущего (HART-терминала или ПЛК) и структурой ответа ведомых устройств (рис. 4.3).



Рис. 4.3. Структура HART-сообщения

Все HART-сообщения передаются побайтно в асинхронном режиме. Передача каждого байта начинается со стартового бита, затем идут биты данных (начиная со старшего), бит чётности и стоповый бит.

Сообщение состоит из следующих элементов (табл. 4.3).

Длина преамбулы зависит от требований ведомых устройств. Ведущее устройство использует самую длинную преамбулу только при первом обращении к ведомому устройству. Затем, запросив требуемую ведомым устройством длину преамбулы, ведущее устройство использует преамбулу требуемой длины.

Длинная преамбула приводит к снижению скорости передачи полезных данных, поэтому практически все современные ведомые устройства используют пятибайтовую преамбулу.

Поле статуса (ST) появляется только в ответе ведомого устройства. Если ведомое устройство не выполнило команду, в поле статуса отображается причина,

например, сообщение получено устройством с ошибкой или устройство не может выполнить полученную команду.

Таблица 4.3.

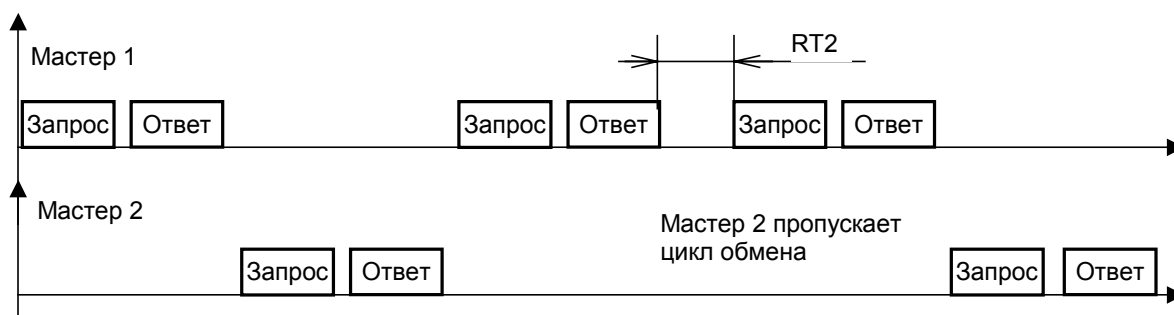
Поля HART-сообщения

Имя поля	Длина, байт	Назначение
PA (Preamble) Преамбула	от 5 до 20	Синхронизация HART-модемов и обнаружение несущей
SD (Start Delimiter) Стартовый байт	1	Признак начала сообщения. Синхронизация приёмника и передатчика
AD (Address) Адрес	1 или 5	Содержит адрес приёмника или признак пакетного режима.
CD (Command) Команда	1	Содержит код HART-команды, которую должно выполнить Slave-устройство
BC (Byte Count) Количество байтов	1	Содержит суммарную длину полей ST и DT в байтах
ST (Status) Статус ведомого устройства	0 – для ведущего устройства, 2 – для ведомого устройства	Ведомое устройство сообщает ведущему устройству о выполнении или не выполнении команды и причину невыполнения
DT (Data) Данные	от 0 до 25	Поле содержит параметры или результат выполнения команды (например, результат измерения)
CHK (Checksum) Контрольная сумма	1	Поле используется для обнаружения ошибок при передаче

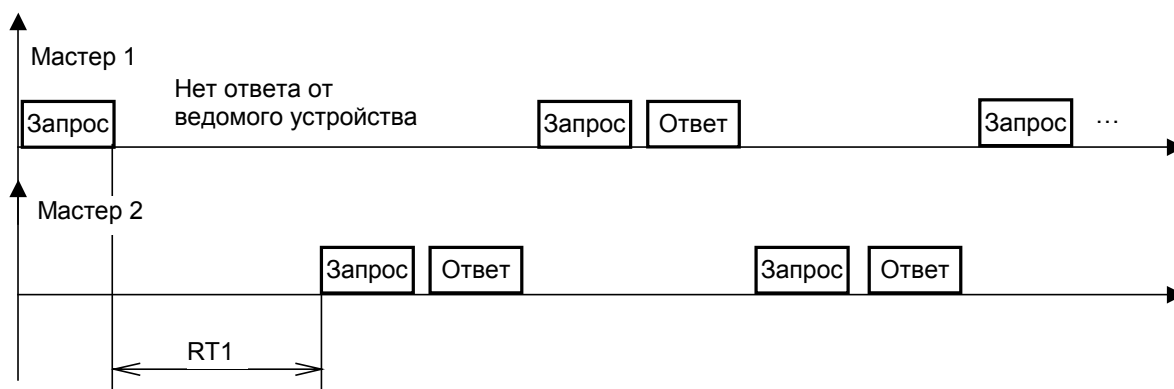
Надёжность передачи данных по HART-протоколу обеспечивается различными мерами контроля, как на уровне байта, так и на уровне пакета. Частота возникновения ошибки на уровне передачи битов составляет одну ошибку на 10^5 бит. Каждый передаваемый байт внутри HART-пакета имеет бит паритета; каждый HART-пакет имеет контрольную сумму, с помощью которой можно распознавать до 3 ошибочных битов.

Управление доступом к среде передачи

Поскольку в сети может быть два ведущих устройства, то должен быть определён протокол доступа этих устройств к среде передачи. Оба ведущих устройства постоянно контролируют сетевой трафик, поочередно посылая запросы и получая ответы от ведомых устройств (рис. 4.4). Если ведущему устройству не нужно ничего передавать, то он не занимает шину. В этом случае ведущее устройство, передававшее последний раз выжидает в течение времени RT₂, после чего начинает новый цикл обмена (рис. 4.4, а) [40].



а) Пропуск цикла одним из ведущих устройств



б) Нет ответа от ведомого устройства

Рис. 4.4. Циклы обмена по HART-протоколу

Если ведущее устройство не получает ответа в течение времени $RT1$ с момента окончания передачи кадра-запроса, то право на запрос переходит к другому ведущему устройству (рис. 4.4, б). Длительность временного интервала $RT1$ определена так, чтобы она была немного больше предельного времени отклика ведомого устройства $TT0$. Если ведомое устройство не успевает ответить в течение интервала $TT0$ с момента окончания передачи кадра-запроса, то оно не отвечает вообще. Благодаря этому не возникает конфликта между ведомым устройством и ведущим устройством, начавшим передачу по истечении времени $RT1$.

Значения временных интервалов, использующиеся в HART-протоколе для синхронизации устройств приведены в табл. 4.4.

Длительность в табл. 4.4 измеряется временным интервалом, который нужен для передачи одного байта (символа). Зная, что каждый байт данных передаётся 11 битами, а скорость передачи 1200 бит/с, можно рассчитать длительность одного символа – 9,167 мс.

Различные значения интервалов $RT1$ для главного и второстепенного ведущих устройств введены для того, чтобы после инициализации сети первым начало передачу главное ведущее устройство.

Таблица 4.4.

Временные интервалы, определённые в HART-протоколе

Обозначение	Описание	Длительность в символах
RT2	Ожидание ведущего устройства перед повторным использованием сети	8
RT1(0)	Ожидание ответа от ведомого устройства главным ведущим устройством	33
RT1(1)	Ожидание ответа от ведомого устройства второстепенным ведущим устройством	41
TT0	Предельное время начала ответа ведомого устройства	28

Команды HART-протокола

Внутри уровня 7 протокол HART использует команды, которые подразделяются на три основных класса.

Универсальные команды. Эти команды используются и поддерживаются всеми ведомыми устройствами. Они служат решению таких общих задач, как, например, считывание первичных значений измерений, диапазона измерений, граничных величин или констант. Имеется 10 таких команд:

- прочитать производителя и тип устройства;
- прочитать главную переменную (ГП) и единицы измерения;
- прочитать силу тока на аналоговом выходе и процент от диапазона;
- прочитать силу тока на аналоговом выходе и до четырёх предопределённых динамических переменных;
- прочитать/записать 8-символьный идентификатор и 16-символьное описание;
- прочитать/записать 32-символьное сообщение;
- прочитать диапазон значений устройства, единицы измерения и время выборки;
- прочитать серийный номер устройства и ограничения;
- прочитать/записать последний шифр комплекта устройств;
- записать адрес запроса.

Стандартные команды. Они используются в большинстве HART-приборов, но не во всех. К этой группе принадлежат прежде всего такие команды, как считывание и запись стандартных и приборных параметров:

- прочитать выборку значений динамических переменных (до четырёх переменных);
- записать константу времени выборки;

- записать диапазон значений устройства;
- калибровать (установка нуля, диапазона);
- установить постоянное значение выходного тока;
- выполнить самотестирование;
- выполнить перезапуск устройства;
- установить ГП в нуль;
- записать единицы измерения ГП;
- установить нулевое значение ЦАП и коэффициент усиления;
- записать функцию преобразования (квадратный корень или линейное);
- записать серийный номер устройства;
- прочитать/записать привязки динамических переменных.

Специфические команды устройств. Эти команды содержат функции, которые ограничиваются данной моделью или типом прибора. К ним относятся команды, связанные с настройкой, вводом в эксплуатацию или работой специфических приборов:

- прочитать/записать уровень обрезки малых значений;
- пуск, останов или общий сброс;
- прочитать/записать коэффициент поправки на плотность;
- выбрать ГП (масса, расход или плотность);
- прочитать/записать информацию о материале и конструкции;
- калибровать датчик;
- включить ПИД-регулятор;
- записать уставку ПИД-регулятора;
- характеристика регулирующего клапана;
- уставка положения регулирующего клапана;
- границы перемещения регулирующего клапана;
- единицы измерения пользователя;
- информация локального дисплея.

Для каждой команды определён код, а также значения полей данных в запросе и ответе. Например, для универсальной команды «Прочитать силу тока на аналоговом выходе и процент от диапазона» код команды – 2, поле данных в запросе будет пустым (длина – нуль байтов). Ответ на эту команду будет содержать поле данных из 8 байтов: байты 0–3 содержат значение тока в миллиамперах в формате с плавающей запятой (4 байта); байты 4–7 – процент от диапазона в формате с плавающей запятой. Подробнее с форматами команд и передаваемыми данными можно познакомиться в [41].

Средства параметрирования и подключения HART-устройств

Для обеспечения нормальной работы устройств различных производителей в системах с HART-протоколом был введён Device Description Language (DDL) – язык описания устройств. Этот язык позволяет производителям HART-устройств однозначно описывать свойства и возможности устройств. Описание на языке DDL осуществляется в текстовом формате, а затем переводится компилятором в двоичную форму (DD). Полученный двоичный образ HART-устройства может быть загружен, например, в портативный HART-терминал или в ПК, где соответствующее прикладное программное обеспечение, предназначенное для конфигурирования и наладки HART-устройств, прочтя этот образ, сможет настроиться на работу с соответствующим устройством. Можно сказать, что описание на языке DDL является своего рода «драйвером» HART-устройства, который ко всему прочему программно совместим с любым прикладным программным обеспечением, предназначенным для работы с HART-протоколом.

Многие фирмы-производители HART-устройств предлагают специальное программное обеспечение как для их конфигурирования, так и для их встраивания в систему управления. Оно позволяет использовать в полной мере возможности современных интеллектуальных датчиков и исполнительных механизмов, а также обеспечивает их удобное конфигурирование и диагностику.

AS-interface

Основная задача сети AS-interface – связать в единую информационную структуру устройства самого нижнего уровня распределённой системы автоматизации, а именно: датчики и разнообразные исполнительные механизмы, имеющие соответствующий сетевой интерфейс. Название сети раскрывает её предназначение: Actuator/Sensor Interface (ASI, AS-i) – интерфейс с исполнительными механизмами и датчиками. Протокол ориентируется в первую очередь на подключение **дискретных** датчиков и исполнительных механизмов.

Впервые ASI-протокол вышел на рынок в конце 1989 года и уже сегодня поддержан рядом известных фирм: IFM, Limberg, Siemens, Pepperl+Fuchs, Allen-Bradley. Существует и одноименная ассоциация по поддержке этой сети – ASI [42].

AS-интерфейс является открытой промышленной сетью нижнего уровня систем автоматизации, которая предназначена для организации связи с датчиками и исполнительными устройствами в соответствии с требованиями европейских нормативов EN 50295 и международного стандарта МЭК 62026-2 [43].

В существовавших ранее решениях для подключения дискретных датчиков и исполнительных механизмов к управляющему устройству использовалось множе-

ство кабельных соединений (жгут), так как каждое такое устройство подключалось к модулю ввода-вывода управляющего устройства отдельной парой проводников. При этом затраты на приобретение кабельной продукции, на её монтаж и эксплуатацию были очень велики. AS-интерфейс позволяет решить задачу подключения датчиков и исполнительных механизмов к системе управления на основе построения сети с использованием одного двухжильного кабеля, с помощью которого обеспечивается как питание всех сетевых устройств, так и опрос датчиков и выдача команд на исполнительные механизмы.

При наличии в системе специальных модулей AS-интерфейс позволяет подключать к системе управления обычные широко распространённые на рынке датчики и исполнительные механизмы. Следующим шагом в развитии сетевых технологий на базе AS-интерфейса стало включение интегральных микросхем ведомого устройства непосредственно в электронную часть датчиков и исполнительных механизмов. Для таких интеллектуальных устройств становится возможной реализация удалённого параметрирования и диагностики без использования дополнительных связей.

Гибкость управления системой достигается за счёт применения различных ведущих устройств. Функции ведущих устройств могут выполнять программируемые логические контроллеры, промышленные компьютеры или модули связи с сетями более высокого уровня — MODBUS, Interbus, CAN, PROFIBUS, DeviceNet.

Передача данных

В качестве среды передачи используется пара обычных проводников. Площадь поперечного сечения каждого проводника, равная $1,5 \text{ мм}^2$, установлена стандартом. При таком сечении гарантируется питающий ток 2 А, таким образом, ведомые устройства могут питаться непосредственно от шины AS-интерфейса.

Оболочка кабеля AS-интерфейса имеет специальный профиль в виде трапеции с выступом (рис. 4.5), исключающий возможность неправильного подключения сетевых компонентов. Желтый плоский кабель стал своего рода рыночным знаком AS-интерфейса. Большая часть сетевых компонентов подключается к кабелю методом прокалывания изоляции, и сделать это можно практически в любой точке соединительного кабеля, что обеспечивает гибкость сетевой архитектуры и сокращает сроки монтажа. Наряду с профилированным кабелем используется и круглый кабель, ориентированный на специальные модули.

Если для исполнительных механизмов требуется дополнительное питание, например постоянное напряжение 24 В, то можно применить аналогичный профилированный кабель черного цвета, который также использует технологию про-

калывания. Для напряжений более 30 В, в частности для 220 В переменного тока, используется кабель с оболочкой красного цвета.

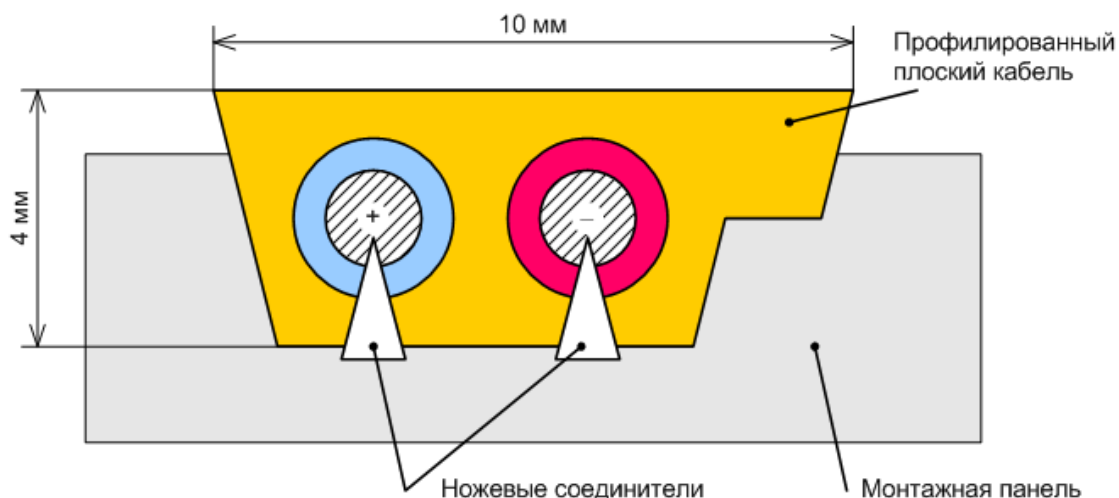


Рис. 4.5. Поперечное сечение кабеля AS-интерфейса и способ подключения устройств

Скорость передачи данных ограничена до 167 кбит/с. Для кодирования данных используется манчестерский код, в котором логические единица и нуль кодируются восходящим и нисходящим фронтом сигнала в середине битового интервала. Такой тип кодирования снижает влияние на ASI-кабель внешних возмущений.

Подключение устройств и топология

Локальная вычислительная система низкого уровня на базе AS-интерфейса может иметь только одно ведущее устройство. По спецификации AS-интерфейса версии 2.0, к ведущему устройству можно подключить 31 ведомое устройство. Каждое ведомое устройство может иметь до 4 входов/выходов.

AS-интерфейс использует метод доступа к ведомым устройствам, основанный на их циклическом опросе. При опросе системы, состоящей из 31 ведомого устройства, время цикла составляет 5 мс. Таким образом, не позднее чем через каждые 5 мс каждый датчик или исполнительный механизм системы будет опрошен ведущим устройством.

Каждый узел ASI-сети должен иметь специальную интерфейсную микросхему с поддержкой ASI-протокола. AS-интерфейс позволяет передавать как данные, так питающую нагрузку к узлам сети, поскольку существует большое число фотоэлектрических и индуктивных датчиков, которые потребляют небольшую мощность и могут питаться непосредственно от сети. Упрощённо ASI-сеть может выглядеть, как на рис. 4.6.

Топология сети AS-интерфейса может быть произвольной (рис. 4.7). Отсутствие терминаторов шины позволяет даже «закольцевать» среду передачи. Един-

ственное, что необходимо учитывать – это ограничение общей длины кабеля – 100 м. Под общей длиной понимается сумма длин всех ветвей сегмента сети, обслуживаемого одним ведущим устройством. Специальный расширитель позволяет удлинить кабель или разделить ветвь на группы. Если требуется большая длина кабеля, то можно использовать до двух повторителей, что обеспечит надёжное соединение при суммарной протяжённости линий связи до 300 м. При этом необходимо учитывать, что каждый сегмент требует отдельного источника электропитания.

Для сетевых устройств должны использоваться только специальные источники, предназначенные для работы с AS-интерфейсом.

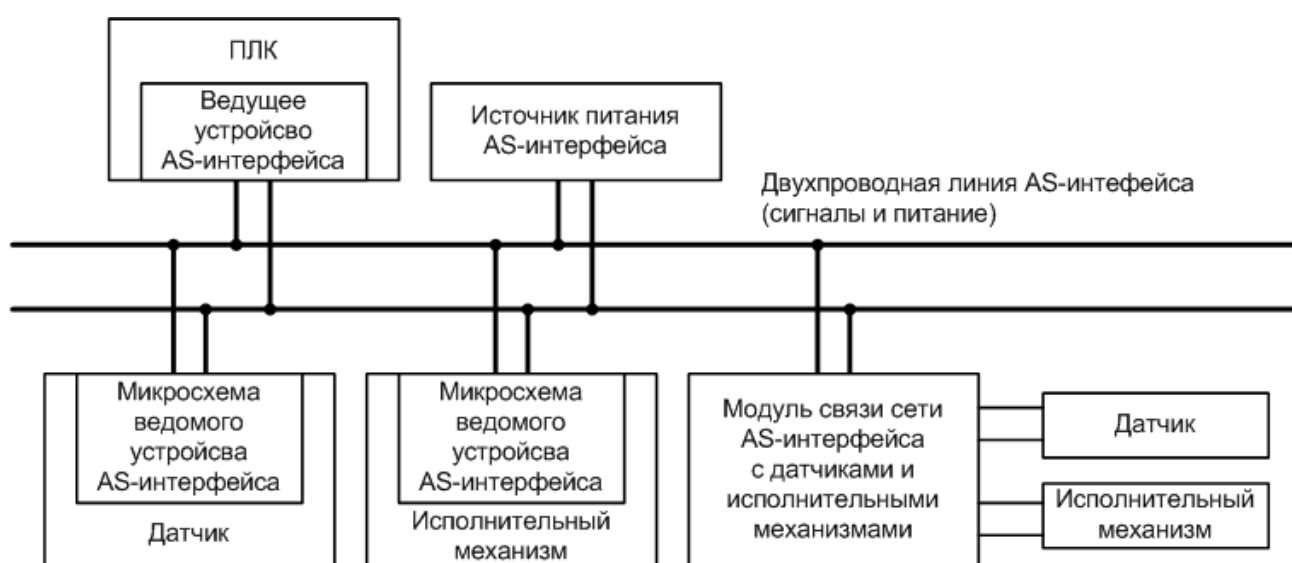


Рис. 4.6. Пример ASI-структуры

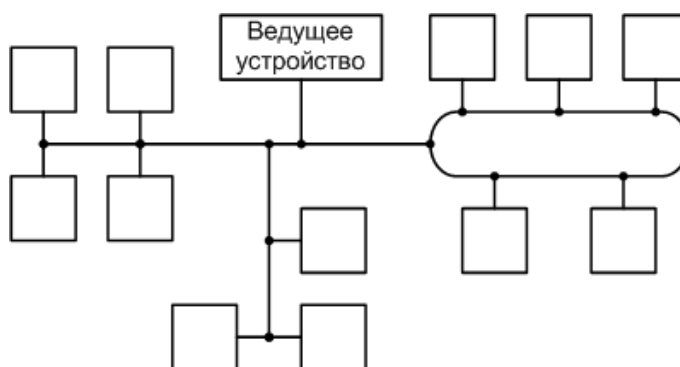


Рис. 4.7. Топология сети AS-интерфейса

Форматы пакетов данных

Протокол AS-интерфейса (рис. 4.8) состоит из запроса ведущего устройства, паузы ведущего устройства, ответа ведомого устройства и, соответственно, паузы ведомого устройства.

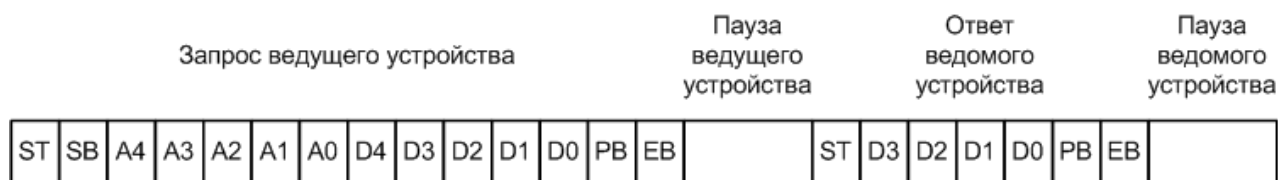


Рис. 4.8. Структура протокола AS-интерфейса

Все запросы ведущего устройства имеют длину 14 бит, все ответы ведомого устройства занимают 7 бит. При этом период времени передачи одного бита составляет 6 мкс. Пауза ведущего устройства может занимать по времени от 3 до 10 тактов передачи бита. Если ведомое устройство было синхронизировано, то есть приняло сообщение ведущего устройства и ответило, то это позволяет начать передачу ответа ведомого устройства через 3 такта. Если ведомое устройство не было синхронизировано, например, это первый запрос в адрес данного ведомого устройства или запрос после воздействия помехи, то требуется на два такта больше, чем это было необходимо в первом случае. Если ведущее устройство после 10 тактов не приняло стартовый бит ответа ведомого устройства, можно сделать заключение, что ответ не проходит, и ведущее устройство может послать следующий запрос, например, ведомому устройству с более высоким адресом.

Дополнительную информацию о назначении и состоянии битов запроса ведущего устройства и ответа ведомого устройства содержат табл. 4.5 и 4.0.

Таблица 4.5.

Биты запроса ведущего устройства

Обозначение	Название	Пояснения
ST	Стартовый бит	Указывает начало запроса ведущего устройства: 0 – действительный стартовый бит, 1 – не допускается
SB	Управляющий бит	Обозначает тип запроса: 0 – запрос данных, параметра, адреса, 1 – команда
A0...A4	Адрес	Адрес вызываемого ведомого устройства (5 разрядов)
D0...D4	Данные	5 информационных разрядов, соответствующих типу запроса (параметр, команда)
PB	Бит чётности	Сумма всех «1» в запросе должна быть чётной
EB	Конечный бит	Указывает конец запроса ведущего устройства: 0 – не допускается, 1 – признак конечного бита

В табл. 4.7 представлены некоторые запросы и команды ведущего устройства спецификации 2.0. Полный список команд и запросов AS-интерфейса можно найти в [44].

Бит D4 (см. табл. 4.7) запроса ведущего устройства определяет, какую именно информацию содержит запрос ведущего устройства (SB=0): если D4 равен ну-

лю, то информационные разряды запроса (D3–D0) содержат значения, которые нужно установить на дискретных выходах ведомого устройства, а если D4 равен единице, то информационные разряды запроса (D3–D0) содержат значения конфигурационных параметров устройства.

Таблица 4.0.

Биты ответа ведомого устройства

Обозначение	Название	Пояснения
ST	Стартовый бит	Указывает начало запроса ведущего устройства: 0 – действительный стартовый бит, 1 – не допускается
D0...D3	Данные	4 информационных разряда, указывают, например, параметр, установленный порт ввода-вывода, статус ведомого устройства
PB	Бит чётности	Сумма всех «1» в запросе должна быть чётной
EB	Конечный бит	Указывает конец запроса ведущего устройства: 0 – не допускается, 1 – признак конечного бита

Таблица 4.7.

Допустимые запросы ведущего устройства

	ST	SB	5 адресных разрядов	5 разрядов информации	PB	EB
Запрос данных	0	0	A4 A3 A2 A1 A0	0 D3 D2 D1 D0	PB	1
Записать параметр	0	0	A4 A3 A2 A1 A0	1 D3 D2 D1 D0	PB	1
Присвоение адреса	0	0	0 0 0 0 0	A4 A3 A2 A1 A0	PB	1
Другие команды						
Сброс	0	1	A4 A3 A2 A1 A0	1 1 1 0 0	PB	1
Удалить адрес	0	1	A4 A3 A2 A1 A0	0 0 0 0 0	PB	1
Считать конфигурацию ввода-вывода	0	1	A4 A3 A2 A1 A0	1 0 0 0 0	PB	1
Считать идентификационный код	0	1	A4 A3 A2 A1 A0	1 0 0 0 1	PB	1
Считать статус	0	1	A4 A3 A2 A1 A0	1 1 1 1 0	PB	1

Микросхема ведомого устройства

Для того чтобы электронная часть ведомого устройства, могла быть встроена непосредственно в датчик или исполнительное устройство, она должна быть компактной и, кроме того, дешёвой. Это возможно только при использовании специализированных микросхем с высокой степенью интеграции.

Через интегральную микросхему ведомого устройства двоичные датчики и исполнительные устройства подключаются к сети AS-интерфейса. Специализированная микросхема (ASI Chip, ASIC) обеспечивает датчик или исполнительное

устройство электропитанием от сети, распознает переданную от ведущего устройства информацию и посылает в ответ собственные данные.

Возможны два способа использования таких микросхем:

- микросхема может быть встроена прямо в датчик или исполнительное устройство, в результате чего получается устройство с интегрированным AS-интерфейсом (рис. 4.9);
- микросхема может быть встроена в специальный модуль таким образом, что к модулю можно подключать обыкновенный датчик или исполнительное устройство, которые при этом становятся устройствами с внешним AS-интерфейсом (рис. 4.10).

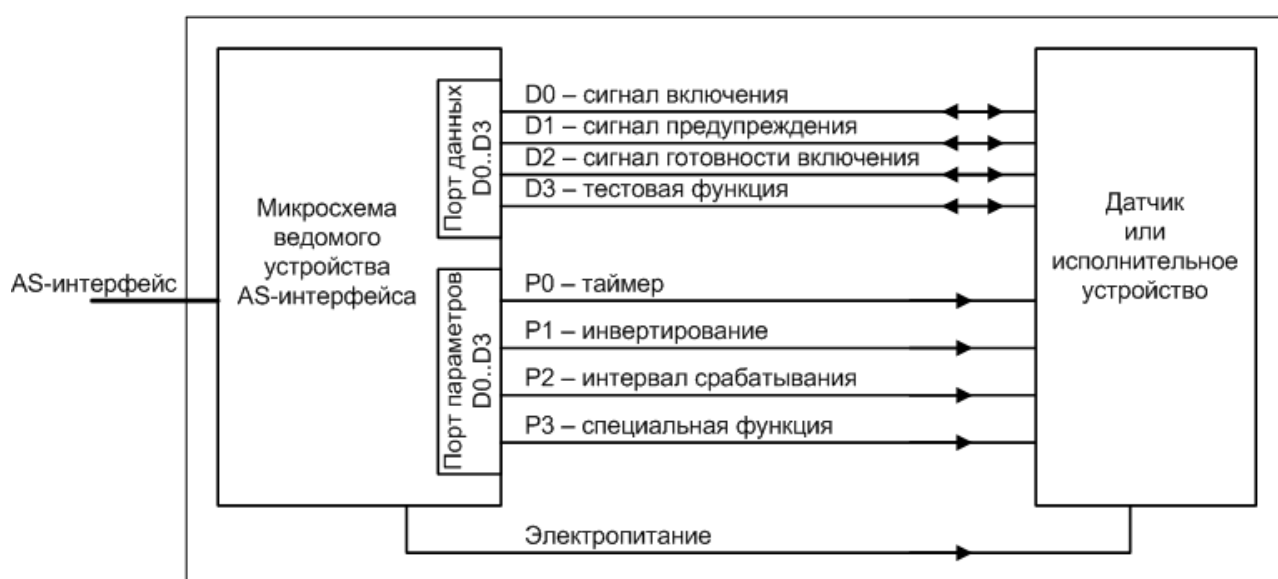


Рис. 4.9. Схема датчика или исполнительного устройства с интегрированным AS-интерфейсом

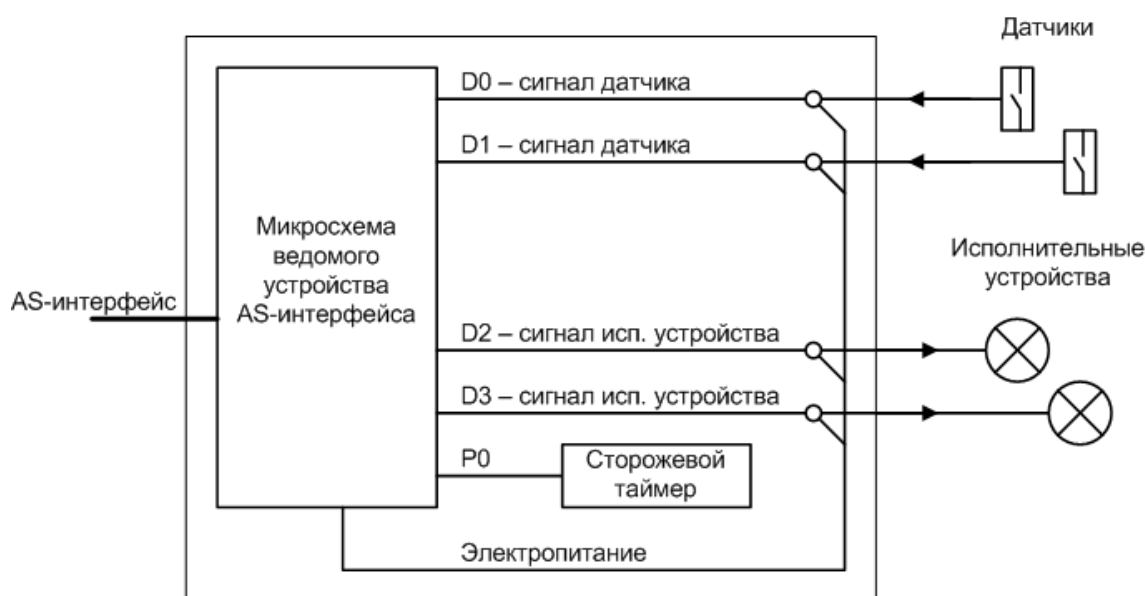


Рис. 4.10. Схема модуля связи сети AS-интерфейса с датчиками и исполнительными механизмами без встроенных микросхем ведомого устройства

Основным производителем микросхем для AS-интерфейса является компания AMS (Austria Microsystems), выпускающая микросхемы ASI3+ и SAP4.

Четыре возможные точки подключения к порту данных микросхемы ведомого устройства (D0–D3, рис. 4.9, 4.10) могут быть сконфигурированы как входы, выходы и двунаправленные порты (входы/выходы). Конфигурирование выполняется производителем AS-устройства и не может быть изменено в процессе эксплуатации.

В основном режиме работы AS-интерфейса ведомые устройства периодически опрашиваются с помощью команды «Запрос данных» (Data Request). При посылке запроса ведущее устройство указывает, какие значения нужно установить на дискретных выходах ведомого устройства. При этом, если какой-то из портов данных сконфигурирован как вход, то переданное для него значение игнорируется. При ответе, ведомое устройство передаёт значения входов, подключённых к соответствующим портам. Для портов, к которым подключены выходы, передаваемые значения не определены (это могут быть нули или единицы).

Например, для устройства на рис. 4.10 порты D0 и D1 сконфигурированы как входы, а порты D2 и D3 – как выходы. Пусть также рассматриваемое ведомое устройство имеет адрес 11_{10} (01011_2). Тогда запрос с командой включить оба исполнительных устройства будет выглядеть так (см. табл. 4.7):

0 0 01011 01100 1 1

а ответ (если на выходе первого датчика установлен логический нуль, а на выходе второго – логическая единица) будет следующим:

0 0010 1 1

С помощью команды «Записать параметр» (табл. 4.7) ведомое устройство получает от ведущего 4 бита данных, соответствующих значениям параметров. С их помощью можно управлять особыми функциями ведомого устройства.

Спецификация AS-интерфейса версии 2.1

Весной 2000 года появилась новая спецификация AS-интерфейса версии 2.1. Количество ведомых устройств в одной сети увеличено до 62 за счёт разделения адресного пространства ведущего сетевого устройства на две подобласти: А и В. Ведомые устройства версии 2.0 стали называться стандартными, а устройства версии 2.1 – устройствами с расширенным режимом адресации.

В стандартном режиме адресации ведомое устройство могло иметь до 4 входов/выходов. Так называемые А/В-устройства (устройства, адресуемые в соответствии со спецификацией версии 2.1) могут иметь до 4 входов или до 3 выходов. Другие технические данные системы на базе AS-интерфейса приведены в табл. 4.8.

Таблица 4.8.

Технические данные системы на базе AS-интерфейса

Параметр	Спецификация версии 2.0	Спецификация версии 2.1
Топология	Магистральная, древовидная, звездообразная и др.	
Число ведомых устройств	До 31	До 62
Число подключаемых датчиков и исполнительных механизмов (ИМ)	До 4 датчиков и 4 ИМ на одно ведомое устройство. До 124 датчиков и 124 ИМ на одно ведущее устройство	До 4 датчиков и 3 ИМ на одно ведомое устройство. До 248 датчиков и 186 ИМ на одно ведущее устройство
Максимальная протяжённость линии связи	Без повторителей – до 100 м С повторителями – до 300 м	
Линия связи	Двухжильный кабель (2x1,5мм ²) специального профиля без экрана, невитой, для одновременной передачи данных и питания	
Электропитание	Через шину AS-интерфейса: 2,8А (ном.), 8А (макс.) / 29,5...31,6 В	
Скорость передачи данных	До 53 кбит/с при общей пропускной способности 167 кбит/с	
Структура сообщений	Одноадресное сообщение ведущего устройства с прямым ответом ведомого устройства	
Время цикла при 31 ведомом устройстве	Не превышает 5 мс (задержка на одно ведомое устройство – порядка 0,15 мс)	
Время цикла при 62 ведомых устройствах	–	Не превышает 10 мс.
Коррекция ошибок	Идентификация ошибок и повторный запрос со стороны ведущего устройства	
Метод доступа	Циклический опрос (сканирование) ведомых устройств	

Для того, чтобы обеспечить совместимость на одной шине устройств спецификации 2.0 и 2.1, формат пакета запроса остался неизменен, а подобласть адресного пространства А или В определяется значением бита D3 в запросе/команде ведущего устройства (рис. 4.8, табл. 4.7). Тогда, на передачу информации от ведущего устройства остаётся только три бита (D2–D0), этим и объясняется уменьшенное количество исполнительных устройств, которые можно подключить к ведомому устройству.

Если в AS-интерфейсе версии 2.1 используются только ведомые устройства одной подобласти адресного пространства (А или В), то время цикла опроса не превышает 5 мс, как и в версии 2.0. В случае использования всего адресного пространства, доступного для версии 2.1, ведомые устройства подобластей А и В обслуживаются по очереди: в первом цикле производится опрос ведомых устройств

подобласти А, во втором — подобласти В, и в такой последовательности циклический процесс опроса повторяется далее. Таким образом, в этом случае суммарное время обслуживания всех ведомых устройств не превышает 10 мс.

Обслуживание ведомых А/В-устройств способны выполнять только ведущие сетевые устройства, поддерживающие спецификацию версии 2.1. Устройства, не поддерживающие данную версию, способны обслуживать не более 31 ведомого устройства (подобласть адресного пространства А).

4.2. ПРОТОКОЛЫ СИСТЕМНОГО УРОВНЯ

Modbus

Этот протокол разработан фирмой Modicon Gould Inc., ныне Schneider Electric [45, 46] для построения промышленных распределённых систем управления. Специальный физический интерфейс для него не определён. Поэтому на физическом уровне протокола Modbus может использоваться любой байт-ориентированный протокол: RS-232C, RS-422, RS-485 или же токовая петля.

Протокол MODBUS работает по принципу «Master – Slave», или «ведущий – ведомый». Конфигурация на основе этого протокола предполагает наличие одного ведущего устройства и до 247 ведомых устройств. Каждому ведомому устройству на этапе конфигурирования сети присваивается уникальный адрес устройства в диапазоне от 1 до 247. Только ведущее устройство инициирует циклы обмена данными (транзакции). Существует два типа транзакций:

- запрос/ответ (адресуется только одно из ведомых устройств);
- широковещательная передача (ведущее устройство, используя нулевой адрес, обращается ко всем остальным узлам сети одновременно без квитирования).

Таким образом, каждая транзакция содержит один кадр запроса и один кадр ответа, либо один кадр широковещательной передачи.

Некоторые характеристики протокола Modbus фиксированы. К ним относятся формат кадра, последовательность кадров, обработка ошибок коммуникации и исключительных ситуаций, и выполнение функций.

Другие характеристики выбираются пользователем. К ним относятся тип средства связи, скорость обмена, проверка на чётность, число стоповых бит, и режим передачи (ASCII или RTU). Параметры, выбираемые пользователем, устанавливаются (аппаратно или программно) на каждом устройстве. Эти параметры устанавливаются одинаковыми для всех устройств сети Modbus и не могут быть изменены во время работы системы.

Режим передачи определяет структуру отдельных блоков информации в сообщении и системы счисления, используемой для передачи данных. В системе

Modbus существуют два режима передачи. Оба режима обеспечивают одинаковую совместимость при связи с ведомыми устройствами. Режим выбирается в зависимости от объединяемого с помощью сети Modbus оборудования. Для каждой сети Modbus должен использоваться только один режим. Смешивание режимов не допустимо. Режимы делятся на **ASCII** (American Standard Code for Information Interchange – Американский стандартный код обмена информацией) и **RTU** (Remote Terminal Unit – модуль удалённого терминала). Основные характеристики режимов ASCII и RTU представлены в табл. 4.9.

Таблица 4.9.

Характеристики режимов ASCII и RTU

Характеристика	ASCII (7-бит)	RTU (8-бит)
Система кодирования	Используются ASCII символы 0–9, A–F	8-битовая двоичная система
Число бит на символ: – стартовый бит; – биты данных; ¹ – бит чётности; – стоповые биты.	1 7 вкл./выкл. 1 или 2	1 8 вкл./выкл. 1 или 2
Контрольная сумма Modbus-кадра	LRC (Longitudinal Redundancy Check – продольный контроль по избыточности)	CRC-16 (Cyclical Redundancy Check – контроль с помощью циклического избыточного кода)

¹ младший бит передаётся первым

Символы ASCII удобнее использовать при отладке, поэтому этот режим удобен для компьютеров, программируемых на языке высокого уровня. Режим RTU обеспечивает более компактный размер кадров, что позволяет сократить длительность транзакции.

В режиме RTU данные передаются в виде восьмиразрядных двоичных символов. В режиме ASCII каждый RTU-символ сначала делится на две четырехразрядных части (старшая и младшая), значение каждой цифры представляется одной 16-тиразрядной цифрой, а затем ASCII-символы этих 16-тиразрядных цифр используются при формировании сообщения. В результате каждый RTU символ представляется двумя ASCII символами, которые могут принимать значения 0–9 и A–F. Их коды ($30_{16} - 39_{16}$ и $41_{16} - 46_{16}$ соответственно) лежат в основной (нижней) части ASCII-таблицы, что позволяет использовать только семь бит для передачи каждого байта (самый старший восьмой бит всегда будет равен нулю, а значит его можно не передавать).

ASCII режим использует в два раза больше символов, чем RTU режим, но декодирование и управление данными – легче. К тому же, в режиме RTU символы

сообщения должны передаваться непрерывным потоком. В режиме ASCII допуска задержка до 1 секунды между двумя соседними символами.

Формат кадра для режимов ASCII и RTU представлен в табл. 4.10.

Таблица 4.10.

Формат кадра для режимов ASCII и RTU

	Старт	Адрес	Функция	Данные	Контр. сумма	Конец
ASCII	1 символ «:»	2 символа	2 символа	2N символов	2 символа	2 символа CR LF
RTU	пауза более 3,5 байт	1 байт	1 байт	N байт	2 байта	пауза более 3,5 байт

В режиме ASCII **начало кадра** обозначается символом двоеточия «:». **Конец кадра** обозначается символами возврата каретки (CR) и перевода строки (LF), указывающими на конец кадра. Символ перевода строки (LF) также служит как синхронизирующий символ, который указывает на то, что передающая станция готова к приёму ответного сообщения.

В режиме RTU приёмное устройство отслеживает время между приёмом символов. Если прошло время, равное периоду следования 3,5 символов, а кадр не был завершён или не поступило нового символа, устройство очищает кадр и предполагает, что следующий принимаемый байт – это адрес устройства в новом сообщении.

Поле адреса следует сразу за началом кадра и состоит из одного восьмиразрядного символа в режиме RTU или 2-х символов в режиме ASCII. Эти биты указывают пользователю адрес ведомого устройства, которое должно принять сообщение, посланное ведущим устройством.

Каждое ведомое устройство имеет уникальный адрес и только адресуемое устройство может ответить на запрос, который содержит его адрес. Когда ведомое устройство посылает ответ, адрес, указываемый в кадре ответа, информирует ведущее устройство, о том, какое ведомое устройство на связи. В ширококвещательном режиме используется адрес 0. Все ведомые устройства интерпретируют такое сообщение как выполнение определённого действия, но без посылки подтверждения.

Поле кода функции указывает адресуемому ведомому устройству какое действие выполнить. Коды функций Modbus специально разработаны для связи персональных компьютеров и промышленных коммуникационных систем Modbus. Старший бит этого поля устанавливается ведомым устройством в единицу в

случае, если ведомое устройство сообщает ведущему устройству, что ответное сообщение ненормальное.

Поле данных содержит информацию, необходимую ведомому устройству для выполнения указанной функции, или содержит данные собранные ведомым устройством для ответа на запрос.

Поле контрольной суммы позволяет Modbus-устройствам проверять сообщение на наличие ошибок. Существует два типа ошибок, которые могут возникать в системах связи: ошибки передачи и программные или оперативные ошибки. Протокол Modbus позволяет определять оба типа ошибок.

Ошибки связи обычно заключаются в изменении бита или бит сообщения. Например, байт 0001 0100 может измениться на 0001 0110. Ошибки связи выявляются при помощи символов начала и конца кадра, контроля по чётности и избыточным кодированием.

Когда обнаруживается ошибка кадрирования, чётности и контрольной суммы, обработка сообщения прекращается. Ведомое устройство не должно генерировать ответное сообщение, поскольку при возникновении ошибки связи, ведомое устройство не может быть уверенно, что сообщение было адресовано именно ему.

Ведущее устройство должно программироваться так, чтобы в случае отсутствия ответного сообщения в течение определённого времени, была зафиксирована ошибка связи. Продолжительность этого времени зависит от скорости обмена, типа сообщения, и времени опроса ведомого устройства. По истечении этого периода ведущее устройство повторяет запрос.

Для обеспечения качества передачи данных протокол Modbus обеспечивает несколько уровней обнаружения ошибок. Для обнаружения множественного изменения битов сообщения протокол использует избыточный контроль: CRC и LRC для режимов RTU и ASCII соответственно.

Вычисление контрольной суммы отличается для ASCII и RTU режимов. Контрольная сумма в режиме ASCII – LRC – это восьмиразрядное число, передаваемое как два ASCII-символа. Контрольная сумма рассчитывается следующим образом [45, 46]. Все байты сообщения, исключая символы начала и конца кадра складываются без учёта переноса, а затем вычисляется дополнительный код полученного числа. В приёмнике все полученные байты сообщения также складываются и, если передача прошла без ошибок, то суммирование полученной суммы с принятым значением контрольной суммы без учёта переноса даст нуль (см. табл. 4.11).

**Пример расчёта LRC для сообщения «Запрос на чтение
первых восьми булевых ячеек из устройства с адресом 02»**

	ASCII-символы сообщения	Значения байтов сообщения в двоичном формате	Примечания	
Сообщение, сформированное передатчиком				
Адрес	0 2	0000 0010		
Функция	0 1	0000 0001		
Адрес первой ячейки (HIGH)	0 0	0000 0000		
Адрес первой ячейки(LOW)	0 0	0000 0000		
Число ячеек(HIGH)	0 0	0000 0000		
Число ячеек(LOW)	0 8	0000 1000		
Расчёт LRC			0000 1011 1111 0100 1111 0101	Сумма Инверсия Доп. код
Контрольная сумма	F 5	1111 0101		
Сообщение, полученное приёмником				
Приёмное устройство	0 2	0000 0010	Сумма всех принятых байт равна нулю (0000 0000), значит сообщение принято без ошибок.	
складывает все байты	0 1	0000 0001		
данных, включая LRC.	0 0	0000 0000		
Сумма может превышать	0 0	0000 0000		
8 бит, в расчёт принимаются	0 0	0000 0000		
только младшие 8 бит.	0 8	0000 1000		
	F 5	1111 0101		

Расчёт CRC-кода несколько сложнее, алгоритм его расчёта может быть найден в [45, 46].

Команды протокола Modbus

Каждое Modbus-устройство рассматривается протоколом как некоторый контроллер с дискретными и аналоговыми входами и выходами, к которым подключены соответственно датчики и исполнительные механизмы. Состояния входов и выходов такого Modbus-устройства представляются в виде четырёх таблиц данных. В каждой таблице располагаются данные определённого типа, см. табл. 4.0.

В таблицах дискретных входов и входных регистров хранятся данные, которые получены от дискретных и аналоговых датчиков. Поскольку эти данные описывают состояния датчиков, то и измениться они могут только при изменении состояний датчиков. Поэтому для таблиц входов определён режим доступа «только чтение».

Типы данных Modbus

Тип данных	Размер поля	Обозначение	Режим доступа
Дискретные входы (Discrete Inputs)	1 бит	%IX	только чтение
Дискретные выходы (Coils – катушки реле) ¹	1 бит	%QX	чтение/запись
Входные регистры (Input Registers)	16-разрядное слово	%IW	только чтение
Регистры временного хранения информации (Holding Registers)	16-разрядное слово	%QW	чтение/запись

¹ Поскольку для включения дискретных исполнительных механизмов чаще всего используются электромагнитные реле, то и дискретные выходы были названы «coils», чтобы подчеркнуть этот факт.

Таблицы, описывающие состояния дискретных выходов и регистров могут использоваться для хранения информации о состояниях дискретных и аналоговых исполнительных механизмов, подключенных к Modbus-устройству. Для управления исполнительными механизмами состояния выходов могут изменяться как программой самого узла Modbus, так и командами, переданными по сети. Кроме того, таблицы дискретных выходов и регистров временного хранения информации могут использоваться для размещения переменных соответствующего типа.

Каждая таблица располагается в соответствующей области памяти Modbus-устройства и может содержать до 65536 значений параметров. Номер параметра, его позиция в таблице определяется числом в диапазоне от 1 до 65536. Адреса параметров при передаче по протоколу Modbus задаются беззнаковым целым числом (unsigned integer) которое принимает значения от 0 до 65535. Таким образом, при формировании запроса необходимо отнимать единицу от адреса запрашиваемого параметра.

Набор команд протокола Modbus описывает функции:

- чтение/запись битов и битовых последовательностей;
- чтение/запись регистров и группы регистров;
- функции диагностики;
- программные функции;
- функции управления списком опроса;
- функция сброса (RESET).

Полный список кодов функций и форматов запросов можно найти в [45, 46]. Основные функции доступа к данным и диагностики, чаще всего реализуемые Modbus-устройствами представлены в табл. 4.13 [47].

Коды функций Modbus

Тип доступа	Область доступа	Название функции	Код
Побитовый доступ	Дискретные входы	Чтение группы дискретных входов	02
	Дискретные выходы или внутренние логические переменные	Чтение группы дискретных выходов	01
		Запись одного дискретного выхода	05
		Запись группы дискретных выходов	15
16-разрядный доступ	Входные регистры	Чтение входного регистра	04
	Выходные регистры или внутренние целочисленные переменные	Чтение нескольких регистров хранения информации	03
		Запись одного регистра хранения	06
		Запись группы регистров хранения	16
Диагностика			08

Функция диагностики реализует серию проверок системы коммуникации между ведущим и ведомым устройствами или проверку на различные внутренние ошибки в ведомом устройстве. Широкое вещание не поддерживается. Функция использует два байта кода подфункции в запросе для определения типа теста, который необходимо провести (например, перезапуск, чтение значения диагностического регистра, переход в режим «только приём»). Ведомое устройство возвращает оба кода функции и подфункции в нормальном ответе. Большинство диагностических запросов используют два байта поля данных для посылки диагностических данных или контрольной информации ведомому устройству. Некоторые результаты диагностики могут возвращаться подчинённым в поле данных нормального ответа.

Пример. Ниже приведена транзакция между ведущим устройством и ведомым с адресом 17_{10} . В результате транзакции происходит чтение значений группы регистров хранения информации (код функции 03) с адресами

%QW109 – %QW112.

Использование функции с кодом 03 требует указания в поле данных запроса адреса первого регистра группы (2 байта, старший байт передаётся первым) и количества считываемых регистров (2 байта, старший байт передаётся первым).

В ответе ведомого устройства возвращается адрес устройства, код функции. В первом байте поля данных указывается количество байтов информационных байтов в поле. Количество информационных байтов определяется количеством считываемых регистров – по два байта на каждый регистр (старший байт, младший байт).

Для формирования запроса необходимо все числа перевести в шестнадцатеричную форму записи:

- адрес ведомого устройства – $17_{10} \rightarrow 11_{16}$,

- код функции – $03_{10} \rightarrow 03_{16}$,
- адрес первого регистра – $(109 - 1) = 108_{10} \rightarrow 6C_{16}$,
- количество считываемых регистров – $04_{10} \rightarrow 04_{16}$.

Запрос в формате Modbus-ASCII будет выглядеть следующим образом.

: 11 03 00 6C 00 04 7D ↵

Ответ ведомого устройства:

: 11 03 08 0B C7 00 F4 1A 20 09 93 48 ↵

После «разбора» ответа ведущим устройством, получаем значения считанных регистров:

- %QW109: $0BC7_{16} \rightarrow 3015_{10}$;
- %QW110: $00F4_{16} \rightarrow 244_{10}$;
- %QW111: $1A20_{16} \rightarrow 6688_{10}$;
- %QW112: $0993_{16} \rightarrow 2451_{10}$.

Как интерпретировать полученные данные зависит от конкретного ведомого устройства. Ведущее устройство должно «знать», какие именно данные лежат в регистрах по запрашиваемым адресам. Их обработка осуществляется прикладной программой ведущего устройства.

Протокол Modbus можно назвать наиболее распространённым в мире. Для работы со своими изделиями его используют десятки фирм. Хотя ограничения этого протокола достаточно очевидны, он привлекает простотой логики и независимостью от типа интерфейса.

Modbus Plus

Компания Schneider Electric предлагает для объединения своих контроллеров закрытый стандарт сети Modbus Plus (MODBUS+, MB+) [48]. Передача данных осуществляется со скоростью 1 Мбит/с, в качестве среды передачи используется экранированная витая пара. Сеть Modbus Plus реализует децентрализованный способ управления доступом к шине по методу передачи маркера. Протокол Modbus Plus позиционируется как высокоуровневый протокол управления каналом передачи данных (High level Data Link Control, HDLC) [47].

Сеть поддерживает до 64 адресуемых узлов. До 32 устройств могут быть подключены к сетевому кабелю непосредственно на расстоянии до 450 метров (рис. 4.11). При помощи повторителей длина сети может быть увеличена до 1800 метров, а количество устройств – до 64. Для больших расстояний имеются волоконно-оптические повторители.

Минимальная длина кабеля между любыми двумя узлами сети составляет 3 метра. Максимальное расстояние между двумя узлами совпадает с максимальной длиной кабельного сегмента – 450 метров. При монтаже двухкабельной сети

Modbus Plus с резервированием максимальная разница длин двух кабелей не должна превышать 150 метров (рис. 4.12).

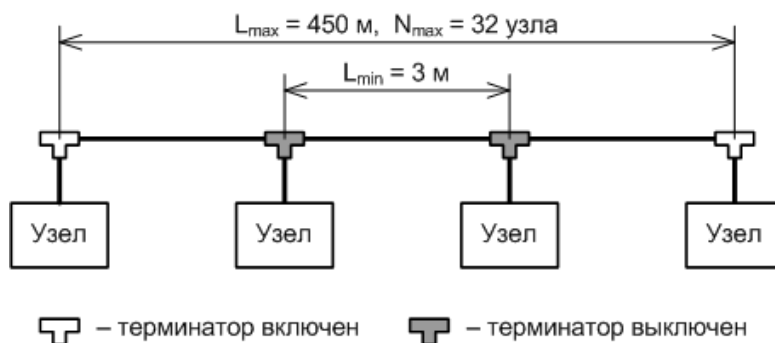


Рис. 4.11. Сегмент сети Modbus Plus с одним кабелем

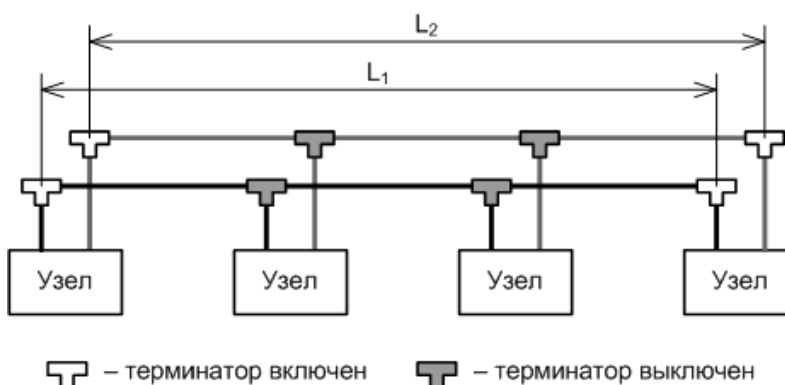


Рис. 4.12. Сегмент сети Modbus Plus с резервированием. $|L_1 - L_2| < 150$ м

Каждый узел подключается к магистральному кабелю с помощью специальных ответвителей (рис. 4.13). Они обеспечивают электрическое соединение магистрального кабеля и кабеля снижения для подключения узла сети. В каждом ответвителе имеются встроенные терминаторы, которые подключаются с помощью перемычек. Подключение терминаторов осуществляется только в ответвителях, расположенных на концах кабельного сегмента, во избежание отражения сигнала. На конце сегмента магистральный кабель подключается с любой стороны ответвителя. Терминатор подключается к противоположным контактам ответвителя.

Устройства, поддерживающие протокол Modbus, а также другие устройства с последовательным протоколом могут быть подключены к сети с помощью мостов-мультиплексоров. Каждый мост-мультиплексор имеет четыре последовательных конфигурируемых порта. Подключенные к мосту-мультиплексору последовательные устройства могут обмениваться данными с устройствами, подключенными в сеть Modbus Plus, а также с другими устройствами, подключенными к последовательным портам моста-мультиплексора. Каждый порт моста-мультиплек-

сора может быть сконфигурирован для работы в режиме ведущего или ведомого устройства сети Modbus.

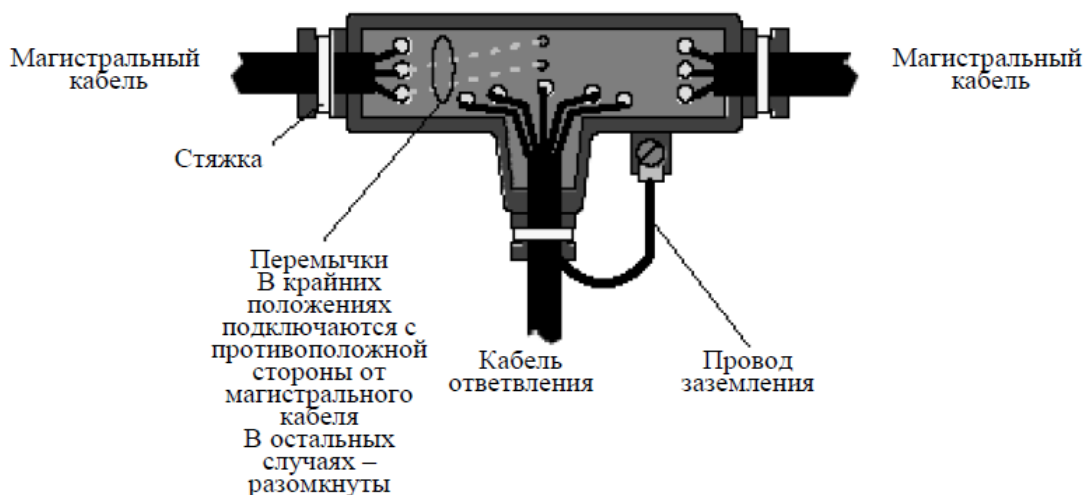


Рис. 4.13. Подключение к сети Modbus Plus с помощью ответвителя

Несколько сетей Modbus Plus могут быть объединены в единую информационную систему с помощью мостов.

Логическая организация Modbus Plus

Каждому узлу сети пользователем задаётся уникальный адрес. Адрес узла не зависит от его физического положения в сети. Адреса устройств должны лежать в диапазоне от 1 до 64.

Все узлы являются равноправными членами логического кольца по которому передаётся маркер. Маркер представляет собой специальное сообщение, которое пересылается от одного узла к другому — от узла с младшим адресом к узлу со старшим. Узел с самым старшим адресом пересылает маркер узлу с самым младшим адресом, замыкая, таким образом, логическое кольцо.

Там, где несколько сетей соединены мостами, маркер не пересылается через мост. Каждая сеть обслуживает свою собственную циклическую последовательность маркера независимо от других сетей (рис. 4.14).

Узел, владеющий маркером, может инициировать транзакции с другими узлами. Каждое сообщение содержит поля маршрута, которые определяют адреса источника и приёмника сообщения, включая маршрут через мосты к узлу-приёмнику в удалённой сети. Каждое сообщение может содержать до 100 регистров (16-тибитных слов) данных. Остальные узлы проверяют сеть на приходящие сообщения.

Когда узел получает сообщение, он немедленно посылает подтверждение узлу-источнику. Если сообщение является запросом данных, то узел-приёмник начинает готовить запрошенные данные для ответного сообщения. Готовое сообще-

ние будет передано узлу, запросившему данные, после получения узлом-приёмником маркера, разрешающего ему передачу. Таким образом, каждое сообщение от узла-источника немедленно квитируется узлом-приёмником, однако ответная посылка сообщения происходит лишь при получении маркера.

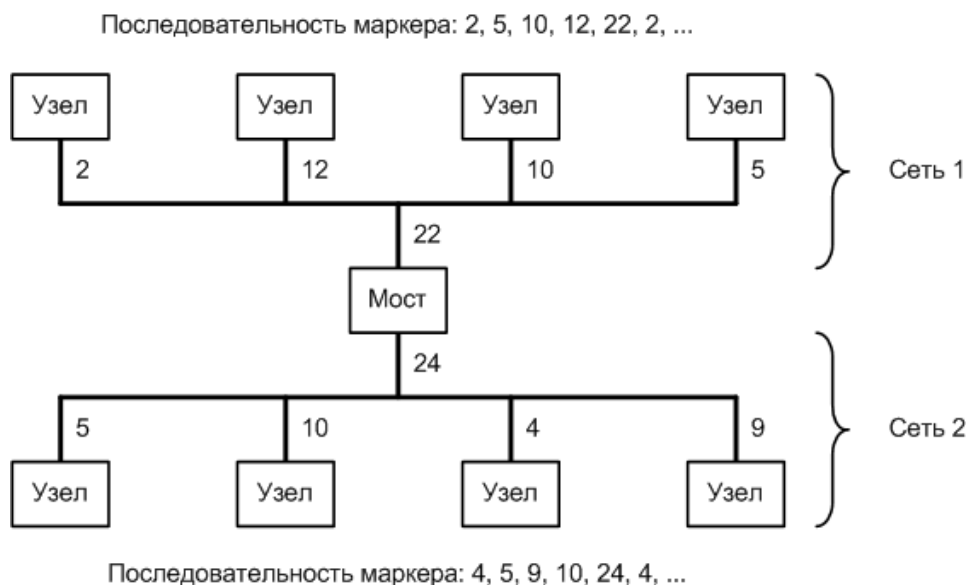


Рис. 4.14. Последовательность вращения маркера в Modbus Plus

Узлы могут также передавать сообщения, содержащие статистические данные и сообщения об ошибках.

После того, как узел отправит все свои сообщения, он пересылает маркер следующему узлу последовательности.

Принципы работы моста Modbus Plus

Мост Modbus Plus подключается как узел в обе сети, которые он соединяет. Он работает как независимый узел в каждой сети, получающий и отправляющий маркеры согласно последовательности циркуляции маркера в каждой сети.

На рис. 4.15 показаны три сети (А, В и С), объединяемые двумя мостами. Один мост принадлежит сети А как узел с адресом 22, а сети В как узел с адресом 25. Другой мост в сети В является узлом 20, а в сети С – узлом 20. Сетевые адреса моста в каждой сети устанавливаются независимо друг от друга.

Когда сообщение, предназначенное для удалённого узла, получено на одном из портов моста, мост запоминает сообщение, а затем направляет его по адресу узла в следующую сеть после получения маркера для передачи в той сети.

Каждый кадр сообщения содержит маршрутную информацию, которая позволяет переслать его через несколько мостов на конечный узел на удалённой сети. Маршрут задаётся при подготовке сообщения узлом-источником. Сообщение

может пройти до четырёх мостов, прежде чем попадёт к конечному адресату. На рис. 4.16 показан пример поля маршрута в кадре типичного сообщения.

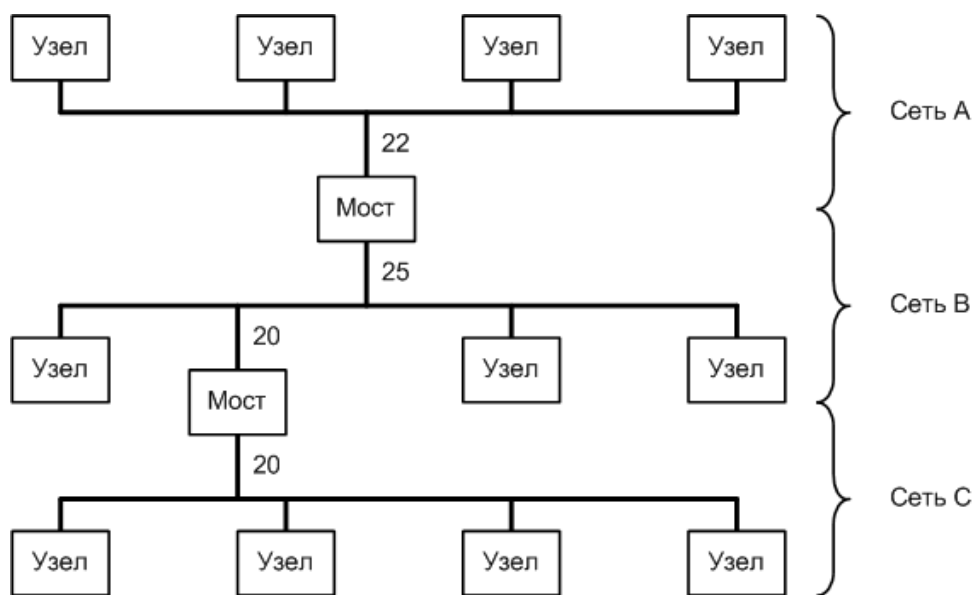


Рис. 4.15. Маршрутизация сообщений в объединённых сетях

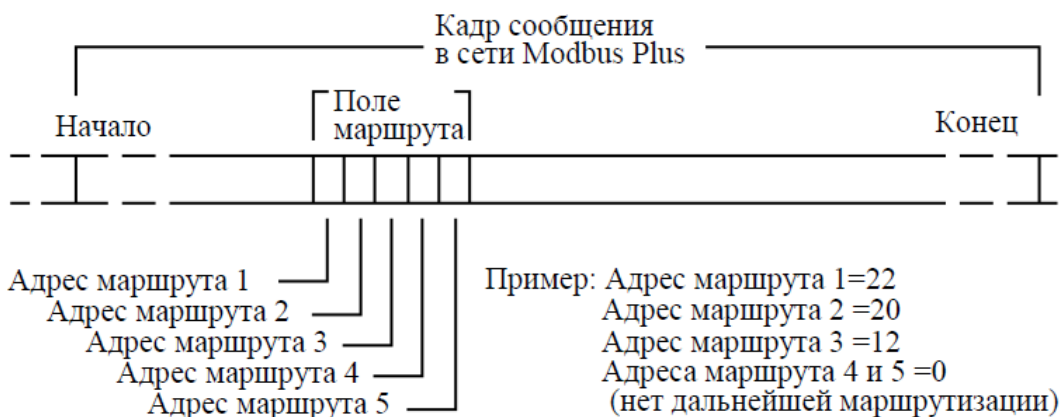


Рис. 4.16. Описание поля маршрута в кадре сообщения

На рис. 4.15 и 4.16 показан пример пересылки сообщения через два моста. Сообщение исходит из узла 5 сети А и предназначено для узла с адресом 12 в сети С. Узел-источник должен задать маршрут передачи сообщения:

- первый адрес маршрута содержит адрес моста (22) в сети А источника сообщения;
- второй адрес маршрута содержит адрес моста (20) в следующей сети В;
- третий адрес маршрута содержит адрес узла-приёмника (12) в конечной сети С;
- остальные поля маршрута будут содержать нули, что означает, что дальнейшая пересылка сообщений не нужна.

Когда первый мост в сети А (22) получает исходное сообщение, он проверяет поле маршрута и определяет, что требуется переправка сообщения на другой сетевой порт, поскольку следующий адрес в поле ненулевой. Мост удаляет свой адрес из поля маршрута, сдвигает оставшиеся адреса в поле маршрута на одно место влево и обнуляет поле справа. Тогда следующий адрес маршрута 20 окажется в первой позиции поля. Когда мост получит маркер для передачи в сети В, он передаст сообщение узлу с адресом (20) в сети В.

Второй мост (20) обрабатывает сообщение таким же образом, удаляя собственный адрес из поля маршрута и сдвигая оставшиеся адреса на одну позицию влево. Узел 12 становится конечным приёмником, т.к. все остальные адреса поля маршрута стали нулевыми. Когда маркер сети С будет получен мостом, он перешлёт сообщение узлу 12 сети С.

Транзакция глобальной базы данных

Когда узел пересылает маркер, он может транслировать до 32 слов (16 бит каждое) глобальной информации всем другим узлам сети. Информация содержится в маркерном кадре. Процесс передачи глобальных данных при пересылке маркера управляется прикладной программой каждого узла независимо.

Глобальные данные становятся доступны всем узлам сети. Каждый узел в своей памяти формирует таблицу глобальных данных, в которую записываются глобальные данные, переданные всеми узлами. Каждый раз при пересылке маркера часть таблицы, соответствующая пересылающему узлу обновляется. Использование глобальной базы данных позволяет быстро обновлять сигналы тревоги, уставки и другую информацию. Скорость обновления данных при использовании глобальной базы данных в среднем в восемь раз выше, чем при обычном обмене сообщениями между узлами. Фактически, передача глобальных данных представляет собой реализацию широковещательной передачи, которая осуществляется каждый раз при смене ведущего узла.

Каждая сеть содержит свою глобальную базу данных, т.к. маркер не пересылается через мост в другую сеть.

Modbus/TCP

Широкое распространение протокола Ethernet-TCP/IP при построении локальных вычислительных сетей привело к их естественному проникновению в системы управления производством. В качестве основного аргумента при этом используется идея «бесшовного» соединения всех уровней классической пирамиды автоматизации: от уровня автоматизации технологических процессов до уровня управления предприятием. Реализация этой идеи потребовала серьезной адаптации Ethernet, особенно в плане поддержки реального времени [49]. Недетерми-

нированные протоколы связи типа HTTP и FTP конечно обеспечивают универсальность и удобство использования, но для применения в промышленности все же пришлось разрабатывать на основе Ethernet специальные прикладные протоколы.

Недавно образованная при поддержке Schneider Electric группа Modbus-IDA (International Development Association – Международная ассоциация развития) [50] предлагает архитектуру Modbus/TCP для распределённых систем управления, использующую структуру сообщений Modbus. Modbus/TCP это симбиоз стандартного протокола Modbus и протокола Ethernet-TCP/IP как средства передачи данных. В результате получился простой, структурированный, открытый протокол передачи. Поскольку функции управления доступом к среде передачи реализуются на уровне протокола Ethernet-TCP/IP, то каждый участник сети может инициировать транзакцию. Инициатор транзакции (клиент) формирует запрос, а устройство, которому адресован запрос (сервер), формирует ответ.

Протокол Modbus позиционируется группой Modbus-IDA как протокол уровня приложений модели взаимодействия открытых систем, который обеспечивает клиент-серверный обмен данными между устройствами, объединёнными различными типами шин или сетей (рис. 4.17). Все три протокола из семейства Modbus (Modbus RTU, Modbus Plus и Modbus/TCP) используют один протокол уровня приложений, что позволяет обеспечить их совместимость на уровне обработки пользовательских данных.

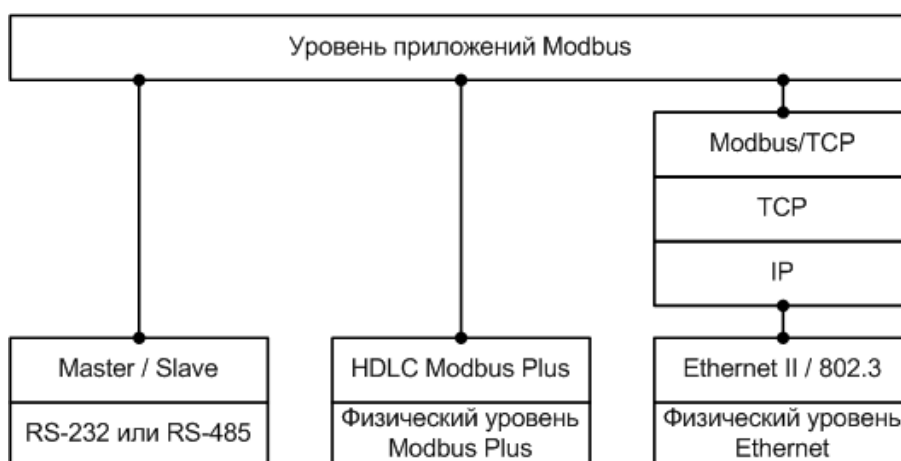


Рис. 4.17. Семейство протоколов Modbus

Протокол уровня приложений определяет простой кадр данных протокола (PDU, Protocol Data Unit), состоящий из двух полей – поля кода функции и поля данных. Коды функций и формат поля данных остаются неизменными для любого протокола семейства Modbus.

Для передачи данных по сети, в зависимости от её типа, используются дополнительные поля кадра данных приложения (ADU, Application Data Unit). На рис. 4.18 представлены форматы кадров данных приложения для различных типов сетей. Можно заметить, что в кадре данных приложения Modbus/TCP отсутствует поле контрольной суммы. Нет в нём и поля адреса. Это связано с тем, что функции адресации и проверки правильности передачи по сети берёт на себя протокол Ethernet-TCP/IP.



а) Формат сообщения Modbus-RTU для сети на базе RS-422/485



б) Формат Modbus-сообщения для передачи данных по TCP/IP

Рис. 4.18. Modbus-сообщение

Для идентификации Modbus-сообщения в сети TCP/IP используется специальный заголовок – MBAP header (Modbus Application Protocol header). Этот заголовок состоит из полей, представленных в таблице 4.14.

Таблица 4.14.

Поля заголовка МВАР

Поле	Длина, байт	Запрос	Ответ
Идентификатор транзакции	2	Формируется при запросе	Копируется в ответ из запроса
Идентификатор протокола	2	Формируется при запросе	Копируется в ответ из запроса
Длина сообщения в байтах	2	Формируется при запросе	Формируется при ответе
Идентификатор устройства	1	Формируется при запросе	Копируется в ответ из запроса

Длина заголовка составляет семь байтов. Поле **идентификатора транзакции** используется для того, чтобы инициатор транзакции мог понять на какой его запрос пришёл ответ. Например, при формировании каждого нового запроса поле идентификатора транзакции увеличивается на единицу. **Идентификатор протокола** всегда равен нулю – признак протокола Modbus. Поле «Длина сообщения в

байтах» содержит количество последующих байтов сообщения, включая поле идентификатора устройства. Поле **идентификатора устройства** может использоваться для задач маршрутизации между сетями семейства протоколов Modbus. Оно используется, например, для обращения к ведомому устройству сети Modbus через шлюз между сетью Modbus/TCP и Modbus на базе последовательного интерфейса.

Все Modbus/TCP-сообщения передаются через зарегистрированный порт 502.

Modbus-IDA это не только протоколы на основе Modbus, это целая архитектура, объединяющая методы построения различных систем автоматизации с распределённым интеллектом и описывающая как структуру системы управления в целом, так и интерфейсы устройств и программного обеспечения в частности. Это обеспечивает вертикальную и горизонтальную интеграцию всех уровней автоматизации с широким использованием web-технологий.

Передача данных в реальном времени обеспечивается использованием стека IDA (рис. 4.17), являющегося надстройкой над TCP/UDP и основанного на протоколе Modbus. Передача некритичных ко времени данных и поддержка web-технологий происходит через стек TCP/IP. Предусмотрена возможность удалённого управления устройствами и системами (диагностика, параметризация, загрузка программ и т.п.) при помощи стандартных протоколов HTTP, FTP и SNMP.

CAN

История этого протокола началась в начале 80-х годов [33]. Технологии того времени подошли к такому этапу, на котором возникла необходимость сбора и обработки результатов от множества датчиков, устанавливаемых в автомобилях, за короткие промежутки времени. Эту задачу можно было решить только при использовании сетевой структуры, объединяющей все компоненты и использующей для этой цели недорогую, последовательную сетевую структуру. По этой причине фирма Robert Bosch GmbH (Германия) разработала протокол «Control Area Network» (CAN, сеть области управления), который был утверждён Международной организацией по стандартизации в качестве стандарта ISO 11898. В настоящее время организация, объединившая производителей различных CAN-устройств – CAN in Automation (CiA, CAN в автоматизации) [51] занимающаяся продвижением CAN-протокола расшифровывает аббревиатуру CAN несколько по иному: Controller Area Network – сеть уровня контроллеров.

Протокол CAN используется для организации последовательной передачи данных по шине с высокой степенью надёжности. Протокол отвечает требованиям задач реального времени. В автомобильной промышленности системы управ-

ления двигателем, датчики, системы АБС и т.п. подключаются к шине CAN со скоростью передачи до 1 Мбит/с.

CAN-протокол описывает уровень канала данных и часть физического уровня модели взаимодействия открытых систем (уровень 2 и, частично, уровень 1 модели ВОО).

На уровне канала данных протокол описывает два подуровня [52]:

- подуровень управления логической связью – Logical Link Control (LLC) – верхний подуровень;
- подуровень управления доступом к среде – Medium Access Control (MAC).

Подуровень LLC предоставляет сервисы для передачи данных и удалённых запросов данных, решает, какие сообщения, полученные LLC подуровнем актуальны для данного узла.

Подуровень MAC в основном реализует протокол передачи данных, т.е. формирует и принимает сообщения, реализует арбитраж, квитирование, проверку ошибок передачи, формирование сообщений об ошибках. Именно на MAC-подуровне контролируется состояние шины – свободна/занята. Кроме того, основные параметры битовой синхронизации описываются на MAC-подуровне.

На физическом уровне CAN-протокол описывает требования к физической передаче сигналов, кодированию битов и синхронизации. Протокол не определяет жестко уровни логических сигналов и электрические характеристики среды передачи, они могут быть выбраны производителем CAN-устройств. Естественно, что в рамках одной сети все устройства должны соответствовать одинаковым требованиям.

Для передачи данных по физическому каналу используется кодирование без возвращения к нулю (NRZ-кодирование). Поскольку при NRZ-кодировании уровень сигнала может оставаться постоянным относительно долгое время, например, при передаче группы битов с одинаковыми значениями, то, чтобы избежать рассинхронизации приёмников и передатчика, после каждых пяти битов одинакового значения в сообщение вставляется один добавочный бит противоположного значения – **правило добавочного бита**. Естественно, приёмник удаляет все добавочные биты перед обработкой полученного сообщения.

По требованиям CAN-протокола, среда передачи должна находиться в одном из двух состояний: логический нуль (нижний уровень, CAN_L) – доминирующее состояние, а логическая единица (верхний уровень, CAN_H) – рецессивное состояние. При одновременной передаче доминирующего и рецессивного битов на шине должно присутствовать доминирующее значение. Этот механизм используется при арбитраже сообщений.

В CAN-протоколе используется метод широковещательной передачи сообщений – каждый приёмник сам решает, нужно ли ему обрабатывать очередное передаваемое сообщение. Содержание каждого сообщения определяется идентификатором сообщения. Идентификатор не определяет узел-приёмник, а описывает «смысл» передаваемых данных (можно сказать, что это имя передаваемой переменной), благодаря чему остальные узлы сети могут решить для себя, должны ли они обрабатывать передаваемые данные или нет.

Благодаря применению метода широковещательной передачи, CAN-устройства не используют никакой информации о конфигурации сети (адреса узлов отсутствуют). Узлы могут подключаться к сети и отключаться от неё без каких-либо изменений в настройках и программном обеспечении остальных узлов. Протокол также не ограничивает количество узлов, которые могут быть подключены к одной сети. На практике максимальное количество узлов сети определяется предельно допустимой нагрузкой на шину.

Когда шина свободна, любой узел может начать передачу. Если несколько узлов начали передачу одновременно, конфликт разрешается с помощью поля идентификатора сообщений (поля арбитража). Во время передачи сообщения, передатчик сравнивает значение передаваемого им бита со значением, установленным на шине. Если передаваемое и принимаемое значение бита совпадают, то узел может продолжать передачу. Если же при передаче рецессивного бита шина находится в доминирующем состоянии, то передатчик должен прекратить передачу (табл. 4.15). Таким образом, право на передачу по шине получает тот узел, который передаёт сообщение с наивысшим приоритетом. Важно понимать, что приоритетным является не передающий или принимающий узел, а сообщение.

Таблица 4.15.

Пример поразрядного арбитража

Идентификатор передаваемого сообщения	Передаваемые узлом данные	Примечание
Узел №1: 10110110101	101 1	узел №1 проигрывает арбитраж и прекращает передачу сообщения
Узел №2: 10101100111	101 0 110 0 111 ...	узел №2 выигрывает арбитраж и продолжает передачу сообщения
Узел №3: 10101101000	101 0 110 1	узел №3 проигрывает арбитраж и прекращает передачу сообщения
Состояние шины	101 0 110 0 111 ...	

Типы сообщений в CAN-протоколе

В настоящее время в CAN-протоколе используются два различных формата сообщений, которые отличаются длиной поля идентификатора [52, 53]. Сообщения с 11-тибитным полем идентификатора называются стандартными, сообщения с 29-тибитным полем идентификатора называются расширенными. Формат расширенных сообщений определён таким образом, чтобы стандартные сообщения выигрывали арбитраж.

Протокол определяет следующие типы сообщений или фреймов (от англ. frame – кадр):

- фрейм данных (Data Frame), перемещает данные с передатчика на приёмники;
- удалённый фрейм (Remote Frame) запрашивает передачу фрейма данных с таким же как у него идентификатором;
- фрейм ошибки (Error Frame) передаётся любым узлом, при обнаружении ошибки CAN-сети;
- фрейм перегрузки (Overload Frame) обеспечивает дополнительную задержку между предыдущим фреймом и следующим фреймом данных или удалённым фреймом.

Фреймы данных и удалённые фреймы могут передаваться как в стандартном, так и в расширенном форматах. Каждый фрейм отделяется от предыдущего с помощью межфреймового интервала.

Фрейм данных состоит из семи битовых полей (рис. 4.19, табл. 4.0).

SF	Arbitration Field	Control Field	Data Field	CRC Field	Ack Field	EF
----	-------------------	---------------	------------	-----------	-----------	----

Рис. 4.19. Структура фрейма данных

Поле «Начало фрейма» состоит из одного доминирующего бита. Станция может начать передачу, только если шина свободна (Bus Idle). В этом случае она находится в рецессивном состоянии, и передний фронт бита используется для синхронизации приёмников.

Поле арбитража и управляющее поле различаются для стандартных и расширенных сообщений (табл. 4.17).

В стандартном формате поле арбитража содержит 11-тибитный идентификатор и RTR-бит (Remote Transmission Request, бит удалённого запроса). В расширенном формате поле арбитража содержит 29-ти битный идентификатор, разбитый на две части – 11 бит и 18 бит, базовый идентификатор и расширенный иден-

тификатор соответственно, SSR-бит (Substitute Remote Request, бит замещения), IDE-бит (Identifier Extension bit, бит-признак расширенного формата) и RTR-бит.

Таблица 4.0.

Формат фрейма данных

Имя поля	Длина, бит	Назначение
SF (Start of Frame) Старт-бит фрейма	1	Синхронизация приёмника и передатчика
Arbitration Field Поле арбитража	11+1 или 29+3	Идентификация сообщений; реализация механизма арбитража. Различается для стандартных и расширенных сообщений
Control Field Управляющее поле	2+4	Содержит длину поля данных в байтах
Data Field Поле данных	от 0 до 64 (кратно 8)	Содержит передаваемые данные
CRC Field Контрольная сумма	15+1	Обеспечивает механизм контроля передаваемых данных с помощью циклического избыточного кода
Ack Field Поле подтверждения	2	Служит для подтверждения приёмником корректной передачи сообщения
EF (End of Frame) Стоп-бит фрейма	7	

Таблица 4.17.

**Поле арбитража и управляющее поле фрейма данных
в стандартном и расширенном форматах**

	Поле арбитража		Управляющее поле				Поле данных		
Стандартный формат	идентификатор	R T R	I D E	r0	DLC				
Длина, бит	11	1	1	1	4				
Значение		D	D	D					
	Поле арбитража						Управляющее поле		
Расширенный формат	базовый идентификатор	S R R	I D E	расширенный идентификатор		R T R	r1	r0	DLC
Длина, бит	11	1	1	18		1	1	1	4
Значение		R	R			D	D	D	4

D – доминирующий бит, R – рецессивный бит.

RTR-бит используется для того, чтобы различать фреймы данных и удалённые фреймы. RTR-бит фрейма данных имеет доминирующее значение, в то время как RTR-бит удалённого фрейма – рецессивное. Таким образом, при одновременной передаче фрейма данных и удалённого фрейма с одинаковыми идентификаторами, фрейм данных выиграет арбитраж.

Замещающий бит расширенного формата, SSR-бит, имеет рецессивное значение и служит для той же цели: чтобы при одновременной передаче фреймов данных в стандартном и расширенном форматах с одинаковыми базовыми идентификаторами стандартный формат выигрывал бы арбитраж.

IDE-бит имеет доминирующее значение для стандартного формата и рецессивное – для расширенного. Благодаря этому удалённые фреймы в стандартном формате будут выигрывать арбитраж у фреймов данных и удалённых фреймов в расширенном формате. IDE-бит входит в состав управляющего поля фрейма данных стандартного формата и в поле арбитража фрейма данных расширенного формата.

Управляющее поле в обоих форматах имеют длину 6 битов. Бит r_0 в стандартном формате и биты r_1 и r_0 в расширенном формате зарезервированы для дальнейшего использования и имеют доминирующее значение. Таким образом, первые два бита управляющего поля имеют доминирующие значения как в стандартном, так и в расширенном форматах.

Оставшиеся четыре бита управляющего поля (DLC, Data Length Code – код длины поля данных) определяют, сколько байтов будет передано в поле данных. При этом поле DLC может содержать значения в диапазоне от 0000_2 до 1000_2 , что соответствует длине поля данных от нуля до восьми байтов.

Поле контрольной суммы имеет длину 16 битов. Первые 15 битов содержат младшие 15 битов CRC-кода, рассчитанного для последовательности битов первых четырёх полей фрейма: поля начала фрейма, поля арбитража, управляющего поля, поля данных. Расчёт CRC-кода можно найти, например, в [52]. Последний бит поля контрольной суммы называется бит-разделитель поля контрольной суммы (CRC Delimiter) и имеет рецессивное значение.

Поле подтверждения имеет длину в два бита и состоит из области подтверждения (ACK Slot) и разделителя поля подтверждения (ACK Delimiter). Передатчик передаёт в этом поле два рецессивных бита, а приёмник, который принял сообщение правильно, устанавливает на шине доминирующее значение во время передачи бита ACK Slot, чем подтверждает, что сообщение принято без ошибок.

Поле конца фрейма содержит семь рецессивных битов.

Удалённый фрейм используется для запроса фрейма данных с требуемым идентификатором. Фрейм может быть передан в стандартном или расширенном

форматах. Удалённый фрейм состоит из шести полей: поле начала фрейма, поле арбитража, управляющее поле, поле контрольной суммы, поле подтверждения, поле конца фрейма. Фактически он формируется точно так же как и фрейм данных за исключением следующего:

1) RTR-бит в поле арбитража принимает рецессивное значение, что является признаком удалённого запроса;

2) независимо от значения в поле DLC, поле данных отсутствует (имеет нулевую длину). DLC-поле может содержать любое значение из допустимого диапазона, равное длине поля данных соответствующего идентификатора.

Каждый фрейм данных или удалённый фрейм отделяется от предыдущего фрейма с помощью специального **межфреймового интервала** (Interframe Spacing). Межфреймовый интервал состоит из двух полей: поля паузы (Intermission) и поля «шина свободна» (Bus Idle). Поле паузы состоит из трёх рецессивных битов, за которыми следует поле «шина свободна» произвольной длины, которое также состоит из рецессивных битов.

Межфреймовый интервал используется для того, чтобы обеспечить возможность передачи фрейма перегрузки, который, в случае необходимости, начинает передаваться во время передачи первого или второго бита межфреймового интервала.

Фрейм ошибки состоит из двух полей: поля флагов ошибки (Error Flags), и поля-разделителя фрейма ошибки (Error Delimiter).

Узел обнаруживший ошибку сразу же начинает передавать последовательность из шести доминирующих битов (флаг ошибки, Error Flag). В итоге сигнал на шине нарушает правило добавочного бита, и все станции понимают, что передача фрейма данных или удалённого фрейма была прервана и также начинают передавать флаг ошибки. В результате поле флагов ошибки представляет собой последовательность из доминирующих битов длиной от шести до двенадцати бит.

Каждый узел, передавший флаг ошибки, начинает передавать рецессивные биты. После того, как шина перейдёт в рецессивное состояние (все узлы завершили передачу флага ошибки), каждый узел передаёт ещё семь рецессивных битов. Таким образом, поле-разделитель фрейма ошибки состоит из восьми рецессивных битов.

Фрейм перегрузки состоит из двух полей: поля флага перегрузки (Overload Flag) и поля-разделителя фрейма перегрузки (Overload Delimiter). Необходимость передать фрейм перегрузки может возникнуть в трёх случаях:

1) по «личным» причинам узла-приёмника, которому требуется задержка перед передачей следующего фрейма данных или удалённого фрейма;

2) при появлении доминирующего значения в первом или втором бите поля паузы межфреймового интервала (это означает, что другой узел начал передавать фрейм перегрузки);

3) при появлении доминирующего бита в последнем бите поля-разделителя фрейма ошибки или поля-разделителя фрейма перегрузки.

Между любыми двумя фреймами данных и/или удалёнными фреймами может быть передано не более двух фреймов перегрузки.

Поле флага перегрузки содержит шесть доминирующих битов, благодаря чему, поле паузы разрушается и остальные узлы определяют начало фрейма перегрузки и, в свою очередь, начинают передавать фрейм перегрузки.

Поле-разделитель фрейма перегрузки состоит из восьми рецессивных битов. Механизм формирования поля-разделителя фрейма перегрузки полностью совпадает с механизмом формирования поля-разделителя фрейма ошибки.

А вот как выглядят описанные четыре типа фреймов, по мнению компании KVASER-AB. Вот что могли бы сказать контроллеры, обращаясь в сеть [54]:

- фрейм передачи данных: «Привет всем, вот данные с идентификатором X, получите»;
- удалённый фрейм запроса данных: «Привет всем, а может ли кто-нибудь выслать данные с идентификатором X?»;
- фрейм ошибки: «Стоп, начнем все с начала»;
- фрейм перегрузки: «Я очень занят, подождите чуть-чуть».

Стандарты протоколов уровня приложений на базе CAN

Как уже говорилось, CAN-протокол описывает уровень канала данных и часть физического уровня модели ВОС. Однако этого зачастую недостаточно для эффективной разработки CAN-сетей. Неопределёнными остаются такие важные на этапе разработки моменты, как адресация узлов, распределение между ними CAN-идентификаторов, интерпретация содержимого фрейма данных, передача данных длиной более 8 байт и др. Поэтому с началом массового выпуска CAN-компонентов и широкого распространения CAN-приложений, рядом независимых компаний и некоммерческих ассоциаций в области систем промышленной автоматизации, транспорта и т.д. проводилась (и продолжается по сей день) работа по созданию и стандартизации спецификаций протоколов верхнего уровня (Higher Level Protocol, HLP) для CAN-сетей [55].

Как правило, большинство существующих на сегодня спецификаций протоколов верхнего уровня для CAN-сетей имеют сжатую трехуровневую архитектуру, включающую два базовых уровня CAN-протокола (физический, часто дополненный более конкретными спецификациями, и канальный) и уровень приложе-

ний или прикладной уровень. Сервисы промежуточных уровней либо отсутствуют, либо включены в прикладной уровень. Уменьшенное число уровней против полных семи позволяет обеспечить предсказуемость задержек прохождения сообщений в сети и повысить её производительность в режиме реального времени.

При разработке CAN-приложений на базе стандартных прикладных протоколов разработчик получает в руки уже готовые механизмы передачи данных любой длины, процедуры начальной инициализации, распределения идентификаторов и т.п. Появляется возможность простой интеграции стандартных модулей независимых производителей и наращивание сети в будущем, а также максимально полной реализации основных преимуществ CAN-протокола, особенно при работе в режиме реального времени.

К настоящему времени известно уже более четырех десятков спецификаций протоколов верхнего уровня для CAN [55]. Среди подобного многообразия наибольшее распространение в системах промышленной автоматизации получили четыре, поддерживаемых ассоциацией CiA и рассмотренных ниже. Это CAL/CANopen, CAN Kingdom, DeviceNet и SDS (Smart Distributed System).

CAL/CANopen

Разработка и поддержка открытого протокола прикладного уровня для сетей промышленной автоматизации были одними из приоритетных целей создания организации CiA в 1992 году. Основой такого протокола послужил HLP, разработанный фирмой Philips, после доработки и усовершенствования которого рабочей группой CiA, в 1993 году была опубликована спецификация CAL (CAN Application Level –уровень приложений CAN-протокола) (CiA DS 20x, DS от Draft Standard – проект стандарта, англ.) [56]. Фундаментом CAL-протокола служит канальный уровень CAN. CAL не является ориентированным на конкретные приложения стандартом протокола, не содержит каких-либо профилей, привязанных к конкретным устройствам, и не определяет содержание передаваемых данных, но предлагает стандартизованные элементы сетевого сервиса прикладного уровня. CAL-протокол включает в себя четыре составные части:

- спецификация CAN-сообщений (CAN Message Specification, CMS);
- сетевое управление (Network Management, NMT);
- распределение идентификаторов (Distributor, DBT);
- управление уровнем (Layer Management, LMT).

Спецификация CAN-сообщений (CMS) определяет типы объектов взаимодействия в рамках объектно-ориентированного подхода, правила и механизмы передачи данных разных типов, посредством CAN-фреймов, включая передачу пакетов длиной более 8 байт.

Сетевое управление построено на взаимодействии типа ведущий – ведомые. Один модуль сети является NMT-мастером, все остальные NMT-ведомые. NMT-мастер управляет NMT-ведомыми, желающими принять участие во взаимодействии, и позволяет им общаться между собой посредством CMS-сервисов.

В задачи сетевого управления также входят контроль ошибок и конфигурирования устройств. Благодаря DBT-сервисам происходит бесконфликтное распределение идентификаторов среди модулей под контролем DBT-мастера. Посредством LMT-сервисов возможны запрос и изменение текущих параметров (значений идентификаторов, скорости передачи, битового квантования и т. п.) в модулях непосредственно из CAN-сети.

Сетевые CAN-приложения, основанные на прикладном уровне CAL, в настоящее время успешно работают в медицинской электронике, системах контроля дорожного движения, на транспорте, в промышленном оборудовании.

Протокол **CANopen** может рассматриваться как приложение прикладного уровня CAL-протокола. Первоначально CANopen предназначался для сетей управления движущимися механизмами в системах промышленной автоматизации. Однако впоследствии протокол нашел применение в медицине, морской электронике, на транспорте и в системах автоматизации зданий.

CANopen базируется на двух уровнях стандарта CAN (ISO 11898, Bosch CAN Specification 2.0 A/B). В дополнение к спецификациям физического уровня ISO 11898 (среда передачи данных двухпроводная дифференциальная линия), CANopen содержит собственные правила битового квантования, а также определяет несколько типов используемых соединителей:

- 9-контактный D-Sub разъём (DIN 41652);
- 15-контактный D-Sub разъём;
- разъём RJ-10 (как в телефонах);
- разъём RJ-45 (как в Ethernet-сетях);
- 5-контактный круглый Mini-разъём;
- 5-контактное открытое клеммное соединение и др.

Разводкой контактов для всех типов соединителей предусмотрена возможность подачи питания на трансиверы узлов, имеющих гальваническую развязку. В сети CANopen определены восемь градаций скоростей передачи данных: 1 Мбит/с, 800 кбит/с, 500, 250, 125, 50, 20 и 10 кбит/с. Поддержка скорости 20 кбит/с является обязательной для всех модулей.

Прикладной уровень представляет собой подмножество CAL и основано на четырёх его сервисных элементах CMS, NMT, DBT и LMT, дополненных профилем соединения (CiA DS 301) [57], определяющим базовые правила обмена данными и структуру словаря объектов. Более развитые механизмы сетевого взаимо-

действия для интеллектуальных устройств (человеко-машинные интерфейсы HMI, PC-контроллеры, PLC, инструментальные средства и т. п.) описаны в дополнении к коммуникационному профилю (CiA DS 302).

В сети CANopen на прикладном уровне модули обмениваются между собой объектами-сообщениями (Communication Object, COB), включающими в себя один или более CAN-фреймов. Всего существует четыре типа таких объектов:

- объекты данных процесса (Process Data Objects, PDO);
- объекты сервисных данных (Service Data Objects, SDO);
- объекты специальных функций (Special Function Objects);
- объекты сетевого управления (Network Management Objects).

SDO-объекты позволяют модулям обмениваться данными любого объёма (при последовательностях более 8 байт благодаря использованию нескольких CAN фреймов) в низкоприоритетном режиме. Как правило, этот тип обмена (обязательный к поддержке всеми модулями) используется для конфигурирования устройств или настройки формата PDO-объектов.

Обмен на основе PDO-объектов используется для синхронной или асинхронной скоростной передачи не более 8 байт. Он имеет более высокий приоритет, чем SDO и применяется для пересылок данных в режиме реального времени. Объекты специальных функций служат для выполнения ряда специальных задач: запуска синхронных процессов, передачи значения абсолютного времени и кодов ошибок. Объекты сетевого управления включают сообщения NMT, LMT и DBT сервисов.

Администрированием сети занимается NMT-мастер, который инициализирует устройства, обеспечивает контроль ошибок, а также производит их периодическую «перекличку» с помощью PDO-сообщений для выявления узлов, находящихся в нерабочем состоянии ввиду физического отсутствия или отключения от шины по счётчику ошибок.

Для максимального упрощения процесса интеграции модулей независимых производителей в единую сеть в CANopen используется концепция профилей. Определён целый ряд профилей:

- модули ввода/вывода (аналоговые и цифровые) (CiA Draft Standard DS 401);
- приводы и модули управления перемещением (CiA DS 402);
- элементы человеко-машинного интерфейса (CiA DS 403);
- измерительные устройства и регуляторы (CiA DS 404);
- программируемые контроллеры, совместимые со стандартом МЭК 61131-3 (CiA DS 405);
- энкодеры (CiA DS 406);

- регулирующие клапаны, гидравлические механизмы (CiA DS 408);
- уклономеры (CiA DS 410);
- медицинские устройства (CiA DS 412);
- ткацкие машины (CiA DS 414);
- дорожные машины (CiA DS 415);
- лифты (CiA DS 417);
- поезда (CiA DS 421);

а также другие профили, практически для всех отраслей, в которых могут применяться системы автоматического управления.

Первым профилем приложения стал WD-407 (IBIS-CAN) для электронных систем управления на общественном транспорте Европы (билетный контроль, подсчёт пассажиров и т. п.), где CAN-сети применяются достаточно широко.

CAN Kingdom [54]

Протокол шведской компании KVASER-AB [58] занимает особое место среди спецификаций протоколов верхнего уровня для CAN-сетей не только из-за своего необычного названия (CAN Королевство), но и в значительной степени благодаря оригинальной концепции сетевого взаимодействия и эффективности CAN-приложений на его основе [55]. Началу работ над первой версией протокола CAN Kingdom в 1990 году предшествовал многолетний опыт компании в области создания систем распределённого управления. Протокол был специально разработан для управления движущимися машинами и механизмами промышленными роботами, текстильными станками, мобильными гидравлическими устройствами, и позволяет достичь высокой производительности в режиме реального времени при удовлетворении жёстких требований безопасности. CAN Kingdom является также основой американского военного стандарта CDA 101 и широко используется в военной технике от надувных лодок и систем наведения на цели до сверхзвуковых истребителей и ракет.

Основной целью создания протокола было предоставление системному разработчику максимальной свободы в реализации своих идей при построении сети, сохранив при этом возможность использования стандартных модулей от независимых производителей. CAN Kingdom не является «готовым» протоколом в том смысле, в каком это справедливо, например, по отношению к стандартам типа CANopen или DeviceNet. Это скорее набор примитивов, метапротокол, с помощью которых можно «собрать» протокол под конкретную сеть модулей. Этим достигается уникальное сочетание простоты интеграции готовых модулей с высокой степенью «закрытости» оригинального протокола.

При разработке спецификации CAN Kingdom авторы отказались от принятого в подобных случаях и широко распространённого следования правилам взаимодействия открытых систем. Семиуровневая модель ВОС создавалась изначально для описания традиционных компьютерных сетей, от которых не требуется работа в реальном масштабе времени, и предназначены они для обслуживания пользователей, требования которых априори (на этапе построения такой сети) неизвестны и непредсказуемы. В системах же управления реального времени ситуация прямо противоположная: на стадии разработки все коммуникационные потребности модулей должны быть известны.

Краеугольным камнем концепции сетевого взаимодействия CAN Kingdom является принцип: «узлы обслуживают сеть» в отличие от принципа «сеть обслуживает пользователей», свойственного компьютерным сетям.

Это означает, что каждый узел, подключаемый к сети должен объявить о своём подключении. После этого остальные узлы обеспечат новый узел информацией, необходимой для выполнения его задачи в сети. Каждый узел, подключаемый к сети уже «знает» о своей роли в системе и запрашивает средства, которые ему необходимы.

Представление CAN-сети в терминах CAN Kingdom (в сравнении с традиционным) дано на рис. 4.20. В CAN Kingdom сеть CAN – это страна (королевство) со своей столицей (центральный контролирующий узел) и провинциальными городами (остальные узлы). Король (управляющая программа-супервизор) управляет всем королевством и отвечает за соблюдение закона и порядка в нём, а за местное управление (в пределах своего узла) отвечают мэры городов (управляющие программы узлов). Каждый город экспортирует или импортирует продукцию-информацию посредством почты, которая циркулирует по почтовому тракту (CAN-шина) и проходит через почтмейстеров (CAN-контроллеры). Типы почтовой корреспонденции (информация, передаваемая по сети) и её соответствие CAN-терминам таковы:

Письмо	CAN-фрейм данных или удалённого запроса
Конверт	CAN-идентификатор
Страница	Поле данных CAN-фрейма
Строка	Байт данных
Элемент строки	Бит данных

Неформальный язык описания протокола позволяет любому специалисту, далекому от вычислительной техники или электроники биологу, механику или врачу благодаря интуитивно понятному описанию сети (как должны функционировать общество или страна, примерно представляют себе все) активно участвовать если не в процессе разработки системы, то хотя бы сознательно формулиро-

вать технические условия и иметь представление о принципах её функционирования.

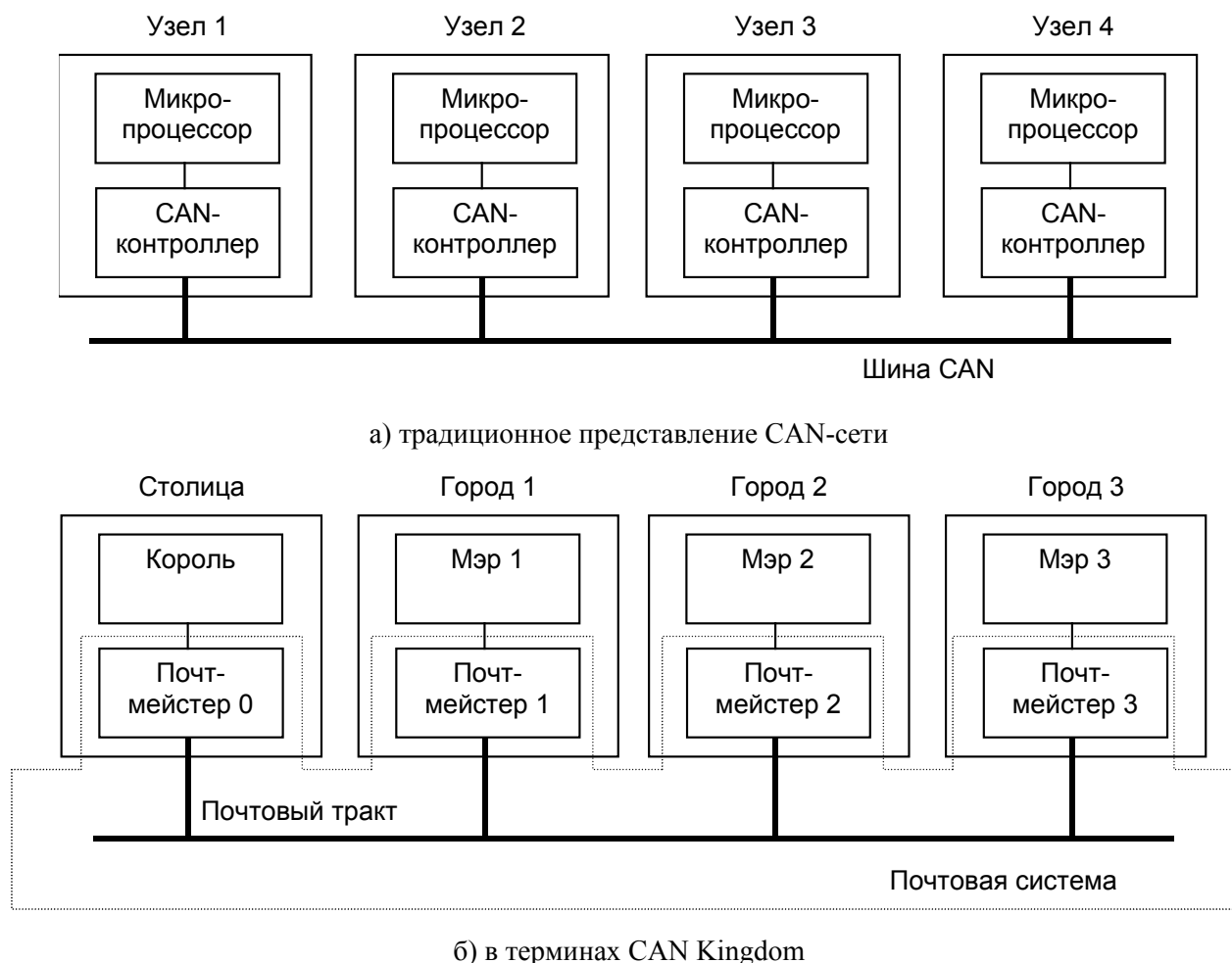


Рис. 4.20. Представление CAN-сети в терминах CAN Kingdom

CAN-система на базе протокола CAN Kingdom обладает следующими особенностями:

- распределение CAN-идентификаторов находится под полным контролем разработчика;
- максимальное время прохождения любого сообщения в сети предсказуемо;
- во время начальной инициализации системы происходит обязательный этап настройки протокола, включая построение форматов данных, начиная с битового уровня, методов управления шиной, распределение идентификаторов и т.д.;
- в системе всегда должен присутствовать (как минимум до завершения настройки протокола) супервизор (король), производящий инициализацию системы, контроль подключенных узлов и т. д. Ни один узел не может принимать участие в сетевом обмене без разрешения Короля.

- перед инициализацией сети каждый узел (город) должен иметь свой номер (CAN Kingdom не описывает конкретный способ установки номера узла, это может быть DIP-переключатель, энергонезависимая память или конфигурация соединителя), «знать» идентификатор сообщения инициализации (королевское письмо) и скорость передачи данных в сети;

- в сеть CAN Kingdom возможна интеграция любых CAN-модулей, включая разработанных для других протоколов, например, DeviceNet или SDS;

- не существует каких-либо рекомендуемых скоростей передачи данных, но за первые 200 мс после подачи питания узел обязан настроиться на прослушивание шины на скорости 125 кбит/с. Допустимы отличающиеся от ISO 11898 спецификации физического уровня.

Наличие одного центра-короля, который содержит всю информацию о системе, избавляет от использования профилей устройств, часто применяемых в других протоколах верхнего уровня CAN-сетей.

Правила идентификации модулей основаны на использовании международного кода EAN/UPC, включающего код производителя и продукта. Среди других особенностей CAN Kingdom можно отметить гибкость режимов передачи и упаковки данных (включая использование поля арбитража для передачи данных), объединение узлов в группы, поддержка часов реального времени, различных режимов доступа к шине.

DeviceNet

DeviceNet (сеть устройств) – протокол, разработанный и опубликованный в 1994 году компанией Allen-Bradley [59] корпорации Rockwell и впоследствии переданный в ведение специально организованной для его поддержки ассоциации ODVA (Open DeviceNet Vendor Association Inc., Открытая ассоциация продавцов DeviceNet-устройств) [60]. DeviceNet недорогое, простое и эффективное решение для объединения разнообразных устройств промышленной автоматизации независимых производителей в единую систему: фото-, термодатчики, стартеры, считыватели штриховых кодов, элементы человеко-машинного интерфейса клавиатуры, дисплейные панели, наряду с управляющими устройствами PLC, компьютерами и т.д. При разработке протокола помимо снижения стоимости также стояла задача упрощения и унификации диагностики подобных устройств. Первые устройства, удовлетворяющие спецификации DeviceNet, появились на рынке в начале 1995 года. DeviceNet также построен на двух нижних уровнях стандарта CAN, дополненных более детальными, чем в других протоколах верхнего уровня для CAN, спецификациями физической среды [55].

Сеть DeviceNet имеет шинную топологию с отводами. Физической средой передачи является 4-проводной кабель (CAN_H, CAN_L, Vcc, Ground), причем возможны две его разновидности: толстый (внешний диаметр 12,2 мм) и тонкий (6,9 мм). Определены лишь три значения скорости передачи данных 125, 250 и 500 кбит/с. Максимальные длины центральной магистрали и отводов в зависимости от скорости передачи и типа кабеля приведены в табл. 4.18.

Таблица 4.18.

Ограничения на протяжённость сети DeviceNet

Скорость передачи, кбит/с	Длина магистрали, м		Длина отводов, м	
	Толстый кабель	Тонкий кабель	Одиночных	Суммарная
125	500	100	6	156
250	250	100	6	78
500	100	100	6	39

Важной особенностью сети DeviceNet является возможность питания модулей непосредственно от сетевого кабеля (24 В, до 8 А на толстом кабеле), а также допускается применение нескольких источников питания в любой точке шины. Всё это даёт возможность построения автономной сети, не зависящей от наличия или качества внешнего питания, а при необходимости позволит легко демонтировать и снова развернуть систему на новом месте. Сеть DeviceNet допускает «горячее» (без обесточивания сети) подключение и отключение узлов.

Стандарт DeviceNet содержит также подробное описание многочисленных типов переходников, разветвителей (одиночных и многопортовых), соединителей (Mini, Micro), сетевых отводов и т.п.

При описании организации типов данных, сетевого поведения узлов в DeviceNet используется объектно-ориентированная модель. Обязательные классы объектов включают в себя следующие:

- объект удостоверения (Identity object). Содержит информацию об устройстве (код производителя, продукта, версия и т. п.);
- объект соединения (Connection object). Логический порт ввода/вывода устройства;
- объект DeviceNet. Включает MAC ID (идентификатор устройства), скорость передачи, состояние устройства и т.п.;
- объект маршрутизатора сообщения (Message router object). Перенаправляет явное сообщение получателю.

Сообщения в сети DeviceNet могут быть двух типов:

- сообщения ввода/вывода (I/O messages) предназначены для целей управления устройствами и передачи данных в реальном времени между узлами в широковещательном или в режиме точка-точка. Используют идентификаторы с высоким приоритетом, которые и определяют содержание сообщения;

- явные сообщения (Explicit messages) для многоцелевого обмена данными в режиме точка-точка. Обеспечивают типичный сервис запрос/ответ. Используют идентификаторы с низким приоритетом и применяются обычно для конфигурирования устройств и целей диагностики. Значение сообщения содержится в поле данных.

При необходимости передачи данных длиной более 8 байт применяется механизм фрагментации. В зависимости от потребностей обмена и возможностей модулей, сеть предлагает три способа организации взаимодействия устройств: ведущий-ведомый (master-slave), мультимастерный (multi-master) или равноправный (peer-to-peer). Пересылки данных могут инициироваться путём опроса, циклически или по изменению их значения. Максимальное число узлов в сети DeviceNet – 64.

В сеть могут быть подключены устройства двух типов: устройства, работающие как независимые менеджеры сообщений, способные устанавливать равноправные соединения с другими модулями (peer-to-peer) и ведомые устройства, требующие меньшей длины кода и производительности управляющего микроконтроллера, которые не могут произвольно выбирать путь соединения, и их объекты соединения конфигурируются при включении устройства.

Для обеспечения взаимодействия устройств разных производителей в рамках единой сети, стандарт DeviceNet, подобно многим протоколам верхнего уровня, определяет ряд профилей устройств. Формированием библиотек профилей занимаются специальные группы (Special Interest Groups) ассоциации ODVA.

SDS (Smart Distributed System)

SDS (Smart Distributed System, «умная» распределённая система) – разработка компании Honeywell Inc. [61]. Наряду со стандартом DeviceNet, SDS представляет собой ещё одно недорогое и законченное решение для сетевого управления интеллектуальными датчиками и исполнительными механизмами от промышленного контроллера в системах промышленной автоматизации [55].

По степени завершённости от спецификаций физической среды до прикладного уровня, ориентировке на снижение стоимости, SDS-стандарт напоминает DeviceNet.

Шинная топология представляет собой линейную шину (магистраль) с короткими отводами. Определены два базовых типа кабельной разводки. При сборке магистрали сети применяется 4-хпроводной кабель с максимальной токовой

нагрузкой 8 А и 5-контактный разъёмы Mini. Для подключения физических устройств к сети используется 4-хпроводной кабель на 3 А, и 4-контактный разъём Micro без отдельного контакта для экрана кабеля.

В сети SDS допускается и обычная проводная разводка с использованием открытых клеммных соединителей. Всеми типами кабельной разводки и соединителей, также как и в сети DeviceNet, предусмотрено подведение питающего напряжения к узлам. Предельные значения длин магистрали и отводов сети SDS в зависимости от скоростей (их четыре) передачи приведены в табл. 4.19.

Таблица 4.19.

Предельные значения длин магистрали и отводов сети SDS

Максимальная длина магистрали, м	Скорость передачи, кбит/с	Максимальная длина отводов, м	Максимальное количество узлов
22,8	1000	0,3	32
91,4	500	0,9	64
182,8	250	1,8	64
457,2	125	3,6	64

Сообщения, циркулирующие в сети SDS, носят название APDU (Application layer Protocol Data Unit, блоки данных протокола прикладного уровня). APDU-сообщение представляет собой CAN-фрейм стандартного формата (расширенный формат фрейма в SDS-сети не применяется), каждый элемент которого имеет своё собственное назначение. APDU-сообщение имеет две формы: укороченную (не используется при передаче данных и содержит в поле DLC нули) и длинную.

При необходимости передачи последовательностей данных более шести байтов используется фрагментированный формат (до 64 фрагментов по четыре байта) длинной формы APDU-сообщения.

Укороченная форма APDU-сообщения используется в следующих сервисах прикладного уровня:

- Change of State (OFF, ON, OFF ACK, ON ACK) – обнаружение изменения состояния логического устройства;
- Write (ON State, OFF State, ON State ACK, OFF State ACK) – управление состояниями логического устройства.

К сервисам, использующим длинную форму APDU, относятся:

- Channel – сервис, обеспечивающий как широковещательный канал соединения (multicast), так и равноправный (peer-to-peer);
- Connection – открытие/закрытие индивидуальных типов соединения;
- Read – чтение атрибутов объектов устройства;
- Write – изменение атрибутов объектов устройства;

- Action – команда объекту устройства выполнить действие;
- Event – сигнализация о событии объектом устройства.

Сеть SDS всегда требует наличия единственного мастера-менеджера сети как минимум на этапе включения для выполнения автонастройки скорости передачи модулей. В процессе работы сети допускается наличие нескольких мастеров на шине, но они должны функционировать в пределах своих адресных доменов, а при включении сети только один из них может брать на себя функцию сетевого менеджера для автонастройки скорости устройств.

PROFIBUS

В 1987 году для немецкой промышленности был разработан и принят стандарт DIN E 19245 PROFIBUS (PROcess Field BUS – полевая шина для управления процессами). В 1996 году этот стандарт стал международной нормой EN 50170 [62].

Говоря о PROFIBUS, необходимо иметь в виду, что под этим общим названием понимается совокупность трёх различных, но совместимых протоколов: PROFIBUS-DP, PROFIBUS-FMS и PROFIBUS-PA.

Архитектура протоколов PROFIBUS базируется на модели взаимодействия открытых систем, в ней реализованы уровни 1,2 и 7.

PROFIBUS-DP

PROFIBUS-DP (Distributed Periphery – распределённая периферия) описывает уровни 1 и 2 модели ВОО, а также пользовательский интерфейс. Уровни с 3 по 7 не используются. Благодаря такой архитектуре достигается быстрая передача данных. Утилита прямого доступа к данным (Direct Data Link Mapper) организует доступ к уровню 2. В основу пользовательского интерфейса положены необходимые пользовательские функции, а также системные и аппаратно-зависимые функции различных типов PROFIBUS-DP-приборов. Этот профиль протокола PROFIBUS оптимизирован для быстрого обмена данными специально для коммуникаций между системами автоматизации и децентрализованной периферией на полевом уровне.

PROFIBUS-FMS

В PROFIBUS-FMS применяются уровни 1,2 и 7. Пользовательский уровень состоит из двух слоёв: FMS (Fieldbus Message Specification – спецификация шинных сообщений) и LLI (Lower Layer Interface – интерфейс с нижним уровнем).

FMS содержит пользовательский протокол и предоставляет в распоряжение коммуникационные службы. LLI реализует различные коммуникационные связи и создаёт для FMS аппаратно-независимый доступ к уровню 2.

PROFIBUS-FMS применяется для обмена данными между контроллерами и компьютерами. FMS-сервисы обеспечивают бóльшую гибкость при передаче больших объёмов данных. Профили PROFIBUS-DP и PROFIBUS-FMS применяют одинаковую технику передачи и единый протокол доступа к шине и поэтому могут работать через общий кабель.

PROFIBUS-PA

Профиль PROFIBUS-PA (Process Automation – автоматизация процессов) применяет расширенный PROFIBUS-DP-протокол передачи данных. Протокол PROFIBUS-PA обеспечивает надёжность передачи данных и питание полевых приборов через шину. Приборы PROFIBUS-PA могут подключаться в сеть PROFIBUS-DP с помощью специальных устройств PROFIBUS-PA-Links.

PROFIBUS-PA – специальная концепция, позволяющая подключать к общей шине датчики и исполнительные механизмы, находящиеся во взрывоопасной зоне.

Физический уровень для DP/FMS на базе RS-485

В качестве основного способа организации сети PROFIBUS-DP/FMS рассматривается объединение устройств с помощью экранированной витой пары. На физическом уровне PROFIBUS используется симметричная передача данных по стандарту EIA/TIA RS-485 (также обозначается H2). По концам шинного сегмента устанавливаются терминаторы (рис. 4.21). В табл. 4.20 приведена максимально допустимая длина провода (длина сегмента) системы PROFIBUS. Эта длина зависит от скорости передачи. Внутри сегмента может быть до 32 узлов.

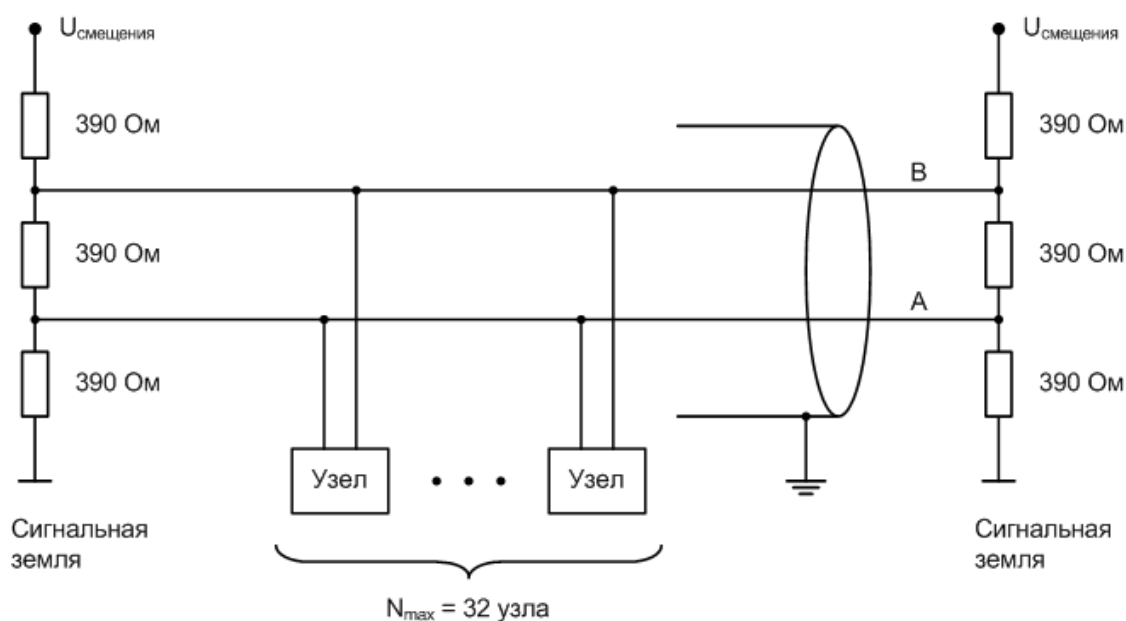


Рис. 4.21. Структура шинного сегмента RS-485

Таблица 4.20.

Максимальная длина сегмента в зависимости от скорости

Скорость передачи, кбит/с	9,6 – 187,5	500	1500	12000
Длина сегмента, м	1000	400	200	100

Для подключения большего количества узлов нужно использовать несколько шинных сегментов, которые соединяются друг с другом через повторители (усилители мощности). Повторитель усиливает уровень передаваемого сигнала. Из-за временных задержек и искажений двоичный сигнал может, согласно EN 50170, проходить максимум три повторителя, которые работают как усилители мощности и включены последовательно. На практике, однако, повторитель-соединитель реализуется как восстановитель сигнала. Поэтому число повторителей, которые можно включить последовательно, зависит от их конструкции. Так, например, можно последовательно включить до девяти повторителей фирмы Siemens типа 6ES7 972-0AA00-0XA0, таким образом, длина сети может быть увеличена в 10 раз, по сравнению с табл. 4.20.

В качестве стандарта для подключения узлов к шине в нормах PROFIBUS EN 50170 рекомендуется 9-и штырьковый разъём D-Sub. На узлах сети устанавливается розеточная часть разъёма, на шинный кабель – вилка разъёма.

Шинные провода данных с обеих сторон замкнуты на согласованные нагрузки – пассивные терминаторы (см. рис. 4.21). Благодаря этим сопротивлениям устанавливается безопасный потенциал покоя на проводах шины, когда участники не обмениваются сообщениями (потенциал покоя между телеграммами). Шинные терминаторы встроены почти во все стандартные разъёмы PROFIBUS и могут быть активизированы с помощью переключателей.

Протокол реализует асинхронную, полудуплексную передачу данных с использованием NRZ-кодирования. Данные передаются побайтно, внутри 11-тиразрядного кадра (старт-бит, 8 битов данных, бит чётности, стоп-бит; старший бит передаётся первым).

Физический уровень для DP/FMS на базе световодов

Помимо использования передачи электрических сигналов, в стандарте PROFIBUS предусмотрено использование оптической передачи – передачи данных с помощью световодов. Световодная техника устойчива к электромагнитным помехам и обеспечивает гальваническую развязку между участниками сети. Благодаря простой технике подключения световодов, специальным пластиковым световодам, эта техника пришла на полевой уровень. Длина линии связи может быть до 15 км при стеклянных световодах и до 80 м при пластиковых.

Для подключения узлов к световоду применяются различные модули:

- модуль-преобразователь интерфейса (OLM, Optical Link Module) похож на повторитель RS-485. Имеет два функционально разделенных электрических канала и выходы для одного или двух оптических каналов. Модули соединяются с отдельными участниками или сегментами шины через интерфейс RS-485 (рис. 4.22);

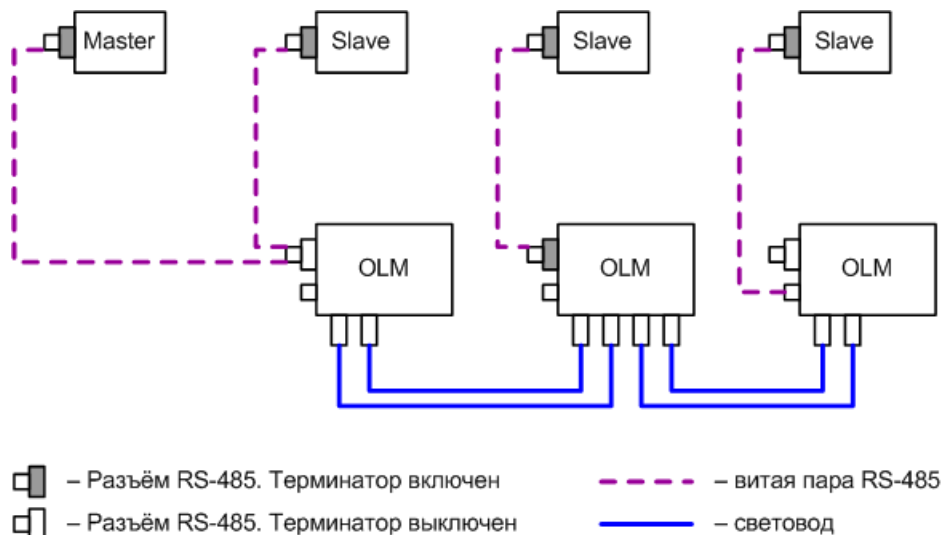


Рис. 4.22. Пример шинной конфигурации с OLM-техникой

- модуль подключения к световоду (OLP, Optic Link Plug). С помощью модулей OLP можно соединять друг с другом оптическим волокном пассивных участников сети – ведомых устройств (Slave). Модули OLP подключаются прямо на девятиштырьковый разъём узла. Питание OLP-модуля осуществляется через разъём узла. Как видно из рис. 4.23, для подключения активных участников шины – ведущих устройств (Master) к OLP-кольцу всегда используется OLM.

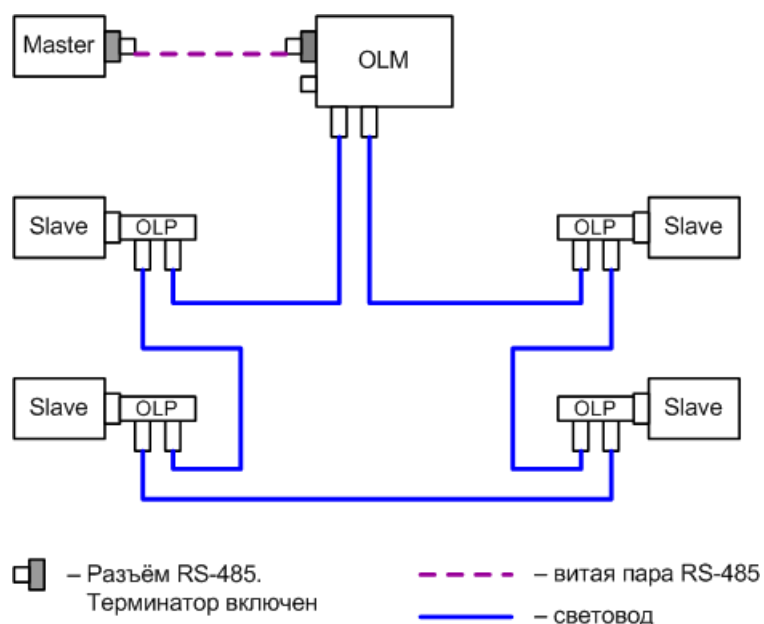


Рис. 4.23. Оптическое кольцо с OLP-техникой

- интегрированное LWL-подключение (Licht Wellen Leiter, световолновод, нем.). Прямое подключение участников PROFIBUS к световоду. Возможно у приборов со встроенным LWL-входом.

Физический уровень для PA

В PROFIBUS-PA используется стандарт передачи МЭК 1158-2, что позволяет использовать сеть PROFIBUS-PA во взрывоопасной зоне. Питание узлов сети осуществляется прямо через шину. Для передачи данных используется бит-синхронизированный с манчестерским кодированием протокол передачи (обозначается также как H1). При передаче данных с помощью манчестерского кода логический ноль передаётся как смена фронта с «0» на «1», а логическая единица – как смена фронта с «1» на «0». Данные передаются с помощью модуляции ± 9 мА тока шины.

Скорость передачи составляет 31,25 кбит/с. В качестве среды передачи используется витой экранированный или неэкранированный провод. Шина (рис. 4.24), состоит из сегментов, к которым подключены узлы, сегменты замкнуты на RC-цепочки. К сегменту шины PA может быть подключено максимум 32 участника. Максимальная длина сегмента сильно зависит от применяемого источника питания, типа провода и потребления тока подключенными участниками.

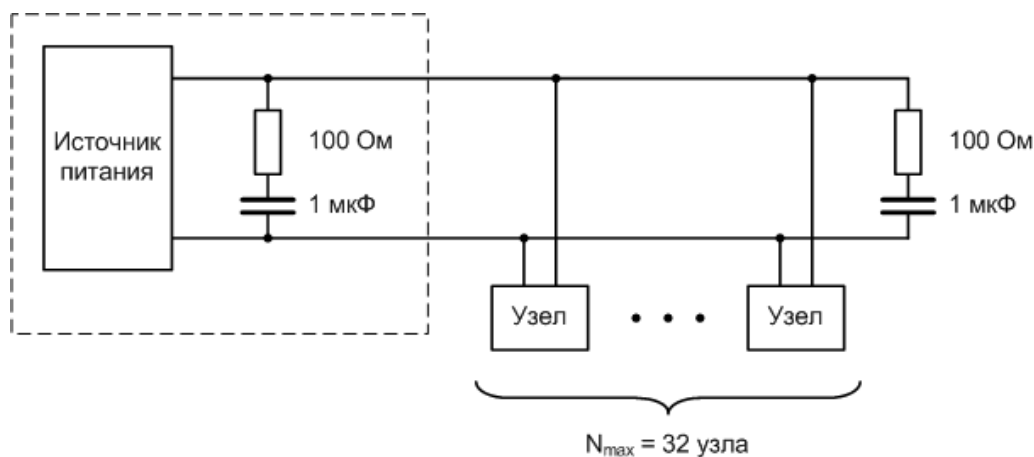


Рис. 4.24. Структура шинного сегмента PA

В качестве среды передачи для PROFIBUS-PA применяется двухжильный кабель, технические данные которого не установлены/не нормированы. Использование рекомендуемого в DIN 61158-2 типа кабеля (тип A) позволяет построить сеть с максимальной длиной 1900 м.

Уровень канала данных для PROFIBUS

Согласно модели BOC на втором уровне реализуется управление доступом к шине, обеспечивается надёжность передачи данных, выполняется формирование

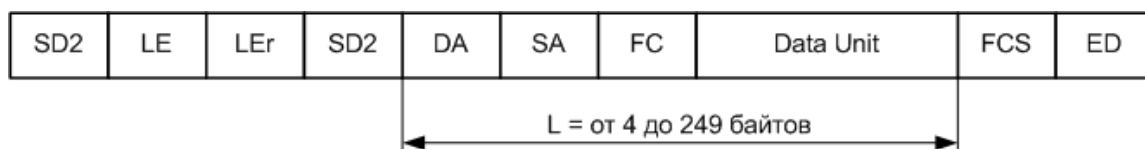
сообщений. Канальный уровень обозначается в PROFIBUS как FDL-уровень (Fieldbus Data Link). На канальном уровне PROFIBUS используется несколько типов сообщений. Форматы сообщений представлены на рис. 4.25. Названия полей сообщений и их назначение описаны в табл. 4.0.



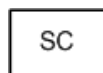
а) Формат сообщения с постоянной длиной информационного поля



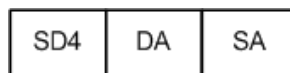
б) Формат сообщения с постоянной длиной информационного блока с данными



в) Формат сообщения с переменной длиной информационного блока



г) Короткое квитирование



д) Сообщение-маркер

Рис. 4.25. Формат PROFIBUS-сообщений

За счёт использования битов чётности и контрольного байта, при передаче может быть распознано до трёх ошибочных битов в одном сообщении. Сообщение, в котором обнаружена ошибка, повторяется, по крайней мере, один раз. Имеется возможность повторять сообщения, проходящие по уровню канала данных до восьми раз (шинный параметр «Retry»).

На канальном уровне PROFIBUS реализуются передачи типа «точка – точка», а также широковещательные передачи (всем участникам сети – Broadcast или группе участников сети – Multicast). В случае широковещательной передачи приём данных не квитируется. Службы, реализованные на канальном уровне, приведены в табл. 4.22. В протоколах DP и PA применяется, соответственно, подмножество служб канального уровня. Так, например, PROFIBUS-DP использует только службы SRD и SDN.

Таблица 4.0.

Поля PROFIBUS-сообщений

Имя поля	Назначение
SC (Single Character) Одиночный символ	Используется только для квитирования (SC=E5h)
SD1–SD4 (Start Delimiter) Стартовый байт	Признак начала сообщения. Служит для отличия различных форматов сообщений (SD1=10h, SD2=68h, SD3=A2h, SD4=DCh)
LE/LEr (LEngth) Байты длины	Указывают длину информационных полей сообщений с переменной длиной
DA (Destination Address) Байт адреса приёмника	Содержит адрес приёмника сообщения
SA (Source Address) Байт адреса источника	Содержит адрес источника сообщения
FC (Frame Control) Байт управления сообщением	Определяет, какая служба канального уровня должна обрабатывать сообщение и приоритет сообщения
Data Unit Поле данных	Содержит передаваемые данные
FCS (Frame Check Sequence) Контрольный байт	Содержит контрольную сумму сообщения, которая образуется путём побайтного сложения без учёта бита переполнения
ED (End Delimiter) Оконечный байт	Признак конца сообщения (ED=16h)

Таблица 4.22.

Службы передачи PROFIBUS

Служба	Функции	DP	PA	FMS
SDA	Посылка данных с квитированием			x
SRD	Данные посылаются и принимаются с квитированием	x	x	x
SDN	Данные посылаются без квитирования	x	x	x
CSRD	Циклическая посылка и приём с квитированием			x

Службы вызываются через точки доступа к службе (SAP, Service Access Point) канального уровня из вышестоящего уровня. В PROFIBUS-FMS эти точки доступа используются для адресации логических коммуникационных связей. В протоколах DP и PA применяемые точки доступа строго упорядочены. У всех активных и пассивных участников можно использовать параллельно несколько точек доступа. Различаются точки доступа источника (SSAP, Source Service Access Point) и точки доступа приёмника (DSAP, Destination Service Access Point).

Прикладной уровень FMS

Прикладной уровень определён только для PROFIBUS-FMS. С помощью введения специальных профилей для FMS-устройств, в качестве надстройки к прикладному уровню, обеспечивается возможность единообразного взаимодействия с устройствами разных производителей. В профилях устройств, подключаемых к PROFIBUS-FMS, определён обязательный набор функций, требующийся для конкретного типа устройств, благодаря чему устройства различных производителей имеют одни и те же коммуникационные функции. Для FMS определены следующие профили:

- коммуникации между контроллерами (3.002). Этот коммуникационный профиль устанавливает, какие FMS-службы применяются для коммуникаций между PLC. В профиле определены службы, параметры и типы данных, которые должен поддерживать каждый ПЛК;
- профиль для автоматизации зданий (3.011). Это отраслевой профиль и основа для многих открытых стандартов в автоматизации зданий. Он описывает, как осуществляется обмен, управление, регулирование, обслуживание, обработка и архивирование сигналов (Alarm) в системах автоматизации зданий через FMS;
- коммутационные низковольтные приборы (3.032). Это отраслевой пользовательский FMS-профиль. Он определяет правила обмена данными с низковольтными коммутационными приборами через FMS.

Пользовательский интерфейс DP и DP-профили

Для PROFIBUS-DP определён пользовательский интерфейс, который позиционируется как надстройка над моделью BOC. Протокол PROFIBUS-DP определяет, как передаются данные по шине между участниками. При этом смысл передаваемых по протоколу пользовательских данных протоколом не определяется (поскольку нет прикладного уровня!). В PROFIBUS-DP также как и в PROFIBUS-FMS определён ряд профилей, благодаря чему приборы разных производителей могут обмениваться информацией. В настоящее время установлены следующие профили PROFIBUS-DP:

- профиль для управления роботами через PROFIBUS-DP (3.052). Профиль позволяет описать движение и программное управление роботом;
- профиль для подключения энкодера (преобразователя угол-код) (3.062). Для энкодеров определены два аппаратных класса основных и дополнительных функций, как, например, масштабирование сигналов и расширенная диагностика;
- профиль для приводов с регулируемой скоростью – PROFIDRIVE-профиль (3.072). Профиль устанавливает, как приводы параметрируются и передают за-

данные и истинные значения, благодаря этому становится возможным обмен данными с приводами различных производителей;

- профиль для управления и наблюдения, HMI (Human Machine Interface) (3.082). Этот профиль используется при подключении к сети устройств для реализации человеко-машинного интерфейса (операторские панели, персональные компьютеры). Профиль устанавливает правила подключения этих устройств через PROFIBUS-DP к компонентам автоматизации.

Управление доступом к шине в PROFIBUS

К управлению доступом к шине PROFIBUS предъявляются два существенных требования. С одной стороны для надёжных коммуникаций между равноправными ПЛК и/или ПК необходимо, чтобы каждый участник в течение определённого временного окна получал доступ к шине для решения своих коммуникационных задач.

С другой стороны для обмена данными между ПЛК или ПК и простой децентрализованной периферией требуется быстрый обмен данными с возможно малыми издержками протокола. Это достигается благодаря использованию гибридного управления доступом к шине, состоящим из децентрализованного обмена маркером между активными участниками (Master-ами) и централизованного обмена Master-Slave для обмена данными между активными и пассивными участниками шины PROFIBUS.

Активный участник, который владеет маркером, берёт на себя функции мастера на шине, чтобы проводить коммуникации с пассивными и активными участниками.

Обмен сообщениями по шине происходит при этом через адресацию участников. Каждому PROFIBUS-участнику назначается однозначный адрес. Адрес назначается из области от 0 до 126. При этом максимальное число участников, находящихся на шине, не превышает 127.

Использование гибридного управления доступом к среде передачи позволяет реализовать следующие конфигурации системы:

- «чистая» система Multimaster (обмен маркером);
- «чистая» система Master-Slave;
- комбинация обоих методов.

Метод доступа к PROFIBUS не зависит от используемой среды передачи.

Активные участники, подключенные к PROFIBUS, образуют упорядоченное по возрастанию их адресов логическое маркерное кольцо (Token Ring) (рис. 4.26) по которому циркулирует маркер (token). Маркер, а с ним и право на доступ к среде передачи, передаётся в специальном маркер-сообщении между активными

участниками. Активный участник с наивысшим на шине адресом HSA (Highest Station Address) передаёт маркер активному участнику с наименьшим шинным адресом, чтобы замкнуть маркерное кольцо. Время прохождения маркера через всех активных участников называется временем обращения маркера и ограничивается с помощью параметра TTR (Time Target Rotation), который задаётся при конфигурации сети.

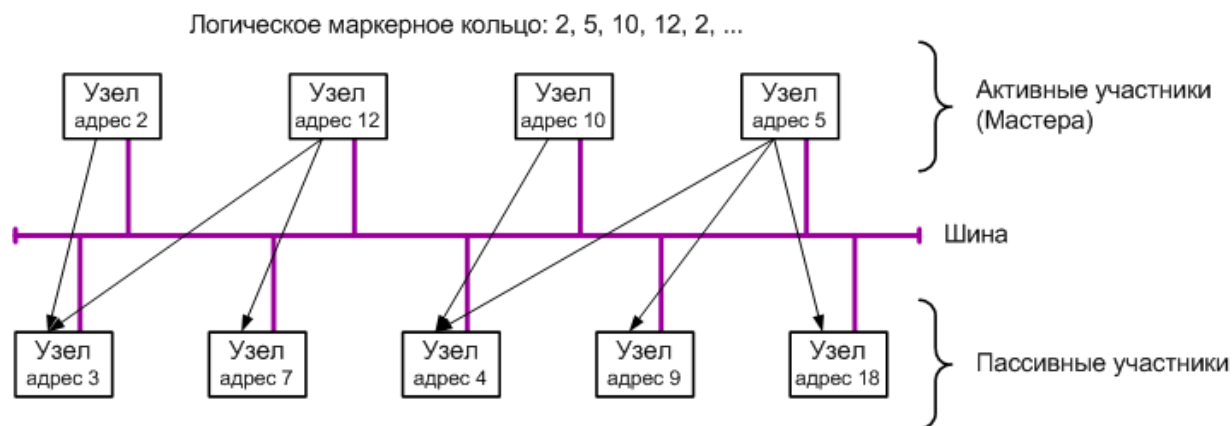


Рис. 4.26. Метод обмена маркером

Управление доступом к шине активных участников осуществляется как на этапе инициализации, так во время функционирования маркерного кольца. При этом устанавливаются адреса всех имеющихся на шине активных участников и заносятся в список активных станций (LAS, List of Active Station). Для управления маркером необходимо знать адреса предыдущей станции, от которой маркер присылается, и следующей станции, которой маркер предназначается. Кроме того, список LAS нужен для того, чтобы при текущей работе исключать из кольца вышедших из строя или дефектных активных участников и, соответственно, принимать вновь появившихся участников без помех текущему обмену данными по шине.

Если логическое маркерное кольцо состоит только из одного активного и нескольких пассивных участников, то это соответствует «чистой» системе Master-Slave. Метод Master-Slave позволяет мастеру (активному участнику), опрашивать назначенных ему Slave (пассивных участников).

4.3. ЗАКЛЮЧЕНИЕ

Существующее в настоящее время разнообразие стандартов промышленных сетей может вызвать желание подождать, пока какой-нибудь один стандарт не окажется, наконец, победителем. К сожалению, ждать придётся до бесконечности, поскольку единого стандарта сети, способного удовлетворить всех, не будет ни-

когда. Ни одна из существующих на сегодняшний день промышленных сетей не может стать доминирующей, поскольку ни одна из них не в состоянии решить всех проблем. С не меньшим успехом можно ожидать, к примеру, наступления тех светлых времен, когда все автомобильные двигатели станут одной и той же мощности, или когда все двери будут иметь одни и те же ширину и высоту.

Можно констатировать, что [31]:

- большинство задач распределённых систем автоматизации под силу практически любому протоколу соответствующего уровня;
- все серьёзные протоколы имеют за плечами поддержку в виде сильных компаний и международных ассоциаций пользователей;
- все они имеют хорошо проработанную аппаратную и программную поддержку;
- заметна тенденция организации шлюзов из одного протокола в другой, например: ASI – PROFIBUS или Interbus-S – PROFIBUS и т.д.

При выборе промышленной сети для построения конкретной системы управления следует сузить выбор до нескольких приемлемых вариантов, после чего сравнить характеристики устройств и изучить сведения о поставщиках, например, по следующим критериям [31]:

- информативность – насколько доступны спецификации протоколов и стандарты, на которые опираются эти протоколы;
- открытость – прежде всего, отсутствие лицензионной платы за использование протокола;
- перспективность – насколько тот или иной протокол представляет собой растущий организм и как он приспособливается под нужды потребителей. Сегодня заметна тенденция в динамическом развитии семейства протоколов PROFIBUS. К тому же на его основе готовится проект так называемого европейского стандарта промышленной сети, который имеет кодовое название EN50170;
- реальная поддержка в стране – это, пожалуй, самый важный компонент успеха любого предложения. Есть ли пользователи, а лучше Ассоциация пользователей, у которых можно получить исчерпывающие ответы на вопросы;
- примеры реализации. Всегда хочется посмотреть своими глазами на то, о чём много говорят. Кроме того, это важный аргумент при принятии решений.

Вопросы для самоконтроля

1. Какие требования предъявляют к промышленной сети?
2. На каких уровнях типовой структуры АСУ ТП применяются промышленные сети?

3. Какие характеристики промышленной сети позволяют судить о соответствии сети требованиям задач реального времени?
4. К какому уровню следует отнести протоколы HART и ASI? Modbus Plus, PROFIBUS FMS? CAN и PROFIBUS DP?
5. Сколько ведущих и ведомых устройств может быть в сети на базе HART-протокола? AS-интерфейса? Modbus? в сегменте сети Modbus Plus? PROFIBUS DP?
6. Каков максимальный объём данных передаваемых в сообщении протокола HART? ASI? Modbus? CAN? PROFIBUS?
7. Какие протоколы реализуют централизованное, децентрализованное, гибридное управление доступом к среде передачи?
8. Какие протоколы поддерживают широковещательные передачи?
9. Физический интерфейс какого протокола основан на методе передачи данных с помощью частотной модуляции?
10. Какой протокол изначально был ориентирован на объединение в сеть дискретных датчиков и исполнительных устройств?
11. Для какого протокола не определён специальный физический интерфейс?
12. Какой протокол использует поле арбитража для определения приоритета передаваемого сообщения?
13. В каком протоколе в состав маркерного кадра входит глобальная база данных для передачи широковещательных сообщений?
14. Какие классы команд описываются HART-протоколом?
15. Каким образом можно подключить к сети на базе AS-интерфейса дискретный датчик или исполнительный механизм?
16. Какую структуру памяти имеет любое устройство, подключенное к сети с точки зрения Modbus-протокола?
17. Как реализуется маршрутизация сообщений в протоколе Modbus Plus?
18. В какой последовательности осуществляется обход логического кольца маркером в сети Modbus Plus? PROFIBUS?
19. В чём состоит отличие форматов сообщений в сети Modbus и Modbus/TCP?
20. Какое состояние среды передачи в протоколе CAN является доминирующим? Для чего вообще в CAN-протоколе определены доминирующий и рецессивный уровни?
21. Какие типы сообщений определены в CAN-протоколе?
22. Перечислите стандарты протоколов уровня приложений на базе CAN.
23. Что может использоваться в качестве среды передачи для протокола PROFIBUS-DP/FMS?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Олссон Г., Пиани Дж. Цифровые системы автоматизации и управления .– СПб.: Невский Диалект, 2001 .– 557 с.
2. Изерман Р. Цифровые системы управления .– М.: Мир, 1984 .– 541 с.
3. Бесекерский В.А. Цифровые автоматические системы .– М.: Наука, 1976 .– 576 с.
4. Иванов В.А., Ющенко А.С. Теория дискретных систем автоматического управления .– М.: Наука, 1983 .– 336 с.
5. Болтон У. Карманный справочник инженера-метролога .– М.: Издательский дом «Додэка-XXI», 2002 .– 384 с.
6. Келим Ю.М. Типовые элементы систем автоматического управления .– М.: Инфра, 2004 .– 384 с.
7. Бессонов Л.А. Теоретические основы электротехники. Изд. 6-е, перераб. и доп. Учебник для студентов энергетических и электротехнических вузов .– М., «Выш. школа», 1973.– 752 с.
8. Денисенко В., Халывко А. Защита от помех датчиков и соединительных проводов систем промышленной автоматизации//Современные технологии автоматизации .– 2001 .– №1. С. 68 – 75. (http://www.cta.ru/pdf/2001-1/notebook1_2001_01.pdf)
9. Николайчук О.И. Системы малой автоматизации .– М.: СОЛОН-Пресс, 2003 .– 256 с.
10. Гук М. Конструктивы и шины компьютеров промышленного назначения (<http://www.neva.ru/mgook/art/mg-inbus.htm>)
11. Бердичевский М. Конструктивы Евромеханики во встраиваемых системах// Современные технологии автоматизации .– 2002 .– №4, С. 52 – 59. (http://www.cta.ru/pdf/2002-4/hardware2_2002_4.pdf)
12. Бердичевский М. Стандартные типоразмеры корпусов электронного оборудования// Современные технологии автоматизации .– 1997 .– №1, С. 96 – 106. (http://www.cta.ru/pdf/1997-1/note1_1997_1.pdf)
13. VITA Open Standards, Open Markets (<http://www.vita.com>)
14. Цифровая и вычислительная техника: Учебное пособие для ВУЗов / Э.В. Евреинов, Ю.Т. Бутыльский, И.А. Мамзелев и др.; Под ред. Э.В. Евреинова .– М.: Радио и связь, 1991 .– 464 с.
15. PICMG Open Modular Computing Specifications (<http://www.picmg.org>)
16. Гук М. Интерфейсы ПК. Справочник .– СПб.: Питер, 1999 .– 416 с.
17. Организация ЭВМ. 5-е изд. / К. Хамахер, З. Вранешич, С. Заки .– СПб: Питер; Киев: Издательская группа BHV, 2003 .– 848 с.
18. Брокарев А.Ж., Петров И.В. Программируемые логические контролеры, МЭК системы программирования и CoDeSys// Автоматизация и производство .– 2006 .– №1
19. PLCopen for efficiency in automation (<http://www.plcopen.org>)
20. ГОСТ Р 51840–2001 (МЭК 61131-1–92). Программируемые контроллеры. Общие положения и функциональные характеристики .– Введ. 2001–12–24 –М.: ИПК Изд-во стандартов, 2002
21. Крылов Е.В., Типикин Н.Г. Принципы функционирования ЭВМ. Учеб. пособие по курсам «Технология программирования» и «Операционные системы» .– Обнинск: ИАТЭ, 1996 .– 77 с.
22. SIEMENS. SIMATIC. Программирование с помощью STEP 7 V5.3. Руководство .– Редакция 01/2004 .– М.: Siemens AG, 2004 .– 602 с. (http://www.automation-drives.ru/as/download/doc/software/base/00_Contents_r.pdf)

23. SIEMENS. SIMATIC. Компоненты комплексных систем автоматизации. Каталог ST70-2005 .– М.: Siemens AG, 2005 .– 811 с.
24. Автоматизация и управление. Платформа автоматизации Modicon Quantum. Unity, Concept & ProWORX 32. Каталог. Июль 2004 .– М.: Schneider Electric, 2004.
25. Бердичевский М. Корпуса, шкафы, конструктивы: степени защиты// Современные технологии автоматизации .– 1996 .– №1. С. 121 – 124.
(http://www.cta.ru/pdf/1996-1/note3_1996_1.pdf)
26. Шелестов И. Надёжность автоматики: проблемы и решения (часть 1)// Автоматизация и производство .– 2007 .– №1, С. 30 – 31.
(<http://www.owen.ru/documents/aip/files/aip0107s30-31.pdf>)
27. Шелестов И. Надёжность автоматики: проблемы и решения (часть 2)// Автоматизация и производство .– 2007 .– №2, С. 32 – 34. (<http://213.234.195.21/pub/AiP/30/aip0207-s32-34.pdf>)
28. Quantum Series Automation. Технические средства. Справочное руководство. 840 USE 100 00 // Компакт-диск «Schneider Electric. Средства автоматизации (версия 5.71)»
(file:///C:/CD_ROM/Info/TSX_Quantum/User_Man/840_USE_100_00_Rus.pdf)
29. Industrial PC to you. Карточка продукта. I-8024. 4-канальный модуль аналогового вывода.
(<http://www.ipc2u.ru/catalog/Q/Q8/11932.html>)
30. SIEMENS. SIMATIC. Описание модулей. Руководство .– М.: Siemens AG, 2002.
31. Гусев С. Краткий экскурс в историю промышленных сетей // Современные технологии автоматизации .– 2000 .– №4, С. 78 – 84. (http://www.cta.ru/pdf/2000-4/note1_2000_4.pdf)
32. Филинов Е.Н. История автоматизированных систем управления предприятиями (АСУП)
(<http://www.computer-museum.ru/histussr/asuphist.htm>)
33. Любашин А.Н. Первое знакомство: краткий обзор промышленных сетей по материалам конференции FieldComms 95 (<http://www.mka.ru/?p=40453>)
34. Модель OSI. (<http://www.citforum.ru/nets/switche/osi.shtml>)
35. Обрежьте жирок с RS-485. (<http://www.gaw.ru/html.cgi/txt/interface/rs485/power.htm>)
36. Дополнительная информация для разработчиков. (<http://www.ultronic.ru/rs485r2a.htm>)
37. RS-422 and RS-485 Application Note.
(<http://www.bb-elec.com/bb-elec/literature/tech/485appnote.pdf>)
38. Локотков А. Интерфейсы последовательной передачи данных. Стандарты EIA RS-422A/RS-485 // Современные технологии автоматизации .– 1997 .– №3, С. 110 – 119.
(http://www.cta.ru/pdf/1997-3/note1_1997_3.pdf)
39. Половинкин В. HART-протокол // Современные технологии автоматизации .– 2002 .– №1, С. 6 – 14. (<http://www.cta.ru/pdf/2002-1/06-15.pdf>)
40. The HART book. International guide to HART-based products & services.
(<http://www.thehartbook.com/technical.htm>)
41. HART. Field communications protocol. Application guide.
(<http://www.hartcomm2.org/download/appguide.pdf>)
42. AS-Interface. (<http://www.as-interface.net/>)
43. Половинкин В. Основные понятия и базовые компоненты AS-интерфейса // Современные технологии автоматизации .– 2002 .– №4, С. 18 – 29.
(http://www.cta.ru/pdf/2002-4/promnet2_2002_4.pdf)
44. Addendum for AS-Interface Compendium
(<http://www.as-interface.net/Download/System/Publications/Chapter-324-Addendum-EN.pdf>)
45. Modbus Protocol. (<http://www.modicon.com/techpubs/toc7.html>)

46. Modbus. Описание протокола (рус.) // Компакт-диск «Schneider Electric. Средства автоматизации (версия 5.71)» (file:///<CD-ROM>/Info/Networks/MODBUS/Modbus_Rus.Doc)
47. Modbus application protocol specification. V1.1b
(http://www.modbus-ida.org/docs/Modbus_Application_Protocol_V1_1b.pdf)
48. Руководство по проектированию и монтажу сети Modicon Modbus Plus 890 USE 100 00 Версия 3.0 // Компакт-диск «Schneider Electric. Средства автоматизации (версия 5.71)» (file:///<CD-ROM>/Info/Networks/Modbus_Plus/User_Man/890_Use_100_00_Rus.pdf)
49. Schwab C. Промышленные сети. Industrial Ethernet. Взгляд сверху на Industrial Ethernet // The Industrial Ethernet book. Issue 23 .– пер. с англ. AKOM .– 2005.
(<http://www.industrialnets.ru/index/?node=625>)
50. Modbus-IDA the architecture for distributed automation (<http://www.Modbus-IDA.org>)
51. CiA (<http://www.can-cia.org>)
52. CAN Specification 2.0, Part B (<http://www.can-cia.org/downloads/specifications/?269>)
53. Карпенко Е. Возможности CAN протокола // Современные технологии автоматизации .– 1998 .– №4, С. 16 – 20 (http://www.cta.ru/pdf/1998-4/promnet1_1998_4.pdf)
54. Lars-Berno Fredriksson A CAN Kingdom Rev. 3.01
(<http://www.kvaser.com/can/info/files/ck301p.pdf>)
55. Щербаков А. Протоколы прикладного уровня CAN-сетей // Современные технологии автоматизации.– 1999 .– №3, С. 6 – 15 (http://www.cta.ru/pdf/1999-3/promnet1_1999_3.pdf)
56. CiA DS 201 to 207 Version 1.1. CAN Application Layer for industrial applications
(<http://www.can-cia.org/downloads/ciaspecifications/?1169>)
57. CANOpen Application Layer and Communication Profile. CiA Draft Standard 301. Version 4.02
(<http://www.can-cia.org/downloads/ciaspecifications/?1390>)
58. KVASER. Advanced CAN Solutions (<http://www.kvaser.com>)
59. Rockwell Automation. Allen Bradley (<http://www.ab.com>)
60. ODVA ((<http://www.odva.org>))
61. Honeywell (<http://www.honeywell.com/sites/honeywell/>)
62. SIEMENS. Сети. SIMATIC NET. PROFIBUS-DP. Руководство. (http://www.automation-drives.ru/as/download/doc/simatic_net/profibus/01_Basic_PROFIBUS_r.pdf)

ОГЛАВЛЕНИЕ

Предисловие.....	3
1. Принципы построения систем цифрового управления технологическими процессами	5
1.1. Роль вычислительной техники в управлении технологическими процессами	5
1.2. Особенности цифрового управления	7
1.3. Типовая структура системы управления	9
1.4. Компоненты интерфейса между процессом и управляющим компьютером	11
Характеристики аналоговых датчиков	12
1.5. Согласование и передача сигналов	16
Согласование сигналов.....	16
Электрические проводники.....	18
Электрические помехи	19
Выбор носителя сигнала: напряжение или ток.....	24
Вопросы для самоконтроля.....	26
2. Аппаратная организация управляющих ЭВМ.....	28
2.1. Основные требования к аппаратной организации управляющих ЭВМ	28
2.2. Программируемые логические контроллеры.....	29
2.3. Конструктив «Евромеханика»	31
2.4. Открытые стандарты системных магистралей.....	35
Стандарт VME	35
Стандарт CompactPCI	37
2.5. Основные принципы организации работы ПЛК.....	39
Разработка программного обеспечения ПЛК.....	39
Принципы функционирования ПЛК	42
2.6. Основные характеристики модулей центральных процессоров	50
2.7. Принципы функционирования и основные характеристики модулей ввода-вывода ПЛК.....	52
Модуль дискретного ввода	52
Модуль дискретного вывода.....	54
Модули счёта импульсов.....	58
Модуль аналогового ввода.....	60
Модуль аналогового вывода	64
Вопросы для самоконтроля.....	67

3. Общие принципы построения промышленных сетей	69
3.1. Предпосылки применения промышленных сетей	69
3.2. Стандартизация в промышленности	70
3.3. Основные сетевые топологии	72
Управление доступом к среде передачи	74
3.4. Модель взаимодействия открытых систем	76
3.5. Уровень приложений в промышленных сетях	80
3.6. Протоколы канального уровня	81
Последовательная передача данных	81
Кодирование бит	83
Синхронизация	86
Асинхронная передача данных	87
3.7. Стандарты электрических интерфейсов применяемых в промышленных сетях	88
Интерфейс RS-232C (EIA-232-D)	89
Интерфейсы RS-422A и RS-485 (EIA/TIA-422A и EIA/TIA-485)	93
Интерфейс «токовая петля»	97
Вопросы для самоконтроля	98
4. Современные стандарты промышленных сетей	99
Области применения промышленных сетей	100
4.1. Протоколы уровня датчиков	101
HART-протокол	101
AS-interface	110
4.2. Протоколы системного уровня	119
Modbus	119
Modbus Plus	126
Modbus/TCP	131
CAN	134
Стандарты протоколов уровня приложений на базе CAN	141
PROFIBUS	152
4.3. Заключение	161
Вопросы для самоконтроля	162
Библиографический список	164