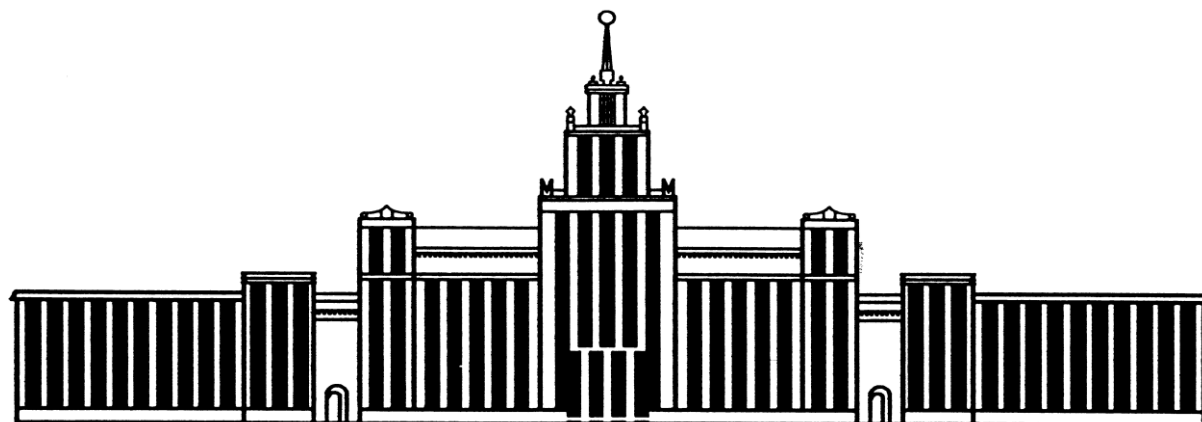

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ



ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

681.5(07)
Б246

Т.А. Барбасова, Е.А. Канашев

ПРОМЫШЛЕННЫЕ СЕТИ И СИСТЕМЫ СВЯЗИ

Учебное пособие

Челябинск
2020

Министерство науки и высшего образования Российской Федерации
Южно-Уральский государственный университет
Кафедра «Автоматика и управление»

681.5(07)
Б246

Т.А. Барбасова, Е.А. Канашев

ПРОМЫШЛЕННЫЕ СЕТИ И СИСТЕМЫ СВЯЗИ

Учебное пособие

Челябинск
Издательский центр ЮУрГУ
2020

УДК 681.51.01(075.8)
Б246

*Одобрено
учебно-методической комиссией
высшей школы электроники и компьютерных наук*

*Рецензенты:
А.Е. Гудилин, О.С. Логунова*

Барбасова, Т.А.

Б246 Промышленные сети и системы связи: учебное пособие / Т.А. Барбасова, Е.А. Канашев. – Челябинск: Издательский центр ЮУрГУ, 2020. – 144 с.

В пособии рассматриваются теоретические основы промышленных сетей, описываются наиболее распространенные промышленные сети и способы связи, применяющиеся для построения распределенных систем управления. Значительное внимание уделяется вопросам практического использования промышленных сетей, особенностям их проектирования, настройки и эксплуатации.

Учебное пособие предназначено для студентов дневной и заочной форм обучения направления 27.03.04 – «Управление в технических системах» при изучении дисциплины «Промышленные сети и системы связи», а также может быть полезно студентам, магистрантам, аспирантам преподавателям и научным работникам, занимающимся вопросами цифровой передачи данных в системах управления.

УДК 681.51.01(075.8)

© Издательский центр ЮУрГУ, 2020

ОГЛАВЛЕНИЕ

Введение.....	5
1. Сетевые технологии в управлении техническими системами	
1.1. Автоматизированные системы управления.....	6
1.2. Концепция комплексной автоматизации.....	6
1.3. Вычислительные сети в структуре АСУ ТП.....	9
Контрольные вопросы	10
2. Основы промышленных сетей	
2.1. Базовые термины и определения.....	10
2.2. Назначение и классификация промышленных сетей.....	11
2.3. Общие требования к промышленным сетям.....	13
2.4. Модель взаимодействия открытых систем	14
2.5. Топология промышленных сетей.....	18
2.6. Взаимодействие приложений	21
2.7. Методы управления доступом к среде передачи.....	23
Контрольные вопросы	28
3. Физический уровень реализации сетей	
3.1. Интерфейс физического уровня	28
3.2. Принципы обмена данными.....	29
3.3. Характеристики процесса передачи данных.....	33
3.4. Проводные линии связи	36
Контрольные вопросы	39
4. Последовательная передача данных	
4.1. Синхронная и асинхронная передача.....	39
4.2. Обмен данными в асинхронной системе	41
4.3. Универсальный асинхронный модем.....	43
Контрольные вопросы	44
5. Интерфейсы последовательной передачи данных	
5.1. Интерфейс RS-232 (TIA/EIA-232).....	45
5.2. Интерфейс RS-423 (TIA/EIA-423).....	54
5.3. Интерфейс RS-422 (ANSI/TIA/EIA-422).....	56
5.4. Интерфейс RS-485 (ANSI TIA/EIA-485).....	58
5.5. Сравнение интерфейсов семейства RS	70
Контрольные вопросы	71
6. Символьные протоколы обмена данными	
6.1. Код ASCII.....	72
6.2. Протокол ADAM ASCII (DCON)	74
Контрольные вопросы	76
7. Сети Modbus	
7.1. Общие сведения	76
7.2. Modbus RTU.....	80
7.3. Modbus ASCII	82

7.4. Modbus TCP	84
7.5. Функции Modbus	85
7.6. Характеристики интерфейсов сети Modbus	96
Контрольные вопросы	98
8. Токовая петля	
8.1. Общие сведения	98
8.2. Аналоговая токовая петля	100
8.3. Цифровая токовая петля	103
Контрольные вопросы	104
9. HART-протокол	
9.1. Общие сведения	105
9.2. Топология сети	109
9.3. Формат сообщений	113
9.4. Команды HART	119
9.5. Язык описания устройств	125
Контрольные вопросы	125
10. AS-интерфейс	
10.1. Общие сведения	126
10.2. Топология сетей AS-i	129
10.3. Передача данных	130
10.4. Модуляция	135
10.5. Полевые устройства	136
10.6. Физические характеристики	139
Контрольные вопросы	141
Библиографический список	142

ВВЕДЕНИЕ

Учебное пособие предназначено для подготовки студентов Южно-Уральского государственного университета (ЮУрГУ) по направлению 27.03.04 при изучении дисциплины «Промышленные сети и системы связи», которая входит в учебный план указанного направления.

В настоящее время учебники по промышленным сетям для направления 27.03.04 отсутствуют. Существующие учебные пособия, выпущенные родственными вузами, имеют небольшой тираж. Кроме того, они либо перегружены излишне большим объемом теоретического материала, либо посвящены рассмотрению достаточно узкого круга способов связи.

Перечисленные обстоятельства обуславливают необходимость и актуальность выпуска учебного пособия, содержащего основные теоретические вопросы организации систем связи и построения промышленных сетей, обеспечивающие подготовку в ЮУрГУ бакалавров по направлению 27.03.04.

Материал, приведенный в данном пособии, соответствует основной образовательной программе направления 27.03.04 и рабочей программе дисциплины «Промышленные сети и системы связи».

В результате освоения материала данного пособия студенты будут:

а) знать особенности архитектуры распределенных систем сбора данных и управления; основные характеристики современных промышленных сетей, их компоненты и протоколы обмена данными; действующие стандарты проектирования и реализации промышленных сетей;

б) уметь осуществлять сбор и производить анализ исходных данных для расчета и проектирования промышленных сетей, выбирать тип системы связи, физический канал и протокол, проектировать средства сопряжения с сетью (что отвечает компетенции ПК–5 ФГОС ВО для направления 27.03.04); производить расчеты параметров линий связей промышленных сетей в соответствии с техническим заданием (ПК–6);

в) обладать способностью выполнять эксперименты на действующих объектах по заданным методикам и обрабатывать результаты с применением современных информационных технологий и технических средств (ПК–1); владеть навыками проектирования промышленных сетей и систем связи.

Учебное пособие поддерживает лекционные, практические и лабораторные занятия, а также самостоятельную работу студентов.

В конце каждого раздела приведены контрольные вопросы, которые используются как один из элементов самостоятельной работы. Качество усвоения данных вопросов проверяется преподавателем в течение семестра в ходе периодического тестирования студентов.

1. СЕТЕВЫЕ ТЕХНОЛОГИИ В УПРАВЛЕНИИ ТЕХНИЧЕСКИМИ СИСТЕМАМИ

1.1. Автоматизированные системы управления

Термин *техническая система* является достаточно широким, в результате чего к таким системам относят отдельные машины, аппараты, приборы, сооружения, ручные орудия, а также их элементы: всевозможные узлы, блоки, агрегаты и т.п. Вместе с тем отличительной функциональной особенностью технических систем является то, что в каждой технической системе существует функциональная часть – *объект управления*, воспринимающий *управляющие воздействия* и изменяющий свое *состояние* в соответствии с ними [1]. Соответственно, для приведения объекта управления (а равно и всей технической системы) к некоторому желаемому состоянию, управляющие воздействия необходимо целенаправленно формировать.

Для простых систем обычно достаточно *локальных регуляторов*, но сложные и многокомпонентные системы, составляющие основу промышленного производства, зачастую требуют комплексного решения – применения *автоматизированных систем управления технологическими процессами* (АСУ ТП).

Согласно стандартам, АСУ ТП – это автоматизированная система для выработки и реализации управляющих воздействий на технологический объект управления в соответствии с принятым критерием качества управления. В свою очередь технологический объект управления – это совокупность технологического оборудования и реализованного на нем по соответствующим инструкциям и регламентам технологического процесса.

1.2. Концепция комплексной автоматизации

Концепция комплексной автоматизации предприятий направлена на создание единой информационной системы управления предприятием, позволяющей хранить и обрабатывать документацию, принимать управленческие решения, формировать бизнес-процессы и всегда иметь доступ к точным данным в любом разрезе деятельности такой организации. Соответственно, на современном производстве существует несколько взаимосвязанных уровней управления: управление финансами и цепочкой поставок, производством и технологическими процессами.

Интеграция – процесс организации отдельных компонентов в единую систему, который обеспечивает их согласованное и целенаправленное взаимодействие для достижения заданной цели. Результатом данного процесса является интегрированная система.

Интегрированная автоматизированная система (ИАС) – совокупность двух или более взаимосвязанных автоматизированных систем (АС), в которой функционирование одной из них зависит от результатов функционирования другой (других) так, что эту совокупность можно рассматривать как единую автоматизированную систему.

Интегрированная автоматизированная система управления (ИАСУ) – автоматизированная система, в которой комбинируются функции управления технологическими процессами и организационного управления производственным предприятием.

Таким образом, ИАСУ предполагает интеграцию АСУ ТП в автоматизированную систему управления предприятием (АСУП), типичное представление структуры которой представлено на рис. 1.1. Такая система состоит из четырех уровней (L4–L1):

- Enterprise Resource Planning (ERP). На этом уровне решаются задачи, связанные с организацией работы предприятия.
- Manufacture Execution System (MES). Задачи этого уровня – логистика и контроль над персоналом, оптимизация работы подчиненных АСУ ТП (уменьшение себестоимости продукции, а также оптимизации работы системы в целом).
- Supervisory Control And Data Acquisition / Human-Machine Interface (SCADA/HMI). Задачи этого уровня – создание условий безотказной работы системы АСУ ТП, адаптация и оптимизация работы оборудования, архивирование и визуализация технологических параметров.
- Local Control And Data Acquisition (LCADA). Выполняет стабилизацию режимных параметров, обработку защит и блокировок и программно-логическое управление отдельными установками.

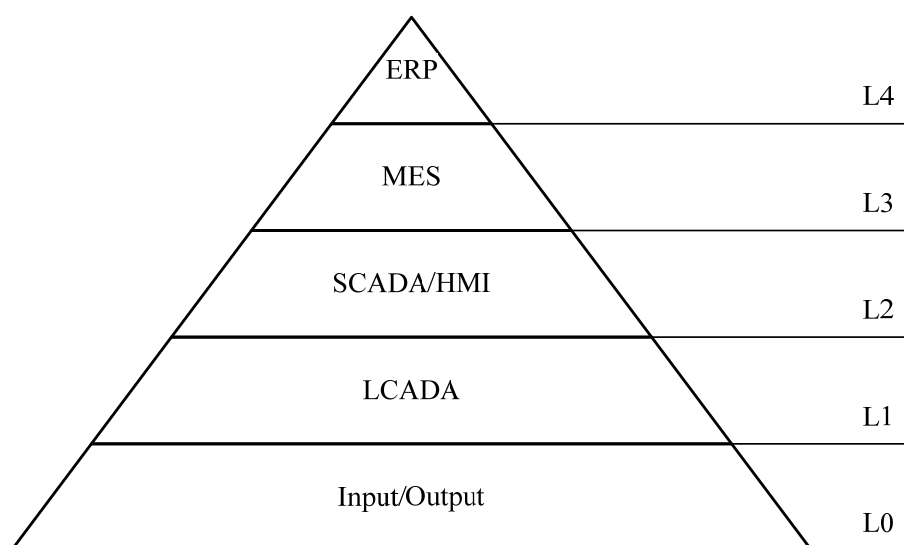


Рис. 1.1. Структура современной интегрированной АСУП

Нулевой уровень (L0) – это уровень датчиков и исполнительных механизмов технологического процесса. Саму же АСУ ТП образуют уровни L2–L0, соответственно, верхний уровень (SCADA/HMI), нижний уровень (LCADA) и полевой уровень.

Комплексная автоматизация затрагивает как горизонтальные, так и вертикальные интеграционные процессы (рис. 1.2), и в соответствии с видами обеспечения выделяют: организационную, функциональную, информационную, программную и техническую интеграцию.

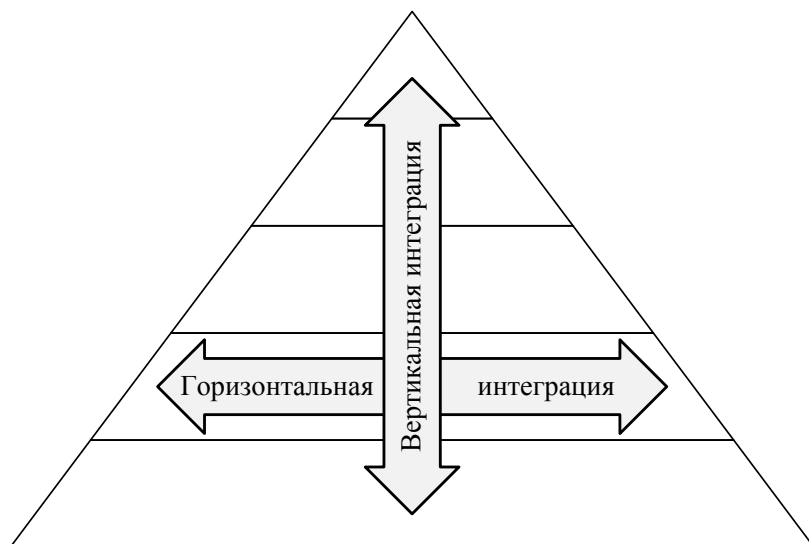


Рис. 1.2. Направления интеграции компонентов АСУП

Организационная интеграция – рациональное сочетание управленческой деятельности персонала по всем уровням ИАСУ и в различных локальных ее подсистемах, которое определяет согласованность управленческих решений.

Функциональная интеграция обеспечивает единство локальных целей функционирования, согласованность функций и критериев эффективности всех компонентов.

Информационная интеграция заключается в создании условий, при которых возможен доступ ко всем необходимым данным для реализации функций системы.

Программная интеграция заключается в обеспечении совместного функционирования отдельных составляющих программного обеспечения с целью их взаимодействия.

Техническая интеграция заключается в сочетании технических средств для обеспечения сбора, передачи и циркуляции технологической, технико-экономической и командной информации между необходимыми составляющими системы.

Одной из основных функций горизонтальной интеграции является обеспечение информационного взаимодействия между существующими

подсистемами одного уровня. На уровне АСУ ТП горизонтальная интеграция предполагает объединение между собой АС управления технологических и производственных процессов, а также административных подразделений предприятия в единую систему. На уровне АСУП горизонтальная интеграция предназначена для объединения рабочих станций организационно-экономического и планового уровня управления предприятием.

Вертикальная интеграция предназначена для интеграции систем различного уровня иерархии управления, например АСУП и АСУТП. Учитывая, что данные системы работают для достижения единой цели, например получения прибыли, их объединение должно улучшить этот процесс.

Все уровни АСУ ТП должны иметь вертикальную информационно-техническую связь друг с другом. Этот обмен информацией происходит в обе стороны: как снизу вверх, так и сверху вниз.

1.3. Вычислительные сети в структуре АСУ ТП

В течение многих лет АСУ ТП строились по традиционной централизованной схеме, в которой имелось одно мощное вычислительное устройство и огромное количество кабелей для подключения оконечных устройств (датчиков, исполнительных механизмов, устройств отображения информации). Такая структура диктовалась высокой ценой электронно-вычислительной техники.

На современном этапе, в условиях бурно растущего производства цифровой техники, стоимость вычислительных устройств зачастую оказывается ниже затрат на кабельно-проводниковую продукцию, и становится экономически целесообразной установка на площадке цеха или участка нескольких локальных контроллеров или интеллектуальных устройств связи с объектом, объединенных в единую сеть передачи данных. Таким образом, современные АСУ ТП являются *распределенными системами управления*, и при построении таких систем нужно учитывать, что на сегодняшний день технологии информационной, технической и программной интеграции базируются на цифровых технологиях с использованием вычислительных сетей. То есть интегрированное производство сегодня – это единая система, которая представляет собой объединение разнотипных микропроцессорных узлов с использованием вычислительных сетей, которые используются на всех уровнях АСУ ТП, но если на верхних уровнях это обычные компьютерные сети, то на нижних уровнях пирамиды автоматизации применяются *промышленные сети* (рис. 1.3).

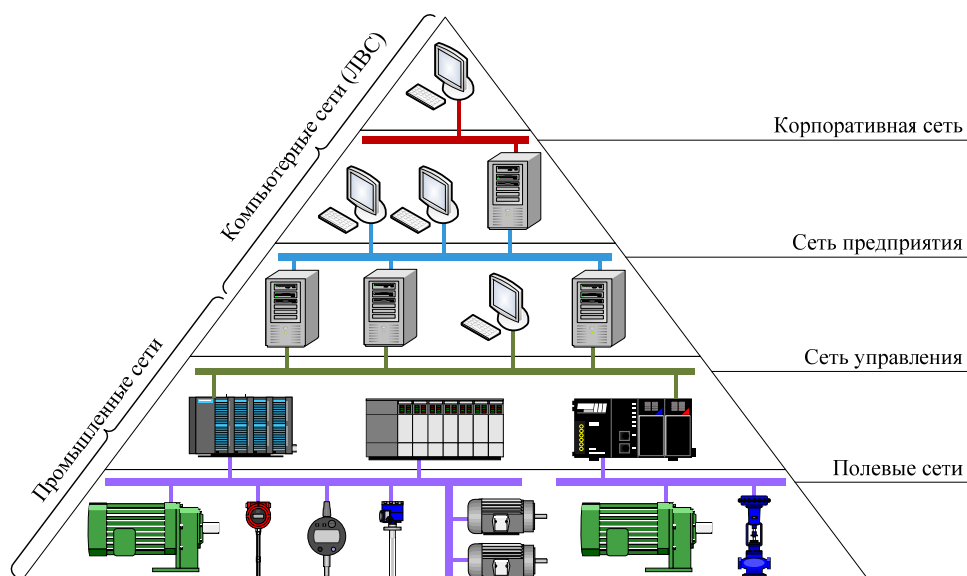


Рис. 1.3. Сети в автоматизированных системах управления

Контрольные вопросы

1. Объясните необходимость интеграции автоматизированных систем.
2. Назовите основные виды интегрированных автоматизированных систем. Из каких подсистем они состоят, и для чего проводится интеграция?
3. Каково назначение горизонтальной интеграции на уровне АСУП и АСУТП?
4. Каково назначение вертикальной интеграции в ИАСУ?
5. В контексте каких видов обеспечения производится интеграция? В чем она заключается?
6. В силу каких обстоятельств возросла актуальность применения промышленных сетей в настоящее время?

2. ОСНОВЫ ПРОМЫШЛЕННЫХ СЕТЕЙ

2.1. Базовые термины и определения

Промышленная сеть – сеть передачи данных, связывающая различные датчики, исполнительные механизмы, промышленные контроллеры и используемая в промышленной автоматизации.

В России промышленные сети появились, следуя тенденциям Запада, где, в соответствии со спецификацией IEC 61158, употребляется термин «Fieldbus», что дословно переводится как «полевая шина». Однако на рус-

ском языке в настоящее время принято использовать термин «промышленная сеть», что более точно отражает суть организации процесса передачи данных и не вводит никого в заблуждение, так как полевая шина фактически является частным случаем построения промышленной сети с использованием соответствующей топологии ее построения.

Полевая шина (Fieldbus) – это локальная сеть с линейной топологией, предназначенная для информационного обмена между территориально рассредоточенными промышленными инструментальными устройствами: датчиками, преобразователями, средствами ручного ввода, исполнительными механизмами, средствами отображения информации и т.д.

То есть, строго говоря, термины «полевая шина» и «промышленная сеть» в общем случае синонимами не являются, но очень часто используются как таковые у различных авторов, особенно при рассмотрении сетей, построенных по шинной топологии.

Соединение промышленной сети с ее компонентами (устройствами, узлами сети) выполняется с помощью интерфейсов.

Интерфейс – совокупность возможностей, способов и методов взаимодействия двух систем, устройств или программ для обмена информацией между ними, определенная их характеристиками, характеристиками соединения, характеристиками сигналов обмена и т. п. Можно рассматривать интерфейс как логическую и (или) физическую границу между устройством и средой передачи информации. Для обмена информацией взаимодействующие устройства должны иметь одинаковый протокол обмена.

Протокол обмена – это набор правил, которые управляют обменом информацией. Протокол обмена определяет синтаксис и семантику сообщений, операции управления, синхронизацию и состояния при коммуникации. Он может быть реализован аппаратно, программно или программно-аппаратно.

Название сети обычно совпадает с названием протокола, что объясняется его определяющей ролью при создании сети.

Обычно сеть использует несколько протоколов, образующих *стек протоколов* – набор связанных коммуникационных протоколов, которые функционируют совместно и используют некоторые или все семь уровней модели взаимодействия открытых систем. Для большинства сетей стек протоколов реализован с помощью специализированных сетевых микросхем или встроен в универсальный микропроцессор.

2.2. Назначение и классификация промышленных сетей

Ранее, говоря о месте вычислительных сетей в структуре современной АСУ ТП, отмечалось, что непосредственно промышленные сети располагаются на нижних уровнях пирамиды автоматизации. Классифицируя промышленные сети можно выделить такие группы [2]:

- сети дискретных датчиков и исполнительных устройств;
- сети устройств с непрерывным воздействием;
- управляющие сети для связи контроллеров и модулей ввода-вывода;

Наиболее простыми являются сети уровня концевых выключателей, кнопок, пускателей двигателей, соленоидов и т.п. Протоколами таких сетей обеспечивается очень маленькое время цикла опроса, поэтому такие сети фактически являются альтернативой дискретному вводу-выводу в контроллерах. Стоимость узла сети относительно низкая.

Сети уровня устройств поддерживают более сложные датчики (датчики давления, уровня, потока, температуры, ...) и исполнительные механизмы с пропорционально-интегральным управлением, моторизованными задвижками и пневматическими блоками позиционирования. Стоимость узла такой сети естественно выше.

Сети уровня управления являются основой для обмена информацией между модулями ввода-вывода, контроллерами, рабочими станциями операторов и административными сетями. Это гетерогенные сети с большими объемами данных и высокими скоростями обмена, которые могут взаимодействовать как с простыми модулями ввода-вывода, так и с интеллектуальными устройствами.

Организация процесса передачи данных между различными узлами промышленной сети во многом определяется характером самих данных. В зависимости от назначения данные подразделяются на два типа: данные процесса и параметрические данные. Данные процесса – это оперативные параметры технологического процесса в режиме реального времени, а параметрические данные – это данные, используемые для программирования, конфигурирования узлов сети и их диагностики.

По формату данные разделяют на аналоговые, дискретные, временные, блочные (структурированные), строковые данные.

По принципам обновления выделяют циклические, периодические и ациклические данные. Циклическим данным свойственно регулярное обновление. Периодические данные имеют цикл постоянной длительности. Ациклические данные, в свою очередь, подразделяются на ациклические данные, обновляющиеся по запросу и ациклические данные, обновляющиеся при смене значений (спорадические).

Автоматизация предприятия – это сложный процесс, который может существенно растянуться по времени. Практически невозможно выполнить все работы сиюминутно, а потому зачастую используется оборудование, выпускаемое различными производителями. Такое положение дел легко объясняется экономическими предпосылками.

Объединение в одну цифровую сеть нескольких устройств от одного производителя обычно не содержит каких-либо существенных проблем, так как программные и аппаратные особенности учтены непосредственно

самим производителем. Когда же речь идет о построении сети из устройств различных производителей, то избежать проблем сопряжения возможно лишь в том случае, если принципы работы, как на программном, так и на аппаратном уровне будут идентичными.

Уникальные системы, которые поддерживает только один производитель, работающие по уникальным протоколам связи, получили название *закрытых систем* (closed/proprietary systems). В противоположность им, *открытые системы* (open systems) реализуют принцип доступности спецификаций на интерфейсы, сервисы и форматы данных для всех производителей оборудования, участвующих в построении такой системы.

Только при использовании принципов открытых систем интеграция изделий разных производителей в одну сеть может быть решена без особых проблем.

Таким образом, промышленные сети должны обладать следующими качествами [2]:

- *взаимосвязанностью* (interconnectivity) – возможностью свободного физического включения в общую сеть устройств от различных производителей;
- *совместимостью* (interoperability) – возможностью построения работоспособной сети на основе включения компонентов от различных поставщиков;
- *взаимозаменяемостью* (interchangeability) – возможностью замены компонентов аналогичными устройствами от других производителей.

В конечном итоге применение промышленных сетей позволяет в несколько раз снизить расходы на кабель и его прокладку, увеличить допустимое расстояние до подключаемых датчиков и исполнительных устройств. В результате упрощается управление сетью датчиков и исполнительных механизмов, упрощается модификация системы при изменении типа датчиков, используемого протокола взаимодействия, добавлении устройств ввода-вывода. Появляется возможность дистанционно настраивать (конфигурировать) датчики и проводить их диагностику.

2.3. Общие требования к промышленным сетям

Промышленные сети логически мало чем отличаются от локальных вычислительных сетей, однако они должны отвечать *специфическим требованиям* [2]:

- жесткая детерминированность (предсказуемость) поведения;
- обеспечение функций реального времени;

- работа на длинных линиях с использованием недорогих физических сред (например, витая пара);
- повышенная надежность физического и канального уровней передачи данных для работы в промышленной среде (например, при больших электромагнитных помехах);
- наличие специальных высоконадежных механических соединительных компонентов;
- возможность организации питания оборудования по кабельной системе;
- свободная топология.

2.4. Модель взаимодействия открытых систем

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 ВОО, OSI-модель, Open System Interconnection model). Модель ВОО предлагает структуру для идентификации и разграничения различных составляющих коммуникационного процесса [3].

Модель взаимодействия открытых систем не связана с конкретными реализациями и описывает коммуникационный процесс в абстрактных понятиях. Модель ВОО – 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 определенную степень совместимости и взаимозаменяемости [4].

Сетевая модель ВОО включает в себя физический, канальный, сетевой, транспортный, сеансовый уровни, уровень представления и прикладной уровни (таблица 2.1).

2.4.1. Физический уровень

Физический уровень (Physical Link Layer) представляет собой физическую среду передачи – электрическую или оптическую, с соответствующими интерфейсами к сопрягаемым объектам, которые называются станциями или узлами. Все вопросы, касающиеся среды передачи, уровней сигналов и частот, кодирования данных, методов передачи, формы и типов разъемов и т.п. рассматриваются на этом уровне.

Основная функция физического уровня – обеспечить прием логической единицы (а не нуля) на стороне получателя в том случае, если на стороне

Таблица 2.1

№	Уровень	Единица обмена		Функции
7	Прикладной	Данные	APDU	Доступ к сетевым службам
6	Представления	Поток	PPDU	Представление и шифрование данных
5	Сеансовый	Сеанс	SPDU	Управление сеансом связи
4	Транспортный	Сегмент	TPDU	Прямая связь между конечными пунктами и надежность
3	Сетевой	Пакет		Определение маршрута и логическая адресация
2	Канальный	Кадр		Физическая адресация
1	Физический	Бит		Работа со средой передачи, сигналами и двоичными данными

отправителя передана единица. На физическом уровне используются понятия: длительность импульса и паузы, длительности фронтов импульсов, количество проводов и цоколевки разъемов, помехозащищенность, волновое сопротивление, полоса пропускания, кодирование информации, синхронизация, модуляция, определение начала кадра, уровень логической единицы и логического нуля.

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.

2.4.2. Канальный уровень

Канальный уровень (Data Link Layer) реализует функции, связанные с формированием и передачей кадров 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 поврежденного 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.

На физическом уровне не учитывается, что канал передачи может быть занят другими устройствами, подключенными к сети. Проверка доступности канала передачи осуществляется на канальном уровне. Канальный

уровень разбивает передаваемые данные на кадры и выполняет функции обнаружения и коррекции ошибок с помощью контрольной суммы, помещаемой в начало или конец каждого кадра. Кадр считается полученным, если контрольная сумма, вычисленная на стороне получателя, совпадает с контрольной суммой, приписанной к переданному кадру. При обнаружении ошибки канальный уровень может выполнить повторную передачу поврежденного кадра. Функция исправления ошибок не является обязательным требованием стандарта.

2.4.3. Сетевой уровень

Основное назначение сетевого уровня (Network Layer) – определение маршрутов пересылки пакетов от источника к приемнику сообщений. Маршруты могут быть жестко заданы или меняться динамически в зависимости от текущей загруженности сети. Сетевой уровень отвечает также за объединение нескольких разных сетей в единую транспортную систему, причем эти сети могут использовать различные принципы передачи сообщений, методы адресации и обладать произвольной структурой связей.

2.4.4. Транспортный уровень

Транспортный уровень (Transport Layer) обеспечивает доставку сообщений с требуемым уровнем качества (надежности). В модели ВОС выделяют на выбор пользователя пять классов сервиса, которые отличаются срочностью доставки сообщений, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, способностью к обнаружению и исправлению ошибок передачи (таких как искажение, потеря или дублирование пакетов).

На транспортном уровне выполняется разбивка сообщений сеансового уровня на пакеты, их нумерация, буферизация принимаемых пакетов, упорядочивание прибывающих пакетов, управление потоками.

2.4.5. Сеансовый уровень

Сеансовый уровень (Session Layer) управляет диалогом (сеансом связи): отслеживает очередность передачи сообщений участниками сети; вставляет метки в длинные сообщения, чтобы в случае потери связи выполнить повторную передачу только утерянной части сообщения; устанавливает способ обмена (дуплексный или полудуплексный).

Этот уровень редко используется на практике, а его функции часто объединяются с функциями прикладного уровня.

2.4.6. Уровень представления

Уровень представления (Presentation Layer) оперирует с формой представления передаваемой по сети информации, не изменяя ее содержания. На этом уровне может задаваться тип кодирования символов (например, ASCII или ANSI), может изменяться синтаксис передаваемых сообщений,

задаваться тип шифрования и дешифрования данных для обеспечения их секретности. Примером такого протокола является Secure Socket Layer (SSL), обеспечивающий секретность сообщений для протокола TCP IP.

2.4.7. Прикладной уровень

Прикладной уровень (Application Layer) – это 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.

Модель ВОС не включает средства взаимодействия между собой приложений, расположенных на разных узлах сети, а описывает только средства, реализуемые операционной системой, системными утилитами и аппаратурой. Поэтому прикладной уровень нельзя путать с уровнем взаимодействия приложений, который в модель ВОС не входит.

Все, что находится выше седьмого уровня модели, это задачи, решаемые в прикладных программах.

Основная идея модели ВОС состоит в следующем. 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, определенным для их уровня. На самом деле объекты запрашивают услуги непосредственно у нижележащего уровня с помощью вызова процедур, 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 (рис. 2.1).

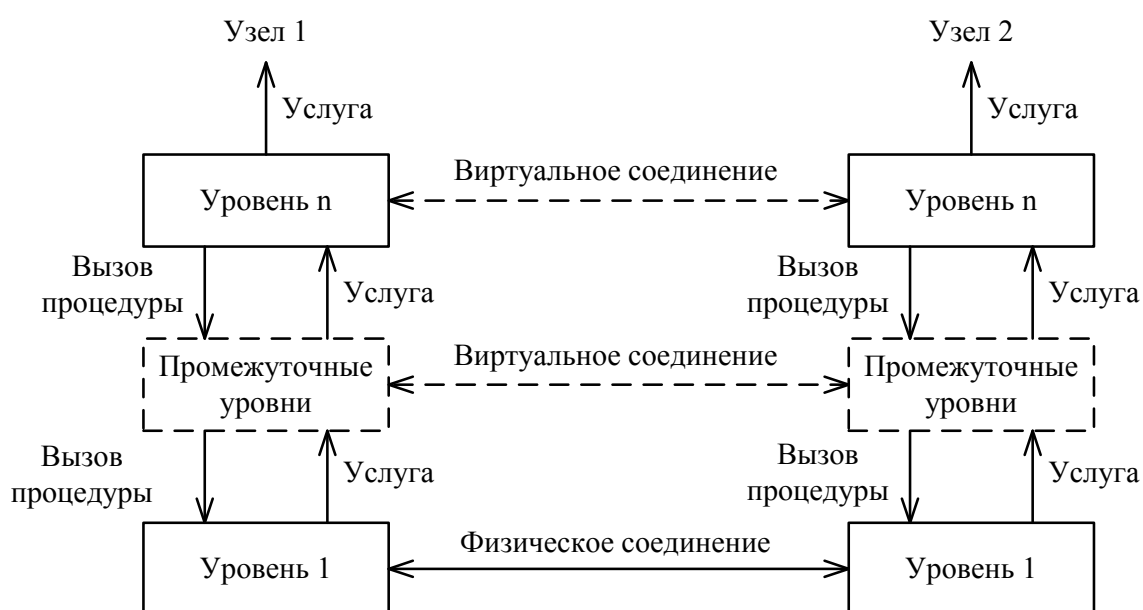


Рис. 2.1. Виртуальное соединение одноранговых объектов в модели ВОС

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 (рис. 2.2). 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.

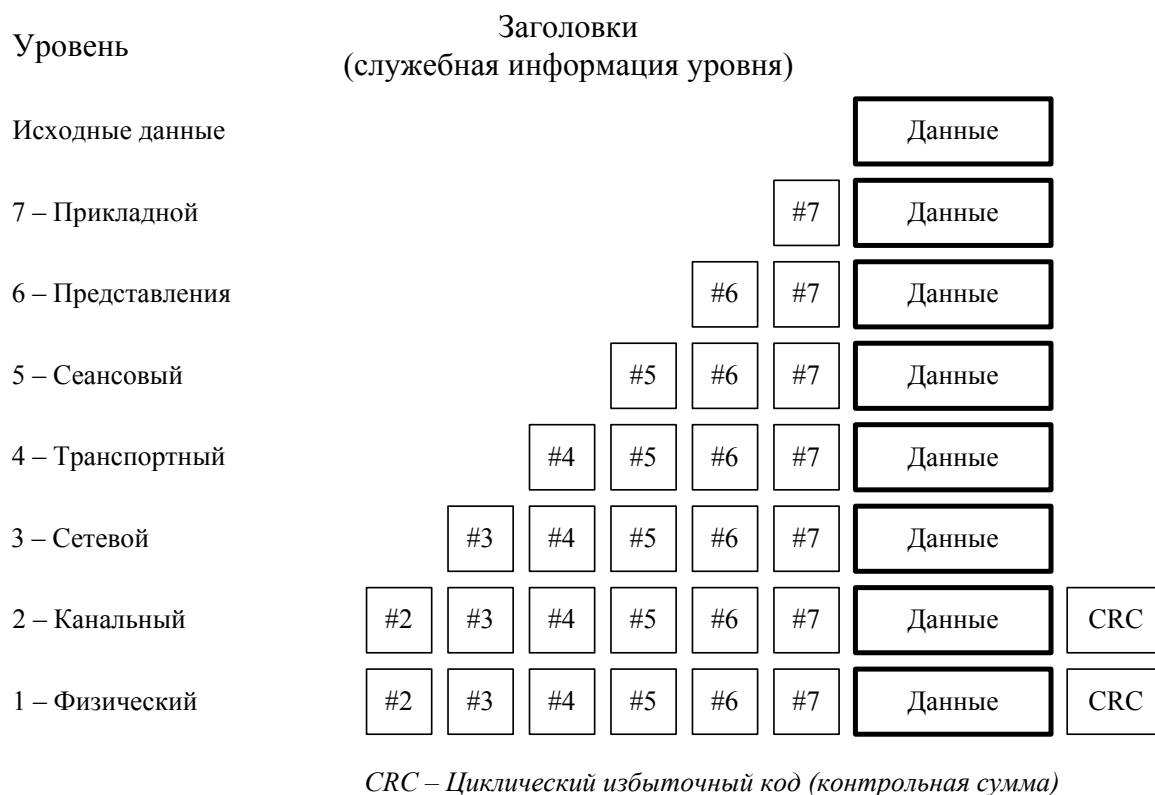


Рис. 2.2. Общая схема передачи информации между уровнями

2.5. Топология промышленных сетей

Топология сети – способ сетевого объединения устройств.

В зависимости от используемой среды передачи, могут применяться различные сетевые топологии:

- точка-точка;
- линия;
- шина;
- кольцо;
- звезда;
- дерево;
- ячеистая топология;
- полносвязная.

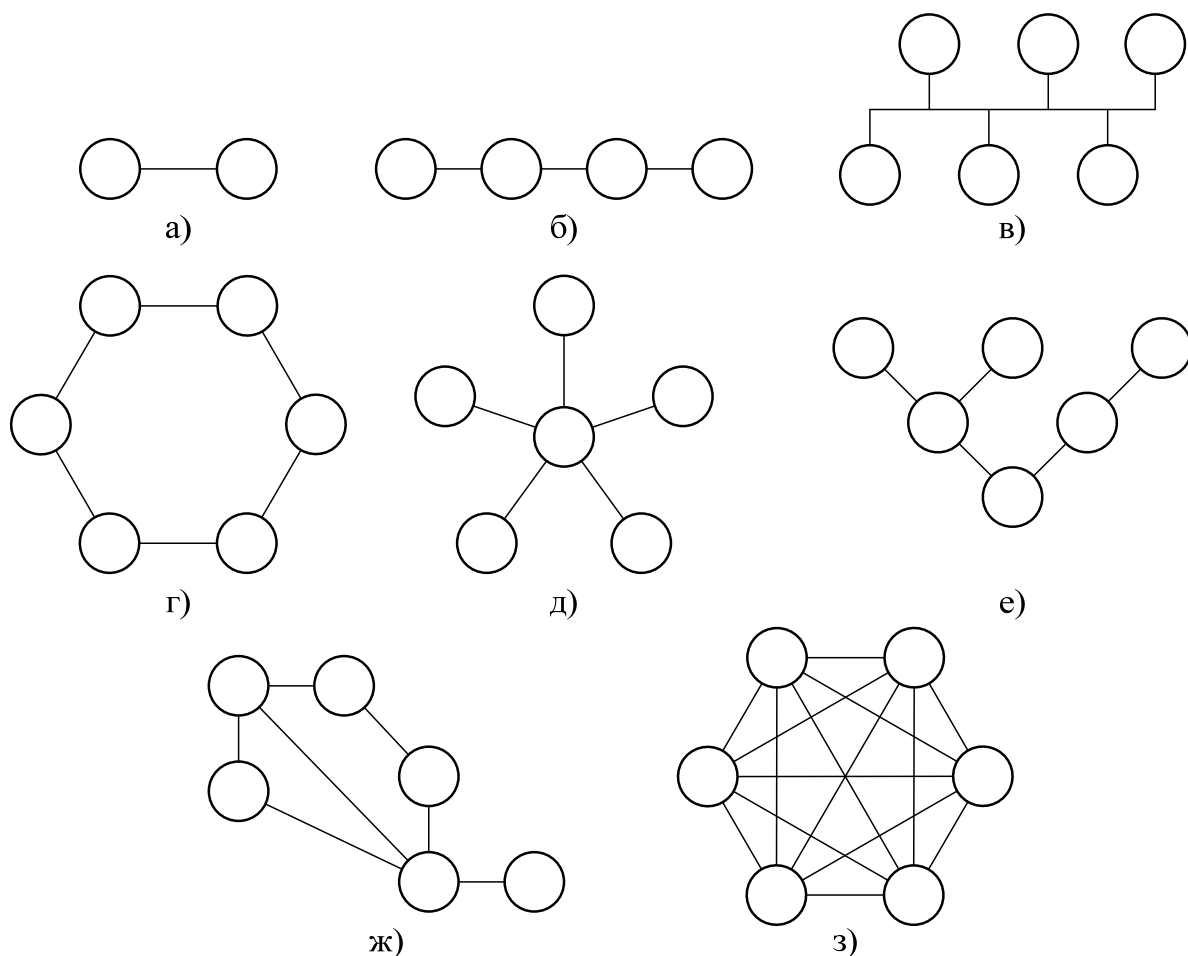


Рис. 2.3. Топологии сетей: а – точка-точка, б – линия, в – шина, г – кольцо, д – звезда, е – дерево, ж – ячеистая, з – полносвязная

Топология «точка-точка» представляет простейший вид сети, при котором два устройства соединяются между собой напрямую через коммуникационное оборудование. Достоинством такого вида соединения является простота и дешевизна, недостатком – соединить таким образом можно только два устройства и не больше.

В структуре линия узел передает сообщение другому узлу, а тот в свою очередь передает следующему узлу и так до тех пор, пока сообщение не придет на целевой узел. Отправленное с одного устройства сообщение может пройти через все устройства в сети.

Данный тип построения сети используется при построении протяженных сетей. Повторители сигналов (репитеры) и шлюзы являются элементами такой сетевой топологии. К недостаткам подобного способа организации сети относится то, что каждое устройство в сети обрабатывает передаваемые данные при приеме, в результате чего скорость передачи данных существенно сокращается, так как передача начнется только после окончания приема очередного фрагмента сообщения. Другой недостаток заключается в том, что при повреждении линии вся сеть приходит в негодность,

так же как и при отключении одного из устройств, так как исключается связующее звено.

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, скажем, 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.

В кольцевой структуре информация передается от узла к узлу по физическому кольцу. Приемник получает данные и передает 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.

В топологии типа «звезда» вся информация передается 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 надежным устройством, так как это слабое место в организации связи – выход узла из строя парализует всю сеть.

Топология типа «дерево» берет свое начало из топологии «звезда», фактически расширяя ее, организуя более разветвленную сеть.

Ячеистая топология (англ. mesh) – сетевая топология на принципе ячеек, в которой каждый узел сети соединяется с несколькими другими узлами этой же сети с возможным принятием на себя функций коммутатора для других узлов. Характеризуется высокой отказоустойчивостью, сложностью настройки и, для проводных сетей, избыточным расходом кабеля. Получается из полносвязной путем удаления некоторых возможных связей. Эта топология допускает соединение большого количества узлов и характерна, как правило, для крупных сетей.

Полносвязная (fully-meshed) топология – это такое размещение узлов, при котором каждый узел соединен с каждым. В настоящее время эта то-

тология применяется лишь при организации работы узлов, соединенных между собой каналами радиосвязи.

При построении проводных сетей основными видами топологий являются звезда, кольцо и шина, которые целесообразно сравнить по трем основным критериям: режим доступа к сети, надежность сети, возможность изменения числа узлов сети. Сравнительные характеристики основных сетевых топологий приведены в таблице 2.2.

Таблица 2.2

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

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.

2.6. Взаимодействие приложений

Как уже отмечалось ранее, модель ВОС описывает только средства, реализуемые операционной системой, системными утилитами и аппаратурой сетевых устройств и не включает средства взаимодействия между собой приложений, расположенных на разных узлах сети. С другой стороны, для того, чтобы некоторое устройство могло осуществлять обмен данными с другими участниками сети, необходимо определить правила и семантику обмена между двумя прикладными сущностями, которые, в свою очередь, обеспечивают обмен между прикладными процессами.

Для того чтобы один прикладной процесс смог общаться с другим, необходимо определить «язык» общения и тип диалога. Сервисы прикладного уровня должны обеспечить необходимую функциональность сети.

Можно выделить следующие функциональные сервисы, поддерживаемые прикладным уровнем:

- обмен данными процесса в реальном времени;
- программирование/конфигурирование узла;
- диагностические сервисы;
- управление состоянием узла;
- резервирование.

Для того чтобы прикладные процессы на узлах могли обмениваться данными, необходимо настроить связь между ними. В зависимости от реализации такой связи можно выделить три модели взаимодействия прикладных процессов:

- Клиент – Сервер (Client – Server);
- Издатель – Подписчик (Publisher – Subscriber);
- Производитель – Потребитель (Producer – Consumer).

Модель «Клиент – Сервер» предполагает взаимодействие только двух прикладных процессов: запрашивающего процесса (клиента) и отвечающего процесса (сервера). Заказ услуг производится с помощью запросов (request), поступающих от клиента. Сервер, обработав запрос, возвращает ответ (response). Структура запроса и ответа зависит от реализации протокола.

Модель «Издатель – Подписчик» обеспечивает связь между несколькими прикладными процессами, один из которых «Издатель», а другие – «Абоненты». Процедура передачи данных называется публикацией (publication). Этот тип обмена наилучшим образом подходит для передачи данных в многоадресном режиме, поскольку прикладных процессов – абонентов может быть несколько. Таким образом, в определенный момент времени данные поступают от прикладного процесса – «Издателя» всем процессам – «Абонентам». В зависимости от того, какой прикладной процесс генерирует публикацию, выделяют два типа модели «Издатель – Подписчик» (рис. 2.4):

- pull model, когда момент публикации определяет прикладной процесс одного из узлов, который в нужный момент отправляет запрос на публикацию (Pull Publishing Manager);
- push model, когда момент публикации определяет прикладной процесс – «Издатель», например, при изменении этих данных или через определенные промежутки времени.

Следует также заметить, что некоторый прикладной процесс в системе для одних данных может быть издателем, а для других – подписчиком.

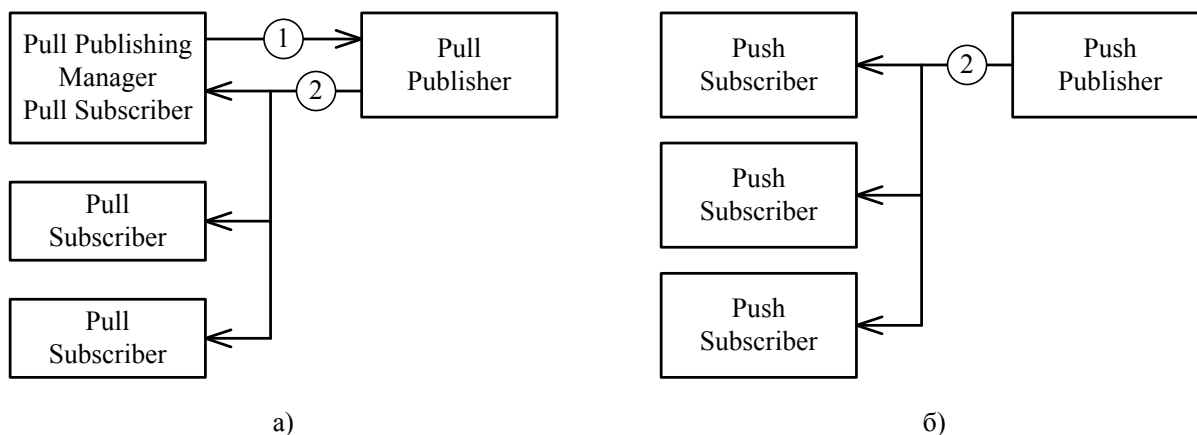


Рис. 2.4. Модель «Издатель – Подписчик»: а – pull model, б – push model,
1 – запрос публикации, 2 – публикация

Модель «Производитель – Потребитель» с точки зрения пользователя аналогична модели «Издатель – Подписчик», за исключением того, что для адресации получателей используется фильтрация по идентификатору сообщений. Прикладной процесс, в широковещательном режиме выдающий данные в сеть, называется производителем, а процесс, их принимающий – потребителем. Все остальные характеристики аналогичны модели «Издатель – Подписчик», поэтому при рассмотрении вопросов функционирования сетей на прикладном уровне эти модели часто отождествляются.

2.7. Методы управления доступом к среде передачи

Одними из значимых требований, предъявляемых к промышленным сетям, являются простота реализации и надежность передачи данных. Практически это достигается за счет выбора надлежащей топологии построения сети и организации процедуры доступа узлов к среде передачи данных.

Подавляющее большинство промышленных сетей, использующих в качестве среды передачи данных проводные линии связи, имеет шинную или древовидную топологию, а сама передача данных является *широковещательной*.

В широковещательных сетях все узлы подключены к линии связи и прослушивают ее, а соответственно получают одинаковые данные – *кадры*, которые посылает в линию связи передатчик. Задача канального уровня каждого из принимающих узлов – отфильтровать кадры, которые не предназначены для конкретного узла (рис. 2.5).

Доставка кадров может проводиться одним из двух способов:

- доставка кадров ориентирована на адрес узлов (модель сообщений «Отправитель – Получатель»);
- доставка кадров ориентирована на идентификатор сообщения (модель сообщений «Производитель – Потребитель»).

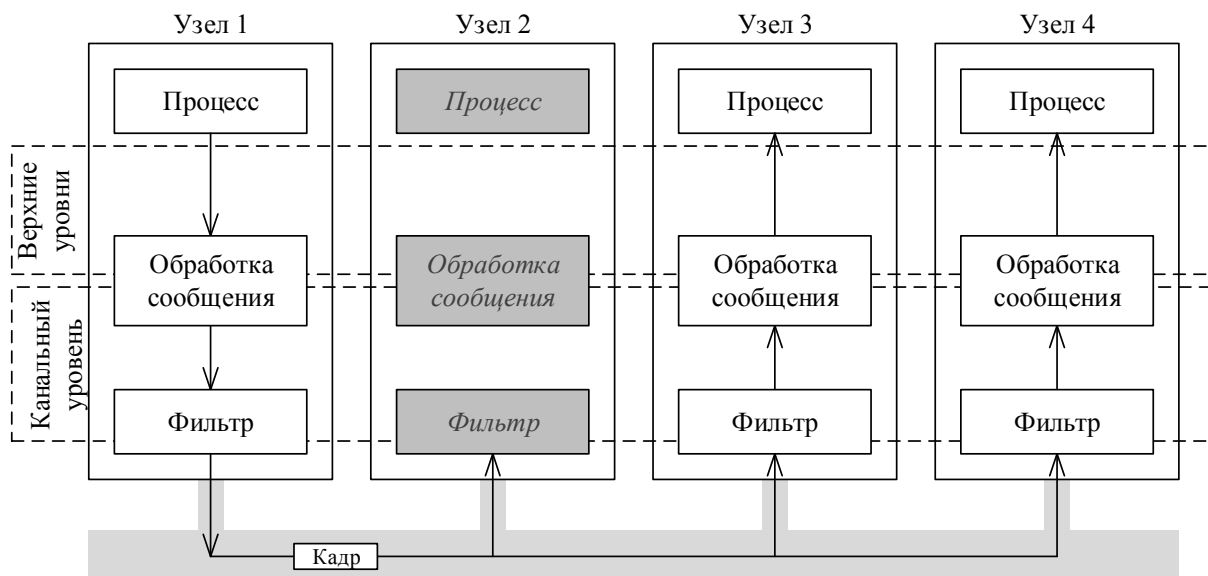


Рис. 2.5. Фильтрация кадров на канальном уровне

Доставка кадров по модели «Отправитель – Получатель» ориентирована на адресацию узлов получателя (иногда отправителя), и является наиболее популярным способом доставки данных. В таких системах каждый узел в сети имеет свой уникальный *адрес*. При формировании кадра, канальный уровень узла отправителя добавляет к нему заголовок с адресом (или адресами) узлов получателей. Все узлы получают этот кадр, но только узлы, с указанным адресом пропустят его для обработки на верхних уровнях модели ВОС. Таким образом, фильтр канального уровня каждого узла настроен на свой адрес. Адрес узла на канальном уровне принято называть *MAC-адрес* (Media Access Control Address).

Ряд сетей с общим доступом используют другой способ фильтрации кадров – на основе *идентификаторов сообщения*. При таком способе адресации доставки кадров, каждый кадр содержит идентификационное поле, по которому проводится фильтрация. Фильтр каждого узла настраивается только на нужное множество идентификаторов. Иначе говоря, фильтр будет пропускать через себя только те кадры, идентификаторы которых соответствуют одному из заданных значений, что соответствует модели «Производитель – Потребитель».

Каждый кадр, передаваемый по сети, должен иметь уникальный идентификатор, который:

- для канального уровня одного из узлов будет настроен как отправитель (Производитель, Producer);
- для канального уровня узлов назначения будет настроен как получатель (Потребитель, Consumer), то есть его фильтр будет пропускать этот кадр.

Если в сети с шинной топологией не будет четко регламентировано, кто в конкретный момент может занимать шину для передачи, то может возникнуть ситуация, когда несколько узлов начнут одновременную передачу данных. Такая ситуация называется *коллизией* или конфликтом. В этом случае приемники не смогут определить, какие данные были переданы, поскольку сигнал в линии связи будет результатом действий нескольких передатчиков. Существует два метода регулирования доступа к шине, известных также под термином *шинный арбитраж*:

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

В случае централизованного контроля доступа к шине (рис. 2.6) выделяется узел с правами мастера (Master, ведущее устройство или *шинный контроллер*). Право инициировать циклы чтения-записи на шине имеет только мастер. Он адресует каждого пассивного участника (Slave, ведомое или подчиненное устройство), обеспечивает их данными и запрашивает у них данные. Очень часто такой способ взаимодействия называют «Ведущий – Ведомый» или «Master – Slave».

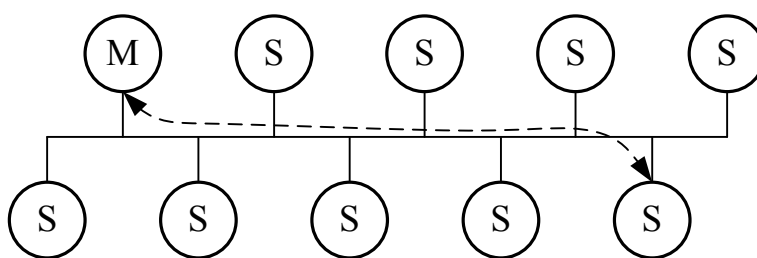


Рис. 2.6. Централизованный доступ к шине: M – Master, S – Slave

Существенным недостатком централизованного контроля шины является то, что во всех операция обмена участвует один узел – мастер. Если по каким-то причинам мастер выйдет из строя, то и циклы обмена данными по шине останавливаются, сеть полностью парализуется.

Именно по этой причине децентрализованный контроль с переходящими функциями мастера от одного узла сети к другому получил наибольшее внимание и развитие.

В случае децентрализованного контроля права мастера назначаются группе устройств сети. Широко приняты и используются две модели децентрализованного доступа:

- модель случайного доступа к шине
- модель с передачей маркера (Token Passing Model).

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). Данный способ применяется, например, при построении локальных вычислительных сетей в соответствии со спецификацией 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 коллизии, и тогда все претенденты «снимают свою заявку». Затем каждый из участников выжидает некоторый случайный интервал времени до следующего момента запроса шины, если она свободна (рис. 2.7).

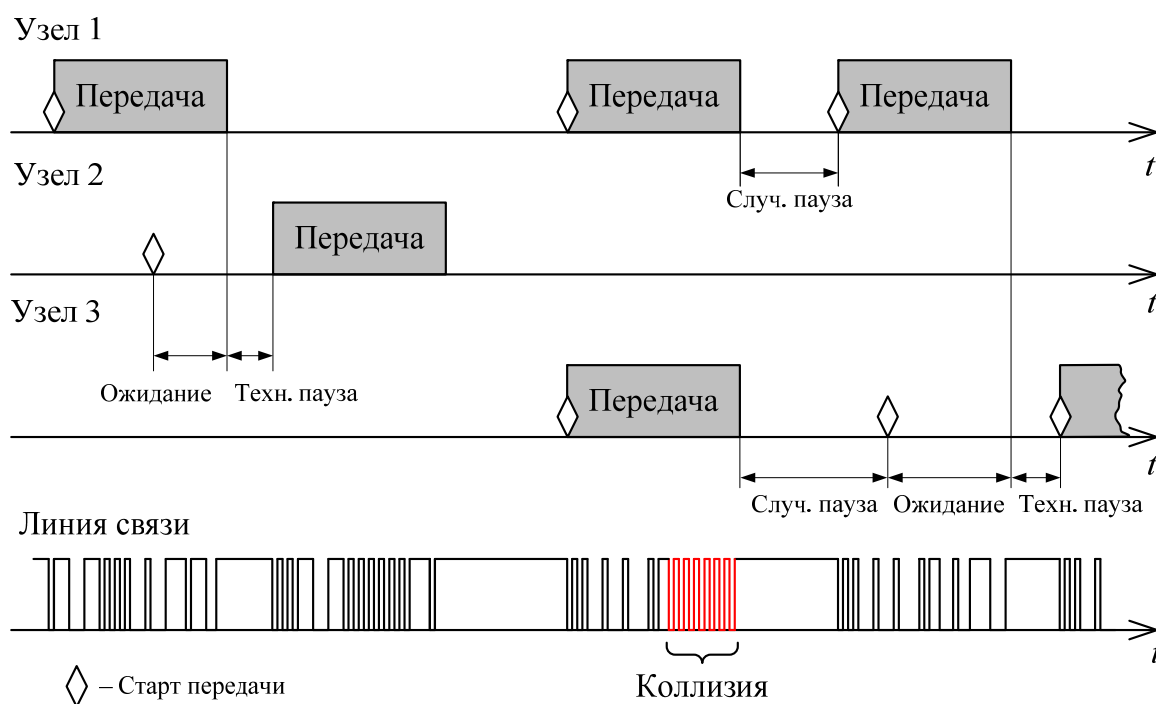


Рис. 2.7. Метод доступа CSMA/CD

Метод доступа CSMA/CD наиболее эффективен в условиях относительно низкой общей загрузки канала передачи данных (менее 30%). В условиях большей загрузки канала выгоднее использовать сети, реализующие иные модели доступа к шине.

В методе множественного доступа с контролем несущей и предотвращением коллизий (Carrier Sense Multiple Access With Collision Avoidance, CSMA/CA) так же, как и CSMA/CD каждый узел сети осуществляет прослушивание линии связи на предмет обнаружения несущей. Узел, который собирается начать передачу, посылает *сигнал преднамеренной помехи* (jam-сигнал), сообщая другим узлам о намерении использовать линию связи. После некоторого периода ожидания, в течение которого другие узлы

также могут послать сигнал преднамеренной помехи, если таковых сигналов не последовало, узел начинает передачу кадра. Если после передачи собственного сигнала преднамеренной помехи узел обнаруживает такой сигнал от другого узла, то передача данных откладывается на отрезок времени случайной длины, а затем попытка повторяется.

В отличие от CSMA/CD для метода CSMA/CA коллизиям подвержены не все пакеты данных, а только короткие jam-сигналы. Улучшение производительности достигается за счет снижения вероятности коллизий и, как следствие, повторных попыток передачи. Но в то же время ожидание jam-сигналов создает дополнительные задержки, поэтому другие методики позволяют достичь лучших результатов. Предотвращение коллизий полезно на практике в тех ситуациях, когда своевременное обнаружение коллизии невозможно – например, при использовании радиопередатчиков.

На рис. 2.8 приведен пример взаимодействия между передатчиком и приемником в соответствии с протоколом 802.11 RTS/CTS, который реализует метод CSMA/CA. В качестве jam-сигналов используются короткие сообщения: Request to Send (запрос на отправку) и Clear to Send (готовность к отправке).

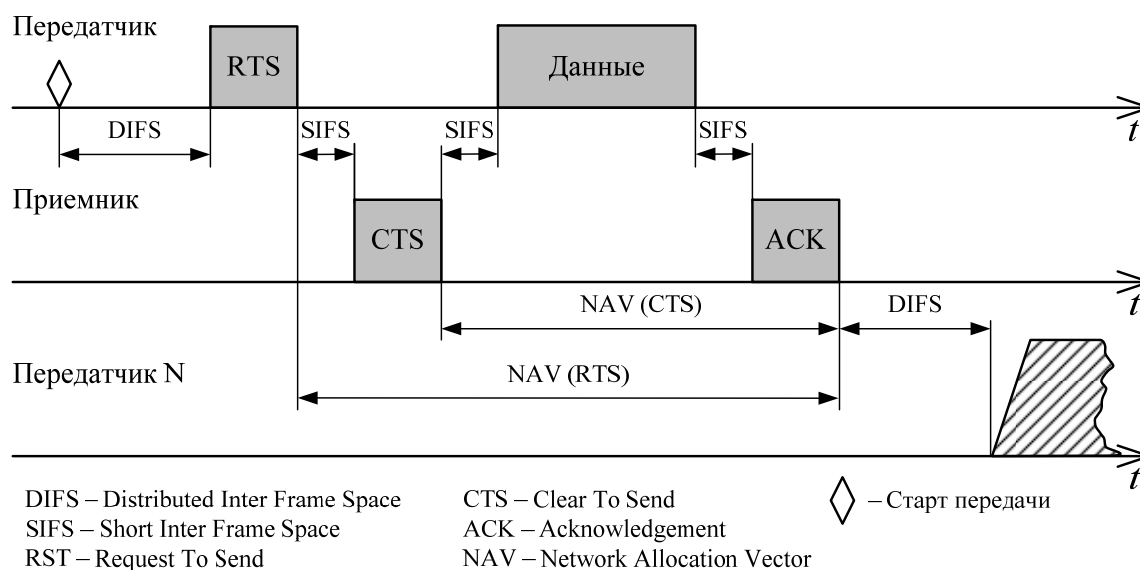


Рис. 2.8. Метод доступа CSMA/CA (802.11 RTS/CTS)

Метод передачи маркера предлагает каждому участнику сети справедливое разделение сетевых ресурсов в соответствии с их запросами. Инициировать процедуру обмена данными может только узел, владеющий *маркером* – специальным пакетом данных. По окончании сеанса связи маркер передается от узла к узлу по логическому кольцу. На рис. 2.9 представлено состояние сети в два разных момента времени t_1 и t_2 ($t_1 < t_2$).

Принцип передачи маркера используется в системах, где реакция на события, возникающие в распределенной системе, должна проявляться за определенное время (системы реального времени).

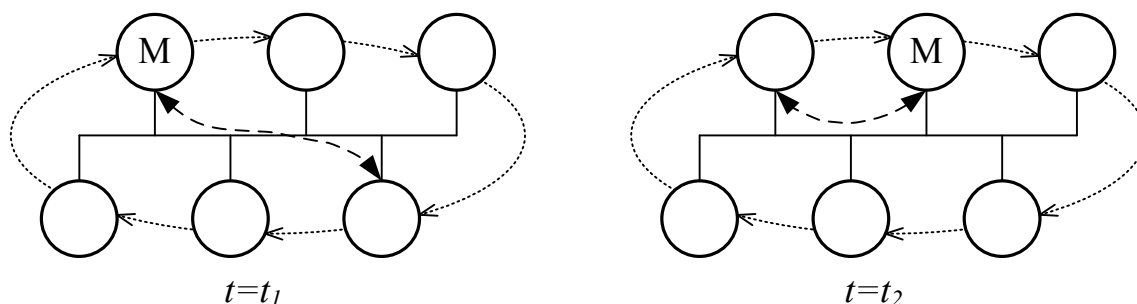


Рис. 2.9. Метод передачи маркера

Контрольные вопросы

1. Объясните назначение физического, канального, сетевого, транспортного и прикладного уровней в контексте модели ВОС.
2. Приведите перечень рабочих характеристик, с помощью которых можно оценить промышленную сеть.
3. Назовите основные топологии сетей.
4. Какие сервисы может поддерживать промышленная сеть на прикладном уровне? Как может быть реализован интерфейс прикладного уровня?
5. Какие модели обмена между прикладными процессами чаще всего используются в промышленных сетях? Как они функционируют?
6. Какие модели функционирования сервисов прикладного уровня можно выделить в промышленных сетях? Как они функционируют?
7. Каким образом организуется доставка кадров узлам в сети?
8. Зачем необходимо регламентировать доступ к среде передачи? Что такое коллизия в сетях с широковещательным доступом? Какие существуют варианты их предотвращения или решения?
9. Объясните, как решается порядок доступа к шине в системах с методом «Ведущий – Ведомый»?

3. ФИЗИЧЕСКИЙ УРОВЕНЬ РЕАЛИЗАЦИИ СЕТЕЙ

3.1. Интерфейс физического уровня

Промышленные сети не используют все уровни модели ВОС, но самый нижний – физический – присутствует всегда. Он является единственной материальной связью между узлами, и с его помощью осуществляется передача электрических или оптических сигналов в кабель или радиоэфир, а

также прием таких сигналов и преобразование их в биты данных. Физический уровень модели ВОС фактически осуществляет интерфейс между сетевым устройством и средой передачи данных – линией связи.

Линия связи – это физическая среда, предназначенная для переноса информации между единицами оборудования, принимающими участие в информационном обмене, включая данные, сигналы управления и синхронизации.

Линии связи подразделяются на электрические линии связи (симметричный кабель, витая пара, коаксиальный кабель), волоконно-оптические линии связи (ВОЛС) и радиоэфир (беспроводная связь).

Канал передачи данных – это совокупность физической среды и технических средств, включая аппаратуру преобразования сигналов, вовлекаемых в процесс передачи информации между оборудованием системы связи.

Формирователь – это электрическая цепь или контакт реле (источник) на передающей стороне линии связи, посредством которого осуществляется передача двоичных цифровых сигналов в оконечную нагрузку по линии связи. Также можно представить формирователь как передатчик двоичных цифровых сигналов.

Оконечная нагрузка – электронная цепь (потребитель) на приемной стороне цепи обмена, посредством которой осуществляется прием двоичных цифровых сигналов от формирователя по линии связи. Также можно представить оконечную нагрузку как приемник двоичных цифровых сигналов.

Стандарты на интерфейсы промышленных сетей определяют параметры, которые позволяют оборудованию разных производителей подключаться и взаимодействовать друг с другом. Такие характеристики интерфейсов физического уровня делятся на *электрические* (или оптические, в зависимости от линии связи) и *механические*.

Электрические характеристики описывают уровни напряжения (или тока) и временные характеристики сигналов, в то время как механические – описывают интерфейсные разъемы и типы линий связи. Оптические характеристики: номинальная длина волны, скорость передачи, уровень чувствительности приемника и т.д.

3.2. Принципы обмена данными

Интерфейсы промышленных сетей используют *последовательный* принцип передачи данных, при котором данные передаются последовательно бит за битом через одну и ту же линию связи. Такой способ отличается от принципа *параллельной* организации связи, когда по отдельным линиям одновременно передается сразу несколько бит данных (4, 8, 16, ...). Несмотря на то, что, казалось бы, параллельный способ существенно уве-

личивает скорость передачи (в 4, 8, 16, ... раз), он обладает двумя существенными недостатками. Во-первых, увеличивается количество проводников, необходимых для прокладки сети, а это в свою очередь значительно удорожает стоимость. И, во-вторых, при высоких скоростях обмена электромагнитные волны, создаваемые током в проводнике, оказывают взаимное влияние на другие сигналы в соседних проводниках, в результате чего форма сигнала видоизменяется, и возникают проблемы декодирования, а, как следствие, достоверность передачи данных падает.

На физическом уровне для передачи данных может использоваться электрический сигнал (электрические кабели), оптический (оптоволокно) или радиоволны различной частоты (включая инфракрасный спектр). Передачу бит электрическим сигналом может осуществляться непосредственно или модулируя несущий сигнал. Среди непосредственных способов наиболее распространены методы NRZ, манчестерского кодирования и дифференциального манчестерского кодирования (рис. 3.1). Передача бит может осуществляться электрическим сигналом по току или по напряжению. При использовании напряжения могут применять асимметричный или симметричный (дифференциальный) способ передачи.

При модулировании сигнала используют амплитудную, фазовую, частотную модуляцию или их сочетание (рис. 3.2). В частности, модуляция применяется при передаче данных через оптоволокно или радиоволны.

Через любой канал связи, соединяющий два устройства, данные можно передавать с использованием одного из трех режимов:

- симплексного;
- дуплексного;
- полудуплексного.

Симплексная система – это такая система, которая приспособлена для передачи сообщений только в одном направлении.

Дуплексная система предназначена для передачи сообщений в обоих направлениях.

В *полнодуплексной системе* данные могут передаваться в обоих направлениях одновременно, а *полудуплексной* считается передача, при которой данные могут передаваться в обоих направлениях, но только в разное время.

Какой бы способ организации передачи данных ни использовался, передача данных подвержена воздействию помех, особенно в промышленных условиях, где могут присутствовать большие электромагнитные помехи, которые могут приводить к искажению получаемых данных.

Основным принципом обнаружения ошибок является вычисление передатчиком контрольного символа, основанное на содержимом оригинального сообщения. Этот символ посылается приемнику вместе с сообщением, и приемник повторяет те же самые вычисления с битами, которые он получает. Если вычисленный контрольный символ не соответствует пере-

данному символу, то можно предположить, что при передаче произошла ошибка.

Простейшей формой контроля за ошибками в асинхронных системах является добавление бита четности, который может быть четным или нечетным.

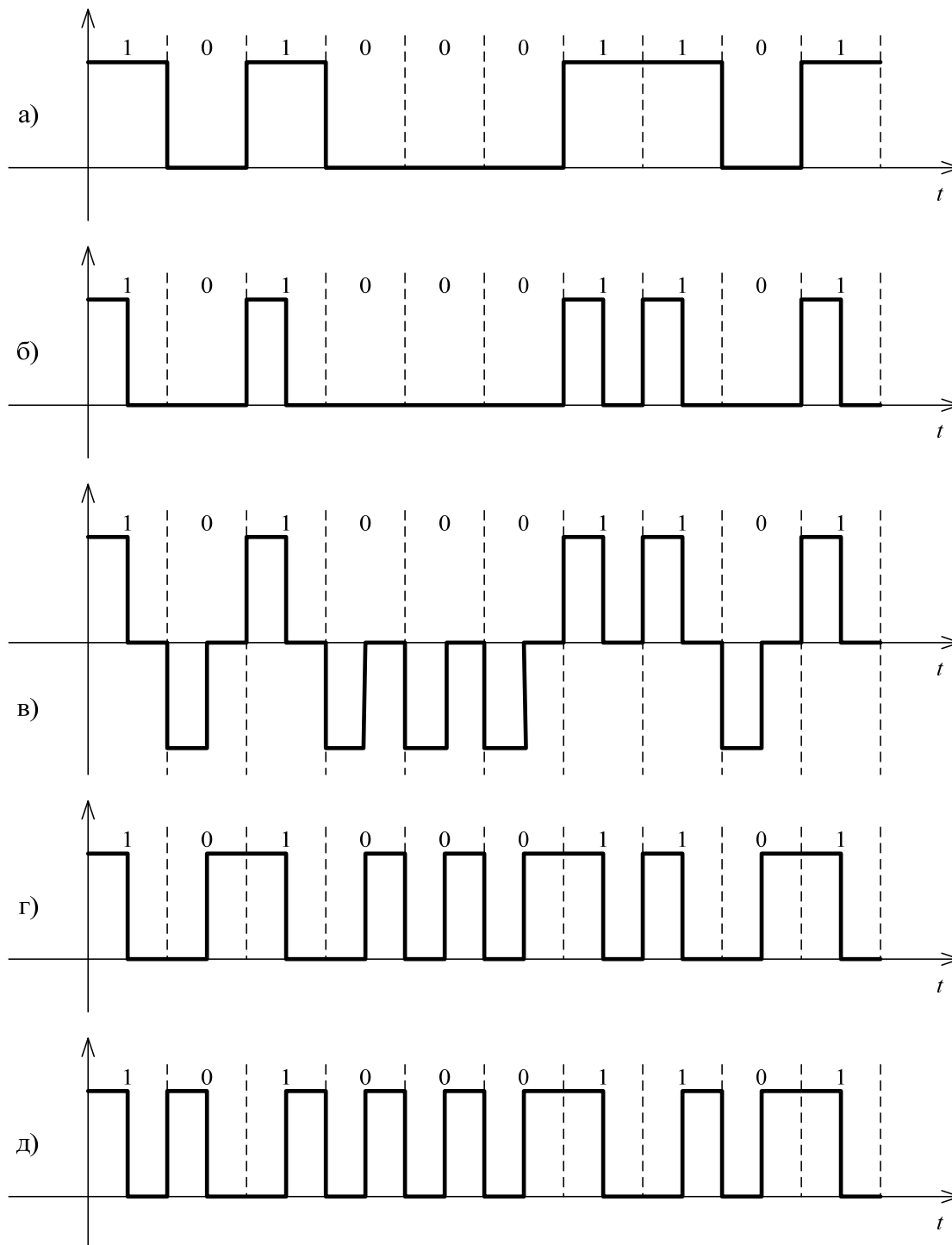


Рис. 3.1. Способы цифрового кодирования: а – прямое двоичное без возврата к нулю (NRZ), б – прямое двоичное с возвратом к нулю (RZ), в – трехуровневое с возвратом к нулю, г – манчестерское кодирование, д – дифференциальное манчестерское

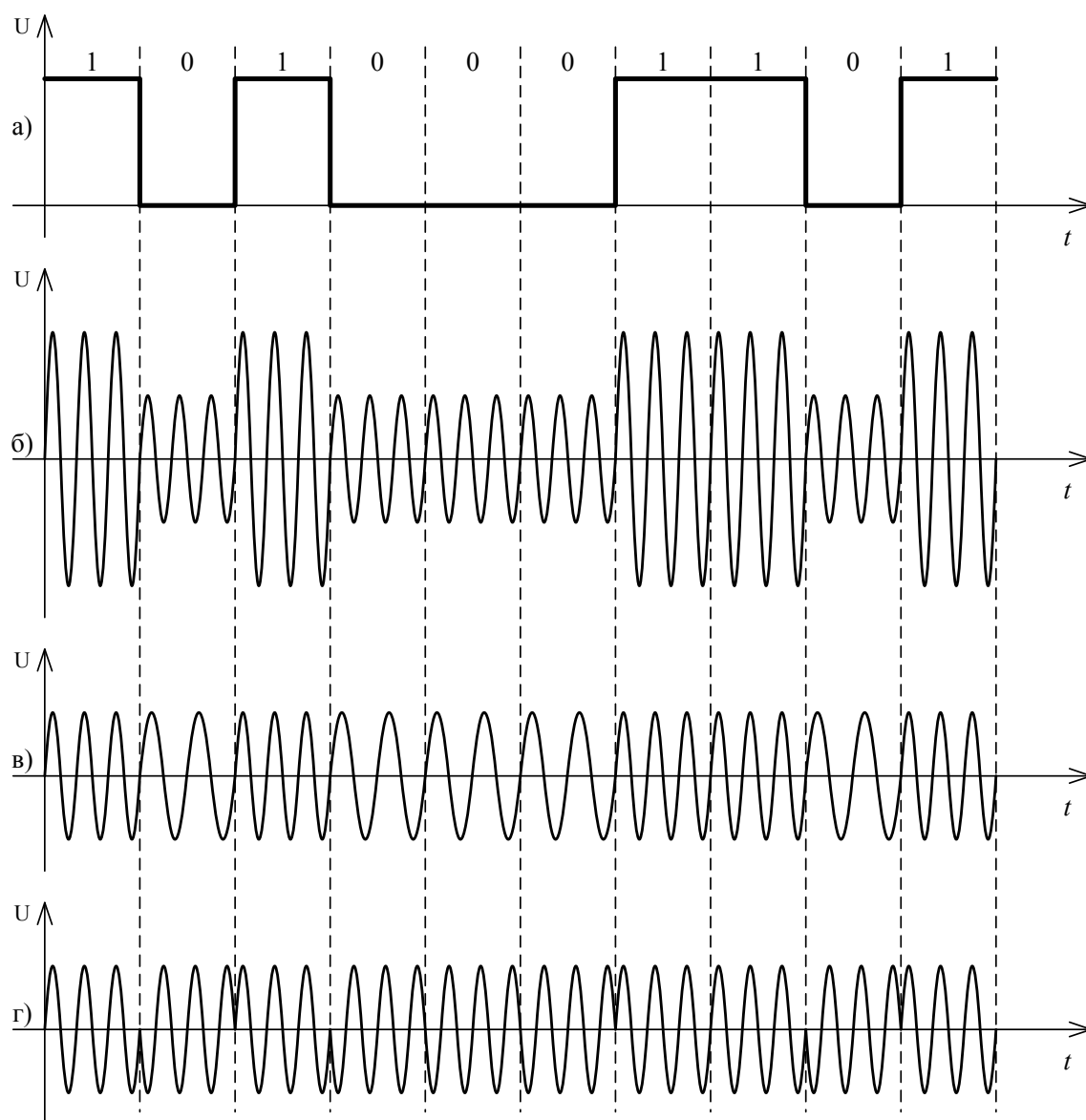


Рис. 3.2. Модуляция несущей: а – двоичный сигнал, б – амплитудная модуляция, в – частотная модуляция, г – фазовая модуляция

Проверка на четность требует, чтобы суммарное количество бит данных, являющихся логической единицей, плюс бит четности равнялось четному числу. Коммуникационное оборудование на передающей стороне вычисляет необходимую четность и устанавливает бит четности, чтобы обеспечить четное число логических единиц.

Проверка на нечетность работает аналогично проверке на четность, за исключением того, что бит четности устанавливается таким образом, чтобы полное количество логических единиц, включая бит четности, равнялось нечетному числу.

Статистически использование бита четности при высокой скорости передачи имеет вероятность обнаружения ошибки только 50%. Этот способ

позволяет обнаружить четное число ошибочных бит, и не позволяет обнаруживать нечетное число ошибочных бит. Вместо бита четности могут использоваться и более сложные алгоритмы контроля над ошибками.

3.3. Характеристики процесса передачи данных

К характеристикам процесса передачи относятся:

- скорость передачи сигналов через канал связи;
- скорость передачи данных;
- ширина полосы канала связи;
- отношение сигнал/шум;
- пропускная способность;
- частота появления ошибок.

Скорость передачи сигналов через канал связи или *символьная скорость* определяется тем, сколько раз физический сигнал изменится в секунду, и измеряется в бодах [6].

Скорость передачи данных или *скорость передачи бит* (*битовая скорость*) выражается в битах в секунду (бит/с) или в кратных единицах: кбит/с, Мбит/с и Гбит/с (кило-, мега- и гигабит в секунду). Эта скорость представляет фактическое число бит данных, передаваемых в единицу времени [6].

В настоящий момент времени существует постоянная путаница с применением этих терминов. Обусловлено это широким распространением передачи цифровых данных по проводным системам с использованием двоичного кодирования. Соответственно, на физическом уровне сигнал может принимать лишь два состояния, соответствующие логическому нулю и логической единице, что существенным образом отличается от популярного в недалеком прошлом использования модемов, где с помощью одного изменения сигнала может кодироваться более одного бита.

В простых системах передачи, скорость передачи сигнала равна скорости передачи бит (1 бод = 1 бит/с).

Ширина полосы обычно выражается в герцах (Гц) и представляет максимальную частоту, с которой может изменяться сигнал, когда ослабление еще не начинает ухудшать сигнал. Поскольку сигнал ослабевает с увеличением расстояния, то каналы связи могут потребовать установки через некоторый интервал повторителей (репитеров), которые необходимы для усиления затухающего сигнала.

Максимальная теоретическая скорость передачи данных определяется формулой Найквиста и включает ширину полосы и количество уровней, используемых для кодирования каждого элемента сигнала [7]:

$$C = 2B \log_2 M, \quad (3.1)$$

где M – количество дискретных сигналов или уровней напряжения, а B – ширина полосы, [Гц].

Если передаваемый сигнал является двоичным (т.е., например, имеет два уровня напряжения, $M = 2$), то скорость передачи данных, которая может поддерживаться при полосе B Гц, равна $2B$ бит/с. Таким образом, если скорость передачи сигнала равна $2B$, то с этой скоростью могут передаваться сигналы с частотами, не превышающими B .

Отношение сигнал/шум канала связи является еще одним важным фактором, ограничивающим скорость передачи. Оно представляет собой отношение мощности сигнала к мощности шума, присутствующего в определенный момент передачи.

$$SNR = \frac{P}{P_{noise}}, \quad (3.2)$$

где P – мощность сигнала, P_{noise} – мощность шума.

Как правило, данное отношение измеряется в приемнике, поскольку именно в этой точке предпринимается попытка обработать сигнал и устранить нежелательный шум.

Очень часто для удобства отношение сигнал/шум представляется в децибелах [7]:

$$SNR_{дБ} = 10 \lg \frac{P}{P_{noise}}, \quad (3.3)$$

Выражения (3.2) – (3.3) показывают превышение уровня сигнала над уровнем шума. Чем больше вычисленное значение, тем более качественный сигнал.

Максимальная реальная скорость передачи данных для коммуникационного канала математически связана с шириной полосы, отношением сигнал/шум и количеством уровней, с помощью которых кодируется каждый элемент сигнала. При уменьшении отношения сигнал/шум уменьшается и скорость передачи бит. Взаимосвязь этих параметров описывается законом Шеннона-Хартли [7]:

$$C = B \log_2(1 + SNR), \quad (3.4)$$

где C – пропускная способность канала [бит/с], B – ширина полосы [Гц].

Поскольку данные всегда передаются в протокольной оболочке, которая может быть кадром символа или представлять гораздо более сложные схемы передачи сообщения, скорость передачи данных всегда будет меньше скорости передачи битов. Количество избыточных данных, входящих в пакет сообщения, увеличивается по мере его прохождения вниз по стеку протоколов сети. Это означает, что отношение неинформативных данных к «реальной» информации может быть важным фактором в определении эф-

фективной скорости передачи, которая иногда называется *пропускной способностью*.

Вычисленная по формуле (3.4) скорость передачи является теоретическим максимумом и практически не достижима в реальных условиях. Одной из причин этого является то, что в формуле учитывается только белый (тепловой) шум и не учитываются импульсные помехи, амплитудные искажения или искажения, вызванные запаздыванием.

Частота появления ошибок связана с такими факторами, как отношение сигнал/шум, наличие шумов и помех.

Наличие шума может приводить к неверному декодированию передаваемых данных вследствие того, что шум, накладывается на цифровой сигнал, может существенно изменить форму передаваемых данных. На рис. 3.3 шум складывается из фоновых помех относительно умеренного уровня и случайных всплесков импульсных помех. Цифровую информацию можно восстановить из принимаемого сигнала путем дискретизации полученной формы сигнала, т.е. ее измерения через определенные промежутки времени и сравнения с пороговым значением. Как легко увидеть, случайного шума достаточно для изменения 1 на 0 или 0 на 1.

Таким образом, наличие шума может привести повреждению одного или нескольких бит. При увеличении скорости передачи продолжительность импульсов становится меньше, а значит, при том же самом шуме будет поражаться большее количество бит. Следовательно, чем выше скорость передачи данных при определенном уровне шума, тем выше уровень ошибок.

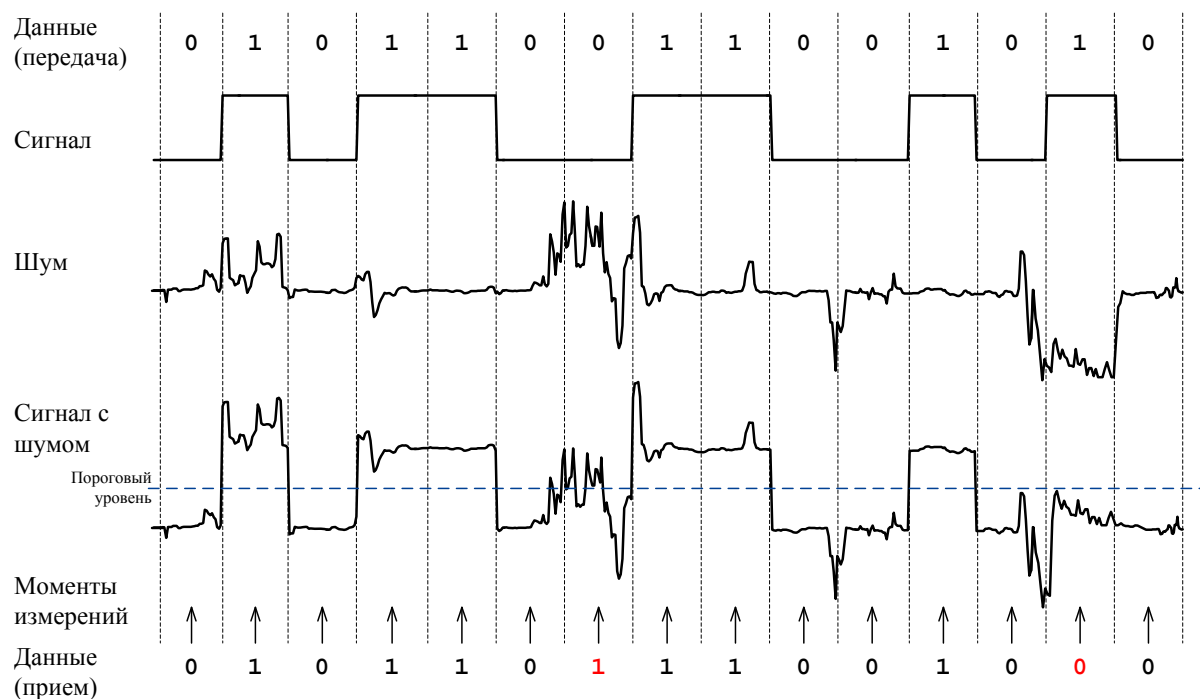


Рис. 3.3. Воздействие шума на цифровой сигнал

Обычно существует компромисс между скоростью передачи и допустимой частотой ошибок, зависящей от типа приложения. Как правило, промышленные системы управления не могут допустить наличие ошибок и поэтому разрабатываются с учетом максимальной надежности передачи данных. Это означает, что промышленная система будет сравнительно медленнодействующей в смысле передачи данных. По мере увеличения скорости передачи имеется точка, в которой количество ошибок становится слишком большим. Протоколы обрабатывают ошибки путем запроса повторной передачи пакетов. Очевидно, что количество повторных передач, в конце концов, достигнет того, что фактически высокая скорость передачи данных понижает реальную скорость передачи сообщений, поскольку будет требоваться много времени на повторные передачи сбойных сообщений.

3.4. Проводные линии связи

При выборе системы обмена данными важно определиться с типом линий передачи и использовать *симметричные* или *несимметричные* линии передачи данных.

В системе, использующей несимметричные линии передачи, общий опорный проводник сигнала одновременно разделяется многими сигналами и электронными схемами. Сигнал в виде напряжения передается только по одному проводу, который измеряется относительно общего провода, иногда называемого сигнальной «землей». Передаваемый сигнал является напряжением между сигнальным и общим проводом (рис. 3.4).

Теоретически несимметричная передача должна работать хорошо, если токи сигнала невелики, и общий провод имеет низкое сопротивление (импеданс). На практике несимметричные системы работают только в качестве коротких каналов связи. Общий провод сигнала имеет характеристики, аналогичные другим проводникам (сопротивление, индуктивность и емкость) и неидеален для использования в качестве базового проводника. Для длинных расстояний общий проводник может не иметь одинакового потенциала во всех его точках или на концах. На общий проводник могут также воздействовать помехи или паразитные напряжения. Иногда в качестве общего провода используется экранирующий проводник, а это также может приводить к очень большому уровню помех, чего следует избегать.

Тот факт, что на общий опорный проводник могут воздействовать различные помехи, означает, что к напряжениям, измеренным у приемника, могут добавляться наводки.

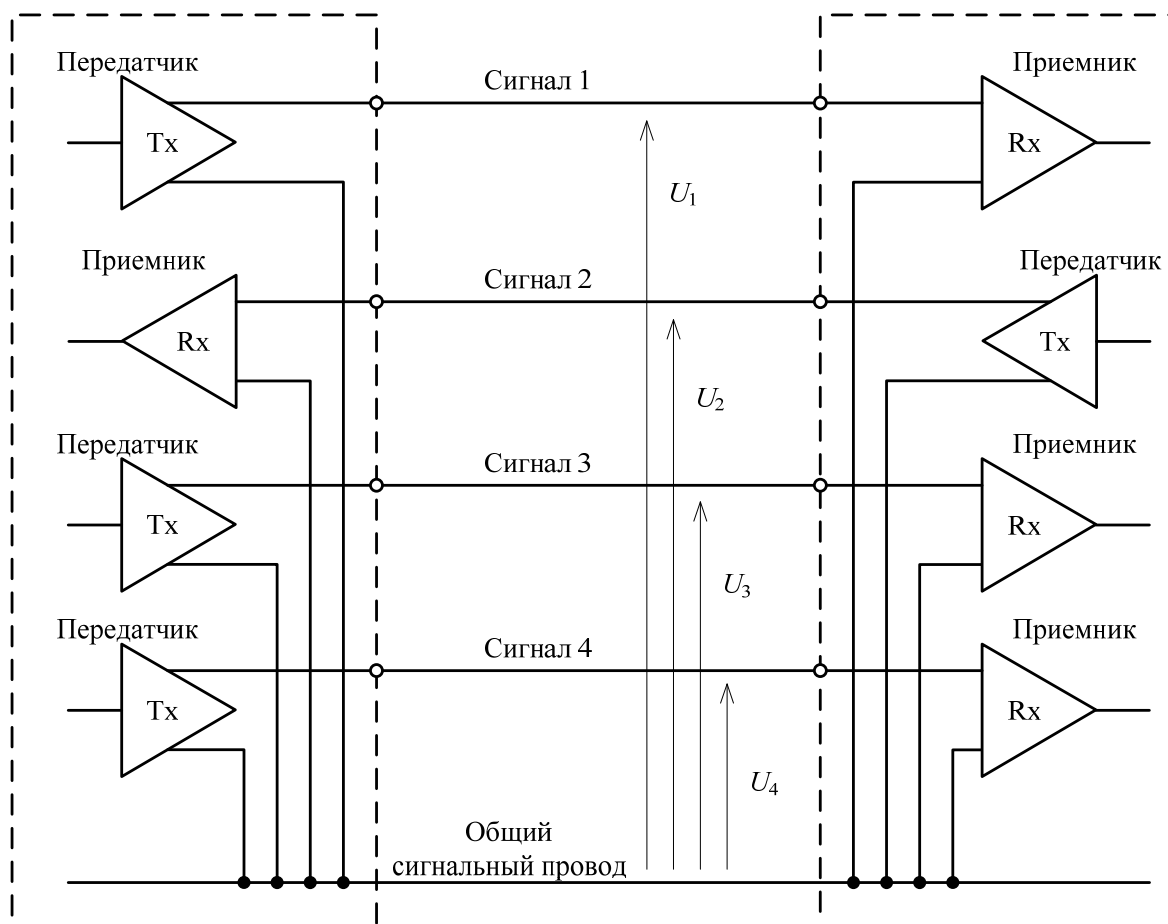


Рис. 3.4. Передача данных по несимметричному интерфейсу

Для симметричной передачи интерфейсы требуют для каждого сигнала по два проводника. Напряжение на принимающем конце является разностью напряжений между этими двумя проводами. Такая система передачи называется симметричной или *дифференциальной* и позволяет устранить многие проблемы наводок, связанные с использованием общего опорного провода (рис. 3.5).

Симметричная линия передачи обеспечивает более высокую скорость передачи данных на длинные расстояния. Дифференциальный способ передачи данных предпочтителен в промышленных условиях, где помехи могут быть основной проблемой. Недостаток этого способа заключается в увеличении необходимого количества проводов.

Успешная передача сигнала напряжения по двум проводам в присутствии помех или падения напряжения основана на том предположении, что эти провода имеют одинаковые характеристики, поэтому воздействие на них будет оказываться одинаковое. Это не означает, что помехи не влияют на симметричную дифференциальную систему. Напряжение на обоих проводах будет увеличиваться или уменьшаться одновременно, а разностный сигнал при этом остается одинаковым. Напряжение между сигнальным и общим проводниками называется *синфазным* напряжением.

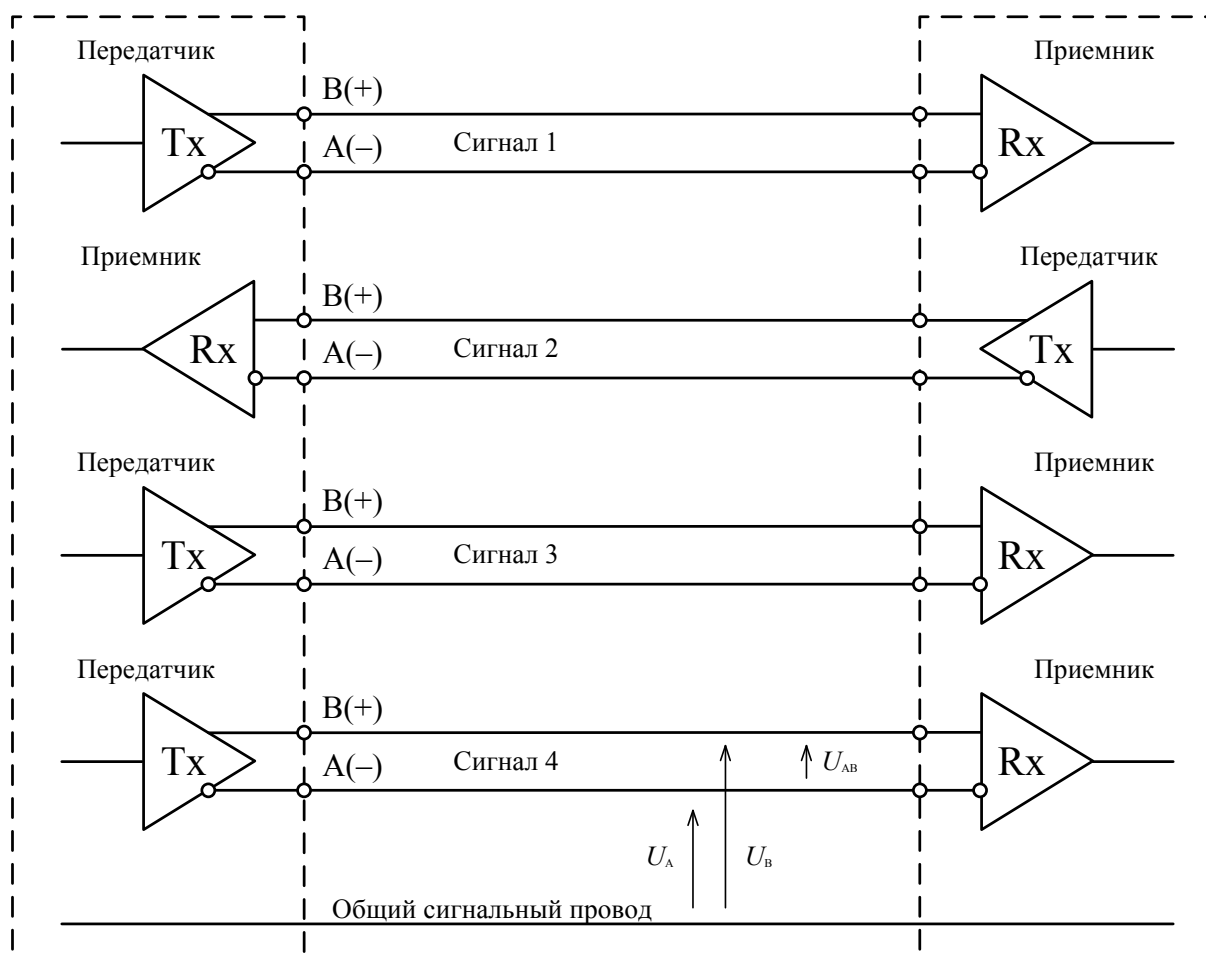


Рис. 3.5. Передача данных по симметричному интерфейсу

Синфазное напряжение является показателем наведенного напряжения или помех на канал связи. В идеальном случае синфазное напряжение на входе приемника будет полностью нивелировано. Однако, чем больше синфазное напряжение, тем выше вероятность искажения выходного напряжения или порчи устройства.

Приемники двухпроводной дифференциальной системы разрабатываются таким образом, чтобы игнорировать или ослаблять синфазное напряжение, используя специальные способы подавления синфазного сигнала. Влияние помех на сигнал измеряется отношением напряжения, пройденного через приемник, к синфазному напряжению. Степень подавления помех приемником измеряется *коэффициентом подавления синфазного сигнала*:

$$CMRR = 20 \log \left(\frac{dU_{\text{ВЫХ}}}{dU_{\text{СИН}}} \right). \quad (3.5)$$

Контрольные вопросы

1. Назовите уровень модели ВОС, который используется в обязательном порядке при построении промышленных сетей.
2. Посредством чего осуществляется перенос информации между оборудованием, участвующим в информационном обмене?
3. Какие компоненты должны находиться в составе устройств для осуществления связи между ним?
4. Укажите характеристики интерфейсов физического уровня, позволяющие оборудованию различных производителей осуществлять обмен данными.
5. В чем заключается преимущество использования последовательного способа организации передачи данных?
6. Какие способы кодирования данных существуют? В чем их особенность?
7. Назовите основные характеристики процесса передачи данных.
8. В чем заключается различие между битовой скоростью и символьной? В каких единицах они измеряются?
9. Какую зависимость определяет формула Найквиста?
10. Взаимосвязь каких параметров описывает закон Шеннона-Хартли?
11. В чем заключается различие между симметричной и несимметричной линией связи?

4. ПОСЛЕДОВАТЕЛЬНАЯ ПЕРЕДАЧА ДАННЫХ

4.1. Синхронная и асинхронная передача

Передавая данные любым из способов, представленных на рис. 3.1 и рис. 3.2, необходимо чтобы приемник осуществлял считывание сигнала в линии связи в определенные моменты времени, соответствующие передаче логического нуля или логической единицы. Если бы и передатчик, и приемник использовали один и тот же источник времени, то никаких сложностей в этом не возникало бы. Однако для реализации подобного способа потребовались бы дополнительные линии связи, передающие тактирующий сигнал, чтобы передатчик и приемник были синхронизированы. В противном случае, приемник не сможет определить, когда поступает первый бит, то есть распознать начало посылки.

Можно поступить иначе. При этом и передатчик, и приемник будут иметь в своем составе тактовый генератор, по сигналу с которого передатчик будет формировать данные в линию связи, а приемник – считывать значения. Но в этом случае возникает другая задача – синхронизация тактовых генераторов, так как даже при начальной синхронизации передатчика и приемника, а также договоренности о времени начала отправки, при

передаче больших последовательностей бит может возникнуть *рассинхронизация* (рис. 4.1), поскольку даже при, казалось бы, абсолютно одинаковых электрических параметрах компонентов тактовых генераторов передатчика и приемника всегда существуют отклонения этих параметров, которые приводят к тому, что с течением времени интервалы тактов в устройствах окажутся смещены, а, следовательно, посылка будет неправильно воспринята.

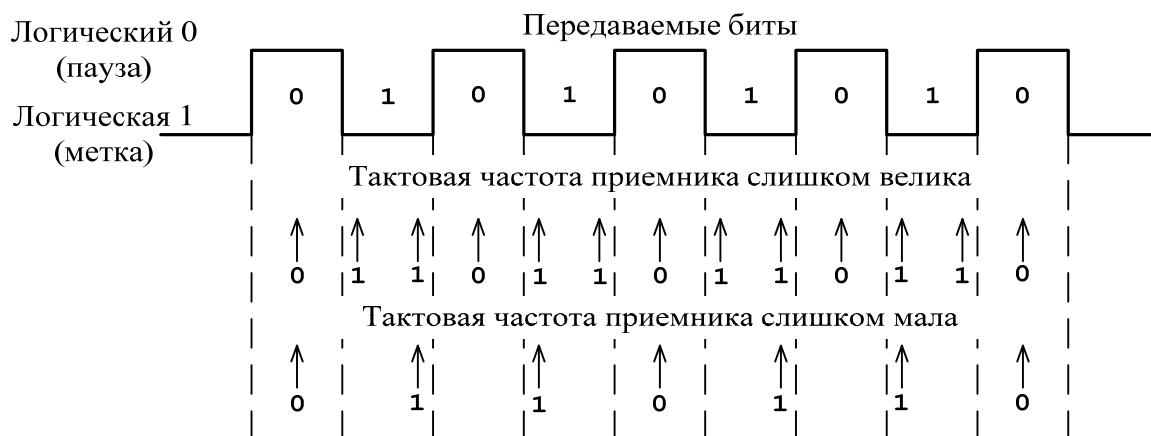


Рис. 4.1. Рассинхронизация передатчика и приемника

При *синхронной передаче* приемник и передатчик постоянно синхронизируются, а передача данных может начаться лишь по сигналу тактового генератора (соответственно, приемник также знает возможные моменты начала передачи). Синхронизация проходит через определенные промежутки времени или данных с помощью синхронизирующей посылки, которая представляет собой импульсный сигнал (набор импульсов) определенной частоты. Этот сигнал генерируется или передатчиком, или приемником, или отдельным устройством. Синхронизирующий сигнал может передаваться по специальному проводу (каналу), или вместе с цифровыми данными (например, в случае RZ-кодирования, при манчестерском кодировании и т.д.), или перед ними (например, в виде преамбулы). В любом случае организация синхронной передачи требует дополнительных затрат, так в первом варианте необходим вспомогательный провод (канал), во втором – вспомогательная полоса пропускания, в третьем – вспомогательные биты.

При *асинхронной передаче*, обмен данными может проходить и без определенного источника времени. Передача начинается в любой момент времени с сигнала начала передачи – *стартового бита*. В это время приемник синхронизируется с передатчиком. Обмен проходит короткими наборами бит (символами), что позволяет не делать дополнительную синхронизацию при передаче. При асинхронной передаче приемник должен

заранее знать все параметры связи, в первую очередь скорость, чтобы правильно идентифицировать поступающие сигналы.

4.2. Обмен данными в асинхронной системе

Одним из способов асинхронной передачи является *символьная передача*. При таком способе передаваемые данные делятся на *символы* по несколько бит и обрамляются служебными битами. Таким образом, *асинхронная система* – это система, в которой каждый символ или байт посылается в виде кадра. Приемник не начинает обнаружения данных до тех пор, пока он не примет первый бит – стартовый бит. Стартовый бит представляется напряжением, противоположным напряжению паузы (отсутствию передачи), и позволяет приемнику синхронизироваться с передатчиком для получения данных, содержащихся в кадре.

Пусть при отсутствии передачи в проводной линии связи содержится логическая единица, а сама передача данных осуществляется при помощи NRZ-кодирования. Использование активного сигнала, несущего некоторую энергию, для обозначения отсутствия передачи зачастую оказывается оправданным, так как позволяет различать ситуации отсутствия передатчика (разрыв линии связи) и отсутствия самой передачи. С началом передачи символа передатчик изменяет сигнал в линии связи на противоположные, то есть логический ноль, и держит его в течение времени, равному времени передачи одного бита. За это время приемник синхронизируется с передатчиком и готовится к приему бит данных, которые передаются сразу после стартового бита (количество бит данных может варьироваться от четырех до восьми, а сами биты передаются, начиная с младшего). Затем передается *бит четности* (бит паритета – parity bit), если он используется при обмене. Как отмечалось ранее, данный бит предназначен для контроля за правильностью передачи данных и выбирается из ряда: четный (even), нечетный (odd), отсутствует (none), всегда 1 (mark) и всегда 0 (space). При выборе трех последних вариантов физически не проводится контроль ошибок. За битом паритета (если он присутствует) идут *стоповые биты*, назначение которых – выдержать минимальную паузу между символами (рис. 4.2). Количество стоповых бит может устанавливаться как 1, 1.5 или 2. При этом указанные значения фактически означают «интервал времени, необходимый для передачи соответствующего количества бит данных». За стоповыми битами может сразу идти следующий символ, то есть стартовый бит следующего символа (рис. 4.3).

Приемник считывает отдельные биты кадра по мере их поступления, контролируя во времени напряжение логического нуля или логической единицы. Тактовая частота должна быть одинаковой у всех устройств, чтобы приемник прочитывал каждый бит именно в тот момент, когда передатчик посылает его. Поскольку тактовые частоты синхронизируются в

начале каждого кадра, то при низких скоростях передачи допускаются некоторые отклонения. По мере увеличения скорости передачи отклонения должны уменьшаться, а при высоких скоростях (свыше 100 кбит/с) асинхронная передача данных может сталкиваться с проблемами синхронизации.

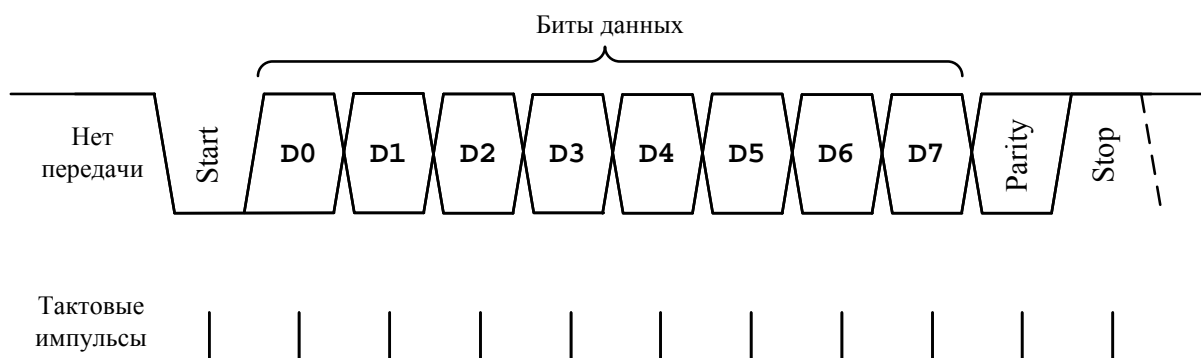


Рис. 4.2. Структура символа (формат кадра)

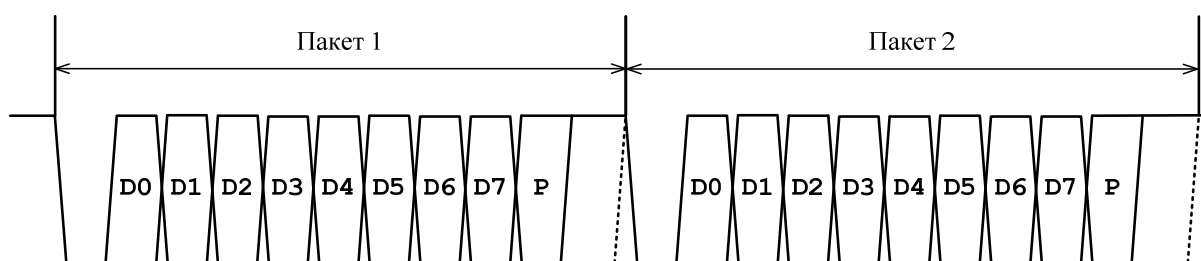


Рис. 4.3. Символьная передача

Передатчик и приемник должны быть настроены на одну и ту же конфигурацию (скорость, количество бит данных и стоповых бит, режим проверки четности), чтобы из кадра можно было корректно извлечь данные. Каждый символ имеет свой собственный кадр, причем эффективная скорость передачи данных получается меньше скорости передачи бит в секунду. Например, с учетом стартового бита, семи бит данных, одного бита четности и одного стопового бита для передачи семи бит данных всего необходимо передавать десять бит. Таким образом, передача полезных данных составляет 70% от битовой скорости.

Стартовые, стоповые биты и биты четности, используемые в асинхронной системе передачи, обычно реально генерируются стандартной интегральной микросхемой, которая является частью интерфейса между шиной микропроцессора и линейным формирователем (или приемником) канала связи. Такой тип микросхем называется *универсальный асинхронный приемопередатчик*.

4.3. Универсальный асинхронный приемопередатчик

Универсальный асинхронный приемопередатчик (Universal Asynchronous Receiver-Transmitter, UART) – это узел вычислительных устройств, предназначенный для связи с другими цифровыми устройствами, который преобразует заданный набор данных в последовательный вид так, чтобы было возможно передать их по однопроводной цифровой линии другому аналогичному устройству. На сегодняшний день UART представляет собой либо специализированную микросхему, которая является частью интерфейса между шиной процессора и передатчиком канала связи, либо UART является частью микроконтроллера. В любом случае, выход UART напрямую к линии связи не подключается, так как выдает сигнал, непригодный для передачи на большие расстояния. Чтобы выдавать и принимать уровни напряжений, соответствующие некоторому каналу связи, используются дополнительные устройства, называемые *линейными формирова-телями* и *линейными приемниками*, являющиеся частью различных интерфейсов последовательной передачи данных (например, RS-232 или RS-485).

При передаче UART последовательно выполняет следующие действия: устанавливает для передатчика битовую скорость, принимает от микропроцессора биты данных символа через параллельную шину, генерирует стартовый бит, последовательно передает биты данных, определяет и передает бит четности, передает стоповые биты, сигнализирует микропроцессору о готовности к обмену следующим символом, при необходимости квитирует данные для микропроцессора.

Во время приема UART: устанавливает для приемника битовую скорость, распознает стартовый бит, считывает биты данных, поступающие от приемника, считывает бит четности (и при необходимости осуществляет проверку на четность или нечетность), передает символ в параллельном коде в микропроцессор, при необходимости квитирует данные, проверяет их на наличие ошибок и при обнаружении устанавливает в регистре состояний флаг ошибки. Встречаются следующие типичные ошибки:

- *переполнение приемника* – байты принимаются быстрее, чем они могут быть прочитаны;
- *ошибка четности* – несоответствие бита четности;
- *ошибка кадра* происходит в том случае, если считанные биты не соответствуют выбранному формату;
- *ошибка разрыва* происходит, если стартовый бит действует в течение времени, большего времени кадра.

Для битовой скорости, с которой могут передаваться символы, существует ряд стандартных значений: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит/с. Это гарантирует, что устройства разных производителей смогут взаимодействовать друг с другом.

Как уже говорилось ранее, для того чтобы организовать передачу данных в асинхронной системе необходимо чтобы все приемопередатчики были настроены одинаково. Помимо скорости передачи данных должны быть зафиксированы: размер символа (количество бит данных), наличие бита паритета и способ его формирования, а также количество стоповых бит.

Настройки параметров передачи обычно обозначают в следующем порядке: скорость передачи, затем, количество бит данных, контроль бита паритета, количество стоповых бит. Например, 9600/8N2 или 19200, 7, E, 2.

Контроль бита паритета обозначается следующим образом:

- N (No parity) – бит четности отсутствует;
- E (Even parity) – дополнение до четного числа единиц;
- O (Odd parity) – дополнение до нечетного числа единиц;
- M (Mark parity) – бит четности всегда 1;
- S (Space parity) – бит четности всегда 0.

Для организации синхронной передачи могут использоваться *универсальные синхронные приемопередатчики* (Universal Synchronous Receiver-Transmitter, USRT). Помимо UART и USART существуют приемопередатчики, которые могут работать как в синхронном, так и асинхронном режиме – *универсальные синхронно-асинхронные приемопередатчики* (Universal Synchronous/Asynchronous Receiver-Transmitter, USART).

Контрольные вопросы

1. Объясните необходимость синхронизации.
2. В чем заключается различие между синхронной и асинхронной передачей данных?
3. Приведите структуру асинхронного пакета передачи данных.
4. Для чего необходим стартовый бит?
5. В какой последовательности передаются биты данных?
6. Какие ошибки приема может зафиксировать UART? В чем они заключаются?
7. С какой скоростью может осуществляться последовательная передача данных через UART?
8. Что означает, что передача данных осуществляется с 1.5 стоповыми битами?
9. Какие режимы паритета можно задать в UART?

5. ИНТЕРФЕЙСЫ ПОСЛЕДОВАТЕЛЬНОЙ ПЕРЕДАЧИ ДАННЫХ

5.1. Интерфейс RS-232 (TIA/EIA-232)

5.1.1. Общие сведения

В 1962 г. американская ассоциация электронной промышленности (Electronic Industries Association, EIA) представила стандарт интерфейса RS-232 для USART, который активно развивался, и в 1969 г. была выпущена третья редакция этого стандарта RS-232C. Префикс «RS», обозначающий «Recommended Standard» – рекомендованный стандарт, долгое время многих вводил в заблуждение, и ставил под сомнение само понимание термина «стандарт». В 1987 г. была выпущена четвертая редакция стандарта, устраняющая некоторые неоднозначности предыдущей редакции, и одновременно подчеркивающая роль EIA в части стандартизации – EIA-232D. Начиная с 1991 г. стандарты выпускаются EIA совместно с ассоциацией телекоммуникационной промышленности (Telecommunications Industry Association, TIA), и в 1997 г. издается действующая до настоящего времени (последняя ревизия в 2012 г.) шестая редакция TIA/EIA-232F. Несмотря на это, в профессиональной среде интерфейс «по старинке» называют RS-232.

Первоначальное назначение интерфейса RS-232 – подключение аппаратуры, передающей или принимающей данные (АПД, DTE – Data Terminal Equipment) к оконечной аппаратуре каналов данных (АКД, DCE – Data Communication Equipment) [8]. Полная схема соединения приведена на рис. 5.1. В качестве АПД может выступать компьютер, принтер, плоттер или любое другое периферийное оборудование, а в качестве АКД – модем.

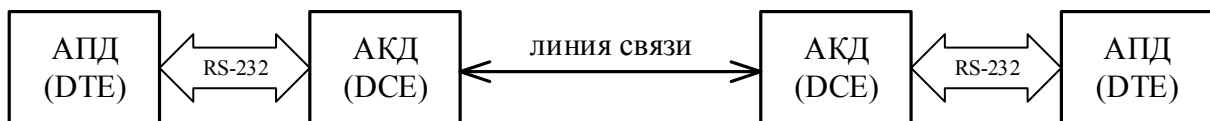


Рис. 5.1. Схема соединения оборудования по RS-232

В конечном итоге, приемопередающее оборудование взаимодействует между собой, несмотря на то, что подключение осуществляется не напрямую, а с использованием линии связи и АКД. В недалеком прошлом линии связи представляли собой коммутируемые телефонные линии связи, которые к настоящему времени практически не используются при построении промышленных сетей. Однако, беспроводные модемные соединения (например, на частоте 433 МГц, сети GPRS, 4G и т.д.) по-прежнему используют аналогичный принцип соединения устройств.

Практически более значимый способ использования интерфейса RS-232 на сегодняшний день заключается в непосредственном соединении АПД, исключая канал связи, и, соответственно, АКД, для чего используется нуль-модемный кабель (рис. 5.2).

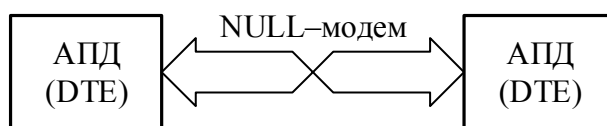


Рис. 5.2. Схема соединения нуль-модемным кабелем

Интерфейс RS-232 функционально эквивалентен многим другим стандартам, таким как: МККТТ (CCITT), V.24/V.28, X.20bis/X.21bis, ISO IS2110, «Стык C2».

Стандарт на интерфейс RS-232 определяют только электрические и механические параметры интерфейса, но не определяют протоколы связи.

5.1.2. Электрические характеристики

Интерфейс RS-232 проектировался для использования несимметричной (несбалансированной) линии передачи данных – сигнал передается относительно общего провода при помощи напряжения кодом NRZ. Все линии интерфейса должны выдерживать замыкание на любой другой контакт без какого-либо ущерба, а максимальное напряжение не должно превышать ± 25 В. Стандартом определено 25 линий электрических подключений, образующих четыре функциональные группы: линии данных, управляющие линии, линии синхронизации, дополнительные линии (таблица 5.1).

Таблица 5.1

№	Обозначение	Тип (для АПД)	Описание
1	PG	Общий	Protective Ground (Защитное заземление)
2	TD	Выход	Transmitted Data (Передача данных) По этой цепи в АПД передаются сигналы, вырабатываемые АКД: а) для передачи данных одной или нескольким удаленным станциям; б) поступающие в АПД с целью проверки со стороны АКД; в) для программирования или управления АПД в процессе последовательного автоматического вызова.
3	RD	Вход	Received Data (Прием данных) По этой цепи в АПД передаются сигналы, формируемые АКД в соответствии: а) с принимаемыми данными, полученными от удаленной станции; б) с проверочными сигналами АПД, а также ответы на сигналы (или «эхо» сигналов) программирования или управления, поступающие от АПД в процессе последовательного автоматического вызова.

Продолжение таблицы 5.1

4	RTS	Выход	<p>Request To Send (Запрос на передачу)</p> <p>Сигналы, передаваемые по этой цепи, управляют в АПД функцией передачи по каналу данных.</p> <p>При состоянии RTS=On АКД переводится в режим передачи по каналу данных.</p> <p>При состоянии RTS=Off АКД переводится в режим отсутствия передачи по каналу данных после того, как закончится передача всех данных, ранее переданных по цепи TD.</p>
5	CTS	Вход	<p>Clear To Send (Очищен для передачи)</p> <p>Сигналы, передаваемые по этой цепи, указывают на готовность АКД передавать данные по каналу данных.</p> <p>Состояние CTS=On указывает, что АКД готова к передаче данных по каналу данных.</p> <p>Состояние CTS=Off указывает, что АКД не готова передавать данные по каналу данных.</p>
6	DSR	Вход	<p>Data Set Ready (Набор данных готов)</p> <p>Сигналы, передаваемые по этой цепи, указывают на готовность АКД к работе.</p> <p>Состояние DSR=On (если цепь ТМ находится в состоянии Off или не используется) указывает на то, что устройство преобразования сигналов или аналогичное устройство подсоединено к линии связи и что АКД готова к взаимодействию по цепям управления для обмена данными.</p> <p>Состояние DSR=On при наличии состояния ТМ=On указывает на то, что АКД готова к взаимодействию по цепям управления для проведения проверки.</p> <p>Состояние DSR=Off при наличии в цепи CTS сигнала On указывает на то, что АКД готова осуществить обмен сигналами данных, связанных с программированием или управлением последовательным автоматическим вызовом.</p> <p>Состояние DSR=Off при наличии в цепи CTS сигнала Off указывает, что:</p> <ul style="list-style-type: none"> а) АКД не готова для передачи данных; б) обнаружено состояние неисправности, которое может быть в сети или в АКД; в) обнаружено разъединение линии от удаленной станции или от сети. <p>Состояние DSR=Off при наличии в цепи ТМ сигнала False указывает, что АКД участвует в проверке со стороны сети или удаленной станции.</p>
7	GND	Общий	<p>Ground (Общий)</p> <p>Цепь представляет собой общий обратный провод для несимметричных цепей RS-232 и устанавливает эталонный потенциал по постоянному току для симметричных цепей.</p>

Продолжение таблицы 5.1

8	DCD	Вход	<p>Data Carrier Detected (Обнаружение несущей)</p> <p>Сигналы, передаваемые по этой цепи, указывают, находится ли уровень принимаемого линейного сигнала канала данных в пределах, установленных соответствующими рекомендациями на АКД.</p> <p>Состояние DCD=On указывает, что уровень принимаемого сигнала соответствует установленным пределам.</p> <p>Состояние DCD=On может быть также во время обмена данными между АКД и АПД при программировании или управлении последовательным автоматическим вызовом.</p> <p>Состояние DCD=Off указывает, что уровень принимаемого сигнала не соответствует установленным пределам.</p>
9	—	—	зарезервирован
10	—	—	зарезервирован
11	—	—	не определен
12	SDCD	Вход	<p>Secondary Carrier Detect (Обнаружение несущей обратного канала)</p> <p>Эквивалентна цепи DCD с той лишь разницей, что она указывает, находится ли уровень принимаемого сигнала по обратному каналу в пределах, установленных соответствующими рекомендациями на АКД.</p>
13	SCTS	Вход	<p>Secondary Clear To Send (Очищен для передачи. Обратный канал)</p> <p>Эта цепь эквивалентна цепи CTS с той лишь разницей, что она указывает на готовность АКД передавать данные по обратному каналу.</p>
14	STD	Выход	<p>Secondary Transmitted Data (Обратный канал передачи данных)</p> <p>Эта цепь эквивалентна цепи TD с той лишь разницей, что она используется для передачи данных по обратному каналу.</p>
15	TST	Вход	<p>Transmitter Signal Timing DCE source (Передатчик сигналов синхронизации)</p> <p>Сигналы, передаваемые по этой цепи, обеспечивают в АПД синхронизацию единичных элементов сигнала.</p> <p>Состояния TST=On должны длиться в течение равных промежутков времени.</p> <p>АПД должно обеспечивать по цепи TD сигнал данных, в котором переходы между единичными элементами сигнала происходят в то же время, что и переходы из состояния On в состояние Off в цепи TST.</p>
16	SRD	Вход	<p>Secondary Received Data (Обратный канал приема данных)</p> <p>Эта цепь эквивалентна цепи RD с той лишь разницей, что она используется для приема данных по обратному каналу.</p>
17	RST	Вход	<p>Receiver Signal Timing (Приемник синхросигнала)</p> <p>Сигналы, передаваемые по этой цепи, обеспечивает в АПД синхронизацию единичных элементов сигнала.</p> <p>Состояния RST=On и RST=Off должны поддерживаться в течение равных промежутков времени. Переход из состояния On в состояние Off должен соответствовать середине каждого единичного элемента сигнала в цепи RD.</p>

Продолжение таблицы 5.1

18	LL	Выход	<p>Local Loopback (Локальная обратная петля)</p> <p>Сигналы, передаваемые по этой цепи, управляют состоянием проверки шлейфом в местном АКД.</p> <p>Состояние LL=On вызывает установление шлейфа в местном АКД.</p> <p>Состояние LL=Off вызывает снятие шлейфа в местном АКД.</p>
19	SRTS	Выход	<p>Secondary Request to Send (Запрос передачи по обратному каналу)</p> <p>Этот сигнал эквивалентен сигналу RTS с той лишь разницей, что он используется в АКД для управления функцией передачи по обратному каналу.</p>
20	DTR	Выход	<p>Data Terminal Ready (Готовность терминала данных)</p> <p>Сигналы, передаваемые по этой цепи, управляют подключением к линии связи или отключением от линии связи устройства преобразования сигналов или аналогичного ему устройства.</p> <p>Состояние DTR=On указывает, что АПД готова к работе, подготавливает АКД к подключению к линии связи устройства преобразования сигналов или аналогичного ему устройства, а также должно указывать на необходимость сохранения соединения, которое было установлено внешними средствами.</p> <p>При состоянии DTR=Off АКД должна отключить от линии связи устройство преобразования сигналов или аналогичное ему устройство после того, как закончится передача всех данных, ранее переданных по цепи TD или по цепи STD.</p> <p>Состояние DTR=Off также может быть использовано для прерывания или прекращения процедуры адресного способа установления соединения в устройстве автоматического вызова АКД.</p>
21	RL	Выход	<p>Remote Loopback (Эксплуатационная проверка)</p> <p>Сигналы, передаваемые по этой цепи, управляют состоянием эксплуатационной проверки.</p> <p>Состояние RL=On вызывает установление состояния проверки.</p> <p>Состояние RL=Off вызывает прекращение состояния проверки.</p>
22	RI	Вход	<p>Ring Indicator (Индикатор звонка)</p> <p>Сигналы, передаваемые по этой цепи, указывают, получен ли АКД сигнал вызова.</p> <p>Состояние RI=On указывает, что сигнал вызова принимается.</p> <p>Состояние RI=Off указывает, что вызов не принимается.</p> <p>Это состояние может также появляться во время прерываний импульсно-модулированного сигнала вызова.</p>
23	DSRS	Выход	<p>Data Signal Rate Selector (Переключатель скорости сигнала данных)</p> <p>Сигналы, передаваемые по этой цепи, служат для переключения скорости передачи данных в случае синхронной АКД, имеющей две скорости или для переключения диапазона скоростей передачи данных в случае асинхронной АКД, имеющей два диапазона скоростей.</p> <p>При состоянии DSRS=On АКД должна перейти на верхнюю скорость или верхний диапазон скоростей.</p> <p>При состоянии DSRS=Off АКД должна перейти на нижнюю скорость или на нижний диапазон скоростей.</p>

Окончание таблицы 5.1

24	TST	Выход	Transmitter Signal Timing DTE source (Передатчик сигналов синхронизации) Сигналы, передаваемые по этой цепи, обеспечивают в АКД синхронизацию единичных элементов сигнала. Состояния TST=On и TST=Off должны поддерживаться в течение равных промежутков времени. Переход из состояния TST=On в состояние TST=Off должен соответствовать середине каждого единичного элемента сигнала в цепи TD.
25	TM	Вход	Test Mode (Тестовый режим) Сигналы в этой цепи указывают, установлено ли состояние проверки в АКД. Состояние TM=On указывает, что АКД находится в состоянии проверки. Состояние TM=Off указывает, что АКД не находится в состоянии проверки.

Логической единице соответствует напряжение на входе приемника в диапазоне $-12...-3$ В относительно общего провода. Для линий управляющих сигналов это состояние называется «On», для линий последовательных данных – «Mark». Логическому нулю соответствует диапазон $+3...+12$ В (состояние «Space» для линий данных и «Off» для линий управляющих сигналов). Диапазоне от -3 до $+3$ В образует зону неопределенности, не соответствующую ни одному из логических сигналов. Решение о приеме логического нуля или же логической единицы должно приниматься только после пересечения указанных границ.

Так как во время передачи сигналов по проводам неизбежна потеря энергии, то стандартом на интерфейс RS-232 устанавливаются большие напряжения для выходов передатчиков: $-15...-5$ В для логической единицы и $+5...+15$ В для логического нуля при любой нагрузке передатчика, лежащей в диапазоне от 3 кОм до 7 кОм, и $-25...-5$ В и $+5...+25$ В соответственно для ненагруженной линии. неопределенный логический уровень передатчика от -5 В до $+5$ В.

Линии данных используются для передачи данных, а управляющие линии – для интерактивного управления устройством. Они управляют потоком данных через интерфейс.

Между допустимой скоростью передачи данных и длиной кабеля, соединяющих два устройства по интерфейсу RS-232, имеется определенная связь. По мере увеличения скорости передачи данных форма передаваемых данных начинает в большой степени зависеть от емкости и индуктивности кабеля, причем емкость кабеля увеличивается с увеличением его длины. Фактически на длину кабеля накладывается ограничение, определяемое количеством ошибок данных, допустимых при передаче. Стандарты EIA-232-D и TIA/EIA-232-E определяют в качестве предельной величины ем-

кость кабеля 2500 пФ. Исходя из погонной емкости кабеля в 160 пФ/м, получается максимальная длина 15 м, а если воспользоваться кабелем с погонной емкостью 50 пФ/м, то 50 м.

Наиболее распространенными скоростями передачи данных для интерфейса RS-232 являются 110, 300, 600, 1200, 2400, 4800, 9600, 19200 и 115200 бит/с. Таблица 5.2 ориентировочно показывает взаимосвязь между выбранной скоростью передачи данных и максимально допустимой длиной кабеля.

Таблица 5.2

Скорость передачи, бит/с	Длина кабеля, м
110	850
300	800
600	700
1200	500
2400	200
4800	100
9600	70
19 200	50
115 200	20

Кабели большей длины можно использовать только при низких скоростях передачи данных. Следует отметить, что максимально достижимая скорость зависит от напряжений передатчика, емкости кабеля, а также от наличия помех.

5.1.3. Механические характеристики

В стандарте RS-232-C определены механические характеристики интерфейса RS-232 между АПД и АКД. В частности определены типы применяемых разъемов. Наиболее распространенными являются DA-15, DB-25, DC-37, DD-50 и DE-9 (рис. 5.3). На АПД устанавливаются вилки, а на АКД – розетки.

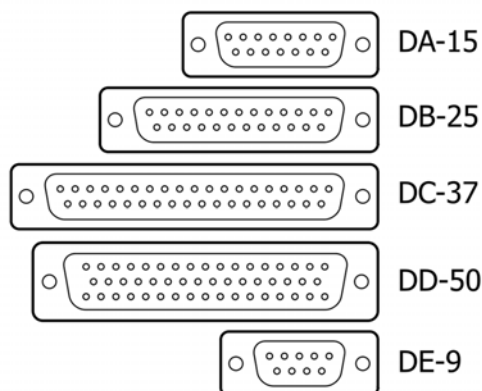


Рис. 5.3. Стандартные разъемы интерфейса RS-232

Разъем типа DE-9 не может коммутировать все сигналы, необходимые для работы интерфейса RS-232 в синхронном режиме, и применяется только в случае UART. Именно такой вариант на сегодняшний день стал стандартом «де-факто» в промышленности. Для подключения АПД к АКД используется прямой кабель, а для соединения двух АПД – нуль-модемный (перекрестный) кабель (рис. 5.4).

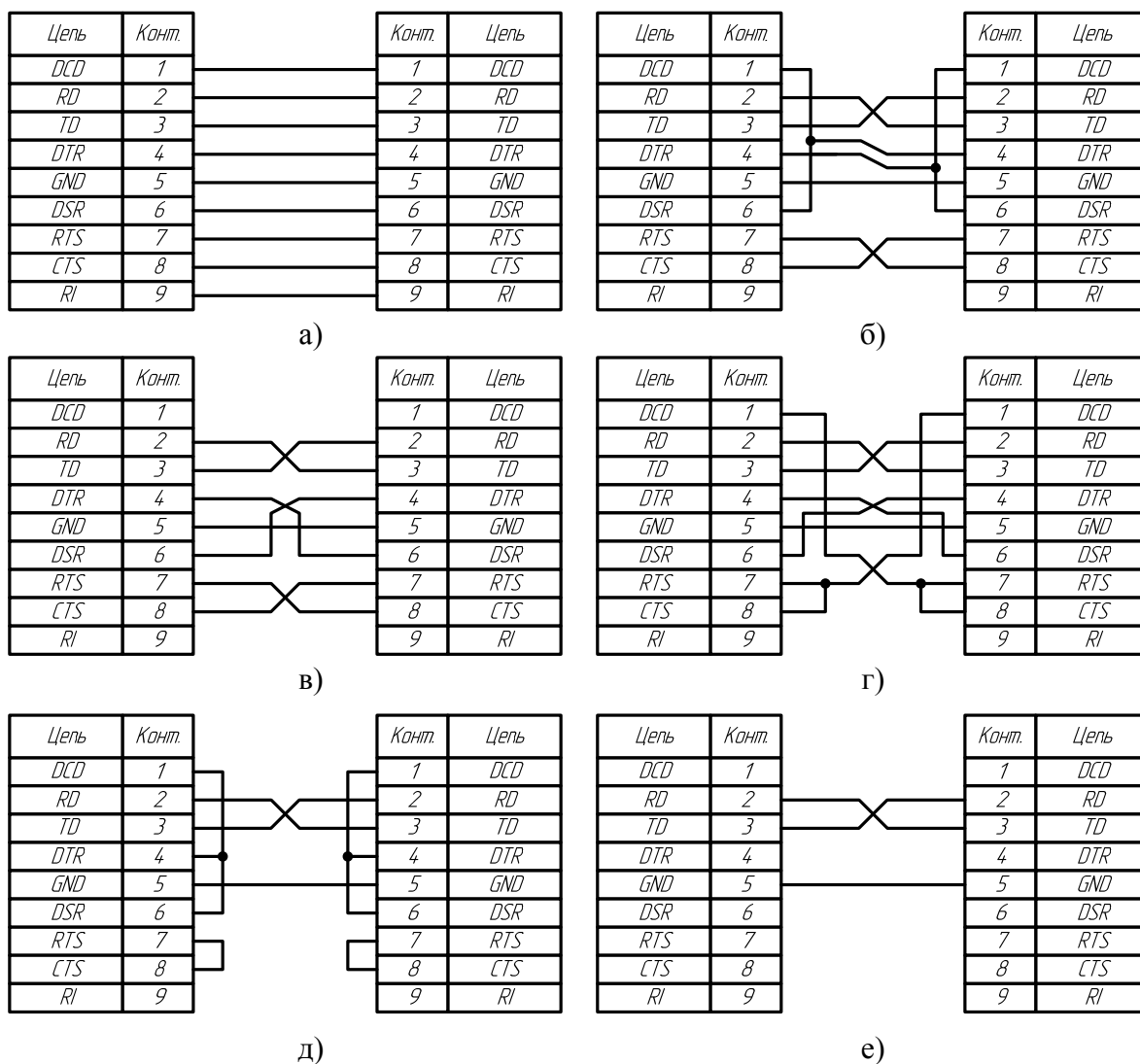


Рис. 5.4. Схемы кабеля RS-232 для DE-9: а – прямой, остальные – перекрестный:
б и в – с полным управлением потоком, г – с частичным управлением потоком,
д – петля, е – без управления потоком

5.1.4. Управление потоком данных

Интерфейс RS-232 в асинхронном режиме может управлять потоком передачи данных. Управление потоком необходимо в том случае, если принимающий узел не успевает обрабатывать входящий поток данных. Он сигнализирует об этом передающему узлу, который прекращает отправ-

лять данные до тех пор, пока не получит на это разрешения от принимающего узла.

Существует два варианта управления потоком данных: аппаратный и программный.

Аппаратный протокол управления потоком реализуется при помощи пары сигналов RTS/CTS или же DTR/DSR. Если ориентироваться на названия сигналов, то на текущий момент пара DTR/DSR явно устарела, и более информативной выглядит пара DTR/DCD [9]. Один из вариантов перекрестных кабелей, представленных на рис. 5.4 учитывает это. А в целом поведение принимающего и передающего узлов определяется идентичным образом независимо от того какой способ соединения двух АПД используется.

Рассмотрим аппаратное управление потоком данных двух АПД, соединенных перекрестным кабелем, на примере сигналов RTS/CTS (рис. 5.5).

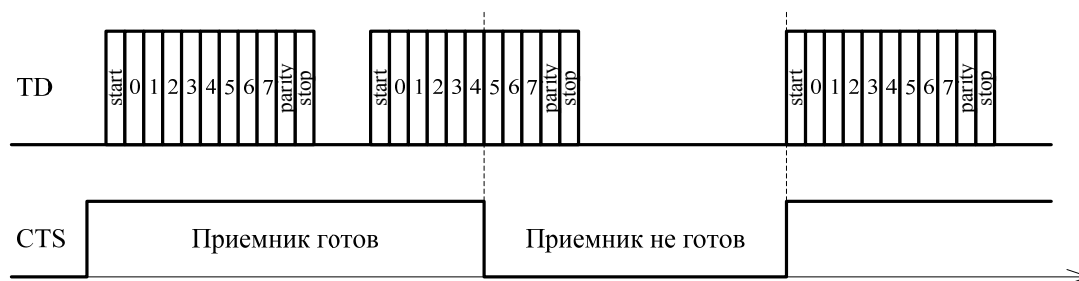


Рис. 5.5. Аппаратное управление потоком (RTS/CTS)

Изначально сигнал CTS определял готовность АКД к приему данных от АПД, для которого сигнал CTS является входным, и значение «On» сигнала CTS разрешает его работу – данные передаются через линию TD. Каждый символ передается в линию связи только, если работа передатчика разрешена (CTS=On). Если же АКД не готова к получению сигнала, то установив каким-либо образом значение «Off» сигнала CTS можно приостановить работу передающего узла. Так как непосредственное подключение двух АПД при помощи перекрестного кабеля не предполагает наличия АКД, то эквивалентом сигнала о том, что устройство может посылать данные является сигнал RTS, поэтому в перекрестном кабеле сигнал RTS на одной стороне соединяется с CTS на другой. В силу того, что данные передаются последовательно бит за битом, может возникнуть такая ситуация, что сигнал CTS изменит свое состояние в тот момент, когда часть бит, образующих очередной передаваемый символ, была уже отправлена в линию связи. В этом случае передатчик будет продолжать посылать данные до тех пор, пока все биты текущего символа не уйдут в линию связи – байт, который уже начал передаваться, задержать сигналом CTS невозможно, что гарантирует целостность посылки.

Аппаратный протокол обеспечивает самую быструю реакцию передающего узла на состояние принимающего узла, но при этом требует дополнительных линий связи.

Программный протокол управления потоком данных может применяться даже в том случае, когда количество проводников в линии связи минимально, но достаточно для организации дуплексного режима.

Если принимающий узел не справляется с потоком данных, поступающим к нему от передающего узла (сигнал TD передающего узла), то такой узел может отправить специальный байт-символ XOFF (ASCII код 13₁₆), который поступит на вход RD передающего узла, и он в свою очередь приостанавливает отправку данных. Для того чтобы передача данных возобновилась, принимающий узел должен отправить символ XON (ASCII код 11₁₆). Время реакции передающего узла на изменение состояния принимающего по сравнению с аппаратным протоколом увеличивается, по крайней мере, на время передачи управляющего символа (XON или XOFF), а также время реакции передающего узла на прием символа (рис. 5.6). Соответственно, данные будут передаваться без потерь лишь в том случае, если принимающий узел имеет дополнительный входной буфер и сигнализирует о заполнении его до некоторого критического предела, например в 90 %.

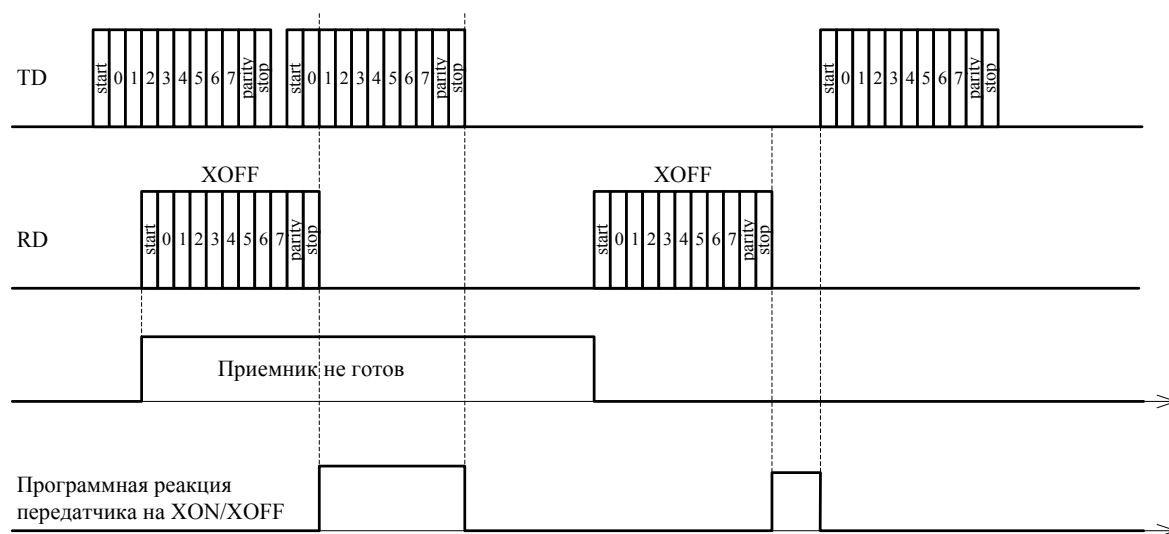


Рис. 5.6. Программное управление потоком (XON/XOFF)

5.2. Интерфейс RS-423 (TIA/EIA-423)

5.2.1. Общие сведения

Стандарт RS-423 определяет несимметричный интерфейс для передачи данных, аналогичный RS-232, но содержащий некоторые улучшения. Он позволяет увеличить длину кабеля между устройствами (до 1200 м), увеличить скорость передачи данных (до 100 кбит/с) и использовать несколь-

ко приемников на одной линии (до 10). Таблица 5.3 показывает практическую связь между выбранной скоростью передачи данных и длиной кабеля. Функциональная схема подключения устройств с интерфейсом RS-423 приведена на рис. 5.7.

Таблица 5.3

Длина кабеля, м	Скорость, кбит/с
9	100
91	10
120	1

Способность поддерживать до 10 приемников достигается увеличением нагрузочной способности линейных формирователей, уменьшением потребляемого тока входных приемников и подключением приемников в дифференциальном режиме. Как и в случае использования интерфейса RS-232, канал связи является несимметричным, и для полной дуплексной передачи требуется как минимум три проводника.

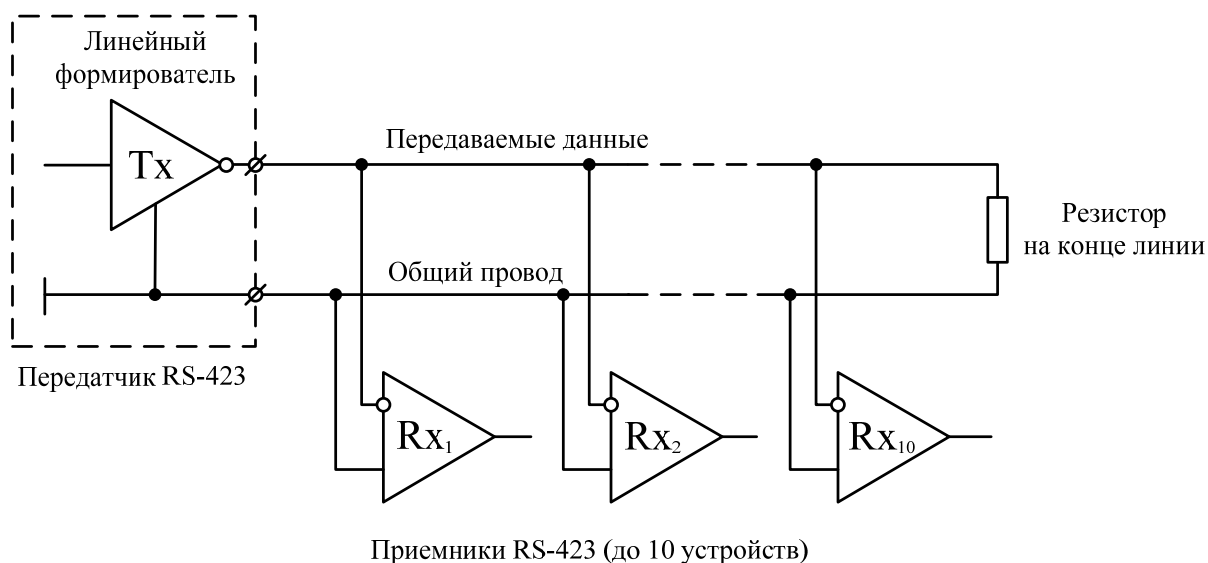


Рис. 5.7. Функциональная схема включения RS-423

5.2.2. Электрические характеристики

Интерфейс RS-423 использует следующие уровни напряжения для кодирования двоичных состояний линии передачи данных:

- Логическая 1 (Mark): от -3.6 В до -6.0 В;
- Логический 0 (Space): от $+3.6$ В до $+6.0$ В.

5.2.3. Механические характеристики

Стандарт интерфейса RS-423 не определяет механические подключения или номера контактов. На практике обычно используется разъем DB-25, с назначением контактов аналогично интерфейсу RS-232.

5.3. Интерфейс RS-422 (ANSI/TIA/EIA-422)

5.3.1. Общие сведения

Интерфейс RS-422 является улучшенным вариантом интерфейса RS-423, определяет симметричный (дифференциальный) интерфейс обмена данными, использующий для каждого сигнала по два отдельных провода, и позволяет:

- передавать данные на расстояние до 1200 м (аналогично RS-423);
- использовать скорости передачи до 10 Мбит/с (увеличена в 100 раз);
- подключать к одному линейному формирователю до 10 приемников.

Функциональная схема подключения устройств с интерфейсом RS-422 представлена на рис. 5.8, а зависимость длины линии связи от скорости передачи – на рис. 5.9.

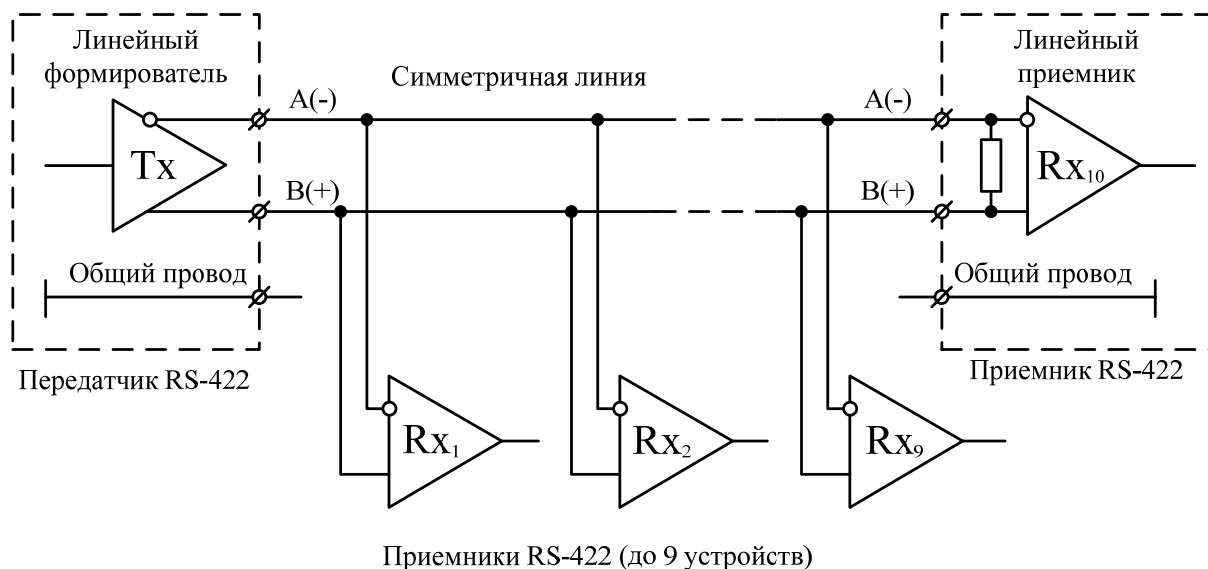


Рис. 5.8. Функциональная схема включения RS-422

Стандарт на интерфейс передачи данных RS-422 появился в начале 1970-х годов, и на сегодняшний день действует его вторая редакция (Revision B), опубликованная в мае 1994 г. и утвержденная Американским национальным институтом стандартов (ANSI). Эта редакция подтверждена TIA в 2005 г. Полное наименование стандарта – ANSI/TIA/EIA-422-B «Электрические характеристики схем дифференциального интерфейса сбалансированного напряжения». Существует международный эквивалент этого стандарта – ITU-T Recommendation V.11 или X.27. Однако, как и в случае с RS-232, чаще всего используется более короткое «историческое» название RS-422.

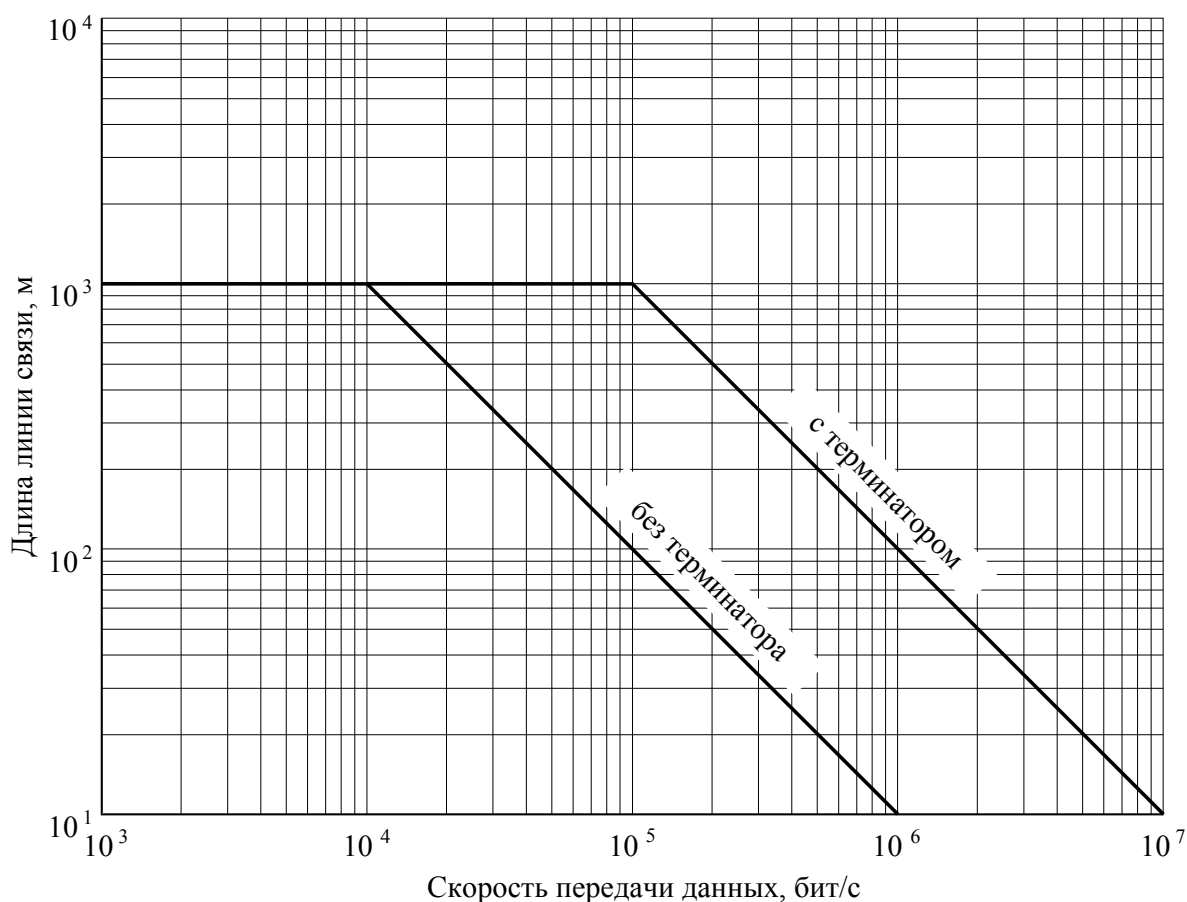


Рис. 5.9. Протяженность линии связи RS-422 в зависимости от скорости передачи

5.3.2. Электрические характеристики

Основной особенностью стандарта RS-422 является использование дифференциальных сигналов, которые позволяют увеличить скорость и помехозащищенность. Каждый сигнал передается по одной паре проводов и представляет разность напряжений между ними. Поскольку земля больше не используется в качестве опорного провода, отпадают проблемы, связанные с различными земляными потенциалами.

Поскольку дифференциальный приемник чувствителен только к разнице напряжений двух сигналов на его входах, то сигналы помех, воздействующие на оба провода, будут сказываться на работе приемника незначительно.

Спецификация определяет линии, как «А» и «В», но их также называют как «DATA-» и «DATA+».

Дифференциальные напряжения между линиями «А» и «В» имеют следующие параметры:

- Логическая 1 (Mark): от -0.2 В до -6 В, ($-6 \text{ В} < U_a - U_b < -0.2 \text{ В}$);

- Логический 0 (Space): от +0.2 В до +6 В, ($0.2 \text{ В} < U_a - U_b < 6 \text{ В}$).

Передатчик должен обеспечивать уровень сигнала 1.5 В при максимальной нагрузке (32 стандартных входа и 2 терминальных резистора) и не более 6 В без нагрузки. На стороне приемника минимальный уровень принимаемого сигнала должен быть не менее 200 мВ.

Для уменьшения уровня помех предпочтительнее использовать общий земляной провод. Таким образом, для симплексного режима требуются три провода, а для дуплексного – пять проводов.

5.3.3. Механические характеристики

Стандарт RS-422 не определяет механические подключения или назначение контактов разъемов. Такие параметры определяются другими стандартами. Например, EIA-530 определяет использование разъема DB-25, а EIA-449 – разъема DC-37. В случае соединения двух АПД возможно применение разъема DE-9. Один из вариантов назначения контактов разъема DE-9 для интерфейса RS-422 с возможностью аппаратного управления потоком данных приведен в таблице 5.4, однако производители оборудования могут использовать и иное назначение контактов. Например, в таблице 5.5 приводится назначение контактов того же разъема DE-9, используемое в устройствах NPort, UPort, MGate фирмы MOXA.

Таблица 5.4

Контакт	Цепь
1	Экран
2	Передаваемые данные В +
3	Принимаемые данные В +
4	Запрос передачи В +
5	Сброс передачи В +
6	Передаваемые данные А –
7	Принимаемые данные А –
8	Запрос передачи А –
9	Сброс передачи А –

Таблица 5.5

Контакт	Цепь
1	TxD – (A)
2	TxD + (B)
3	RxD + (B)
4	RxD – (A)
5	GND
6	–
7	–
8	–
9	–

5.4. Интерфейс RS-485 (ANSI TIA/EIA-485)

5.4.1. Общие сведения

Стандарт ANSI TIA/EIA-485 Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems (Электрические характеристики передатчиков и приемников, используемых в сбалансированных цифровых многоточечных системах) был утвержден в 1998 г. Последнее к текущему моменту времени подтверждение стандарта датировано 2003 г. Разработчиками стандарта являются Electronics Industries Asso-

ciation (EIA) и Telecommunications Industry Association (TIA). Аналогично рассмотренным ранее стандартам, несмотря на все усилия идентифицировать происхождение стандарта, среди разработчиков наиболее популярным остается RS-485.

Интерфейс RS-485 является расширением стандарта RS-422 и поддерживает те же расстояния (см. рис. 5.9) и скорости передачи, но допускает большее количество передатчиков и приемников, подключаемых к одной линии. Существует две версии интерфейса RS-485: двухпроводная (рис. 5.10) и четырехпроводная (рис. 5.11). В любом случае используется дифференциально-симметричный канал связи, как и в интерфейсе RS-422.

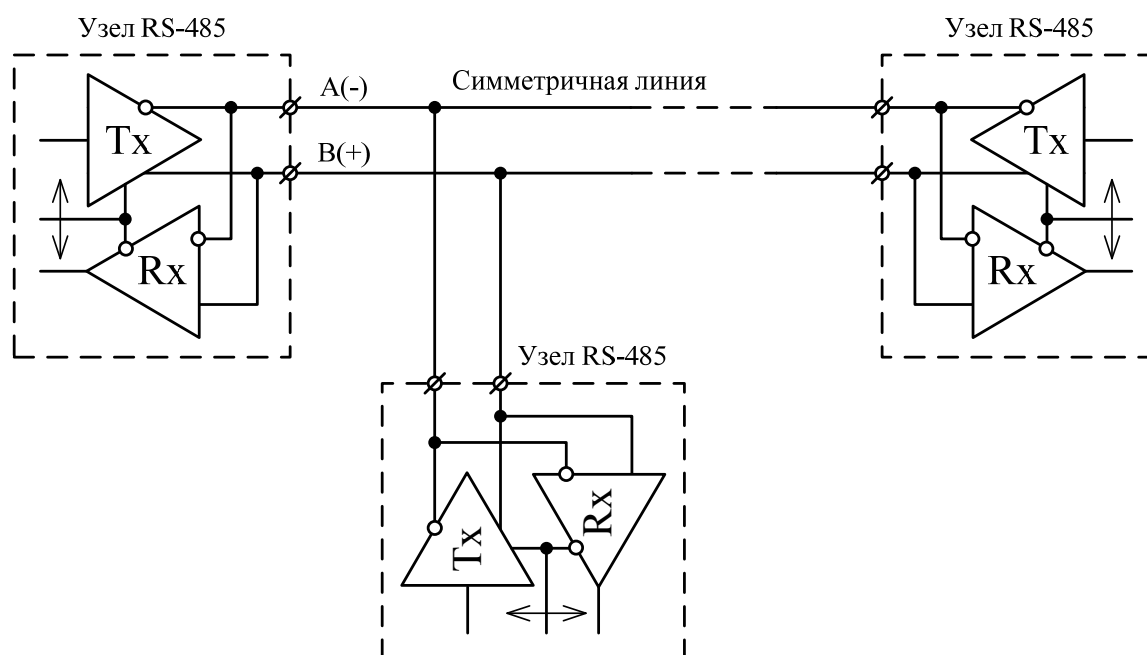


Рис. 5.10. Двухпроводная схема включения устройств с интерфейсом RS-485

Интерфейс RS-485 позволяет обеспечить:

- расстояние до 1200 м;
- скорость передачи до 10 Мбит/с;
- подключение до 32 линейных формирователей на одну линию;
- подключение до 32 приемников на одну линию.

Возможность подключения до 32 передатчиков к линии связи обусловлена тем, что в отличие от интерфейса RS-422, линейный формирователь передатчика может работать с тремя состояниями: логический нуль, логическая единица и высокоимпендансное (высокоомное) состояние, в котором формирователь фактически отключается от линии за счет дополнительных цепей управления.

Наличие в двухпроводной сети RS-485 нескольких потенциальных передатчиков может спровоцировать коллизию в передаче данных, поэтому в таких сетях необходимо разграничение доступа узлов к линии связи на основе какого-либо метода.

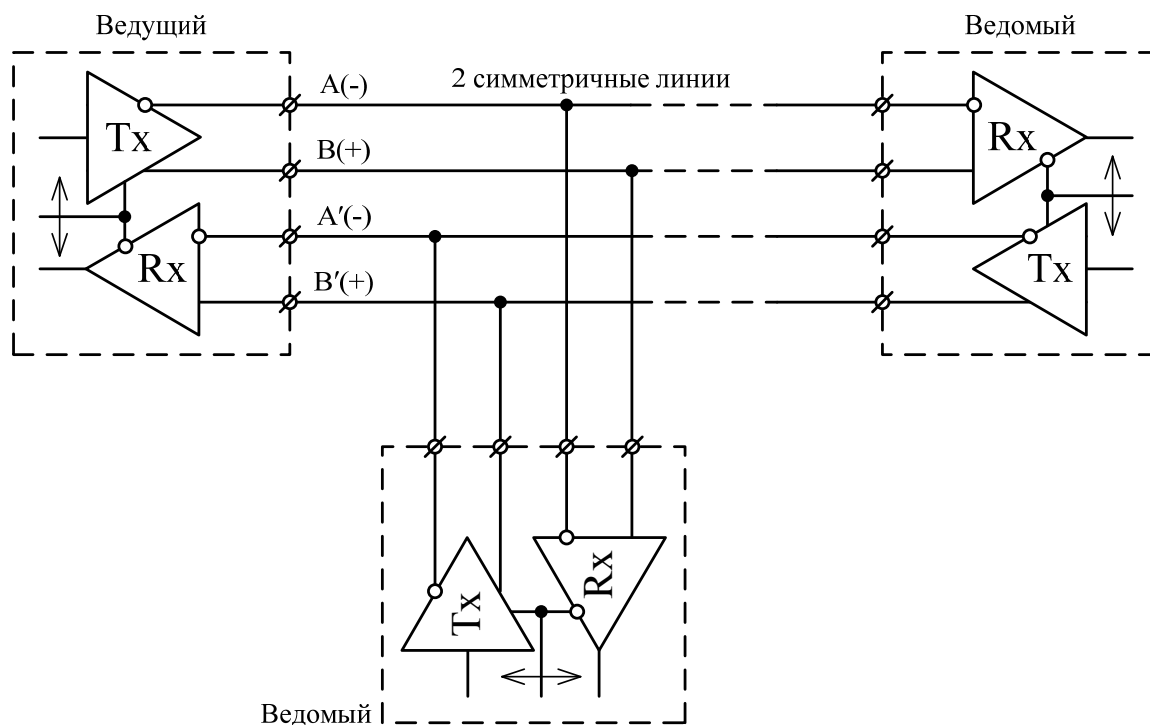


Рис. 5.11. Четырехпроводная схема включения устройств с интерфейсом RS-485

Четырехпроводная схема интерфейса RS-485 требует, чтобы один из узлов был ведущим (фиксированный мастер), а все остальные – ведомыми. Ведущий узел производит обмен данными со всеми ведомыми, но ведомый узел может общаться только с ведущим. В силу особенности подключения, ведомые узлы никогда не получают сообщений от других ведомых, а только от ведущего узла, а потому не может ответить некорректно другим ведомым узлам. Это является достоинством, которое особенно выгодно проявляется при работе со смешанными протоколами на базе интерфейса RS-485. Другим преимуществом четырехпроводной схемы RS-485 является дуплексный режим – возможность одновременной передачи и приема данных, что бывает необходимо при реализации некоторых сложных протоколов обмена.

К недостаткам четырехпроводной схемы следует отнести необходимость явного указания ведущего и ведомых устройств на стадии проектирования системы, в то время как в двухпроводной схеме любое устройство может выступать как в роли ведущего, так и ведомого, а также увеличенное количество проводников.

5.4.2. Электрические характеристики

Аналогично стандарту на интерфейс RS-422, спецификация на интерфейс RS-485 именуется проводники дифференциальной линии передачи дан-

ных как «А» и «В» и устанавливает те же самые уровни напряжений в линии связи:

- Логическая 1 (Mark): от -0.2 В до -6 В, ($-6 \text{ В} < U_a - U_b < -0.2 \text{ В}$);
- Логический 0 (Space): от $+0.2$ В до $+6$ В, ($0.2 \text{ В} < U_a - U_b < 6 \text{ В}$).

Линия «В» положительна (по сравнению с «А»), когда передается логическая единица, и в самом стандарте линия «А» называется «инверсной», а линия «В» соответственно – «не инверсной», но ряд производителей интегральных схем считают иначе, и используют прямо противоположное обозначение сигналов, что привносит существенную путаницу. Поэтому использование обозначений А/В фактически дискредитировало себя, и более предпочтительным является явное указание полярности сигнала – «DATA +» и «DATA –» или «D +» и «D –». В тех случаях, когда используется четырехпроводная схема включения, отдельно указывают сигналы линии передатчика (TxD +/TxD –) и приемника (RxD +/RxD –) узла сети.

Следует отметить, что интерфейсы RS-485 и RS-422 электрически совместимы, а передатчики RS-422/485 совместимы с приемниками RS-423. Схема соединения двух узлов интерфейса RS-485 по четырехпроводной схеме («точка – точка») и схема соединения двух узлов интерфейса RS-422 эквивалентны.

5.4.3. Механические характеристики

Стандарт RS-485 не определяет механические характеристики интерфейса. Аналогично RS-422, подключения или назначение контактов разъемов определяются другими стандартами, применяемыми совместно с RS-485, или же назначаются производителями оборудования. В таблице 5.6 приводится назначение контактов разъема DE-9, используемое в устройствах NPort, UPort, MGate фирмы MOXA для случая использования двухпроводной и четырехпроводной схемы включения, а в таблице 5.7 – для разъема RJ-45 (8p8c), используемого в устройствах NPort 5630 той же фирмы.

Таблица 5.6

Контакт	Цепь	
	4-х пров.	2-х пров.
1	TxD – (A)	–
2	TxD + (B)	–
3	RxD + (B)	DATA + (B)
4	RxD – (A)	DATA – (A)
5	GND	GND
6	–	–
7	–	–
8	–	–
9	–	–

Таблица 5.7

Контакт	Цепь	
	4-х пров.	2-х пров.
1	–	–
2	–	–
3	TxD + (B)	–
4	TxD – (A)	–
5	RxD – (A)	DATA – (A)
6	RxD + (B)	DATA + (B)
7	GND	GND
8	–	–

5.4.4. Практические аспекты построения сети

При проектировании сети RS-485 следует учитывать ряд весьма важных факторов: количество передатчиков и приемников, скорость передачи данных, способ объединения устройств и максимальную протяженность линии связи.

Нагрузочная способность

В силу того, что в сети RS-485 к одному передатчику может быть подключено несколько приемников, полезная энергия формирователя тратится на каждый из них. Стандартом на интерфейс RS-485 установлено максимально допустимое число приемников (32 приемника) и минимальное входное сопротивление каждого из них (12 кОм). Для простоты выполнения расчетов принято использовать понятие *единица нагрузки* (UL, Unit Load). Одна единица нагрузки соответствует приемнику с импедансом примерно 12 кОм.

Современные приемопередатчики часто имеют пониженную нагрузку, соответствующую $\frac{1}{2}$ UL, $\frac{1}{4}$ UL и даже $\frac{1}{8}$ UL, что позволяет увеличить количество узлов сети соответственно в два, четыре и восемь раз, расширяя их количество с 32 до 256. При этом допускается любая комбинация типов приемопередатчиков при условии, что суммарная нагрузка не превышает 32 UL.

Согласование линии

Интерфейс RS-485 (как и RS-422) позволяет осуществлять передачу данных на значительные расстояния при высоких скоростях обмена. В случае, когда время распространения сигнала становится соизмеримым с временными характеристиками передаваемых двоичных импульсов, линию связи необходимо рассматривать как длинную линию с распределенными параметрами, для которой характерны эффекты отражения. Электромагнитная волна, достигая конца линии связи, отражается от него и возвращается к источнику сигнала, отражается от источника и опять проходит к концу линии [4]. Считается, что лишь три цикла прохождения волны оказывают существенное влияние на форму сигнала, так как из-за потерь амплитуда сигнала с каждым разом уменьшается. На рис. 5.12 приведена осциллограмма сигналов, снятая на выходе формирователя, передающего импульсы со скоростью 200 кБит/с, а на рис. 5.13 – на входе приемника. Несмотря на то, что форма импульсов существенно искажена, дифференциальный способ передачи данных позволяет правильно декодировать передаваемые данные.

Чтобы исключить влияние отраженной электромагнитной волны на форму сигнала, необходимо минимизировать эффект отражения – согласовать линию связи, то есть линия связи должна быть нагружена на сопротивление, равное волновому сопротивлению кабеля:

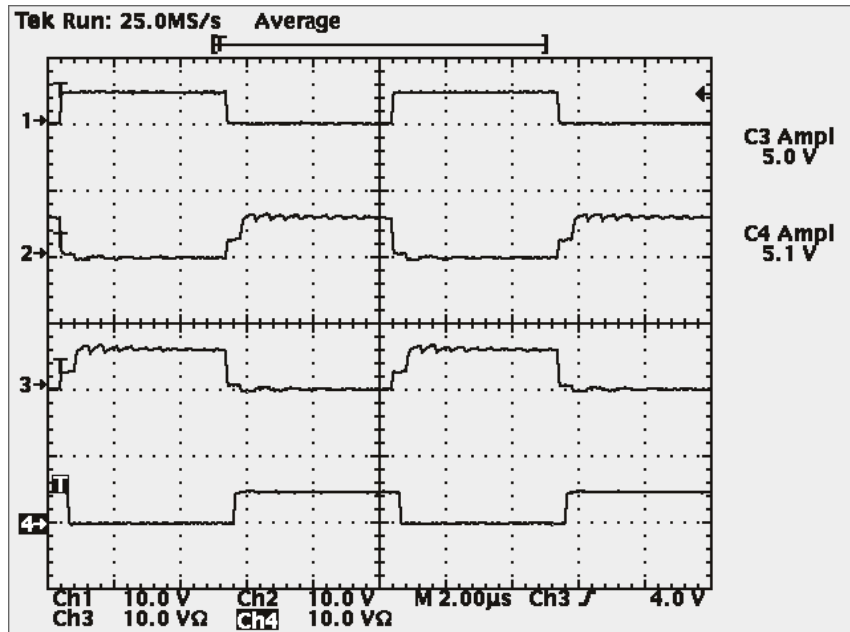


Рис. 5.12. Осциллограмма сигналов передатчика: 1 – передаваемый сигнал, 2 – напряжение в линии A(-), 3 – напряжение в линии B(+), 4 – инверсный выход приемника

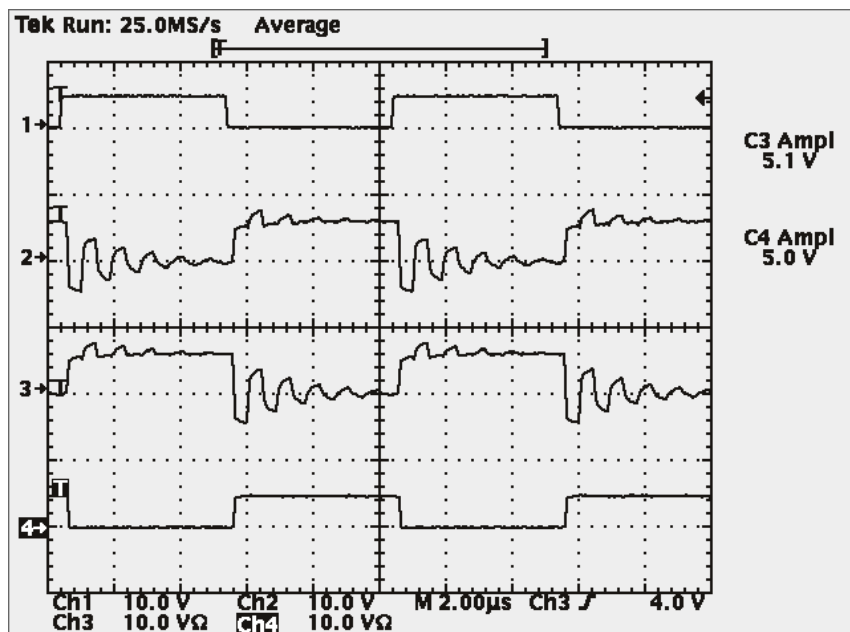


Рис. 5.13. Осциллограмма сигналов приемника: 1 – передаваемый сигнал, 2 – напряжение в линии A(-), 3 – напряжение в линии B(+), 4 – инверсный выход приемника

$$Z_0 = \sqrt{\frac{R_0 + j\omega L_0}{G_0 + j\omega C_0}}, \quad (5.1)$$

где R_0, L_0, G_0, C_0 – погонные сопротивления, индуктивность, проводимость и емкость кабеля соответственно, $j\omega$ – комплексная круговая частота. Для

согласования линии используются терминальные (концевые) резисторы с сопротивлением R_T . Если $R_T = Z_0$, то коэффициент отражения по напряжению

$$K_U = \frac{R_T - Z_0}{R_T + Z_0} \quad (5.2)$$

обращается в нуль, что соответствует идеальному согласованию линии.

Промышленность выпускает кабели, предназначенные для организации связи по интерфейсу RS-485 с волновым сопротивлением 120 Ом, а производители приемопередатчиков также ориентируется на это значение, хотя сам стандарт на интерфейс RS-485 не устанавливает жестких требований на этот счет, и могут применяться кабели с волновым сопротивлением в диапазоне от 100 до 150 Ом.

Для интерфейса RS-485 терминальные резисторы должны устанавливаться на противоположных краях кабеля (рис. 5.14), а в случае интерфейса RS-422 – только на стороне самого удаленного приемника.

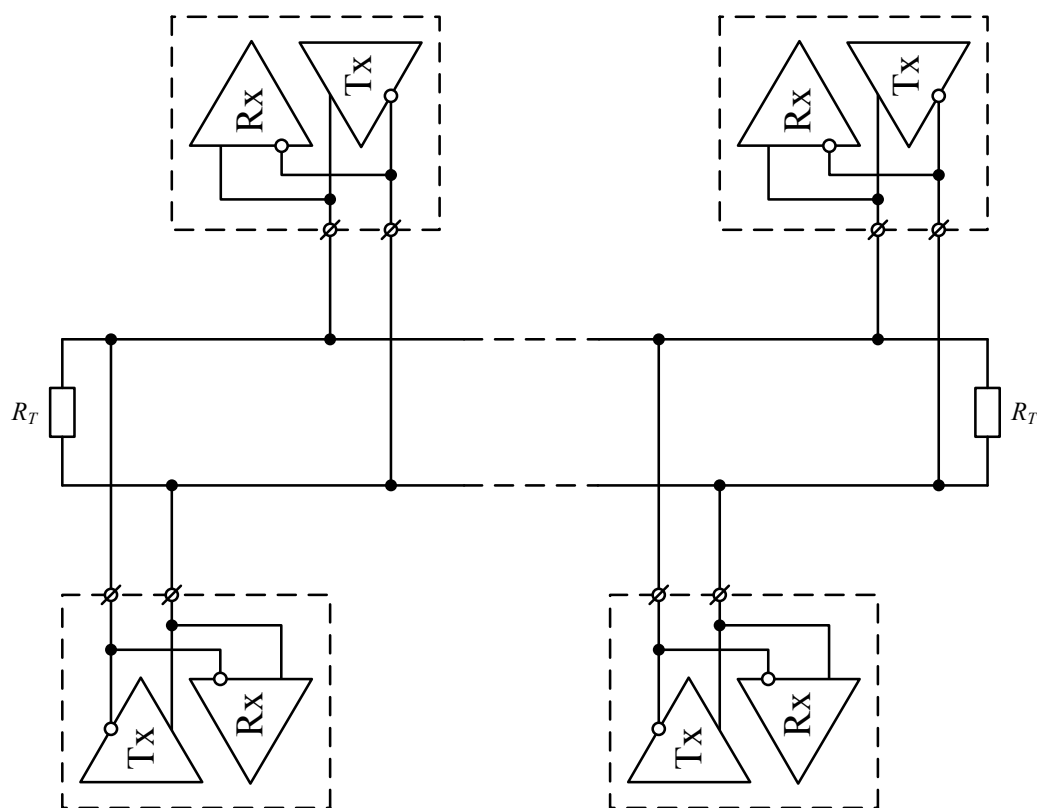


Рис. 5.14. Применение терминальных резисторов для согласования линии передачи

При эксплуатации сети RS-485 в условиях сильных электромагнитных помех каждый из терминальных резисторов рекомендуется заменить двумя последовательно соединенными вдвое меньшего номинала, образующими фильтр нижних частот (рис. 5.15), что обеспечивает дополнительное подавление синфазной помехи. В этом случае необходимо также обеспечить

минимальный разброс параметров, чтобы обеспечить одинаковую частоту среза обоих фильтров.

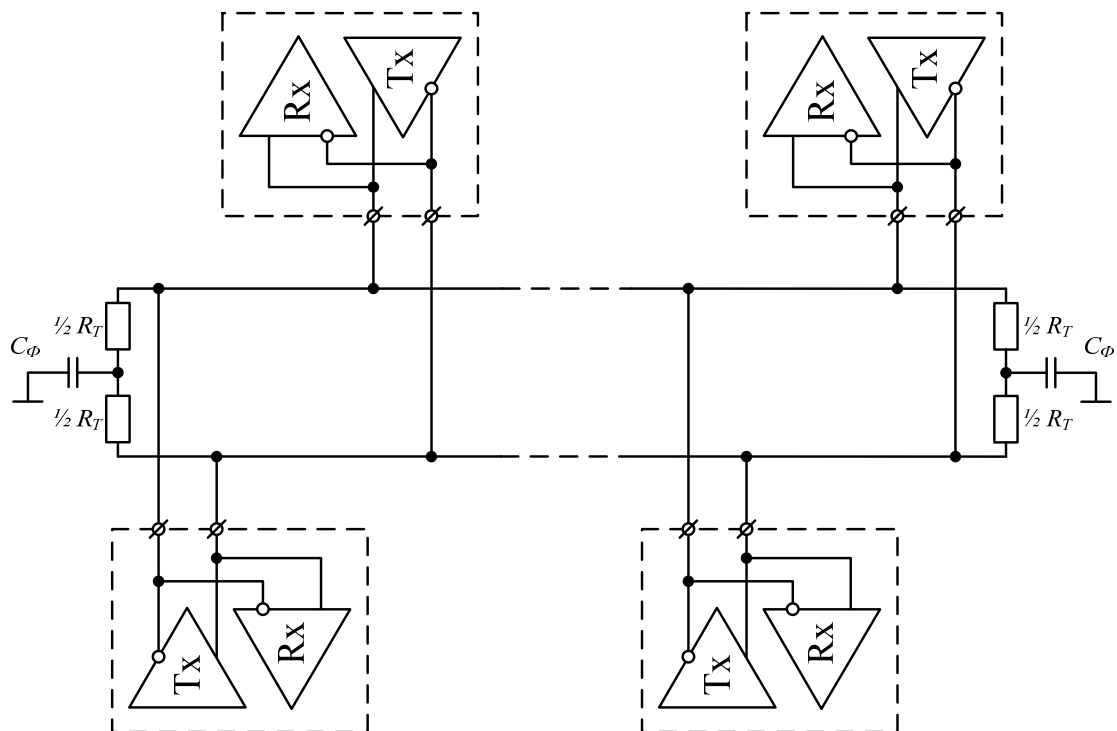


Рис. 5.15. Применение ФНЧ для согласования линии передачи

Так как для RC-фильтра нижних частот частота среза определяется выражением

$$f_0 = \frac{1}{2\pi RC}, \quad (5.3)$$

то значение C_ϕ легко рассчитать, зная частоту следования импульсов f ($f < f_0$).

Однако способы, показанные на рис. 5.14 и 5.15 обладают существенным недостатком – передатчики интерфейса оказываются нагруженными по постоянному току на нагрузку $\frac{1}{2}R_T$, что приводит к дополнительным потерям энергии. На рис. 5.16 приведен другой вариант согласования линии передачи – по переменному току, где конденсаторы C_T предотвращают течение постоянного тока.

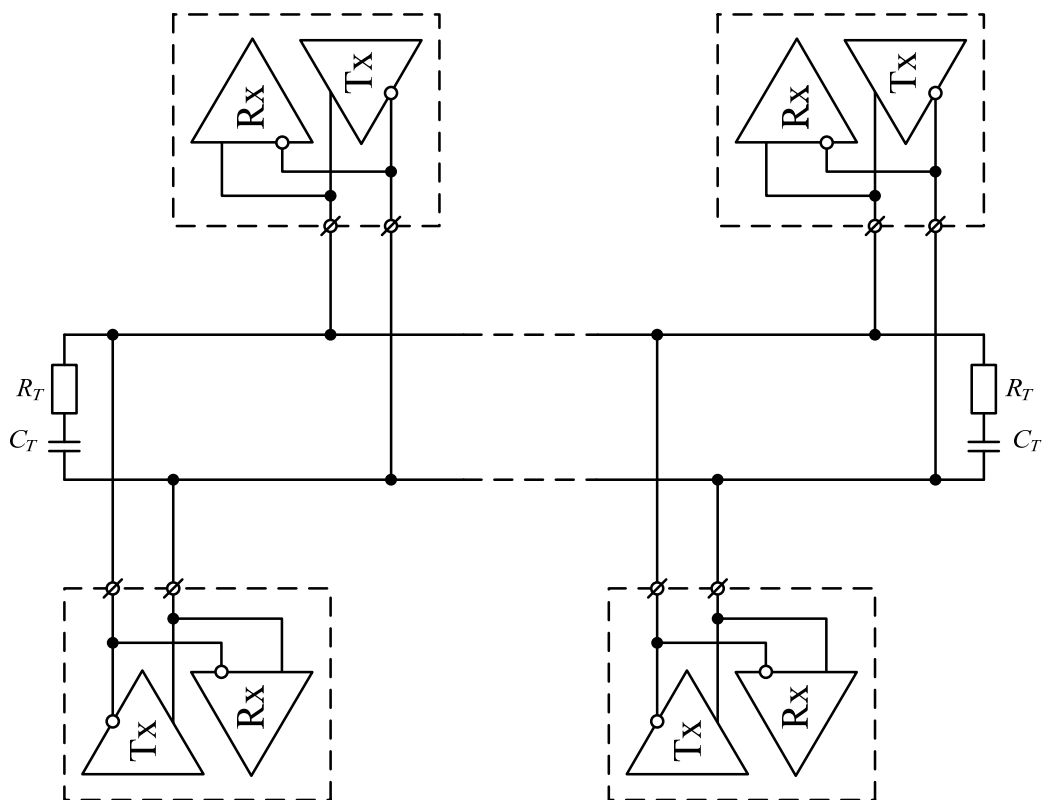


Рис. 5.16. Согласования линии передачи по переменному току

Отводы

Для того чтобы исключить влияние отражения электромагнитной волны от мест подключения приемопередатчиков на протяжении всей линии необходимо чтобы длина отводов (расстояние между приемопередатчиком и кабельной линией связи) была таковой, чтобы электромагнитная волна преодолевала это расстояние примерно за 10% времени фронта импульса полезного сигнала

$$L_{отв} \leq \frac{t_{\phi}}{10} \times \frac{v_{ном}}{100\%} \times c, \quad (5.4)$$

где $L_{отв}$ – длина отвода, [м]; t_{ϕ} – длительность фронта импульса, [с]; $v_{ном}$ – номинальная скорость распространения электромагнитной волны в кабеле, [%]; c – скорость света в вакууме [м/с].

Длительность переднего фронта импульса (время нарастания от 10% до 90% от максимального значения амплитуды) зависит от параметров используемого кабеля и протяженности линии связи

$$t_{\phi} = 2,2 \cdot R_{\text{экв}} \cdot C_k \cdot L, \quad (5.5)$$

где C_k – погонная емкость кабеля, [пФ/м]; L – протяженность линии связи, [м]; $R_{\text{экв}}$ – эквивалентное активное сопротивление нагрузки формирователя, определяемое следующим образом:

$$R_{\text{экв}} = L \cdot r_k^* + \frac{1}{\frac{n}{R_{\text{ex}}} + \frac{2}{R_T}} \approx Z_0, \quad (5.6)$$

r_k^* – погонное сопротивление выбранного кабеля, [Ом/км]; R_T – сопротивление терминирующих резисторов, [Ом]; n – количество приемников с сопротивлением R_{ex} , [Ом].

Отказоустойчивость

Отказоустойчивая работа представляет собой способность приемника принимать определенное состояние на выходе при отсутствии входного сигнала. К потере сигнала могут приводить три возможные причины:

1. Обрыв, вызванный разрывом провода или отключением приемопередатчика от шины.
2. Короткое замыкание, вызванное замыканием проводов дифференциальной пары друг с другом из-за нарушения изоляции.
3. Бездействие шины, возникающее при отсутствии активности шинных формирователей всех узлов сети.

В таких условиях на входе оконечных нагрузок (приемников) фактически присутствует нулевой входной сигнал, однако сами значения, снимаемые с них, могут принимать случайные состояния из-за действия шума в линии связи или непосредственно самих устройствах. Для устранения такой неопределенности используют защитные цепи смещения, которые принудительно устанавливают выход приемника в заданное состояние в случае пропадания сигнала в шине. Такие цепи смещения могут быть интегрированы в микросхему приемопередатчика (внутренние цепи) или могут быть организованы с использованием внешних элементов (внешние цепи).

Недостатком использования внутренних цепей является низкий запас по помехозащищенности, составляющий в худшем случае только 10 мВ.

Внешние цепи отказоустойчивости могут обеспечить больший уровень, и состоят из резистивного делителя напряжения (см. рис. 5.17), обеспечивающего дифференциальное напряжение в шине, достаточное для перевода выхода приемника в заданное определенное состояние.

Для обеспечения достаточного уровня помехозащищенности дифференциальное напряжение U_{AB} на входе приемника должно быть не меньше максимального значения дифференциальной помехи с учетом порогового уровня приемника 200 мВ, то есть $U_{AB} = U_{\text{Noise}} + 200 \text{ мВ}$. Тогда защитное сопротивление R_3 можно определить исходя из выражения:

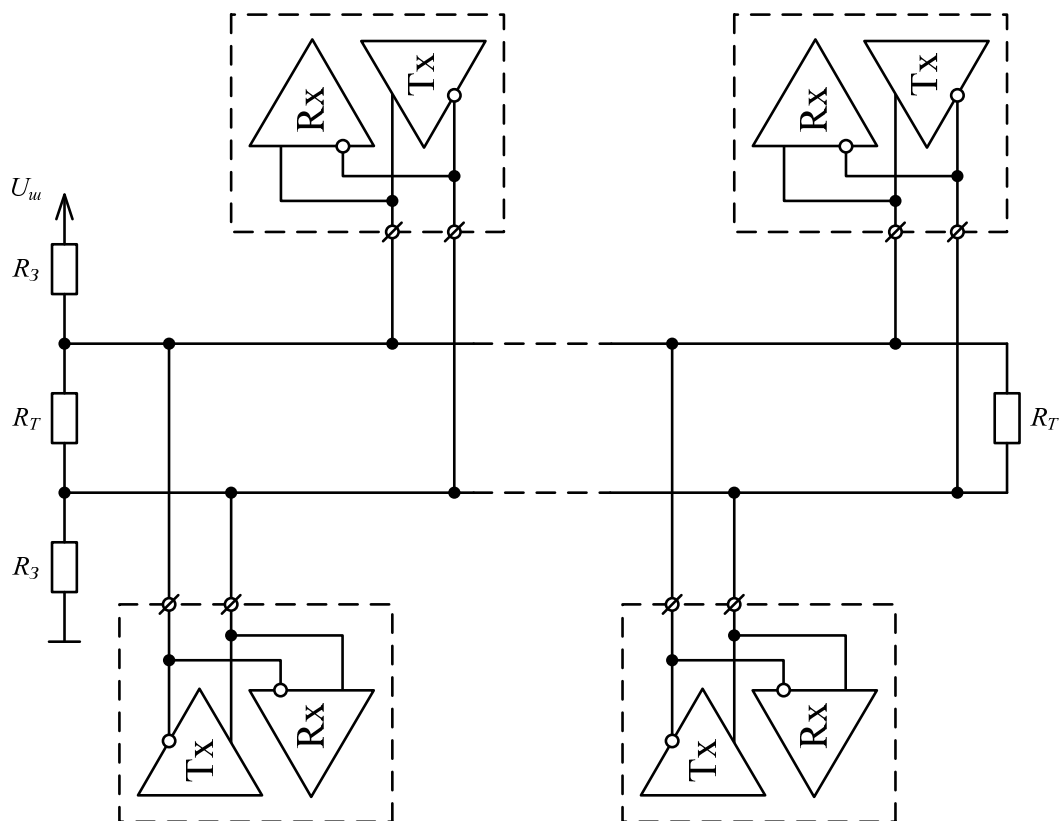


Рис. 5.17. Обеспечение отказоустойчивости линии RS-485

$$R_3 = \frac{U_{u}^{\min}}{U_{AB} \times (1/375 + 4/Z_0)}. \quad (5.7)$$

Цепи отказоустойчивости расширяют условия применения интерфейса RS-485, однако обладают существенным недостатком – соответствует 20 стандартным единицам нагрузки, что в итоге снижает количество узлов в сети

$$n = \frac{32UL_{\text{стнд}} - 20UL_{\text{откз}}}{UL_{\text{прм}}}, \quad (5.8)$$

где $UL_{\text{стнд}}$ – стандартные единицы нагрузки, $UL_{\text{откз}}$ – единицы нагрузки, отводимые на цепи отказоустойчивости, $UL_{\text{прм}}$ – нагрузка приемников в сети.

Кабель

Методика выбора кабеля для интерфейса RS-485 описана в [16], и состоит из нескольких шагов.

1. Исходя из требуемой скорости обмена C , вычислить длительность информационного бита $T_b = \frac{1}{C}$.

2. Задать минимальное напряжение сигнала U_0 , [В], которое должно присутствовать на входе самого удаленного приемника.
3. Задать максимально допустимый уровень искажений сигнала (δ^* , %) на входе самого удаленного приемника.
4. Задать максимальное требуемое значение длины кабеля L , [м].
5. Вычислить максимальное допустимое значение омического сопротивления кабеля длиной L :

$$R_L = \frac{R_T(U_{\min}^{ППД} - U_0)}{U_0},$$

где $R_T \approx Z_0$ – сопротивление терминатора, $U_{\min}^{ППД}$ – минимальное напряжение сигнала на выходе передатчика, U_0 – минимальное напряжение сигнала, которое должно присутствовать на входе самого удаленного приемника, [В].

6. Вычислить погонное сопротивление кабеля $r_k = \frac{R_L}{L}$.
7. Руководствуясь справочными данными, выбрать кабель, волновое сопротивление которого равно принятому в п. 5, а погонное сопротивление – не более вычисленного в п. 6.
8. Вычислить длительность переднего фронта импульса по формуле (5.5) с учетом (5.6).
9. Определить реальное значение уровня искажений сигнала на входе самого удаленного приемника (δ , %) по диаграмме на рис. 5.18.

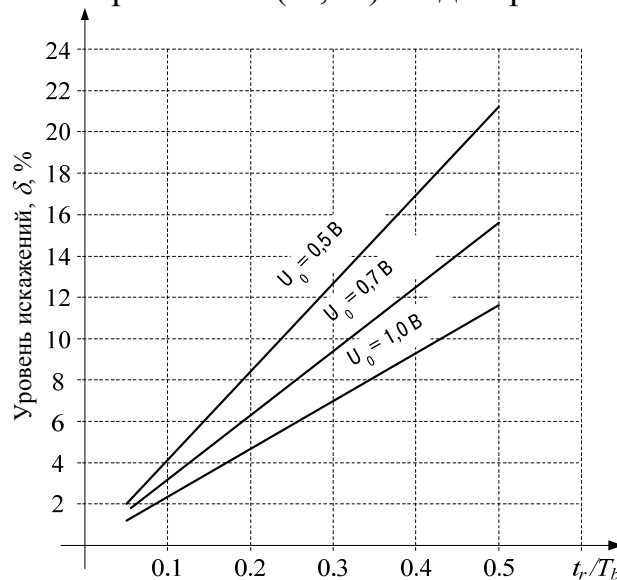


Рис. 5.18. График зависимости уровня искажений сигнала на входе приемника от минимального напряжения сигнала на его входе и от отношения длительности переднего фронта к длительности информационного бита

Если полученный уровень искажений δ превышает допустимый δ^* , следует повторить процедуру выбора кабеля. При этом кабель должен иметь меньшие значения погонного сопротивления и погонной емкости, чем выбранный в п. 7. Если не удастся выбрать кабель с лучшими параметрами, следует снизить значение скорости обмена либо сократить протяженность линии связи.

Повторители сигналов

Как было показано ранее, сеть RS-485 способна работать с 32 стандартными узлами, а скорость передачи данных и протяженность линии связи – это взаимосвязанные параметры. Если необходимо продлить линию связи с сохранением скорости, или использовать большее количество узлов сети, то следует использовать повторители сигнала RS-485, разбивая всю сеть на отдельные сегменты (рис. 5.19). При этом каждый сегмент фактически представляет собой отдельную сеть RS-485, при проектировании которой необходимо соблюдать все ранее приведенные правила.

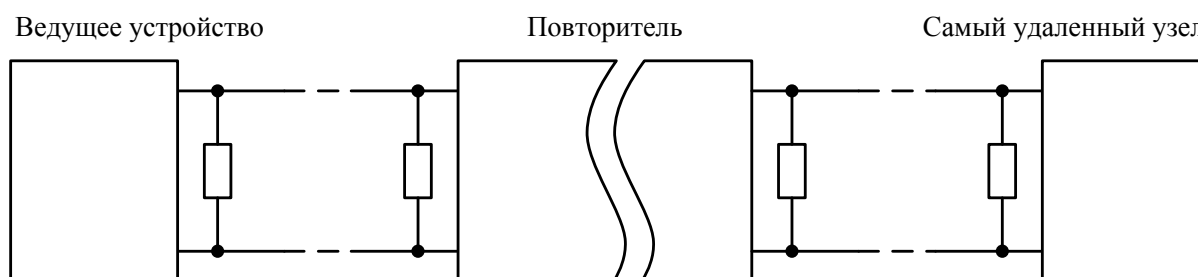


Рис. 5.19. Применение повторителей сигнала

5.5. Сравнение интерфейсов семейства RS

Основные технические характеристики рассматриваемых в данном разделе интерфейсов приведены в таблице 5.8.

Таблица 5.8

	RS-232	RS-423	RS-422	RS-485
Тип линии	Несимметричный	Несимметричный	Дифференциальный	Дифференциальный
Максимальное число формирователей	1	1	1	32
Максимальное число приемников	1	10	10	32
Режим передачи	полудуплексный, дуплексный	полудуплексный	полудуплексный	полудуплексный
Топология	точка-точка	шина	шина	шина

Окончание таблицы 5.8

Максимальная длина (по стандарту)	15 м	1200 м	1200 м	1200 м
Максимальная скорость на расстоянии 12 м	20 кбит/с	100 кбит/с	10 Мбит/с	35 Мбит/с
Максимальная скорость на расстоянии 1200 м	(1 кбит/с)	1 кбит/с	100 кбит/с	100 кбит/с
Максимальная скорость нарастания	30 В/мкс.	настраиваемая	–	–
Входное сопротивление приемника	3...7 кОм	≥ 4 кОм	≥ 4 кОм	≥ 12 кОм
Сопротивление нагрузки формирователя	3...7 кОм	≥ 450 Ом	100 Ом	54 Ом
Чувствительность входа приемника	$\pm 3,0$ В	± 200 мВ	± 200 мВ	± 200 мВ
Входной диапазон приемника	± 15 В	± 12 В	± 10 В	$-7...12$ В
Максимальное выходное напряжение	± 25 В	± 6 В	± 6 В	$-7...12$ В
Минимальное выходное напряжение (с нагрузкой)	± 5 В	± 3.6 В	± 2.0 В	± 1.5 В

Контрольные вопросы

1. Какой интерфейс поддерживает самую большую длину линии связи?
2. Какие характеристики интерфейса зависят от длины линии связи?
3. Какие способы соединения между устройствами доступны при использовании интерфейса RS-232?
4. Сколько приемников можно подключить к передатчику RS-422?
5. Сколько приемопередатчиков можно объединить с помощью интерфейса RS-232? RS-422? RS-485?
6. Перечислите электрические характеристики интерфейса RS-232. Какие основные недостатки данного интерфейса?
7. Какие типы разъемов определены интерфейсом RS-232? Расскажите о назначении всех контактов разъема DE-9.
8. Перечислите электрические характеристики RS-422. Какие основные преимущества данного интерфейса по сравнению с RS-232?
9. Перечислите электрические характеристики RS-485. Какие основные преимущества данного интерфейса по сравнению с RS-422?
10. Какие схемы соединения используются с интерфейсом RS-485?
11. Возможно ли построение смешанной сети как на базе устройств RS-423, так и RS-422?
12. В чем заключается главный недостаток интерфейса RS-423?

6. СИМВОЛЬНЫЕ ПРОТОКОЛЫ ОБМЕНА ДАННЫМИ

6.1. Код ASCII

Использование кода ASCII (American Standard Code for Information Interchange) является самым простым вариантом для применения в промышленных сетях с одним ведущим и несколькими ведомыми узлами. Данное решение может применяться только в небольших сетях с невысокими требованиями к скорости обмена данными, и реализуется с помощью универсального асинхронного приемопередатчика (UART), что фактически не увеличивает стоимости приемопередающих узлов.

Обмен данными может производиться с использованием интерфейсов:

- RS-232 (или RS-422) – двухточечная сеть, в которой расширение осуществляется за счет последовательного подключения узлов. Обмен осуществляется последовательно от ведущего узла к ведомому, и далее по цепи, пока узел не определит свой адрес в пакете и не отошлет ответ о приеме сообщения.
- RS-485 – многоточечное подключение, работающее в полудуплексном режиме.

Таблица ASCII представляет собой кодировку для представления десятичных цифр, латинского и национального алфавитов, знаков препинания, управляющих символов. Таблица была разработана и стандартизована в США в 1963 году. Стандартный набор символов ASCII использует только 7 битов для каждого символа ($2^7=128$ символов). Добавление 8-го разряда позволяет увеличить количество кодов таблицы ASCII до $2^8=256$. Коды от 128 до 255 представляют собой расширение таблицы ASCII. Эти коды используются для кодирования символов национальных алфавитов, а также символов псевдографики.

Основная таблица ASCII представлена в таблице 6.1. Первые 32 символа, а также последний символ этой таблицы являются управляющими, и их подробное назначение приведено в таблице 6.2.

Таблица 6.1

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1_	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2_		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3_	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4_	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5_	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7_	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Таблица 6.2

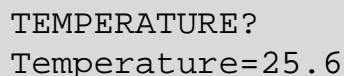
Обозн.	Hex	Наименование	Назначение
NUL	00	Null character	Пустой символ
SOH	01	Start of Header	Начало заголовка
STX	02	Start of Text	Начало текста (сообщения)
ETX	03	End of Text	Конец текста (сообщения)
EOT	04	End of Transmission	Конец передачи
ENQ	05	Enquire	Запрос подтверждения
ACK	06	Acknowledgment	Подтверждение
BEL	07	Bell	Звонок (звуковой сигнал)
BS	08	Backspace	Возврат на один символ
HT	09	Horizontal Tab	Горизонтальная табуляция
LF	0A	Line feed	Перевод строки
VT	0B	Vertical Tab	Вертикальная табуляция
FF	0C	Form feed	Прогон страницы (новая страница)
CR	0D	Carriage return	Возврат каретки
SO	0E	Shift Out	Выключить сдвиг
SI	0F	Shift In	Включить сдвиг
DLE	10	Data Link Escape	Освобождение канала от данных
DC1	11	Device Control 1 (XON)	Управление устройством: 1, 2 – включить, 3, 4 – выключить XON / XOF – управление приемом/передачей
DC2	12	Device Control 2	
DC3	13	Device Control 3 (XOF)	
DC4	14	Device Control 4	
NAK	15	Negative Acknowledge	Негативное квитирование (обр. ACK)
SYN	16	Synchronous Idle	Символ синхронизации
ETB	17	End of Transmission Block	Конец текстового блока
CAN	18	Cancel	Отмена (переданного ранее)
EM	19	End of Medium	Конец носителя
SUB	1A	Substitute	Подстановка (на место утерянного)
ESC	1B	Escape	Меняет значение следующего символа
FS	1C	File Separator	Разделитель файлов
GS	1D	Group Separator	Разделитель групп
RS	1E	Record Separator	Разделитель записей
US	1F	Unit Separator	Разделитель элементов
DEL	7F	Delete	Стереть последний символ

Символьный обмен подразумевает использование только основной таблицы ASCII (таблица 6.1), то есть передачу с использованием семи бит данных, что отличает данный способ от *бинарного обмена*, при котором задействуются восемь бит, то есть возможна передача байта данных целиком.

Существуют большое количество протоколов, построенных на основе ASCII, которые используются в промышленных сетях для подключения различного оборудования. Эти протоколы в основном отличаются форма-

том кадров передаваемых сообщений и допустимым набором символов, при помощи которых организуется передача.

В самом простом случае сеанс обмена данными между устройствами может представлять собой обмен *текстовыми данными*. Преимущество такого подхода заключается в том, что передаваемые данные достаточно легко наблюдаемы человеком, а также могут быть легко им сгенерированы. В качестве примера на рисунке 6.1 приведен фрагмент сеанса взаимодействия с датчиком температуры в текстовом формате.



```
TEMPERATURE?  
Temperature=25.6
```

Рис. 6.1. Текстовый обмен данными

Так как подобного рода протоколы не являются стандартными, то целенаправленно говорить о них не представляется возможным. При необходимости организации взаимодействия с такими устройствами, задачу формирования и декодирования кадров сообщения необходимо решать на уровне приложений в каждом случае отдельно.

6.2. Протокол ADAM ASCII (DCON)

Протокол ADAM ASCII, известный также как DCON, был разработан в середине 1980-х годов компанией Advantech для взаимодействия со своим оборудованием серии ADAM. Этот протокол, как и все ASCII-протоколы, основан на UART. Кроме того, протокол DCON не является стандартным протоколом, однако он очень широко распространен в России, и существует большое количество как отечественных, так и зарубежных производителей промышленного оборудования, реализующих протокол DCON в своих изделиях.

К достоинствам этого протокола следует отнести низкую стоимость, простоту и независимость от специализированных информационных систем. Вместе с тем отсутствие единого стандарта породило большое количество вариаций этого протокола, но в целом идеология и основные принципы являются совместимыми.

Протокол ADAM ASCII (DCON) имеет в своей архитектуре один ведущий узел и до 255 ведомых узлов. Режим приема-передачи – полудуплексный. В модели ВОС задействует физический и прикладной уровень.

Требования к физической среде определяются стандартом на интерфейс RS-485. Передаваемые сообщения на прикладном уровне соответствуют формату, приведенному на рис. 6.2. Для кодирования всех передаваемых данных используются ASCII-символы цифр 0 – 9 (0x30 – 0x39) и букв от A

до F (0x41 – 0x46). Каждая команда оканчивается символом возврата каретки <CR> (ASCII-код 0x0D).

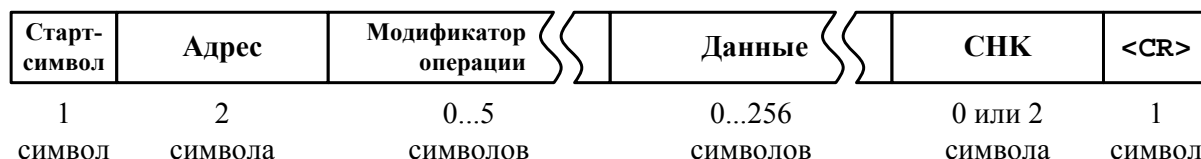


Рис. 6.2. Формат кадра протокола DCON

Стартовый символ определяет группу операций, и зависит от того, кто отправляет кадр сообщения – ведущее или ведомое устройство. Наиболее распространенные символы групп операций представлены в таблице 6.3.

Таблица 6.3

Ведущее устройство		Ведомое устройство	
\$	Определяет группу команд	!	Начало кадра подтверждения
#			Команда проигнорирована
%		?	Команда неправильная
@		>	Команда выполнена
~			

Адрес включает в себя два ASCII-символа, которые в шестнадцатеричной системе счисления задают номер устройства в сети DCON в диапазоне от 0x00 до 0xFF.

Модификатор операции совместно со стартовым символом определяет непосредственную команду, передающуюся от ведущего устройства к ведомому. Он зависит от типа оборудования, и в некоторых случаях может отсутствовать. Модификатор операции не используется в кадрах, поступающих от ведомых устройств.

Поле данных определяется командой. Также зависит от типа используемого оборудования, и в некоторых случаях может отсутствовать.

Для повышения надежности передачи данных может использоваться контрольная сумма (CHK), представляемая двумя символами ASCII. Если оборудование сконфигурировано на использование контрольной суммы, то проверка контрольной суммы и передача ее значения в поле кадра DCON является обязательным. Соответственно, два ASCII-символа позволяют закодировать в шестнадцатеричной системе счисления один байт данных, который является младшим байтом результата суммирования кодов всех ASCII-символов кадра DCON, непосредственно предшествующих полю контрольной суммы.

Например, если нужно передать ведомому устройству команду

\$05M<CR>,

то контрольная сумма CHK равна сумме кодов символов «\$», «0», «5», «M», то есть $CHK = 0x24 + 0x30 + 0x35 + 0x4D = D6$. Таким образом, окончательно команда будет иметь вид:

\$05MD6<CR>,

а в линию связи должны быть отправлены байты с кодами: 0x24, 0x30, 0x35, 0x4D, 0x44, 0x36, 0x0D.

Контрольные вопросы

1. Сколько различных символов кодируется основной таблицей ASCII?
2. Какой код ASCII имеет символ «@»? Какой символ кодируется кодом 0x30?
3. Укажите максимальное количество ведомых устройств, которое может быть в сети DCON.
4. Является ли контрольная сумма обязательным полем кадра протокола DCON?
5. При вычислении контрольной суммы кадра DCON было получено значение 0x2DE1. Как должно быть передано это значение при условии, что на поле контрольной суммы отводится 2 ASCII-символа?

7. СЕТИ MODBUS

7.1. Общие сведения

Протокол Modbus имеет клиент-серверную архитектуру и был разработан компанией Modicon (в настоящий момент входит в состав Schneider Electric) для использования в своих программируемых логических контроллерах (PLC) в 1979 г. С 2004 г. развитием и поддержкой занимается некоммерческая организация Modbus-IDA, позднее переименованная в Modbus Organization.

Несмотря на то, что интерфейс не был стандартизован, протокол Modbus и одноименная сеть широко распространены в мире в области построения АСУ ТП, а сам протокол стал стандартом де-факто в промышленности. Высокой популярности сетей Modbus способствовало то, что для построения сети не требуется каких-либо специализированных интерфейсных контроллеров, а алгоритм взаимодействия узлов имеет простую программную реализации. Тексты стандартов протокола Modbus являются полностью открытыми и доступны на сайте www.modbus.org.

Особенности построения сетей Modbus описываются рядом документов [13]:

- *MODBUS Protocol Specification* (предыдущее название – *Modbus Application Protocol*) – содержит спецификацию прикладного уровня сетевой модели ВОС: описание элементарного пакета протокола (PDU), который един для всех физических уровней, коды функций и состав PDU каждой функции.
- *Modbus Serial Line Protocol and Implementation Guide (Modbus over serial line)* – содержит спецификацию канального и физического уровней сетевой модели ВОС для физических уровней RS-485 и RS-232. Может использоваться любой физический уровень, основанный на асинхронном приеме-передаче.
- *MODBUS TCP/IP (MODBUS Messaging on TCP/IP Implementation Guide)* – содержит спецификацию ADU для передачи данных на основе стека протокола TCP/IP.
- *MODBUS Security Protocol* – опубликованный в мае 2018 г. протокол обеспечения безопасности передачи данных в сетях TCP/IP, добавляющий алгоритмы защиты транспортного уровня (TLS) к традиционным пакетам Modbus.

В документе MODBUS Protocol Specification описываются основные особенности протокола. В частности, взаимодействие узлов сети осуществляется по принципу «Клиент – Сервер» и используется централизованный метод доступа к среде передачи данных («Ведущий – Ведомый»). Ведущее устройство может адресовать запрос индивидуально любому ведомому или инициировать передачу широковещательного сообщения, адресованного сразу всем ведомым устройствам. При индивидуальном запросе ведомое устройство, опознав свой адрес, отвечает на запрос, адресованный именно ему, а при получении широковещательного запроса ответ ведомыми устройствами не формируется.

Сама структура запросов ведущего узла и ответов ведомых устройств описывается в этом же документе. Основой передачи является *элементарный пакет протокола*, так называемый PDU (Protocol Data Unit). Структура PDU не зависит от типа линии связи и включает в себя код функции и поле данных (рис. 7.1).



Рис. 7.1. Структура PDU протокола Modbus

Код функции кодируется однобайтовым полем и может принимать значения в диапазоне 1...127. Диапазон значений 128...255 зарезервирован для кодов ошибок. Поле данных может быть переменной длины. В целом, размер пакета PDU ограничен 253 байтами.

Спецификация протокола Modbus определяет четыре типа данных, которыми могут оперировать сетевые устройства (см. пример на рис. 7.2). К бинарным типам данных относятся:

- Discrete Input (входы) – предназначены только для чтения,
- Coils (катушки реле = выходы) – предназначены для чтения и записи.

К 16-разрядным данным относятся:

- Input Registers (входные регистры) – предназначены только для чтения,
- Holding Registers (регистры хранения) – предназначены для чтения и записи.

На чтение/изменение данных каждого типа данных в протоколе существуют соответствующие команды – функции.

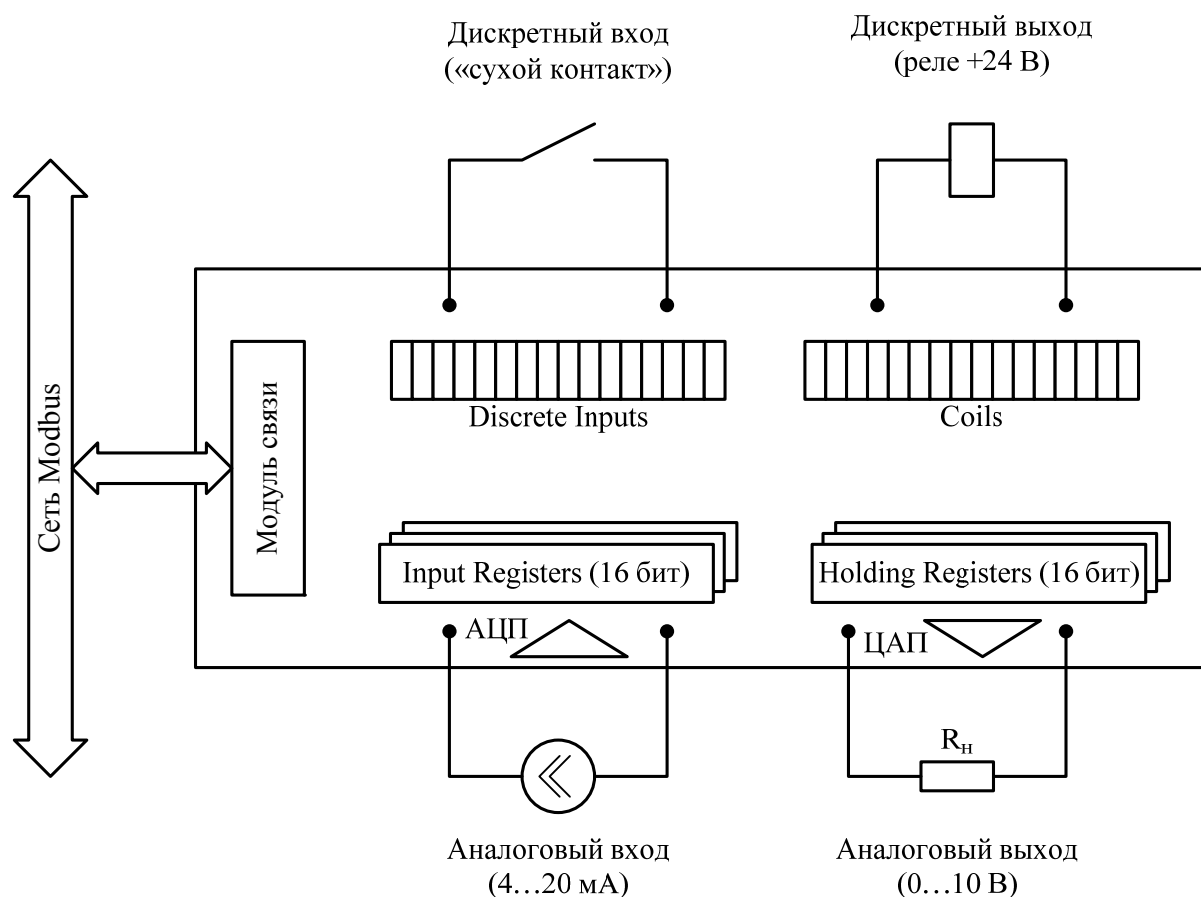


Рис. 7.2. Пример подключения подчиненного устройства Modbus

Ведомое устройство Modbus может содержать в своем составе различные типы данных в различных количествах. Для доступа к конкретным

ячейкам памяти используется значение адреса (номера) регистра. Как было уже сказано, изначально протокол Modbus был разработан для контроллеров Modicon, а потому, даже сейчас, можно встретить документацию от производителей оборудования, в которой используется адресация регистров, использовавшаяся в контроллерах Modicon. Ее особенностью является то, что каждый тип данных имеет свой собственный, непересекающийся диапазон адресов, который записывается в десятичном виде, где первая цифра фактически обозначает группу регистров или же тип данных (см. табл. 7.1). Адреса в пределах каждой группы начинаются с единицы.

Так как для каждого типа данных используется своя функция, определяющая действия, то в существующей спецификации протокола Modbus используется независимая адресация в пределах каждой группы регистров, которая записывается двухбайтовым шестнадцатеричным числом в диапазоне от 0x0000 до 0xFFFF.

Таблица 7.1

Тип данных	Режим	Группа регистров	Адрес Modicon	Адрес Modbus
Coils	RW	0x	00001–09999	0x0000–0xFFFF
Discrete Input	R	1x	10001–19999	0x0000–0xFFFF
Input Registers	R	3x	30001–39999	0x0000–0xFFFF
Holding Registers	RW	4x	40001–49999	0x0000–0xFFFF
Extended Memory	RW	6x	60001–69999	0x0000–0xFFFF

Вне зависимости от того, какой тип протокола Modbus используется (Modbus RTU, Modbus ASCII или Modbus TCP), PDU помещается в другой пакет – ADU (Application Data Unit), структура которого *зависит* от типа линии связи и способа передачи данных. Существуют три варианта ADU, каждый из которых будет рассмотрен при обсуждении соответствующей реализации протокола Modbus. Для всех разновидностей ADU общим является то, что помимо полезных данных (PDU), в них содержится адресная информация, позволяющая узлам сети определить, кто должен обрабатывать принятый пакет данных, а также некоторые виды ADU содержат дополнительные данные, позволяющие диагностировать сбои в процессе передачи данных.

В сетях Modbus каждое ведомое устройство должно иметь уникальный адрес в диапазоне от 1 до 247. Адрес 0 используется для организации широковещательной передачи данных. Адреса в диапазоне 248...255 – зарезервированы и не используются.

Поле адреса ведомого устройства каждого кадра ADU состоит из одного байта информации (в зависимости от особенностей построения ADU может передаваться в линии связи одним или двумя байтами данных). В

кадрах запросов это поле идентифицирует ведомое устройство, к которому обращен запрос.

Ведомые устройства, получив сообщение по сети, должны реагировать на них только если удостоверятся в целостности ADU, и адрес, переданный в сообщении, совпадает с адресом самого устройства (или указана широковещательный адрес). На широковещательные команды ведомые устройства не выдают никакой ответ, а на индивидуальные запросы формируют ADU кадры-ответы, в поле адреса ведомого которых указывают свой индивидуальный адрес.

7.2. Modbus RTU

Modbus RTU (Remote Terminal Unit – удаленное терминальное устройство) является базовым вариантом протокола построения сетей Modbus по последовательным линиям связи. Для формирования PDU используется весь диапазон возможных значений для 8-битных данных (0–255 или 0x00–0xFF).

Структура передаваемого по сети кадра ADU для варианта Modbus RTU приведена на рис. 7.3: сначала передается адрес ведомого устройства (1 байт), затем – PDU, и завершает передачу контрольная сумма (2 байта).

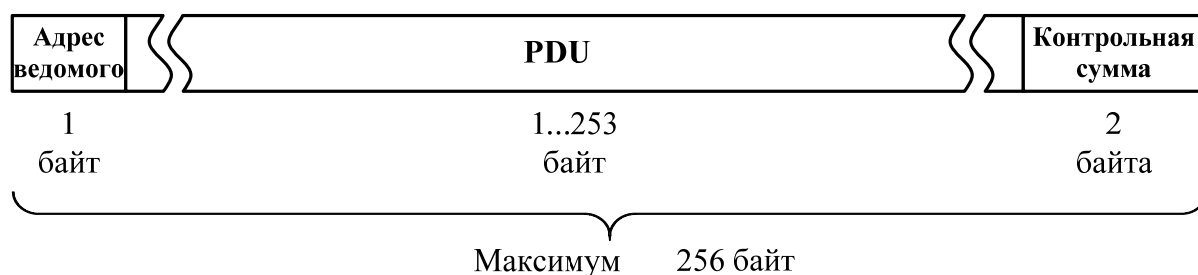


Рис. 7.3. Структура ADU для Modbus RTU

Сообщения Modbus RTU передаются по сети в виде кадров, которые в явном виде не содержат символов начала и конца. Признаком начала (и окончания) ADU является интервал тишины (рис. 7.4) – время по истечении которого принимающее устройство делает вывод об окончании процесса передачи данных, и может начать (если это разрешено) передавать свой кадр ADU.

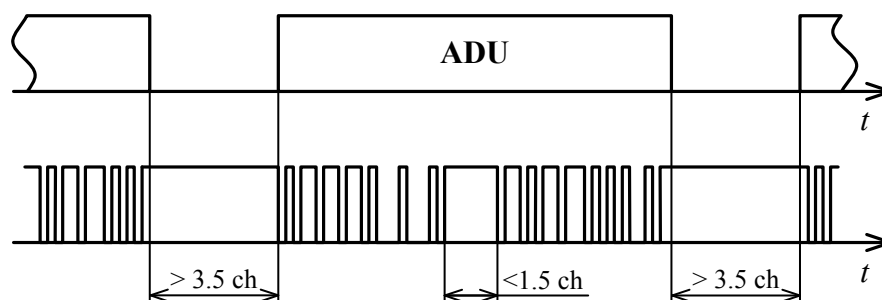


Рис. 7.4. Передача данных Modbus RTU

В соответствии со спецификацией Modbus RTU интервал тишины должен составлять не менее трех с половиной символов на выбранной скорости обмена данными. То есть, если в параметрах коммуникационного обмена установлена скорость 9600 бит/с и режим передачи данных 8e1, то интервал тишины

$$t_{3.5} = 3.5 \times \frac{1+8+1+1}{9600} \approx 4 \text{ мс.}$$

Вообще кадр ADU должен передаваться непрерывно, когда сразу по завершении передачи одного символа в линию связи передается другой, но Modbus RTU допускает задержку в передаче между символами на время, не превышающее времени передачи полутора символов. То есть для того же примера это время не должно превышать

$$t_{1.5} = 1.5 \times \frac{1+8+1+1}{9600} \approx 1.7 \text{ мс.}$$

Если при передаче кадра обнаруживается пауза более 1.5 символов, то считается, что кадр содержит ошибку и должен быть отклонен принимающей стороной. Если ADU начнется раньше, чем гарантированный интервал тишины в 3.5 символа, то оно будет восприниматься как часть предыдущего кадра, что в конечном итоге опять приведет к детектированию ошибки.

Эти интервалы связаны со скоростью передачи данных и должны строго соблюдаться при скоростях до 19200 бит/с (включительно), однако при более высоких скоростях для сокращения вычислительной нагрузки на микроконтроллер сетевого устройства рекомендуется использовать фиксированные значения: минимальный интервал тишины – 1.75 мс, максимальный межсимвольный интервал – 750 мкс.

Modbus RTU имеет два уровня контроля ошибок в сообщении: контроль бита паритета для каждого байта (опционально) и контроль целостности ADU при помощи двухбайтовой циклической контрольной суммы CRC с полиномом $P(x)=x^{16}+x^{15}+x^2+1$.

Контрольная сумма вычисляется передающим устройством и добавляется к сообщению. Принимающее устройство также вычисляет CRC в процессе приема и сравнивает вычисленную величину с полем контрольной

суммы пришедшего сообщения. Если суммы не совпали, то имеет место ошибка, если же оба значения совпадают, то считается, что сообщение не содержит ошибок и получатель получил исходный кадр ADU.

Алгоритм вычисления CRC16

1. Шестнадцатеричный регистр загружается числом $FFFF_{16}$ (все 1), и используется далее, как регистр CRC.
2. Первый байт сообщения складывается по ИСКЛЮЧАЮЩЕМУ ИЛИ с содержимым регистра CRC. Результат помещается в регистр CRC.
3. Регистр CRC сдвигается вправо (в направлении младшего бита) на 1 бит, старший бит заполняется 0.
4. Если младший бит 0, то повторяется шаг 3 (сдвиг), если младший бит 1, то делается операция ИСКЛЮЧАЮЩЕЕ ИЛИ регистра CRC и полиномиального числа $A001_{16}$.
5. Шаги 3 и 4 повторяются восемь раз.
6. Повторяются шаги со 2 по 5 для следующего байта сообщения. Это повторяется до тех пор, пока все байты сообщения не будут обработаны.
7. Финальное содержание регистра CRC и есть контрольная сумма.

В структуре ADU Modbus RTU сначала передается младший байт контрольной суммы, а затем старший.

7.3. Modbus ASCII

В отличие от Modbus RTU в составе ADU протокола Modbus ASCII могут использоваться только ASCII-символы, которыми кодируются шестнадцатеричные числа, т.е. байты с кодами $0x30$ – $0x39$ для кодирования цифр от 0 до 9 и байты с кодами $0x41$ – $0x46$ для кодирования букв A–F. В итоге, каждый байт PDU сообщения в Modbus ASCII содержит два 4-х битных шестнадцатеричных числа, каждое из которых кодируется одним байтом в ADU. Помимо этих 16 значений используются еще 3: символ «двоеточие (:)» ($0x3A$), «перевод строки» ($0x0A$ или $\langle LF \rangle$), «возврат каретки» ($0x0D$ или $\langle CR \rangle$).

Таким образом, в Modbus ASCII используется только ограниченное подмножество символов таблицы ASCII, которое можно визуально наблюдать при работе в терминальной программе, а потому сообщения Modbus ASCII могут быть легко декодированы человеком. Более того, так как все допустимые значения символов в Modbus ASCII находятся в основной таблице ASCII, то очень часто передача данных осуществляется с использованием семи бит данных, например, $7e1$ или $7n2$. Кстати, если бит паритета отсутствует, то обычно передается 2 стоповых бита, а если бит паритета вычисляется, то стоповый бит только один.

Еще одним отличием Modbus ASCII от Modbus RTU является то, что каждый кадр ADU помечается разделителями, обозначающими начало и конец сообщения – сообщения начинаются с символа двоеточие «:» и заканчиваются последовательностью символов «возврат каретки», «перевод строки» (рис. 7.5). Интервалы тишины в данном случае не используются, а в процессе передачи данных допускается до 1 с вне зависимости от скорости обмена данными.

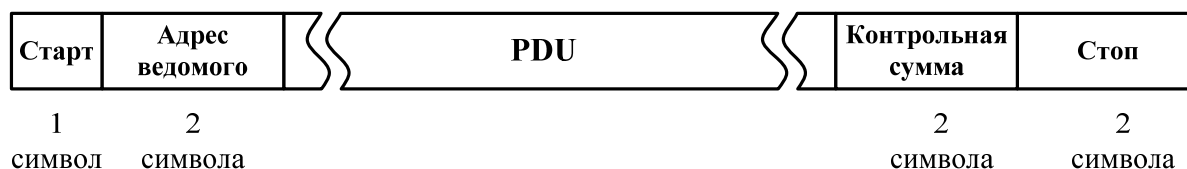


Рис. 7.5. Структура ADU для Modbus ASCII

Приемник ведомого устройства в сети Modbus ASCII должен непрерывно отслеживать передачу символа «двоеточие». Как только он принят, продолжается прием адреса ведомого устройства, данных PDU и контрольной суммы до приема символов <CR><LF>.

Modbus ASCII так же имеет два уровня проверки достоверности передачи данных: контроль бита паритета для каждого символа (опционально) и контроль целостности ADU при помощи однобайтовая контрольная сумма LRC (Longitudinal Redundancy Check), которая в составе ADU передается двумя символами.

LRC вычисляется передающим устройством и добавляется к концу сообщения. Принимающее устройство также вычисляет LRC в процессе приема и сравнивает вычисленную величину с полем контрольной суммы пришедшего сообщения. Если суммы не совпали – то имеет место ошибка.

Алгоритм вычисления LRC 8

1. Сложить все байты сообщения, исключая стартовый символ <:> и конечные <CR><LF>, складывая их таким образом, чтобы перенос отбрасывался.
2. Отнять получившееся значение от числа FF_{16} – это является первым дополнением.
3. Прибавить к получившемуся значению 1 – это второе дополнение.

Пусть в качестве примера требуется передать от ведущего устройства ведомому с адресом 01 сообщение со следующим PDU

03	00	6B	00	03
----	----	----	----	----

Тогда

$$LRC = FF - (01 + 03 + 00 + 6B + 00 + 03) + 1 = FF - 72 + 1 = 8E.$$

И в конечном итоге, в сети будут передаваться следующие байты

3A	30	31	30	33	30	30	36	42	30	30	30	33	38	45	0D	0A
:	01	03	00	6B	00	03	8E	\r	\n							
Старт	Адрес	Protocol Data Unit											LRC	Стоп		

7.4. Modbus TCP

Modbus TCP предназначен для использования в сетях TCP/IP по зарезервированному номеру порта 502, и, так же как и Modbus RTU, является бинарной версией протокола, допускающей в процессе передачи использование всех возможных однобайтных значений (0–255 или 0–FF₁₆). Структура ADU для сети Modbus TCP приведена на рис. 7.6.

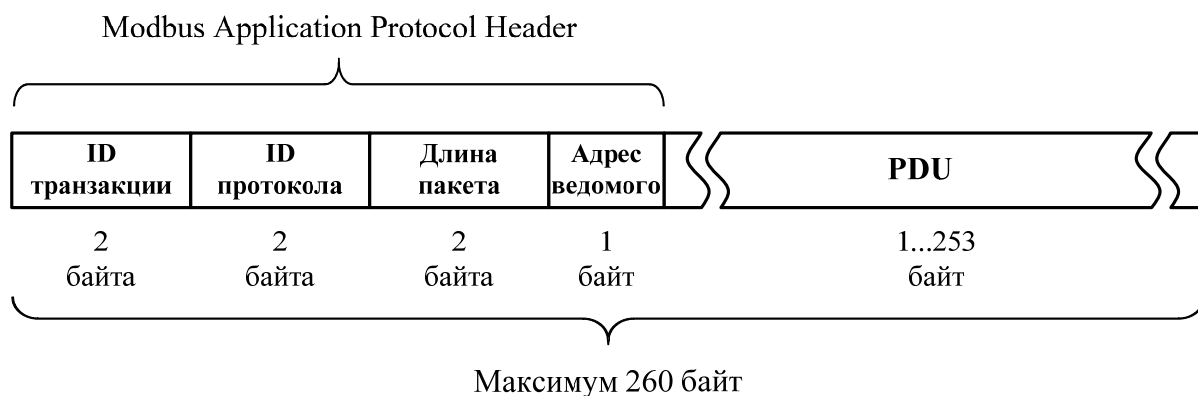


Рис. 7.6. Структура ADU для Modbus TCP

То есть передаче PDU предшествует передача заголовка протокола Modbus-приложения.

ID транзакции – два байта, предназначены для идентификации запроса и ответа, обычно нули.

ID протокола – два байта, используется преобразователями протоколов в смешанных сетях, обычно нули.

Длина пакета – два байта, длина следующей за этим полем части пакета.

Адрес ведомого – адрес подчиненного устройства, к которому адресован запрос. Обычно игнорируется, если в сети только одно устройство или уже установлено TCP-соединение с конкретным устройством. Поле может использоваться, если соединение установлено с мостом, который связан с сетью Modbus RTU/ASCII.

Следует обратить внимание, что поле контроля ошибок в Modbus TCP отсутствует, так как целостность данных обеспечивается за счет стека протокола TCP/IP.

7.5. Функции Modbus

Элементарный пакет данных – PDU протокола Modbus содержит код функции, который непосредственно определяет действия, которые необходимо выполнить ведомому устройству. Если ведомое устройство способно выполнить запрашиваемую функцию, то поле функции его ответа будет эхом оригинального запроса. В противном случае старший разряд в поле функции ответного PDU будет установлен в единицу, что является признаком произошедшей ошибки.

Во время обмена данными между ведущим и ведомыми устройствами в сети могут возникать ошибки, связанные с искажениями при передаче данных, которые детектируются контрольными кодами, а также логические ошибки. Существуют следующие базовые принципы, позволяющие функционировать устройствам в сети и детектировать логические ошибки:

- если ведомое устройство принимает корректный запрос и может его нормально обработать, то возвращается нормальный ответ, в поле функции ADU ответа повторяется код функции;
- если ведомое устройство не принимает какого-либо сообщения, никакой ответ не отправляется, ведущее устройство диагностирует ошибку связи по таймауту;
- если ведомое устройство принимает запрос, но обнаруживает ошибку целостности ADU (ошибка бита паритета, LRC или CRC), ADU игнорируется, никакой ответ ведущему устройству не отправляется, ведущее устройство диагностирует ошибку связи по таймауту;
- если ведомое устройство принимает запрос, но не может его обработать (обращение к несуществующему регистру, запрос не реализуемой устройством функции и т.д.), то оно отправляется ответ, содержащий в себе сведения о причинах ошибки, в поле функции ADU ответа повторяется код функции и старший бит поля функции устанавливается в единицу. Стандартные коды ошибок Modbus приведены в таблице 7.2.

В действующей в настоящее время спецификации протокола определяются три категории кодов функций:

Стандартные команды. Их описание опубликовано и утверждено Modbus Organization. Эта категория включает в себя как уже определенные, так и неиспользуемые в настоящее время коды.

Пользовательские команды. Два диапазона кодов (от 65 до 72 и от 100 до 110), для которых пользователь может назначить произвольную функцию. При этом не гарантируется, что какое-то другое устройство не будет использовать тот же самый код для выполнения другой функции.

Зарезервированные. В эту категорию входят коды функций, не являющиеся стандартными, но уже используемые в устройствах, производимых различными компаниями. Это коды 9, 10, 13, 14, 41, 42, 90, 91, 125, 126 и 127.

Таблица 7.2

Код	Название	Описание
01	ILLEGAL FUNCTION	Принятый код функции не может быть обработан на ведомом устройстве.
02	ILLEGAL DATA ADDRESS	Адрес данных указанный в запросе не доступен данному ведомому устройству
03	ILLEGAL DATA VALUE	Величина, содержащаяся в поле данных запроса, является не допустимой величиной для ведомого устройства
04	SLAVE DEVICE FAILURE	Имела место неустраняемая ошибка пока ведомое устройство пыталось выполнить затребованное действие
05	ACKNOWLEDGE	Ведомое устройство приняло запрос и обрабатывает его, но это требует много времени. Этот ответ предохраняет ведущее устройство от генерации ошибки таймаута. Ведущее устройство может выдать команду Poll Program Complete для обнаружения завершения обработки команды
06	SLAVE DEVICE BUSY	Ведомое устройство занято обработкой команды. Ведущее устройство должно повторить запрос позже, когда ведомое освободится
07	NEGATIVE ACKNOWLEDGE	Ведомое устройство не может выполнить программную функцию, принятую в запросе. Этот код возвращается для неудачного программного запроса, использующего функции с номерами 13 или 14. Ведущее устройство должно дополнительно запросить диагностическую информацию или информацию об ошибках с ведомого
08	MEMORY PARITY ERROR	Ведомое устройство пытается читать расширенную память, но обнаруживает ошибку паритета. Ведущее устройство может повторить запрос, но обычно в таких случаях требуется ремонт

Рассмотрим стандартные команды, позволяющие выполнить все основные действия над всеми основными типами данных, применяющимися в Modbus. Номера функций для этих команд представлены в таблице 7.3.

Таблица 7.3

Тип данных	Чтение	Запись	
		одного	многих
Coils (0x)	01	05	15
Discrete Inputs (1x)	02	—	—
Input Registers (3x)	04	—	—
Holding Registers (4x)	03	06	16

7.5.1. Функция 01 – Чтение дискретных выходов (Read Coils)

Функция с кодом 01 используется для считывания от 1 до 2000 смежных состояний дискретных выходов (катушек) на удаленном устройстве. Запрос PDU указывает начальный адрес, то есть адрес первого выбранного выхода и количество выходов.

Поле данных кадра ответа состоит из количества байт, при помощи которых кодируется состояние выходов, за которым следуют байты данных с состоянием выходов. Один бит соответствует одному выходу. Значение бита 1 соответствует состоянию ВКЛЮЧЕНО, значение 0 – ВЫКЛЮЧЕНО. Младший бит первого байта данных содержит выходные данные, адресованные в запросе. Состояние последующих выходов передается следующими битами этого же байта в направлении от младшего к старшему. Если запрашиваемые сведения передаются несколькими байтами (запрошено более 8 выходов), то биты последующих байт также в направлении от младшего к старшему соответствуют этим выходам. Если запрашиваемое количество выходов не кратно 8, то оставшиеся биты (в направлении к старшему) в последнем байте сообщения будут установлены в «0».

PDU запроса

Function code	1 байт	0x01
Starting Address	2 байта	0x0000 – 0xFFFF
Quantity of coils (Q)	2 байта	0x0001 – 0x07D0

PDU ответа

Function code	1 байт	0x01
Byte count	1 байт	$N = \lceil Q/8 \rceil$
Coil status	N байт	

PDU ответа ошибкой

Function code	1 байт	0x81
Exception code	1 байт	01, 02, 03 или 04

На рис. 7.7 приведен пример считывания состояния 37 дискретных выходов начиная с 19-ой ведомого устройства с адресом 0x11 с использованием протокола Modbus RTU. Минимальное число, кратное 8 и превосходящее запрошенное число дискретных выходов – это 40. Т.е. ответ содержит 5 байт данных, кодирующих состояние выходов:

13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37
1	0	1	1	0	0	1	1	1	1	0	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1	0	0	0	0	1	1	0	1	1
CD								6B								B2								0E								1B				

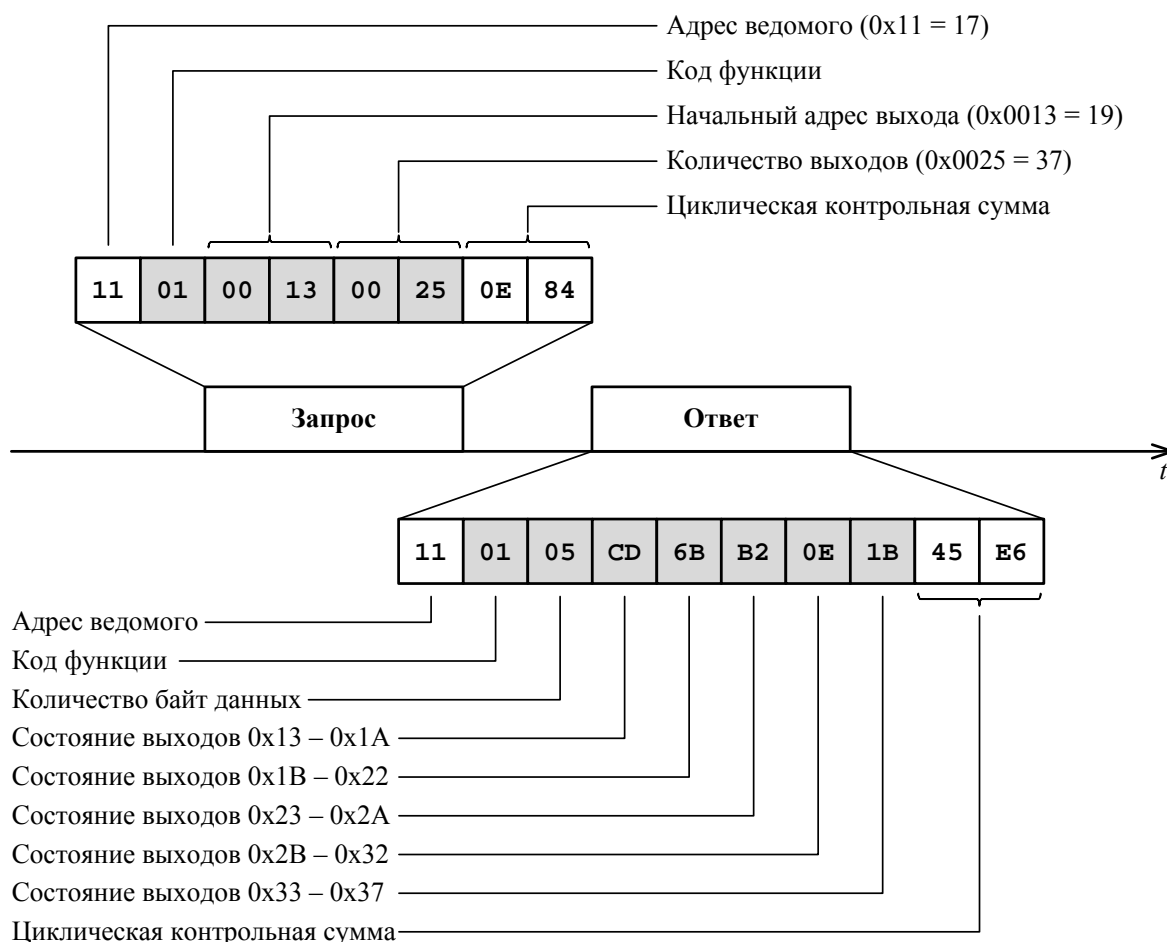


Рис. 7.7. Пример считывания состояния катушек (ADU для Modbus RTU)

В дальнейшем подобные примеры будем представлять в более компактной форме

Запрос		Ответ	
Поле	(Hex)	Поле	(Hex)
Function	01	Function	01
Starting Address (Hi)	00	Byte Count	05
Starting Address (Lo)	13	Coils 0x13 – 0x1A	CD
Quantity of Coils (Hi)	00	Coils 0x1B – 0x22	6B
Quantity of Coils (Lo)	25	Coils 0x23 – 0x2A	B2
		Coils 0x2B – 0x32	0E
		Coils 0x33 – 0x37	1B

Рис. 7.8. Пример считывания состояния катушек

7.5.2. Функция 02 – Чтение входов (Read Discrete Inputs)

Функция с кодом 02 используется для считывания от 1 до 2000 смежных состояний дискретных входов на удаленном устройстве. Запрос PDU указывает начальный адрес, то есть адрес первого выбранного входа и количество запрашиваемых входов.

Поле данных кадра ответа состоит из количества байт, при помощи которых кодируется состояние входов, за которым следуют байты данных с состоянием входов. Один дискретный вход соответствует одному биту в байте данных. Значение бита 1 соответствует состоянию ВКЛЮЧЕНО, значение 0 – ВЫКЛЮЧЕНО. Младший бит первого байта данных содержит выходные данные, адресованные в запросе. Состояние последующих дискретных входов передается следующими битами этого же байта в направлении от младшего к старшему. Если запрашиваемые сведения передаются несколькими байтами (запрошено более 8 дискретных входов), то биты последующих байт также в направлении от младшего к старшему соответствуют этим входам. Если запрашиваемое количество входов не кратно 8, то оставшиеся биты (в направлении к старшему) в последнем байте сообщения будут установлены в «0».

Фактически формат PDU функций 02 и 01 идентичен.

PDU запроса

Function code	1 байт	0x02
Starting Address	2 байта	0x0000 – 0xFFFF
Quantity of Inputs (Q)	2 байта	0x0001 – 0x07D0

PDU ответа

Function code	1 байт	0x02
Byte count	1 байт	$N = \lceil Q/8 \rceil$
Input status	N байт	

PDU ответа с ошибкой

Function code	1 байт	0x82
Exception code	1 байт	01, 02, 03 или 04

На рис. 7.9 приведен пример считывания состояния дискретных входов. Указанные в ответе значения соответствуют установленным в 1 дискретным входам 0xC4, 0xC5, 0xC6, 0xC7, 0xCE, 0xCF, 0xD1, 0xD5, 0xD7.

<i>Запрос</i>		<i>Ответ</i>	
Поле	(Hex)	Поле	(Hex)
Function	02	Function	02
Starting Address (Hi)	00	Byte Count	03
Starting Address (Lo)	C4	Inputs 0xC4 – 0xCB	0F
Quantity of Inputs (Hi)	00	Inputs 0xCC – 0xD3	2C
Quantity of Inputs (Lo)	14	Inputs 0xD4 – 0xD7	0A

Рис. 7.9. Пример считывания состояния дискретных входов

7.5.3. Функция 03 – Чтение регистров хранения (Read Holding Registers)

Эта функция используется для считывания содержимого непрерывного блока регистров хранения на удаленном устройстве (от 1 до 125). Запрос PDU указывает адрес начального регистра и количество считываемых регистров.

В ответном сообщении значения регистров передаются двумя байтами на регистр, причем двоичное содержимое выровнено по правому краю в каждом байте. Для каждого регистра первый байт содержит биты старшего разряда, а второй – биты младшего разряда (Hi-Lo).

PDU запроса

Function code	1 байт	0x03
Starting Address	2 байта	0x0000 – 0xFFFF
Quantity of Registers (Q)	2 байта	0x0001 – 0x007D

PDU ответа

Function code	1 байт	0x03
Byte count	1 байт	N = Q×2
Register value	N байт	

PDU ответа с ошибкой

Function code	1 байт	0x83
Exception code	1 байт	01, 02, 03 или 04

В следующем примере (рис. 7.10) ведущее устройство запрашивает содержимое трех регистров хранения начиная с адреса 0x006B. Содержимое регистров: 0x006B – 0x022B (555), 0x006C – 0x0000 (0) и 0x006D – 0x0064 (100).

<i>Запрос</i>		<i>Ответ</i>	
Поле	(Hex)	Поле	(Hex)
Function	03	Function	03
Starting Address (Hi)	00	Byte Count	06
Starting Address (Lo)	6B	Register 0x006B value (Hi)	02
Quantity of Registers (Hi)	00	Register 0x006B value (Lo)	2B
Quantity of Registers (Lo)	03	Register 0x006C value (Hi)	00
		Register 0x006C value (Lo)	00
		Register 0x006D value (Hi)	00
		Register 0x006D value (Lo)	64

Рис. 7.10. Пример считывания значений регистров хранения

7.5.4. Функция 04 – Чтение входных регистров (Read Input Registers)

Функция Modbus с кодом 04 используется для считывания от 1 до 125 последовательно расположенных входных регистров на удаленном устройстве.

В PDU запроса указывается адрес начального регистра и количество регистров.

В ответном PDU значения регистров передаются двумя байтами на регистр, причем двоичное содержимое выровнено по правому краю в каждом байте. Для каждого регистра первый байт содержит биты старшего разряда, а второй – биты младшего разряда (Hi-Lo).

Таким образом, структура PDU запросов и ответов функции 04 совпадает со структурой PDU функции 03.

PDU запроса

Function code	1 байт	0x04
Starting Address	2 байта	0x0000 – 0xFFFF
Quantity of Input Registers (Q)	2 байта	0x0001 – 0x007D

PDU ответа

Function code	1 байт	0x04
Byte count	1 байт	N = Q×2
Register value	N байт	

PDU ответа с ошибкой

Function code	1 байт	0x84
Exception code	1 байт	01, 02, 03 или 04

На рис. 7.11 приведен пример запроса (и ответа) на чтение одного входного регистра по адресу 0x0008. Для представленных в ответе данных его значение 0x000A или 10 в десятичной системе счисления.

<i>Запрос</i>		<i>Ответ</i>	
Поле	(Hex)	Поле	(Hex)
Function	04	Function	04
Starting Address (Hi)	00	Byte Count	02
Starting Address (Lo)	08	Register 0x0008 value (Hi)	00
Quantity of Registers (Hi)	00	Register 0x0008 value (Lo)	0A
Quantity of Registers (Lo)	01		

Рис. 7.11. Пример считывания значений входных регистров

7.5.5. Функция 05 – Запись дискретного выхода (Write Single Coil)

Функция предназначена для изменения состояния одного дискретного выхода на удаленном устройстве. Состояние ВКЛЮЧЕНО или ВЫКЛЮЧЕНО задается константой в поле данных PDU запроса. Значение 0xFF00 устанавливает адресуемый дискретный выход в состояние ВКЛЮЧЕНО, а значение 0x0000 – в состояние ВЫКЛЮЧЕНО. Все остальные значения недопустимы и не влияют на выход.

В случае успешного выполнения функции ведомое устройство отвечает сообщением с PDU являющимся копией PDU запроса.

PDU запроса

Function code	1 байт	0x05
Output Address	2 байта	0x0000 – 0xFFFF
Output Value	2 байта	0x0000 или 0xFF00

PDU ответа

Function code	1 байт	0x05
Output Address	2 байта	0x0000 – 0xFFFF
Output Value	2 байта	0x0000 или 0xFF00

PDU ответа с ошибкой

Function code	1 байт	0x85
Exception code	1 байт	01, 02, 03 или 04

На рис. 7.12 представлен PDU запроса, включающего дискретный выход по адресу 0x00AC, и ответное PDU.

<i>Запрос</i>		<i>Ответ</i>	
Поле	(Hex)	Поле	(Hex)
Function	05	Output Address (Hi)	05
Output Address (Hi)	00	Output Address (Lo)	00
Output Address (Lo)	AC	Output Value (Hi)	AC
Output Value (Hi)	FF	Output Value (Lo)	FF
Output Value (Lo)	00	Output Address (Hi)	00

Рис. 7.12. Пример управления дискретным выходом

7.5.6. Функция 06 – Запись одного регистра (Write Single Register)

Данная функция позволяет ведущему устройству изменить содержимое одного регистра хранения ведомого устройства.

В PDU запроса указывается адрес регистра и записываемое значение.

Если ведомое устройство способно записать новое значение в заданный регистр, то PDU ответа является копией PDU запроса. В противном случае будет возвращен ответ о возникновении исключительной ситуации.

PDU запроса

Function code	1 байт	0x06
Register Address	2 байта	0x0000 – 0xFFFF
Register Value	2 байта	0x0000 – 0xFFFF

PDU ответа

Function code	1 байт	0x06
Register Address	2 байта	0x0000 – 0xFFFF
Register Value	2 байта	0x0000 – 0xFFFF

PDU ответа с ошибкой

Function code	1 байт	0x86
Exception code	1 байт	01, 02, 03 или 04

На рис. 7.13 приведен пример PDU запроса и ответа, записывающего в регистр хранения по адресу 0x007A шестнадцатеричное значение 0xABCD.

<i>Запрос</i>		<i>Ответ</i>	
Поле	(Hex)	Поле	(Hex)
Function	06	Function	06
Register Address (Hi)	00	Register Address (Hi)	00
Register Address (Lo)	7A	Register Address (Lo)	7A
Register Value (Hi)	AB	Register Value (Hi)	AB
Register Value (Lo)	CD	Register Value (Lo)	CD

Рис. 7.13. Пример записи регистра хранения

7.5.7. Функция 15 – Запись дискретных выходов (Write Multiple Coils)

Функция позволяет ведущему устройству установить значения последовательности (до 1968) дискретных выходов удаленного устройства в состояние ВКЛЮЧЕНО или ВЫКЛЮЧЕНО.

Поле данных PDU запроса состоит из адреса начального выхода, количества переключаемых выходов, информации о количестве байт с данными и, собственно, самих байт с данными, каждый бит которых соответствует одному дискретному выходу. Если количество выходов не кратно 8, то оставшиеся старшие биты в последнем байте PDU сообщения должны быть установлены в 0.

При успешном выполнении команды подчиненное устройство формирует PDU, содержащее код функции, начальный адрес измененных выходов и количество выходов.

PDU запроса

Function code	1 байт	0x0F
Starting Address	2 байта	0x0000 – 0xFFFF
Quantity of Outputs (Q)	2 байта	0x0000 – 0x07B0
Byte Count	1 байт	$N = \lceil Q/8 \rceil$
Outputs Value	N байт	0x0000 – 0xFFFF

PDU ответа

Function code	1 байт	0x0F
Starting Address	2 байта	0x0000 – 0xFFFF
Quantity of Outputs	2 байта	0x0000 – 0x07B0

PDU ответа с ошибкой

Function code	1 байт	0x8F
Exception code	1 байт	01, 02, 03 или 04

В примере на рис. 7.14 осуществляется запись 20 дискретных выходов начиная с адреса 0x0015, причем выходы 0x0015 – 0x001F и 0x0024 – 0x0028 устанавливаются в состояние ВКЛЮЧЕНО, а выходы в диапазоне 0x0020 – 0x0023 – в состояние ВЫКЛЮЧЕНО.

<i>Запрос</i>		<i>Ответ</i>	
Поле	(Hex)	Поле	(Hex)
Function	0F	Function	0F
Starting Address (Hi)	00	Starting Address (Hi)	00
Starting Address (Lo)	15	Starting Address (Lo)	15
Quantity of Outputs (Hi)	00	Quantity of Outputs (Hi)	00
Quantity of Outputs (Lo)	14	Quantity of Outputs (Lo)	14
Byte Count	03		
Outputs Value (Byte #1)	FF		
Outputs Value (Byte #2)	87		
Outputs Value (Byte #3)	0F		

Рис. 7.14. Пример записи нескольких дискретных выходов

7.5.8. Функция 16 – Запись регистров хранения (*Write Multiple Registers*)

Функция с кодом 16₁₀ (0x10₁₆) позволяет ведущему устройству установить содержимое одного или нескольких (до 123) последовательно расположенных регистров хранения ведомого устройства.

Поле данных PDU запроса содержит адрес начального регистра, количество записываемых регистров, информацию о количестве передаваемых байт с данными и, собственно, сами данные, задающие содержимое регистров хранения. Каждый регистр формируется двумя байтами данных.

При успешном выполнении команды подчиненное устройство формирует PDU, содержащее код функции, адрес начального регистра, количество записанных регистров.

PDU запроса

Function code	1 байт	0x10
Starting Address	2 байта	0x0000 – 0xFFFF
Quantity of Registers (Q)	2 байта	0x0000 – 0x007B
Byte Count	1 байт	$N = Q \times 2$
Registers Value	N байт	0x0000 – 0xFFFF

PDU ответа

Function code	1 байт	0x10
Starting Address	2 байта	0x0000 – 0xFFFF
Quantity of Registers	2 байта	0x0000 – 0x007B

PDU ответа с ошибкой

Function code	1 байт	0x90
Exception code	1 байт	01, 02, 03 или 04

<i>Запрос</i>		<i>Ответ</i>	
Поле	(Hex)	Поле	(Hex)
Function	0F	Function	0F
Starting Address (Hi)	03	Starting Address (Hi)	00
Starting Address (Lo)	20	Starting Address (Lo)	15
Quantity of Registers (Hi)	00	Quantity of Outputs (Hi)	00
Quantity of Registers (Lo)	02	Quantity of Outputs (Lo)	14
Byte Count	04		
Register 0x0320 value (Hi)	12		
Register 0x0320 value (Lo)	34		
Register 0x0321 value (Hi)	CA		
Register 0x0321 value (Lo)	FE		

Рис. 7.15. Пример записи нескольких регистров хранения

В примере на рис. 7.15 осуществляется запись двух регистров хранения. В регистр по адресу 0x0320 записывается шестнадцатеричное значение 0x1234, а в следующий регистр (0x0321) – значение 0xCAFE.

7.6. Характеристики интерфейсов сети Modbus

Для успешного построения сетей Modbus на основе оборудования, выпускаемого различными производителями, спецификация на Modbus также предъявляет требования к *физическим* характеристикам сетевых интерфейсов.

В сетях Modbus TCP ведущее и ведомые устройства в современных условиях подключаются к сети посредством унифицированного телекоммуникационного разъема 8p8c (часто неверно называемого RJ-45). При этом на самих устройствах устанавливается розетка, а подключение осуществляется при помощи патч-кордов к сетевому коммутатору. Ряд производителей встраивает небольшие 2-портовые сетевые коммутаторы непосредственно в полевое устройство, что существенно упрощает прокладку сетевых соединений.

Для сетей Modbus RTU/ASCII могут применяться следующие типы физических сетевых интерфейсов:

- винтовые клеммы;
- разъем 8p8c;
- разъем DE-9.

В случае использования винтовых клемм производитель сетевого оборудования обязан описать подключение в документации. Применение разъемов 8p8c и DE-9 определяется документом Modbus over Serial Line [14].

При использовании унифицированного телекоммуникационного разъема 8p8c на ведомом устройстве устанавливается экранированная розетка. В случае применения разъема DE-9 на ведомом устройстве так же должна быть установлена экранированная розетка.

В виду того, что на физическом уровне в сетях Modbus RTU/ASCII обычно используется интерфейс ANSI TIA/EIA-485, предусматривающий как двухпроводное, так и четырехпроводное подключение, в спецификации на сети Modbus отдельно регламентируется назначение контактов разъемов 8p8c и DE-9 в каждом из этих случаев. В табл. 7.4 приведено назначение контактов разъема 8p8c для двухпроводной сети, в табл. 7.5 – для четырехпроводной, в табл. 7.6 – назначение контактов разъема DE-9 для двухпроводной сети, а в табл. 7.7 – для четырехпроводной.

Для минимизации ошибок при монтаже рекомендуется использовать провода определенных цветов для каждого типа сигналов (табл. 7.8).

Таблица 7.4

Конт.	Сигнал	Требование	Описание
3	PMC	Опция	Управление режимом порта
4	D1	Обязательно	Напряжение U_1
5	D0	Обязательно	Напряжение U_0
7	VP	Рекомендуется	Напряжение питания +5...+24V
8	Common	Обязательно	Общий

Таблица 7.5

Конт.	Сигнал	Требование	Описание
1	RXD0	Обязательно	Напряжение U_a'
2	RXD1	Обязательно	Напряжение U_b'
3	PMC	Опция	Управление режимом порта
4	TXD1	Обязательно	Напряжение U_a
5	TXD0	Обязательно	Напряжение U_b
7	VP	Рекомендуется	Напряжение питания +5...+24V
8	Common	Обязательно	Общий

Таблица 7.6

Конт.	Сигнал	Требование	Описание
1	Common	Обязательно	Общий
2	VP	Рекомендуется	Напряжение питания +5...+24V
3	PMC	Опция	Управление режимом порта
5	D1	Обязательно	Напряжение U_1
9	D0	Обязательно	Напряжение U_0

Таблица 7.7

Конт.	Сигнал	Требование	Описание
1	Common	Обязательно	Общий
2	VP	Рекомендуется	Напряжение питания +5...+24V
3	PMC	Опция	Управление режимом порта
4	RXD1	Обязательно	Напряжение U_b'
5	RXD0	Обязательно	Напряжение U_a'
8	TXD1	Обязательно	Напряжение U_a
9	TXD0	Обязательно	Напряжение U_b

Таблица 7.8

Сигнал	Цвет
D1 или TXD1	Желтый
D0 или TXD0	Коричневый
Common	Серый
RXD0	Белый
RXD1	Синий

Контрольные вопросы

1. Какие типы данных поддерживает протокол Modbus?
2. Какова разрядность регистров хранения Modbus?
3. В чем заключается основное различие в передаче данных между Modbus RTU и Modbus ASCII?
4. Укажите максимально адресуемое количество ведомых устройств в сети Modbus.
5. Каким образом в Modbus определяются действия, которые необходимо выполнить полевым устройствам сети?
6. Каким образом в сети Modbus TCP решается вопрос детектирования искажений передаваемых данных?
7. Что происходит, если ведомое устройство по каким-либо причинам не может выполнить запрос ведущего устройства?
8. Каким образом устройства Modbus объединяются в сеть?

8. ТОКОВАЯ ПЕТЛЯ

8.1. Общие сведения

Еще одним из распространенных интерфейсов, применяющимся для организации связи в промышленности является интерфейс типа «*токовая петля*» [4].

В 50-х годах XX века интерфейс токовая петля с уровнями сигналов 0 – 60 мА использовался в системах организации связи военных подразделений. Высокое энергопотребление послужило причиной того, что в 1962 г. был разработан телеграфный аппарат, в котором максимальный уровень тока был уменьшен до 20 мА. Соответственно, с тех же пор токовая петля используется для передачи данных от измерительных преобразователей в системах автоматики. Особое влияние на применение токовой петли в промышленных условиях оказало то, что такой тип связи, обладая низкой стоимостью реализации, высокой помехоустойчивостью, обеспечивает передачу сигналов на достаточно большие расстояния [18].

Высокая помехозащищенность интерфейса обусловлена тем, что информация передается не уровнем напряжения, а величиной силы тока, протекающего в контуре – петле.

В системах передачи данных уровнем напряжения на величину сигнала на приемной стороне оказывают влияние физические параметры линии связи, и прежде всего – активное сопротивление кабеля $R_{каб}$ (рис. 8.1). Во-первых, $R_{каб}$ зависит от протяженности линии связи, и определяет активные потери энергии. Соответственно, величина напряжения на нагрузке R_n для разной протяженности линии передачи будет отличаться при прочих равных параметрах. Во-вторых, активные сопротивления, как кабеля, так и нагрузки R_n , зависят от температуры, которая может меняться под воздействием различных факторов. Таким образом, для точной передачи сигнала необходимо постоянно подстраивать систему.

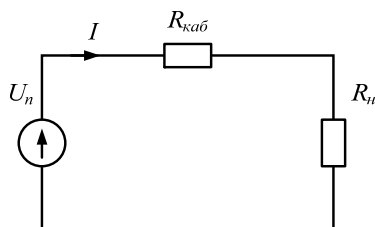


Рис. 8.1. Система связи на основе токовой петли

На рис. 8.2 представлена система связи на основе токовой петли. В отличие от рис. 8.1, в данной схеме используется управляемый стабилизированный источник тока на стороне передатчика, задающий передаваемое значение. Сила тока источника не будет зависеть ни от сопротивления нагрузки R_n , ни от сопротивления кабеля $R_{каб}$, ни от ЭДС индуктивной помехи $E_{инд}$, и будет одинаковой во всех неразветвленных участках цепи. Разве только емкостная наводка $E_{емк}$, которая приложена параллельно источнику тока и утечки в кабеле могут внести искажения в передаваемый сигнал. Использование экранированной витой пары во многом решает и эти проблемы.

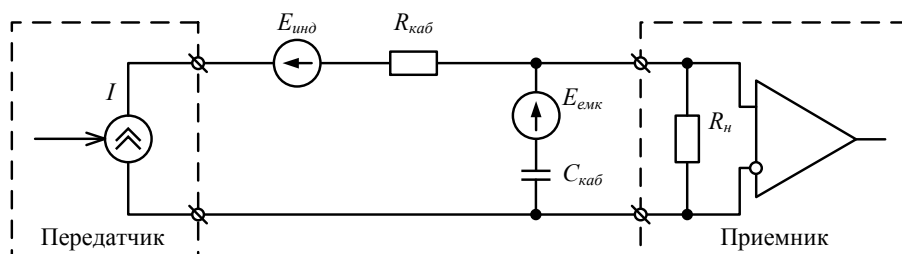


Рис. 8.2. Воздействие помехи на систему связи на базе токовой петли

В итоге основным недостатком токовой петли является низкая скорость передачи данных, которая проистекает из-за низкой скорости заряда емко-

сти кабеля $C_{каб}$ от источника тока, так как его максимальное значение ограничено, в отличие от тока, создаваемого источником напряжения. Например, если взять кабель симметричный с индивидуально экранированными жилами для одиночной прокладки Спецкабель КАСГЭфВ 1×2×0.78, то его погонная емкость 102 пФ/м, а погонное сопротивление 5.9 Ом/км [19]. Соответственно, при длине кабеля 1 км эквивалентная емкость составит 102 нФ, а суммарное сопротивление – 5.9 Ом. Так как время заряда емкости в цепях постоянного тока определяется выражением

$$t = RC, \quad (8.1)$$

то при входном сопротивлении приемника в 250 Ом сопротивление кабеля практически не оказывает влияния, и заряд паразитной емкости кабеля потребует

$$t = (R_n + R_{каб}) \times C_{каб} = (250 + 5.9) \times 102 \approx 26.1 \text{ мкс.}$$

Соответственно, скорость передачи данных ($C_{передачи}$) должна быть такой, чтобы ширина информационного импульса была более 26.1 мкс, то есть

$$C_{передачи} < \frac{1}{t} = \frac{1}{26.1} \approx 38300 \text{ бит/с.}$$

Ну а так как емкостная составляющая кабеля негативно сказывается на форму сигнала, то фактически для реального использования скорость передачи данных должна быть еще меньше, и обычно не превышает 9600 бит/с.

В промышленности интерфейс токовая петля применяется в двух версиях: аналоговая токовая петля – для передачи аналоговых сигналов от датчиков и исполнительных механизмов, и цифровая токовая петля – для передачи двоичных данных между терминальными устройствами.

8.2. Аналоговая токовая петля

Аналоговая токовая петля достаточно широко применяется в промышленности, но в основном для организации связи между различными измерительными преобразователями, исполнительными механизмами полевого уровня АСУТП и вычислителями (контроллерами, панелями оператора, промышленными ПК и т.д.).

ГОСТ 26.011–80 – «Средства измерений и автоматизации. Сигналы тока и напряжения электрические непрерывные входные и выходные» определяют различные виды токовых сигналов и систем передачи данных на их основе. В соответствии с этим стандартом могут использоваться системы с пределами изменения силы тока:

– 0...5 мА;

- $-5...+5$ мА;
- $0...20$ мА;
- $4...20$ мА;
- $-20...+20$ мА;
- $-100...+100$ мА,

однако наибольшее распространения к настоящему моменту времени имеют системы с токовым интерфейсом $4...20$ мА.

Логическое соответствие уровней сигналов измеряемым физическим величинам или формируемым сигналам может быть произвольным, но традиционно меньший уровень тока соответствует меньшему уровню измеряемого значения. Например, выходной сигнал уровнемера, контролирующего уровень заполнения резервуара, равный 4 мА, будет соответствовать пустому баку, а 20 мА – полностью заполненному. Значение силы тока для любого промежуточного значения также не сложно вычислить, так как такая зависимость будет линейной

$$X = \frac{X_{\max} - X_{\min}}{I_{\max} - I_{\min}} \times (I - I_{\min}), \quad (8.2)$$

где X – это некоторая измеряемая физическая величина, X_{\min} , X_{\max} – соответственно, минимальное и максимальное значение этой величины, I – измеренное значение силы тока, I_{\min} , I_{\max} – соответственно, нижняя и верхняя граница токового диапазона. То есть для заполненного наполовину резервуара ток в выходной цепи уровнемера будет

$$I = \frac{I_{\max} - I_{\min}}{X_{\max} - X_{\min}} \times X + I_{\min} = \frac{20 - 4}{100 - 0} \times 0.5 + 4 = 12 \text{ мА}.$$

Основными преимуществами аналоговой токовой петли с диапазоном $4...20$ мА являются:

- простота подключения оборудования – необходим кабель всего с двумя проводниками;
- высокая точность передачи сигнала – сила тока может контролироваться передающим устройством;
- независимость качества связи от длины линии, которая влияет только на максимальную скорость передачи данных;
- возможность идентификации аварийных ситуаций в линии связи – значения тока менее 3.8 мА свидетельствуют об обрыве линии связи, а ток свыше 20.5 мА – о коротком замыкании;
- возможность питания оборудования – полевые устройства могут потреблять ток до 4 мА для организации собственной работы;
- теоретически неограниченная дальность связи – максимальная протяженность линии связи ограничена лишь скоростью передачи данных и характеристиками самого кабеля связи.

На рис. 8.3 представлены два варианта построения аналоговой токовой петли. В первом варианте используется внешний источник питания. Обычно такой вариант используется, когда передатчик – это аналоговый датчик, а приемник – это контроллер. Источник питания E_{num} в такой схеме может выбираться исходя из требований решаемой задачи. Во втором случае источник питания встроенный, что более характерно для случая исполнительным механизмом (приемник) от аналогового выхода контроллера (передатчик).

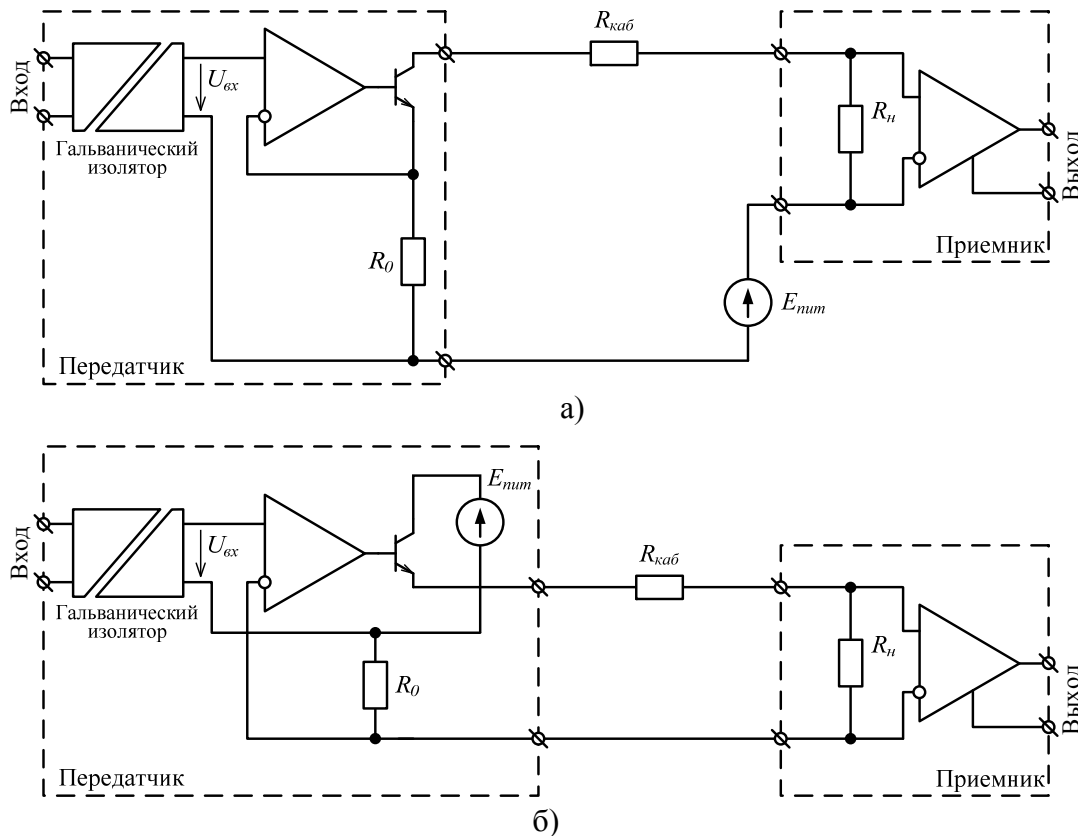


Рис. 8.3. Построение аналоговой токовой петли: а – с внешним источником питания, б – с внутренним источником питания

Принцип действия обеих схем заключается в том, что при бесконечно большом коэффициенте усиления операционного усилителя напряжение между его входами равно нулю, и поэтому ток через резистор R_0 равен $I_{R_0} = U_{ex} / R_0$. Так как у идеального операционного усилителя ток входов равен нулю, то в соответствии с первым законом Кирхгофа, ток через резистор R_0 будет равен току через нагрузку R_n (и сопротивление кабеля $R_{каб}$ конечно же). Значит, напряжение на входе приемника определяется как

$$I_{R_0} R_n = U_{ex} \frac{R_n}{R_0}. \quad (8.3)$$

Напряжение источника питания $E_{пит}$ выбирается таким, чтобы транзистор передатчика работал в ненасыщенном режиме, и можно было скомпенсировать падение напряжения в кабеле и на резисторах R_0 , R_n

$$E_{пит} > I(R_0 + R_{каб} + R_n) + U_{нас}, \quad (8.4)$$

где $U_{нас}$ – напряжение насыщения транзистора (1...2 В) [4].

В соответствии с ранее упомянутым ГОСТ 26.011–80 входное сопротивление приемника должно быть не более 250 Ом, а кабель КАСГЭфВ 1×2×0.78 длиной 1 км (как мы уже выяснили) имеет сопротивление 5.9 Ом. Тогда если взять $R_0 = 500$ Ом, то напряжение источника питания должно быть не менее 17 В. В промышленности самым распространенным номиналом источников питания постоянного тока является 24 В.

8.3. Цифровая токовая петля

Системы связи на основе цифровой токовой петли могут быть построены по тому же принципу, что и системы аналоговые, с упрощением аппаратуры как на передающей, так и принимающей стороне, так как нет необходимости обеспечивать передачу множества уровней значений, а достаточно всего лишь двух, соответствующих логическому нулю и логической единице. Тогда на принимающей стороне можно установить пару компараторов, настроенных на соответствующие уровни, и ток равный 4 мА может соответствовать логическому нулю, а ток 20 мА – логической единице.

Однако гораздо проще реализовать токовую петлю в этом случае в диапазоне 0...20 мА – можно использовать источник тока с не очень большим внутренним сопротивлением и низкой точностью (рис. 8.4).

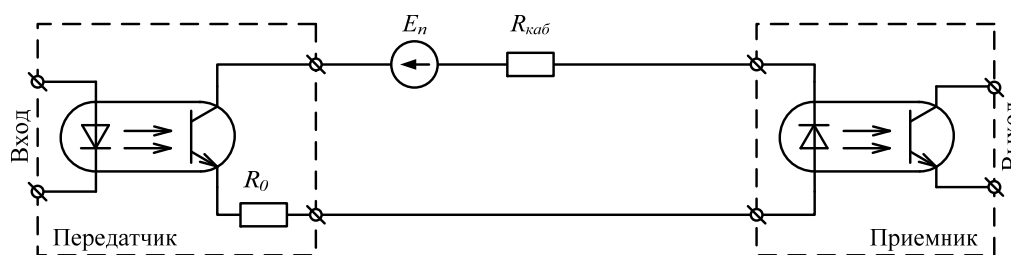


Рис. 8.4. Принцип реализации цифровой токовой петли

Единственное что нужно сделать для подобной схемы – подобрать сопротивление токоограничивающего резистора R_0 . И так как вся информация передается в цифровом виде, то следует выбирать кабель с меньшей погонной емкостью, чем при аналоговой передаче, что позволит увеличить скорость передачи, но увеличит активные потери в кабеле. Например, можно использовать широко применяющийся при прокладке компьютерных сетей кабель КВПУ–5е 4×2×0.52, у которого электрическая емкость

рабочей пары не более 56 пФ/м, но электрическое сопротивление 190 Ом/км [19]. Если максимальную силу тока в контуре принять 20 мА, то для источника питания 24 В для такого кабеля с учетом падения напряжения на открытых полупроводниковых элементах $R_0 \approx 1$ кОм.

Очевидно, что в случае применения как аналоговой, так и цифровой токовой петли, к одному передатчику может быть последовательно подключено несколько приемников одновременно (рис. 8.5). Это особенно удобно если требуется сигнал с датчика завести не только в контроллер системы управления, но и обеспечить индикацию показаний на локальных приборах.

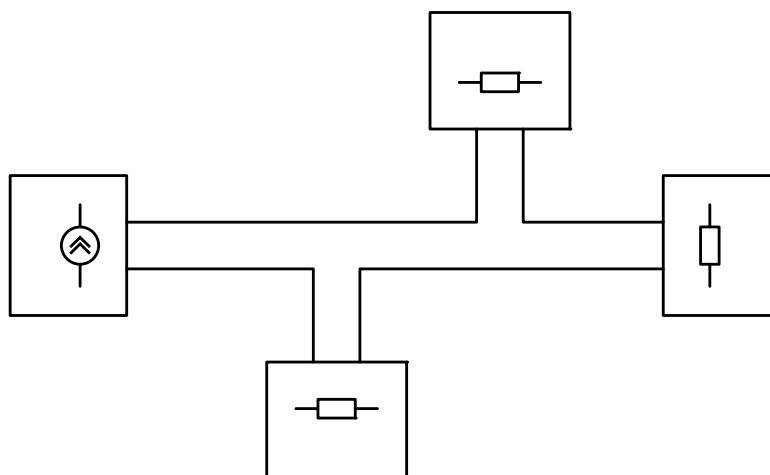


Рис. 8.5. Множество приемников в цепи токовой петли

Однако, как и любой другой интерфейс, токовая петля имеет ряд недостатков и ограничений. Основным из них является возможность передачи по одному кабелю только одного сигнала, что может стать проблемой при большом количестве устройств, так как это приведет не только к увеличению затрат на организацию линии связи, но параллельные линии могут создавать взаимные наводки, что негативно сказывается на помехоустойчивости. Другой серьезный недостаток также уже упоминался – низкая скорость передачи данных.

Несмотря на это интерфейс токовая петля до сих пор не утратил своей актуальности. В частности он применяется при построении сетей на основе протокола HART.

Контрольные вопросы

1. Какие типы токовой петли существуют?
2. В чем заключается основное преимущество интерфейса токовая петля? На каких принципах оно основано?
3. Какие преимущества обеспечивает перенос нижней границы токового диапазона аналоговой петли из 0 мА в 4мА?

9. HART-ПРОТОКОЛ

9.1. Общие сведения

HART – это аббревиатура от Highway Addressable Remote Transducer – магистральный адресуемый удаленный преобразователь. Стандарт HART был разработан в 1980 г. фирмой Rosemount Inc. в качестве проприетарного протокола цифровой связи для своих интеллектуальных полевых приборов [20]. Спецификации протокола были открыты в 1986 г., а возможности протокола были расширены путем последовательных пересмотров спецификаций. С 1993 г. развитием HART и его стандартизацией занимается HART Communication Foundation (HCF) – независимая некоммерческая организация, объединяющая производителей и пользователей HART по всему миру [21]. С 1 января 2015 г. HART Communication Foundation и Fieldbus Foundation объединились в более масштабную организацию FieldComm Group, в которую входят ведущие производители, университеты и исследовательские организации, совместно направляющие разработку и внедрение стандартов [22]. Членство открыто для всех, кто заинтересован в использовании технологий, но, тем не менее, спецификации протокола HART, определяющие все различные аспекты протокола и процедуры тестирования для подчиненных устройств, предоставляются в виде набора из 17 документов на коммерческой основе [23].

Термин HART-протокол (или протокол HART) – это достаточно устоявшееся выражение, которое, тем не менее, не совсем корректно, так как спецификации HART включают в себя не только протокол взаимодействия устройств, но и требования к аппаратуре канала связи.

Устройства, использующие протокол HART, используют как аналоговый стандартный токовый сигнал 4–20 мА, так и цифровой сигнал, то есть HART – это гибридный стандарт связи. Основная информация с полевых устройств (обычно это основная измеряемая величина – *первичная переменная*) передается в аналоговом виде $A(t)$ по токовой петле, а дополнительная информация (например, единицы измерения, диагностическая информация и т.д.) передается при помощи цифровых кодов импульсами тока $D(t)$, которые накладываются на основной сигнал (рис. 9.1). Передача таких импульсов осуществляется с использованием частотной манипуляции (FSK-сигнал) амплитудой около 1 мА. Цифровой сигнал передается двумя частотами, значения которых соответствуют американскому стандарту BELL 202 на телефонные каналы связи. Частота 1200 Гц используется для передачи значения логической единицы и 2200 Гц – логического нуля. При смене частоты фаза колебаний остается непрерывной. Поскольку среднее значение FSK-сигнала равно нулю, передача данных не влияет на достоверность передаваемого аналогового сигнала в токовой петле.

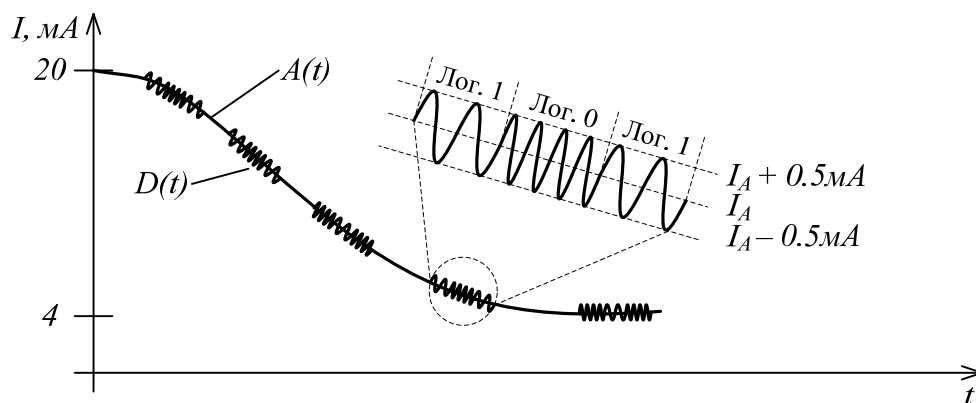


Рис. 9.1. HART-сигнал накладывается на аналоговый токовый сигнал

Протокол HART – это типичный протокол связи с явно выделенным мастером, реализующий принцип обмена «запрос-ответ». Во время нормальной работы время отклика составляет приблизительно 500 мс для каждого полевого устройства, не мешая передаче аналогового сигнала.

Одной из особенностей HART является то, что в одной цепи может существовать два ведущих устройства.

В качестве основного ведущего (*первичный мастер*) может выступать:

- DCS (распределенная система управления);
- ПЛК (программируемый логический контроллер);
- ПК (персональный компьютер).

Вторичный мастер – это обычно портативный переносной терминал, предназначенный для параметризации полевого устройства (рис. 9.2).

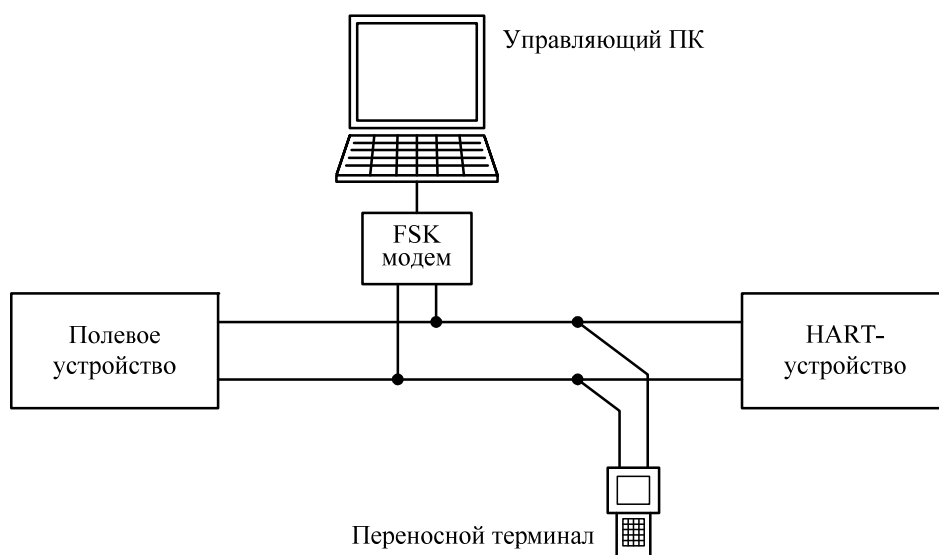


Рис. 9.2. Подключение HART-устройств

На рис. 9.3 приведена упрощенная структура взаимодействия HART-устройств, где цветом выделена часть, относящаяся к цифровому каналу передачи данных.

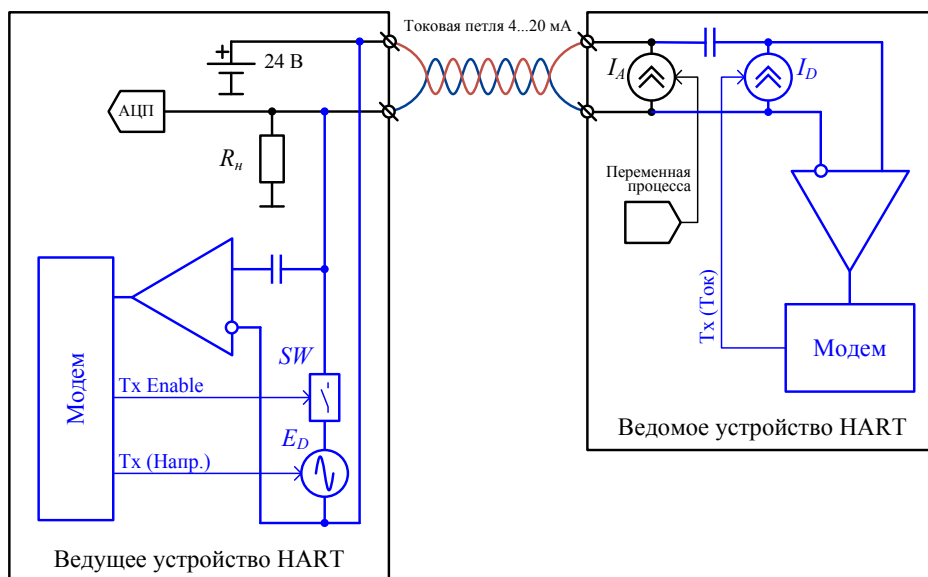


Рис. 9.3. Принцип работы HART-протокола на физическом уровне

Обмен информацией по аналоговому каналу ничем не отличается от рассмотренного ранее интерфейса токовая петля. Ведомое устройство (полевой датчик) изменяет значение силы тока в контуре при помощи управляемого источника тока I_A в зависимости от значения основной переменной процесса. Ведущее устройство (контроллер) HART обнаруживает это изменение тока путем измерения напряжения на измерительном резисторе R_n . Ток петли варьируется в диапазоне от 4 до 20 мА на частотах обычно ниже 10 Гц.

Цифровой сигнал HART накладывается на аналоговый сигнал 4–20 мА. Ведущее устройство HART сигналом Tx Enable через ключ SW подключает к токовой линии управляемый источник напряжения E_D , который генерирует синусоидальные сигналы с частотами, соответствующими передаваемым цифровым сигналам. Это напряжение амплитудой примерно 500 мВ, будучи приложенным к тому же резистору R_n , приводит к изменению тока в контуре, которое может считать ведомое устройство. Когда необходимо ведомому устройству осуществить передачу цифровых сигналов, то задействуется дополнительный источник тока I_D . Таким образом, принято считать, что ведущее устройство HART осуществляет цифровую передачу посредством напряжения, в то время как ведомое – током [27].

Традиционно передача цифровых данных HART осуществляется на скорости 1200 бит/с, что позволяет организовать 2–5 транзакций за секунду. Однако начиная со спецификации 6.0, HART поддерживает когерентную восьмипозиционную фазовую манипуляцию C8PSK, когда каждый передаваемый символ состоит из одного периода частотой 3200 Гц с одной из 8 дискретных фаз. Это позволяет на скорости 3200 бод одновременно передавать информацию о трех последовательных битах сообщения, что в

конечном итоге увеличивает скорость до 9600 бит/с, и позволяет организовать 10–12 транзакций в секунду [30]. Однако, наличие режима фазовой манипуляции не является обязательным для HART- устройств, в отличие от частотно-манипулированного.

Электрические характеристики интерфейса HART-устройств приведены в таблице 9.1.

Таблица 9.1

Сопротивление нагрузки R_n , Ом	230...1100
Сигнал от ведущего устройства, мВ	400...600
Сигнал от ведомого устройства, мВ	0,8...1,2
Минимальное значение сигнала ведомого устройства при $R_n = 230$ Ом, мВ	184
Максимальное значение сигнала ведомого устройства при $R_n = 1100$ Ом, мВ	1320
Чувствительность приемника, мВ	120...2000
Порог приемника, мВ	0...80

В качестве среды передачи сигнала используется кабель типа экранированная витая пара. В опасных зонах могут быть использованы барьеры искробезопасности, которые пропускают HART-сигнал [32].

Для облегчения подключения в полевых условиях устройства HART конструируются таким образом, чтобы их можно было подключать параллельно друг другу. Это устраняет необходимость разрывать цепь (контур) всякий раз, когда необходимо подключить коммуникатор HART для настройки или диагностики оборудования.

Как правило, инструкции производителя для коммуникаторов HART рекомендуют прямое параллельное соединение с приборами HART (рис. 9.4).

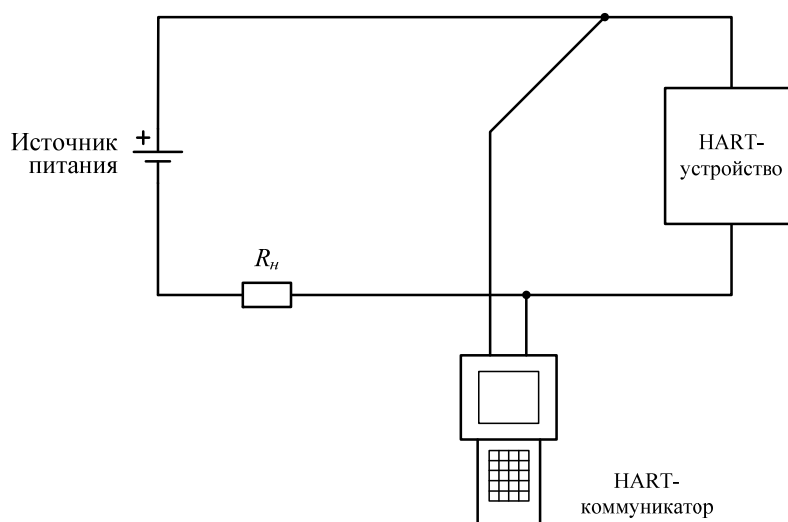


Рис. 9.4. Подключение HART-коммуникатора параллельно прибору

Однако также очень распространено и подключение коммуникатора параллельно с резистором, как показано на рис. рис. 9.5. Это связано с тем, что клеммы резистора обычно более доступны, нежели какие-либо другие.

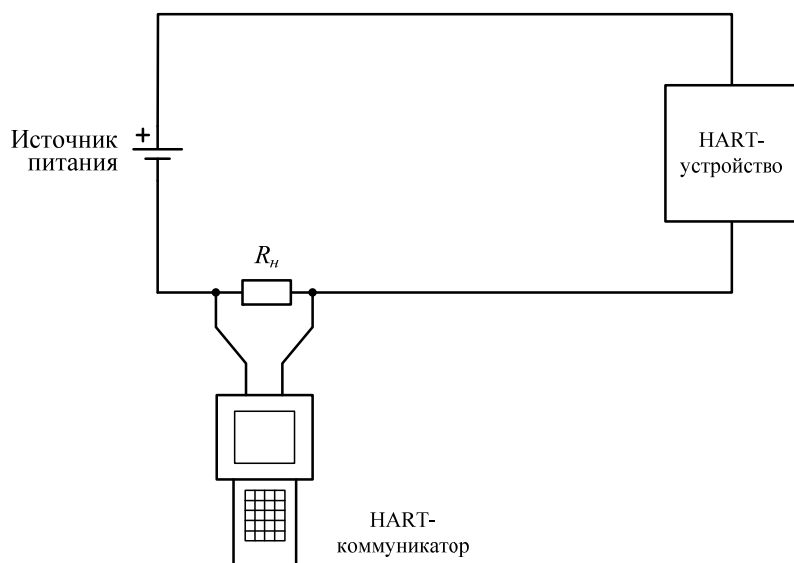


Рис. 9.5. Подключение HART-коммуникатора параллельно нагрузке

9.2. Топология сети

В сети HART может быть достаточно разнообразная топология: точка – точка, звезда, шина в зависимости от организованного режима передачи данных.

Режим передачи цифровой информации одновременно с аналоговым сигналом представлен на рис. 9.6. Обычно такой способ используется для построения систем управления с аналоговыми каналами.

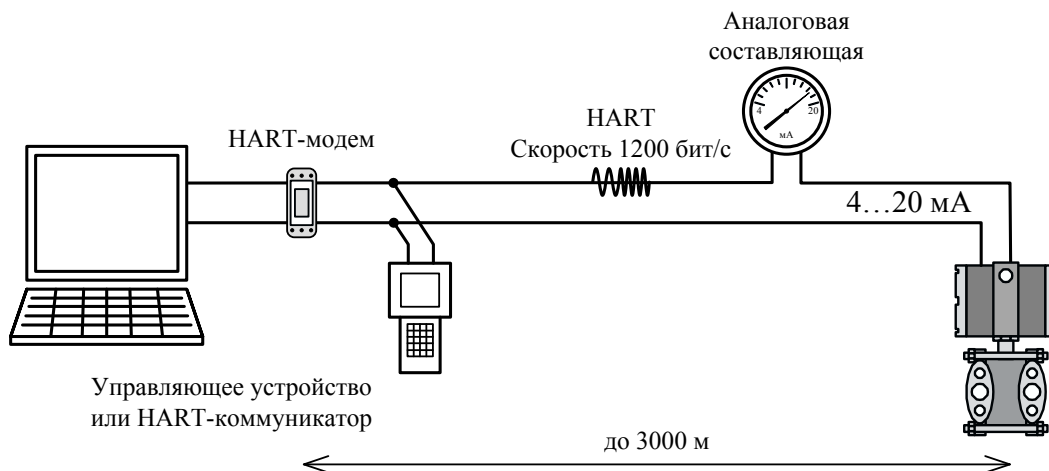


Рис. 9.6. Топология точка – точка

В этом режиме (точка – точка) в АСУТП используется аналоговая составляющая сигнала, а цифровой обмен по протоколу HART осуществляется посредством HART-коммуникатора или компьютера с использованием HART-модема и соответствующего программного обеспечения, и служит для удаленной настройки и конфигурирования полевого оборудования.

Разумеется, только одно HART-устройство может присутствовать в такой топологии, а использование HART-контроллеров при построении АСУТП в роли ведущего устройства неоправданно дорого.

Если в АСУТП используется несколько локальных контуров управления, использующих только один информационный сигнал для выработки управляющего значения, то в каждом контуре такая система управления может быть с успехом построена на интерфейсе токовая петля 4...20 мА.

В случае применения интеллектуальных полевых приборов HART как раз получится случай, отраженный на рис. 9.6. Когда таких приборов много, то настройка и конфигурирование при помощи HART-коммуникатора или модема требует последовательного подключения коммуникационного устройства к каждой линии 4...20 мА, идущей от соответствующих приборов, что неудобно. Выходом из такой ситуации является использование HART-мультиплексоров [33]. В этом случае (рис. 9.7) полевое оборудование по-прежнему использует токовый сигнал 4...20 мА для организации работы системы управления, а конфигурация приборов может осуществляться с использованием одного компьютера. При этом можно объединить в сеть около 500 приборов (например, 30 шт. 16-канальных мультиплексоров соединенных по RS-485) [32].

В шинной топологии (рис. 9.8) в многоточечном режиме к одной паре проводников может быть подключено до 15 полевых устройств HART (в соответствии со спецификацией HART 5, или до 63 устройств в соответствии со спецификацией HART 6, или до 64 – для версии HART 7).

В многоточечном режиме задействован только цифровой обмен данными. Аналоговый сигнал не используется, его значение фиксируется на минимальном значении – 4 мА и используется только для питания устройств. Информация о переменных процесса считывается по HART-протоколу. Так как в этом случае устройств в сети несколько, то для их идентификации используется уникальный идентификатор (адрес). В HART 5.0 приборы имеют адреса от 1 до 15, в версии 6.0 – от 1 до 63, а в версии 7.0 – от 0 до 63. Коммуникатор или система управления определяет все датчики, подключенные к линии, и может работать с любым из них.

В частном случае, при наличии только одного полевого устройства, многоточечный режим вырождается в режим «точка – точка». Более того, так как именно многоточечный режим позволяет использовать преимущества цифровой связи в полной мере, то именно этот режим и будет обсуждаться в дальнейшем.

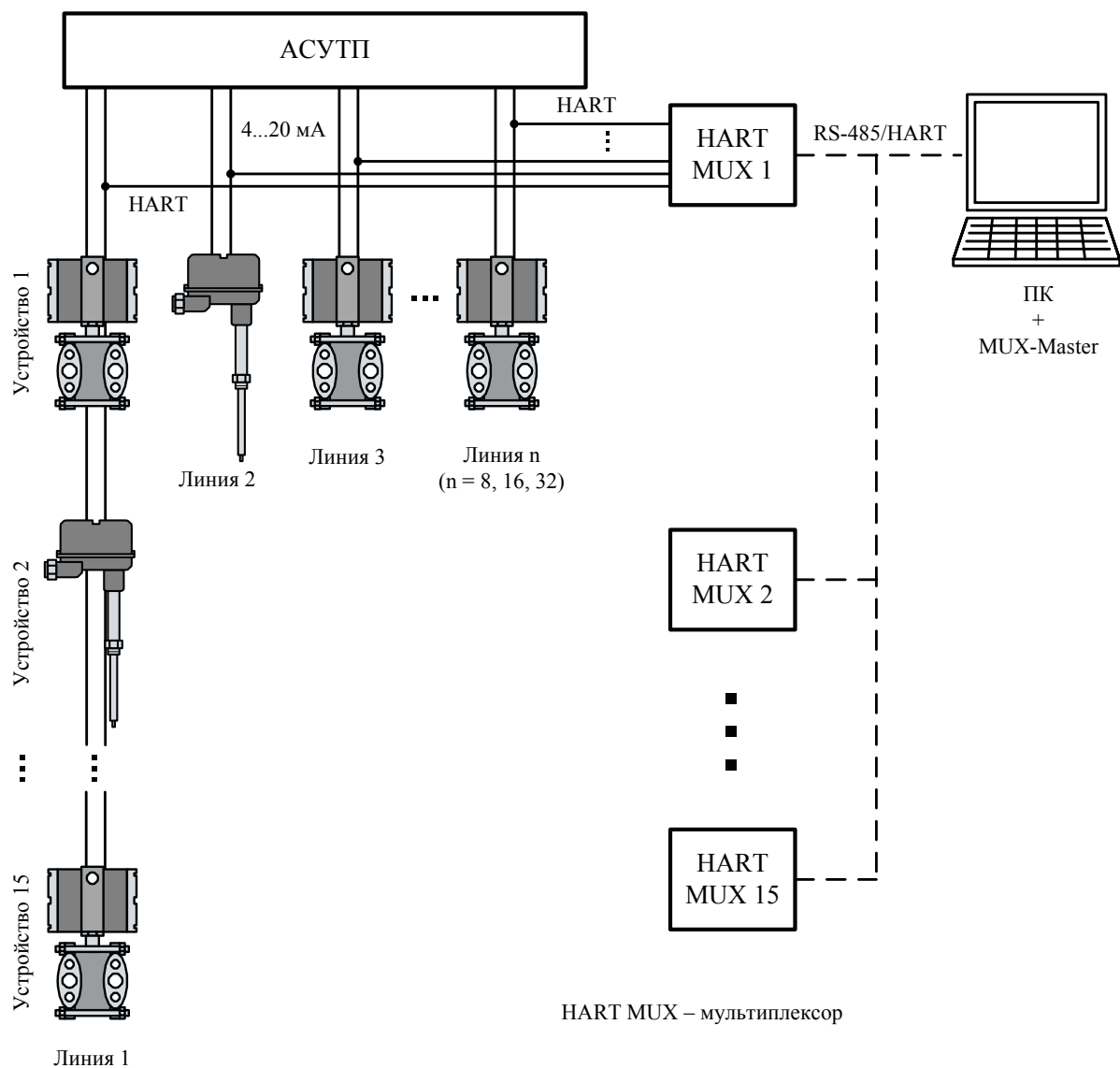


Рис. 9.7. Топология звезда

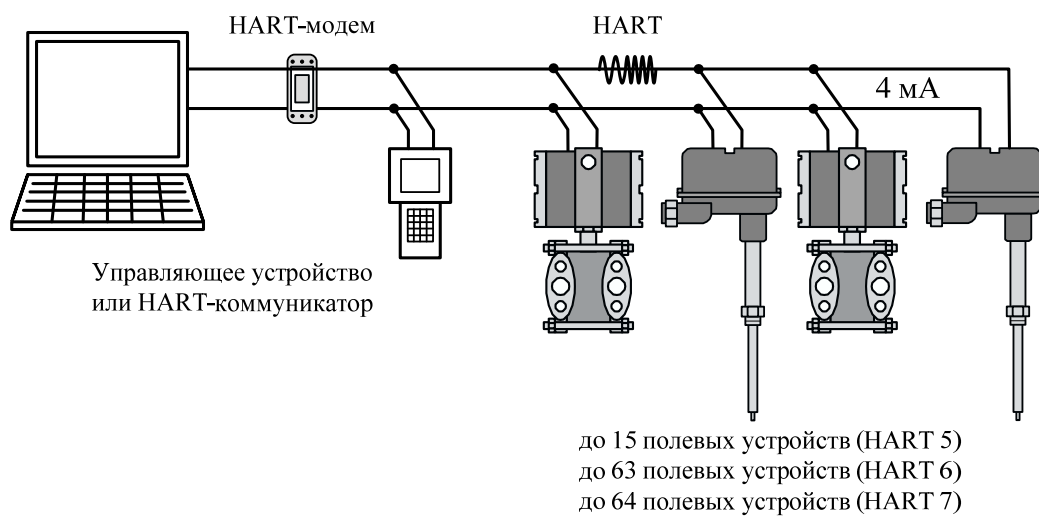


Рис. 9.8. Топология шина

Как было указано при обсуждении интерфейса токовая петля, для питания полевого оборудования с токовым выходом необходим или внешний источник, или он может быть встроен в сам полевой прибор. Соответственно, для HART ситуация аналогичная. Схема подключения двухпроводного передатчика HART приведена на рис. 9.9, а. Некоторые HART устройства могут иметь дополнительные линии подключения к источнику питания, то есть имеют активный источник тока. Схема подключения таких устройств приведена на рис. 9.9, б. На практике возможно организовать смешанное подключение (рис. 9.9, в). Верхние HART-устройства питаются от источника в двухпроводном контуре, а нижние, являющиеся активными источниками тока, питаются отдельно (на схеме не указано).

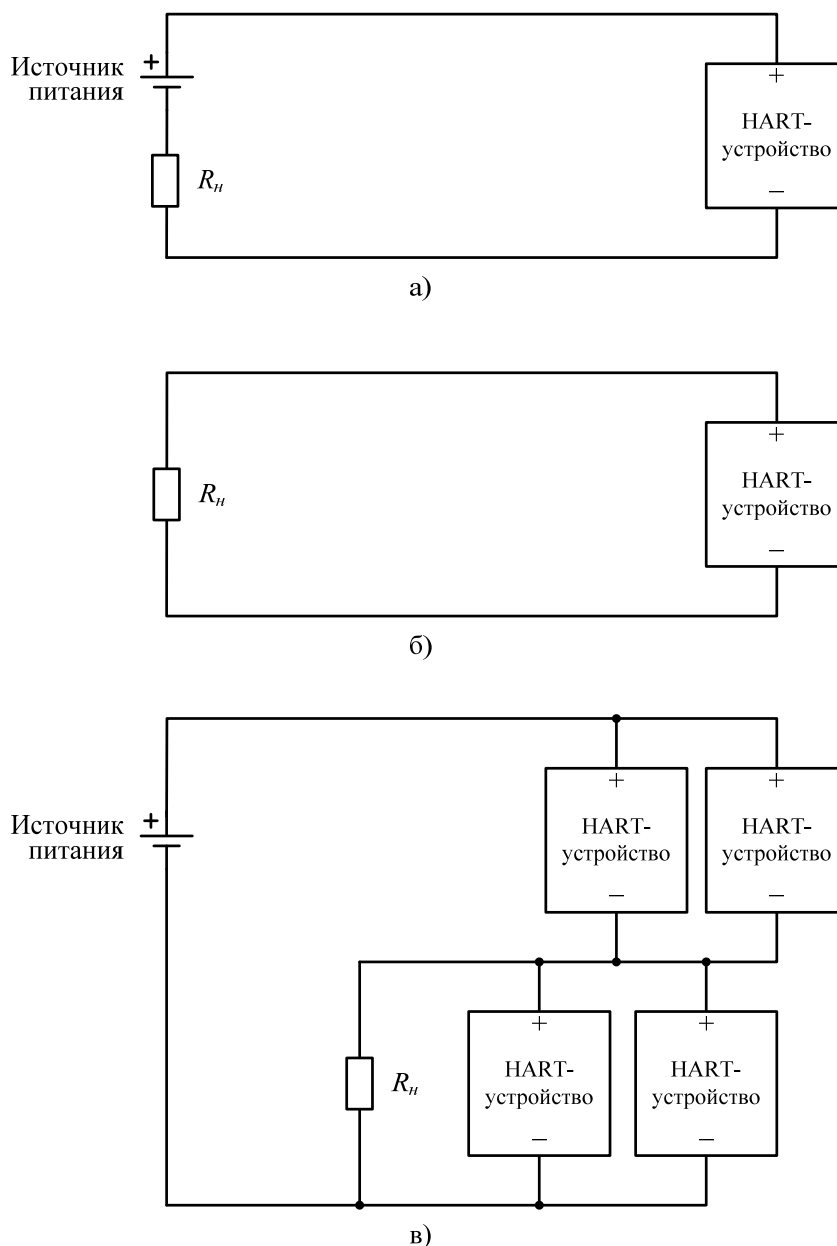


Рис. 9.9. Схема подключения HART-устройств

Организация такой схемы подключения требует трехпроводной линии, однако чаще всего используются две отдельные экранированные витые пары.

9.3. Формат сообщений

HART-сообщения кодируются сериями 8-битных символов, или байтов, которые передаются последовательно по линиям связи, с использованием метода передачи аналогичного UART: передается стартовый бит, 8 бит данных, бит контроля четности (с дополнением до нечетного числа единиц – odd parity) и стоповый бит. Скорость передачи данных определяется стандартом Bell 202 и равна 1200 бит/с (1200/801).

Надежность передачи данных по HART-протоколу обеспечивается различными мерами контроля, как на уровне байта, так и на уровне всего пакета. Частота возникновения ошибки на уровне передачи битов составляет одну ошибку на 10^5 переданных бит.

HART протокол является полудуплексным, соответственно, для нормального обмена данными используется режим запрос/ответ, когда ведущее устройство инициирует обмен, отправляет свое сообщение, и ожидает ответное или подтверждающее сообщение.

Некоторые HART-устройства поддерживают пакетный (или монопольный) режим связи (burst mode), при котором ведомое устройство циклически отправляет сообщение с запрошенными данными через короткие временные промежутки в 75 мс. Этим достигается высокая скорость передачи данных – до четырех сообщений в секунду, в то время как в обычном режиме – только два. В общем случае пакетный режим может быть полезен только при подключении одного ведомого HART-устройства в токовый контур, так как только одно устройство в токовой петле может находиться в таком режиме.

Формат HART-сообщения во всех режимах работы одинаков и приведен на рис. 9.10. В квадратных скобках на данном рисунке обозначены элементы, которые могут отсутствовать в зависимости от типа сообщения. Так, например, поле статуса ST отсутствует при передаче сообщения от ведущего устройства, и, наоборот, присутствует при передаче от ведомого к ведущему.

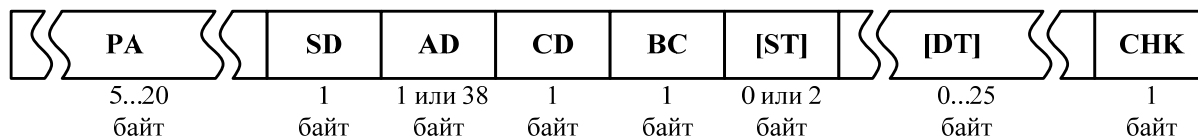


Рис. 9.10. Формат HART-сообщения

Преамбула (PA) состоит от 5 до 20 шестнадцатеричных символов 0xFF (в двоичном коде это все единицы), и позволяет приемнику синхронизировать частоту сигнала и входной поток байт.

Стартовый байт (ST) в HART-сообщениях определяет формат используемого кадра сообщения, источник сообщения, и режим передачи (нормальный или монополярный). Размерность поля – 1 байт. Ранние спецификации протокола HART (вплоть до версии HART 4) описывали только короткий кадр, который позволял адресовать только 15 устройств, так как использовалось всего 4 бита. В версии HART 5 был введен «длинный» формат кадра, в котором ведомые устройства имеют 38-разрядный уникальный идентификатор, состоящий из кода изготовителя оборудования, кода типа устройства, и идентификационного номера устройства (заводского номера).

Длинный формат кадра обеспечивает дополнительную защиту от ошибок при приеме, возникающих из-за внешних помех или перекрестных искажений. Кроме того, при этом расширяется адресное пространство HART-протокола, что позволяет значительно большему числу удаленных ведомых устройств подключиться к крупным сетям, использующим, например, общий канал радиосвязи.

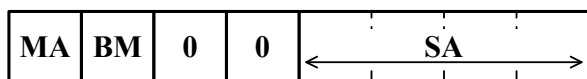
Новые ведомые устройства используют длинный формат кадра, в то время как более ранние устройства используют короткий формат. Для обеспечения совместимости ведущие устройства должны поддерживать оба формата, чтобы работать корректно как с уже существующими приборами, так и с новыми. Новые ведомые устройства должны по-прежнему выполнять команду 0 – «Считать уникальный идентификатор» в коротком формате, поскольку ответ на эту команду включает версии системы команд устройства.

Приемные устройства определяют начало сообщения по концу преамбулы (символов 0xFF), принимая стартовый символ в качестве первого символа сообщения. Он может принимать лишь несколько возможных значений, представленных в таблице 9.2.

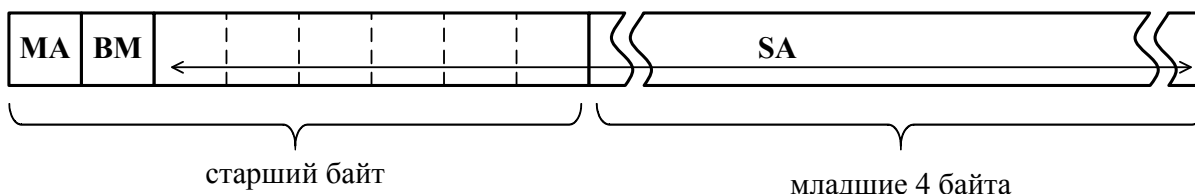
Таблица 9.2

Тип сообщения	Короткий кадр	Длинный кадр
От ведущего к ведомому	0x02	0x82
От ведомого к ведущему	0x06	0x86
В пакетном режиме от ведомого	0x01	0x81

Поле адреса (AD) содержит как адрес ведущего устройства, так и адрес ведомого. При организации связи с использованием короткого формата кадра поле адреса содержит 1 байт, а при использовании длинного – 5 байт (рис. 9.11).



а)



б)

Рис. 9.11. Формат поля адреса: а – короткого кадра, б – длинного кадра

В любом формате самый старший разряд (МА) поля адреса содержит информацию о типе ведущего устройства. Значение 1 – это первичное ведущее устройство, 0 – вторичное ведущее устройство. Для сообщений в пакетном режиме разряд МА попеременно устанавливается в 0 и 1, давая возможность каждому ведущему устройству вмешаться в работу пакетного режима.

Следующий разряд (ВМ) используется для идентификации пакетного режима (0 – нормальный режим, 1 – пакетный).

Далее два разряда зарезервированы и не используются, их значения всегда 0.

Для короткого формата кадра последние 4 разряда (SA), позволяющие задать значения в диапазоне от 0 до 15, определяют короткий адрес устройства. Нулевое значение адреса, аналогично Modbus, означает широко-вещательный адрес, и сообщение получают и обрабатывают все устройства в сети.

В формате длинного кадра адрес ведомого устройства занимает 38 бит (рис. 9.11, б).

Поле команды (CD) имеет размер 1 байт и содержит целое число в диапазоне от 0 до 253 (0xFD), которое задает одну из HART-команд. Значение 254 (0xFE) определено как код расширения. В этом случае за байтом CD следует другой байт, что в совокупности позволяет кодировать более 256 команд при необходимости. Существует три категории команд:

- универсальные – команды, которые должны обрабатываться любым HART-устройством;
- распространенные – команды, используемые лишь в тех устройствах, которые реализуют соответствующую функцию (например, выполнить сброс);

- специфические – команды, которые ограничиваются конкретной моделью или типом HART-устройства.

Полученный ведомым устройством код команды передается обратно в ответном сообщении.

Количество байт (BC) – это поле размером 1 байт, которое содержит целое число от 0 до 27, обозначающее количество байт, при помощи которых формируется оставшаяся часть сообщения (исключая поле контрольной суммы СНК). Указанные значения образуются из-за ограничений на длину поля данных в 25 байт. С помощью поля BC осуществляется проверка целостности сообщения и определяется его конец.

Поле статуса (ST) включается только в ответные сообщения от ведомых устройств. Оно состоит из 2 байт, сообщающих об ошибках коммуникации, статусе полученного сообщения (например, «устройство занято» или «команда не распознана») и оперативном состоянии устройства. Старший бит первого байта поля статуса определяет какая именно информация содержится в поле. Если этот бит установлен в 1, то в поле статуса содержится информация, связанная с приемом сообщений, а отдельные разряды первого байта указывают какие именно ошибки были диагностированы во время передачи (все разряды второго байта сброшены в 0). HART-устройство не возвращает данные, если были обнаружены ошибки связи. Кодирование статуса ошибок приема/передачи приведено в таблице 9.3.

Таблица 9.3

Бит	Наименование	Описание
<i>Первый байт статуса</i>		
6	Ошибка четности	Четность одного или более байтов полученных устройством была неправильна
5	Ошибка переполнения	Ошибка — по меньшей мере один байт данных в приемном буфере микросхемы UART был перезаписан до того как он был прочтен
4	Ошибка формирования кадра	Стоповый бит одного или более байтов полученных устройством не были обнаружены UART
3	Ошибка контрольной суммы	Четность по длине, вычисленная устройством, не такая как Байт Четности по длине в конце сообщения
2	(зарезервировано)	Зарезервировано, всегда равен нулю.
1	Ошибка переполнения буфера	Сообщение было слишком длинным для буфера принимающего устройства
0	(не определено)	Нет определения этого бита на сегодня
<i>Второй байт статуса</i>		
7...0		Все значения 0

Если старший бит поля статуса сброшен в 0, то оставшиеся разряды 6...0 первого байта содержат статус полученной команды (таблица 9.4), а во втором байте могут содержаться сведения о неисправностях полевых устройств (таблица 9.5).

Таблица 9.4

Значение	Описание
0	Ошибка, не характерная для команд
1	(не определено)
2	Неверный выбор
3	Переданный параметр слишком велик
4	Переданный параметр слишком мал
5	Получено слишком мало байт данных
6	Ошибка команды, специфической для датчика
7	В режиме, защищенном от записи
8	Неудача при обновлении Обновление в процессе работы Значение установлено равным ближайшему возможному
9	Параметр процесса слишком велик Нижнее значение процесса слишком велико Не в режиме фиксированного тока
10	Параметр процесса слишком мал Нижнее значение процесса слишком мало Не поддерживается режим моноканала
11	В режиме моноканала Неверный код переменной датчика Верхнее значение диапазона слишком велико
12	Неверный код единиц измерения Верхнее значение диапазона слишком мало
13	Оба значения диапазона выходят за ограничения
14	Введенное верхнее значение диапазона выходит за ограничение Интервал слишком мал
16	Доступ ограничен
32	Устройство занято
64	Команда не задействована

Таблица 9.5

Бит	Наименование	Описание
7	Неисправность прибора	Устройство обнаружило аппаратную ошибку или сбой. Дополнительная информация может быть получена по команде 48 – Считать дополнительный статус
6	конфигурация изменена	Сигнализирует, что с датчиком были произведены команды записи или изменения параметров
5	Холодный старт	С прибора было снято питание, которое затем было восстановлено, что привело к сбросу параметров настройки. Этот флаг может быть также установлен после Основного Сброса или Самотестирования. Ведущее устройство должно сначала сбросить этот флаг командой 38.
4	Доступно больше статусов	Может быть получено больше информации о статусе устройства, чем это возможно при помощи поля ST. Необходимо задействовать команду 48 – Считать дополнительный статус
3	Ток фиксирован	Аналоговый токовый выходной сигнал по первичной переменной зафиксирован на требуемом значении и не отражает измеренное прибором значение
2	Насыщение аналогового выхода	Выходной сигнал по первичной переменной достиг насыщения, то есть пределов шкалы, и больше и не отражает измеренное прибором значение
1	Вторичная переменная вне диапазона	Вторичный параметр процесса, измеряемый датчиком, вышел за пределы, допустимые для датчика. Для дополнительной информации может использоваться команда 48 – Считать дополнительный статус
0	Первичная переменная вне диапазона	Основной параметр процесса (первичная переменная), измеряемый датчиком, вышел за пределы, допустимые для датчика

Поле данных (DT) может достигать максимальной длины 25 байт, однако не все команды и ответы на них содержат данное поле. Данные могут

быть в форме целых чисел без знака, чисел с плавающей точкой (IEEE754) или строк символов ASCII. Число байт и конкретный тип данных, используемый для каждого элемента, указывается для каждой команды.

Поле контрольной суммы (CHK) составляет 1 байт, который рассчитывается как продольная четность байт (операция «Исключающее ИЛИ») всех предшествующих ему байт в сообщении, начиная со стартового байта включительно. Используется для проверки целостности принятого кадра данных.

9.4. Команды HART

Универсальные команды HART-протокола приведены в таблице 9.6, а распространенные команды в таблице 9.7. По возможности, приведены все доступные команды, однако за более подробным описанием назначения отдельных байт данных следует обращаться к документации на соответствующие полевые приборы. Обозначение типов данных, использованных в таблицах 9.6 – 9.7 приведено в таблице 9.8.

Таблица 9.6

Универсальные команды

№	Команда	Данные (тип данных)	
		Запрос	Ответ
0	Считать уникальный идентификатор	нет	Байт 0 254 (расширение)
			Байт 1 Код изготовителя
			Байт 2 Код типа устройства
			Байт 3 Число преамбул
			Байт 4 Версия универсальных команд
			Байт 5 Версия специфических команд
			Байт 6 Версия программного обеспечения
			Байт 7 Версия аппаратного обеспечения
			Байт 8 (H) Флаги функций устройств
			Байт 9–11 (B) Идентификатор устройства
1	Считать первичную переменную	нет	Байт 0 Код единиц измерения первичной переменной
			Байт 1–4 (F) Значение первичной переменной
2	Считать ток и процент диапазона	нет	Байт 0–3 (F) Ток, мА
			Байт 4–7 (F) Процент диапазона

Продолжение таблицы 9.6

№	Команда	Данные (тип данных)	
		Запрос	Ответ
3	Считать ток и значение четырех (предопределенных) переменных	нет	Байт 0–3 (F) Ток, мА
			Байт 4 Код единиц измерения 1-й переменной
			Байт 5–8 (F) Значение 1-й переменной
			Байт 9 Код единиц измерения 2-й переменной
			Байт 10–13 (F) Значение 2-й переменной
			Байт 14 Код единиц измерения 3-й переменной
			Байт 15–18 (F) Значение 3-й переменной
			Байт 19 Код единиц измерения 4-й переменной
			Байт 20–23 (F) Значение 4-й переменной
6	Записать сетевой адрес устройства	Байт 0 Адрес устройства	как в запросе
11	Считать уникальный идентификатор, ассоциированный с заданным тегом	Байт 0–5 (A) Тег	как для команды 0
12	Считать сообщение	нет	Байт 0–23 (A) Сообщение
13	Считать тег, дескриптор, дату	нет	Байт 0–5 (A) Тег
			Байт 6–17 (A) Дескриптор
			Байт 18–20 (D) Дата
14	Считать информацию сенсора первичной переменной	нет	Байт 0–2 Серийный номер сенсора
			Байт 3 Код единиц измерения
			Байт 4–7 (F) Верхний предел измерения сенсора
			Байт 8–11 (F) Нижний предел измерения сенсора
			Байт 12–15 (F) Чувствительность сенсора

Окончание таблицы 9.6

№	Команда	Данные (тип данных)	
		Запрос	Ответ
15	Считать выходную информацию о первичной переменной	нет	Байт 0 Код выбора тревоги
			Байт 1 Код функции преобразования первичной переменной
			Байт 2 Код единиц измерения
			Байт 3–6 (F) Верхний предел первичной переменной
			Байт 7–10 (F) Нижний предел первичной переменной
			Байт 11–14 (F) Величина демпфирования первичной переменной, ед./с
			Байт 15 Код защиты от записи
			Байт 16 Код изготовителя
16	Считать составной номер устройства	нет	Байт 0–2 Составной номер
17	Записать сообщение	Байт 0–23 Сообщение (A)	как в запросе
18	Записать тег, дескриптор, дату	Байт 0–5 Тег (A)	как в запросе
		Байт 6–17 Дескриптор (A)	
		Байт 18–20 (D) Дата	
19	Записать составной номер устройства	Байт 0–2 Составной номер	как в запросе

Таблица 9.7

Распространенные команды

№	Команда	Данные (тип данных)			
		Запрос		Ответ	
33	Считать переменные датчика	Байт 0	Код переменной датчика для слота 0	Байт 0	Код переменной датчика для слота 0
		Байт 1	Код переменной датчика для слота 1	Байт 1	Код единиц измерения для слота 0
		Байт 2	Код переменной датчика для слота 2	Байт 2–5 (F)	Значение переменной для слота 0
		Байт 3	Код переменной датчика для слота 3	Байт 6	Код переменной датчика для слота 1
				Байт 7	Код единиц измерения для слота 1
				Байт 8–11 (F)	Значение переменной для слота 1
				Байт 12	Код переменной датчика для слота 2
				Байт 13	Код единиц измерения для слота 2
				Байт 14–17 (F)	Значение переменной для слота 2
				Байт 18	Код переменной датчика для слота 3
				Байт 19	Код единиц измерения для слота 3
				Байт 20–23 (F)	Значение переменной для слота 3
34	Записать величину демпфирования	Байт 0–3 (F)	Величина демпфирования, с	как в запросе	
35	Записать значения диапазона	Байт 0	Код единиц измерения диапазона	как в запросе	
		Байт 1–4 (F)	Верхнее значение диапазона		
		Байт 5–8 (F)	Нижнее значение диапазона		
36	Установить верхнее значение диапазона (эквивалентно нажатию кнопки SPAN)	нет		нет	
37	Установить нижнее значение диапазона (эквивалентно нажатию кнопки ZERO)	нет		нет	

Продолжение таблицы 9.7

№	Команда	Данные (тип данных)	
		Запрос	Ответ
38	Сбросить флаг «Конфигурация изменена»	нет	нет
39	Управление ЭСППЗУ	Байт 0 Код управления 0 – прожечь ЭСППЗУ 1 – скопировать ЭСППЗУ в ОЗУ	как в запросе
40	Войти/выйти из режима фиксированного тока	Байт 0–3 (F) Ток, мА или 0 – выйти из режима	как в запросе
41	Выполнить самодиагностику датчика	нет	нет
42	Выполнить сброс главного устройства	нет	нет
43	Установить ноль первичной переменной	нет	нет
44	Записать единицы измерения первичной переменной	Байт 0 Код единиц измерения первичной переменной	как в запросе
45	Настроить ноль ЦАП	Байт 0–3 (F) Измеренный ток, мА	как в запросе
46	Настроить коэффициент усиления ЦАП	Байт 0–3 (F) Измеренный ток, мА	как в запросе
47	Записать функцию преобразования	Байт 0 Код функции преобразования	как в запросе
48	Считать дополнительный статус	нет	Байт 0–24 Дополнительный статус. Реальный размер поля данных и кодирование поля зависит от оборудования
51	Записать присвоения динамической переменной	Байт 0 Код первичной переменной датчика	как в запросе
		Байт 1 Код 2-й переменной датчика	
		Байт 2 Код 3-й переменной датчика	
		Байт 3 Код 4-й переменной датчика	
52	Установить ноль переменной датчика	Байт 0 Код переменной датчика	как в запросе
53	Записать единицы измерения датчика	Байт 0 Код переменной датчика	как в запросе
		Байт 1 Код единиц измерения переменной датчика	

Окончание таблицы 9.7

№	Команда	Данные (тип данных)			
		Запрос		Ответ	
54	Считать информацию о переменной датчика	Байт 0	Код переменной датчика	Байт 0	Код переменной датчика
				Байт 1–3	Серийный номер сенсора
				Байт 4	Код единиц измерения переменной датчика
				Байт 5–8 (F)	Верхний предел переменной датчика
				Байт 9–12 (F)	Нижний предел переменной датчика
				Байт 13–16 (F)	Величина демпфирования переменной датчика, с
55	Записать величину демпфирования переменной датчика	Байт 0	Код переменной датчика	как в запросе	
		Байт 1–4 (F)	Величина демпфирования переменной датчика, с		
56	Записать серийный номер сенсора переменной датчика	Байт 0	Код переменной датчика	как в запросе	
		Байт 1–3	Серийный номер сенсора переменной датчика		
57	Считать тег, дескриптор и дату	нет		Байт 0–5 (A)	Тег
				Байт 6–17 (A)	Дескриптор
				Байт 18–20 (D)	Дата
58	Записать тег, дескриптор и дату	Байт 0–5 (A)	Тег	как в запросе	
		Байт 6–17 (A)	Дескриптор		
		Байт 18–20 (D)	Дата		
59	Записать число преамбул в ответе	Байт 0	Число преамбул в ответе	как в запросе	
108	Номер команды в пакетном режиме	Байт 0	Номер команды в пакетном режиме	как в запросе	
109	Управление пакетным режимом	Байт 0	Код управления пакетным режимом	как в запросе	

Таблица 9.8

Обозначение	Тип данных
A	Упакованная ASCII строка (4 символа 3 байтами)
B	Побитовые флаги
D	Дата (3 байта: день, месяц, год – 1900)
F	Число с плавающей точкой (4 байта, IEEE–754)
H	xxxxx ууу (xxxxx = версия аппаратного обеспечения, ууу = код формирования физических сигналов)
(нет)	8-, 16- или 24-разрядные целые числа, включая перечислимые табличные значение

9.5. Язык описания устройств

HART-устройство может иметь уникальный набор команд, устанавливаемых изготовителем или специфическую последовательность команд, необходимых, например, для выполнения процедуры калибровки. Ведущее устройство должно знать эти команды. Один из возможных вариантов передачи описания команд от изготовителя пользователю - с помощью текстового описания в руководстве по эксплуатации. Другим путем является описание устройства с помощью специального языка описания устройств – Device Description Language (DDL), который распространяется и поддерживается FieldComm Group.

Описание устройства на языке DDL представляет собой текстовый файл, который может быть прочитан компилятором языка и преобразован в двоичный формат, воспринимаемый ведущим HART-устройством. Описание устройства может быть представлено компилятором в форме, удобной для системного интегратора. Применение DDL устраняет необходимость написания кода для общения с HART-устройствами.

Язык DDL поддерживается не всеми производителями и пока не получил достаточно широкого распространения.

Контрольные вопросы

1. Каким образом информация от полевых HART-устройств доставляется в АСУТП?
2. Каким способом кодируются данные, передающиеся в цифровом виде в сетях HART?
3. Допускается ли несколько ведущих устройств в сети HART? Какие условия должны быть соблюдены для этого?
4. Каково сопротивление нагрузки, использующееся в сетях HART?
5. Какие виды команд HART существуют? Для чего они предназначены?

10. AS-ИНТЕРФЕЙС

10.1. Общие сведения

Традиционные способы подключения дискретных датчиков и исполнительных механизмов в АСУТП предполагают, что к каждому такому устройству подводится отдельный кабель, содержащий как минимум пару проводов. Как уже отмечалось в разделе 1, в современных условиях затраты, связанные с кабельной продукцией: приобретение, монтаж, эксплуатация становятся достаточно высокими.

В конце 1980-х начале 1990-х группа компаний разработала технические требования к последовательному интерфейсу передачи дискретных сообщений от датчиков и исполнительных механизмов к вычислительным управляющим устройством. По мнению разработчиков, применение систем, построенных на основе такого интерфейса, должно упростить структуру кабельного хозяйства предприятия и снизить затраты на технические средства и монтажные работы. В результате получился AS-интерфейс, или AS-i (Actuators/Sensors interface – интерфейс исполнительных устройств и датчиков). Оригинальная спецификация была опубликована в 1994 г. и имеет версию 2.04. В 1998 г. спецификация была обновлена до версии 2.11. К 2005 г. возможности протокола связи существенно расширились, что было закреплено в ныне действующей спецификации 3.0 [34]

AS-интерфейс – это открытая промышленная сеть нижнего уровня, которая предназначена для организации связи с исполнительными устройствами и датчиками в соответствии с требованиями европейских нормативов EN 50295 и международного стандарта IEC 62026–2 [35].

AS-интерфейс позволяет решить задачу подключения датчиков и приводов 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.

Оболочка кабеля AS-интерфейса имеет специальный профиль, исключая возможность неправильного подключения сетевых компонентов. 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.

При наличии в системе специальных модулей AS-интерфейс позволяет подключать к системе управления обычные широко распространенные на рынке датчики и исполнительные механизмы. Следующим шагом в развитии сетевых технологий на базе AS-интерфейса стало включение инте-

гральных микросхем ведомого устройства непосредственно в электронную часть датчиков и исполнительных механизмов. Для таких интеллектуальных устройств становится возможной реализация удаленного параметрирования и диагностики без использования дополнительных связей.

Гибкость управления системой достигается за счет применения различных ведущих устройств. Функции ведущих устройств могут выполнять программируемые логические контроллеры, промышленные компьютеры или модули связи с сетями более высокого уровня – ModBus, Interbus, CANopen, PROFIBUS, DeviceNet и т.п.(рис. 10.1).



Рис. 10.1. Схема подключения устройств к AS-интерфейсу

Взаимодействие узлов сети AS-интерфейса осуществляется по принципу «Клиент – Сервер», при этом используется централизованный метод доступа к среде передачи данных. Соответственно, одно единственное ведущее устройство в сети может взаимодействовать с некоторым количеством ведомых устройств. В оригинальной спецификации 1994 г. (версия 2.0) можно подключить максимум 31 ведомое устройство, которое могло иметь до 4 дискретных входов и 4 дискретных выходов. По спецификации версии 2.1 стандарта на AS-интерфейс количество ведомых устройств в одной сети увеличено вдвое до 62 за счет разделения адресного пространства ведущего сетевого устройства на две подобласти: А и В. Такие А/В-устройства могут иметь до 4 входов и 3 выходов. Спецификация 3.0 допускает работу AS-интерфейса с ведомыми А/В-устройствами, в составе которых могут быть до 8 входов и 8 выходов. Соответствующее максимальное число входов и выходов, а также другие технические данные системы на базе AS-интерфейса приведены в таблице 10.1.

Таблица 10.1

	Спецификация	
	версия 2.0	версия 2.1
Топология	Магистральная, древовидная, звездообразная и др.	
Число ведомых устройств	До 31	До 62
Число подключаемых датчиков и исполнительных устройств	До 4 датчиков и 4 исполнительных устройств на одно ведомое устройство До 124 датчиков и 124 исполнительных устройств на сегмент (одно ведущее устройство)	До 4 датчиков и 3 исполнительных устройств на одно ведомое устройство До 248 датчиков и 186 исполнительных устройств на сегмент (одно ведущее устройство)
Максимальная протяженность линии связи	Без повторителей/расширителей до 100 м С повторителями/расширителями до 300 м	
Линия связи	Двухжильный кабель (2×1,5мм ²) специального профиля без экрана, с параллельно идущими жилами, для одновременной передачи данных и питания	
Метод подключения	Подключение сетевых устройств выполняется методом прокалывания изоляции кабеля AS-интерфейса	
Электропитание	Через шину AS-интерфейса: 2.8 (ном.), 8 А (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, циклическая передача данных в память центрального процессора контроллера/компьютера или пересылка в обратном направлении	
Функции управления ведущего устройства	Инициализация сети, идентификация ведомых устройств, ациклическая передача значений параметров ведомым устройствам, диагностика передачи данных и ведомых устройств, сообщения об ошибках	
Степень защиты оборудования системы	До IP67	

AS-интерфейс использует метод доступа к ведомым устройствам, основанный на их циклическом опросе (polling). 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 ведомым устройством.

При использовании оборудования AS-интерфейса версии 2.1 время цикла опроса также не будет превышать 5 мс, но только в том случае, если ведомые устройства принадлежат одной подобласти адресного пространства А или В. В случае использования всего адресного пространства, ведомые устройства подобластей А и В обслуживаются по очереди: сначала опрашиваются все ведомые устройства подобласти А, а затем подобласти В, после чего опрос повторяется далее. Таким образом, в этом случае суммарное время обслуживания всех ведомых устройств не превышает 10 мс.

Ведущие устройства, соответствующие спецификации 2.1 (или 3.0) могут опрашивать каждую из подобластей ведомых А/В-устройств, а ведущие устройства спецификации 2.0 могут опрашивать только 31 ведомое устройство, даже если это ведомые А/В-устройства (доступна подобласть адресного пространства А).

10.2. Топология сетей AS-i

Сети AS-интерфейса могут использовать достаточно разнообразные топологии: «шина», «звезда», «кольцо» или «дерево» (рис. 10.2).

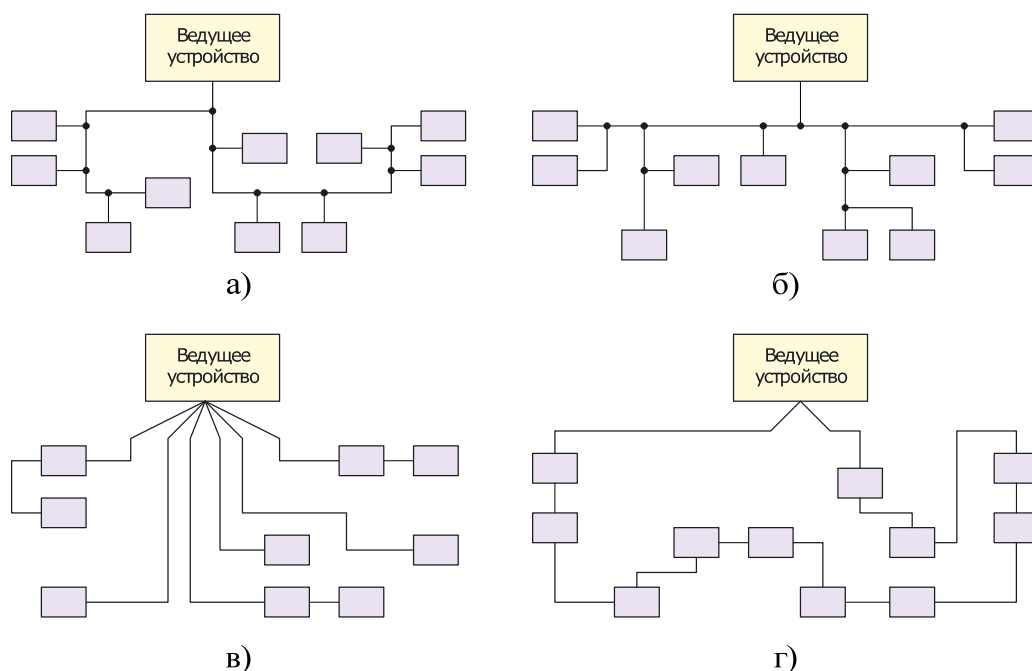


Рис. 10.2. Различные формы топологии сети AS-интерфейса:

а – шина, б – дерево, в – звезда, г – кольцо

Главное ограничение, которое необходимо учитывать, заключается в том, что общая протяженность кабеля не должна превышать 100 м. Под общей длиной понимается сумма длин всех ветвей сегмента сети, обслуживаемого одним ведущим устройством. При помощи специальных расширителей можно увеличить протяженность кабеля или разделить ветвь на группы. Кроме того, в сетях AS-i можно использовать до двух повторителей, при помощи которых суммарная протяженность линий связи может достигать 300 м, однако каждый сегмент требует отдельного источника электропитания. Разумеется, должны использоваться только специальные источники, предназначенные для работы с AS-интерфейсом.

10.3. Передача данных

Процедура передачи данных с использованием AS-интерфейса осуществляется в режиме «Запрос – Ответ» (рис.10.3) с использованием кадров запроса ведущего устройства и кадров ответа ведомых устройств. Кадры имеют фиксированную структуру и, соответственно, фиксированное время передачи. Кадры данных, аналогично протоколу Modbus RTU, разделяются при помощи интервалов тишины, однако эти интервалы выражены относительно времени передачи одного бита данных.

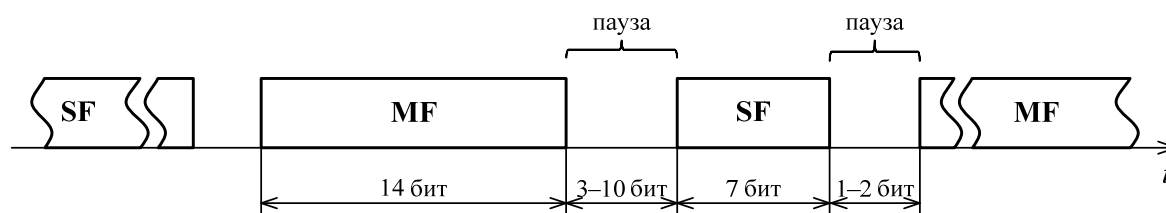


Рис. 10.3. Структура протокола передачи данных AS-интерфейса

Все запросы ведущего устройства (MF) имеют длину 14 бит, все ответы ведомого устройства (SF) занимают 7 бит. Время передачи одного бита – 6 мкс. Пауза ведущего устройства может занимать по времени от 3 до 10 интервалов времени передачи бита. Если ведомое устройство было синхронизировано, то есть уже принимало участие в информационном обмене, то ведомое устройство начинает ответную передачу через 3 такта. Если ведомое устройство по каким-либо причинам не было синхронизировано (например, это первый запрос в адрес данного ведомого устройства или запрос после воздействия помехи), то требуется на два такта больше для выполнения процедуры синхронизации. В том случае, когда ведущее устройство не приняло стартовый бит ответного сообщения ведомого устройства даже после 10 тактов, ведущее устройство должно диагностировать отсутствие связи с ведомым устройством, и может сформировать и отпра-

вить следующий запрос, например, ведомому устройству с более высоким адресом.

Структура кадров передачи данных приведена на рис. 10.4.

ST	SD	A4	A3	A2	A1	A0	I4	I3	I2	I0	PB	EB
----	----	----	----	----	----	----	----	----	----	----	----	----

а)

ST	I3	I2	I0	PB	EB
----	----	----	----	----	----

б)

Рис. 10.4. Структура кадров: а – ведущего устройства, б – ведомого

Назначение полей кадров запроса приведено в таблице 10.2. В таблице 10.3 приведено назначение полей кадра ответа.

Таблица 10.2

Обозн.	Наименование	Назначение
ST	Стартовый бит	Маркирует начало запроса ведущего устройства: 0 – действительный стартовый бит, 1 – не допускается
SB	Управляющий бит	Обозначает тип запроса (запрос данных, параметра и т.п.): 0 – запрос данных, параметра, адреса, 1 – запрос команды
A4...A0	Адрес	Адрес вызываемого ведомого устройства (5 разрядов)
I4...I0	Информация	5 информационных разрядов, соответствующих типу запроса (параметр, команда)
PB	Бит паритета	Сумма всех «1» в запросе ведущего устройства должна быть четной
EB	Конечный бит	Формирует конец запроса ведущего устройства: 0 – не допускается, 1 – признак конечного бита

Таблица 10.3

Обозн.	Наименование	Назначение
ST	Стартовый бит	Маркирует начало ответа ведомого устройства: 0 – действительный стартовый бит, 1 – не допускается
I3...I0	Информация	4 информационных разряда указывают, например, параметр, установленный порт ввода-вывода, статус ведомого устройства
PB	Бит паритета	Сумма всех «1» в ответе ведомого устройства должна быть четной
EB	Конечный бит	Формирует конец ответа ведомого устройства: 0 – не допускается, 1 – признак конечного бита

В таблице 10.4 представлены все допустимые запросы и команды ведущего устройства для спецификации 2.0. Остальные кодовые комбинации зарезервированы для будущего использования.

Таблица 10.4

	ST	SB	Адрес	Информация	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
Сброс ведомого устройства	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	1	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

- Запрос и запись данных осуществляется с помощью команды «Data Exchange». Этот запрос ведущего устройства используется, чтобы передать последовательность битов на выходы данных запрашиваемого ведомого устройства и затем прочесть ответ ведомого устройства, содержащий биты логического состояния входов данных ведомого устройства. Направление порта данных ведомого устройства (вход, выход, двунаправленный порт) задается при установке конфигурации ввода-вывода.
- Команда «Записать параметр» («Write Parameter»). Эта команда ведущего устройства устанавливает выходы параметров ведомого устройства.
- Команда «Присвоение адреса» («Address Assignment»). Данная команда позволяет ведущему устройству устанавливать новое значение адреса ведомого устройства.
- Команда «Сброс ведомого устройства» («Reset Slave»). С помощью этой команды ведомое устройство устанавливается в исходное состояние (аналогично сбросу при включении питания). Ведомое устройство квитирует (подтверждает) безошибочный прием этой команды ответом 0x06. Процесс сброса должен длиться максимум 2 мс.
- Команда «Удалить адрес» («Delete Address»). Эта команда служит для предварительного зануления рабочего адреса ведомого устройства и требуется в связке с командой «Присвоение адреса», потому что команда «Присвоение адреса» («Address Assignment») может быть выполнена только для ведомого устройства с адресом 0x00. Например, если ведомое устройство с установленным адресом 0x10 перепрограммируется на новый адрес 0x07, эту процедуру можно выполнить только с помощью последовательности команд «Delete Address (0x10)» и «Address Assignment (0x07)». В этом случае ведомое устройство подтверждает безошибочный прием первой команды ответом 0x06, после чего оно становится доступным под адресом 0x00; только после этого с помощью второй команды можно записать новый адрес 0x07. Ранее записан-

ный старый адрес можно восстановить с помощью команды «Reset Slave».

- Команда «Считать конфигурацию ввода-вывода» («Read I/O Configuration»). Ведущее устройство может с помощью этой команды считать установленную конфигурацию каналов ввода-вывода ведомого устройства. Код конфигурации передается в ответе ведомого устройства на данную команду и служит совместно с ответом на команду «Read ID_Code» для однозначной идентификации ведомого устройства.
- Команда «Считать идентификационный код» («Read ID-Code»; в версии 2.1 для 62 ведомых устройств используются дополнительно два кода: ID1-Code, ID2-Code). С помощью данной команды считывается ID-код ведомого устройства. Четырехбитовый ID-код программируется один раз при изготовлении ведомого устройства и не может в последующем изменяться. Он служит для обозначения принадлежности ведомого устройства определенному установленному профилю (profile) – совокупности принятых для данного типа устройств формализованных описаний. Все ведомые устройства, параметры и данные которых не соответствуют какому-либо профилю, должны иметь идентификационный код 0xF.
- Команда «Считать статус» («Read Status»). Считывается регистр состояния соответствующего ведомого устройства.

AS-интерфейс изначально был ориентирован на передачу двоичной информации, соответственно, все датчики и исполнительные механизмы, применявшиеся в AS-интерфейсе, оперировали только дискретными значениями. Свойства, параметры, функциональные особенности полевых устройств имеют формализованное описание в виде так называемого *профиля*. Профили ведомых устройств являются «жесткими», так как они не подлежат изменению. Соответствие между профилями и устройствами устанавливается с помощью ID-кода, который совместно с конфигурацией ввода-вывода вводится изготовителем в каждое ведомое устройство и тоже больше не изменяется.

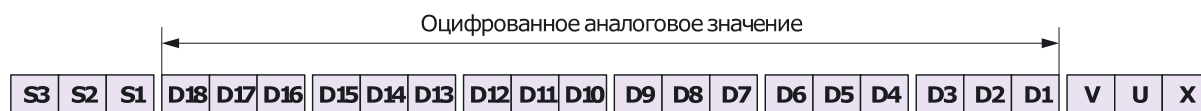
На рис. 10.5 приведен пример фрагмента профиля для интеллектуального датчика с функцией контроля работоспособности. Пользователь-программист знает, что при включении датчика с этим профилем сигнал включения всегда передается в качестве первичной полезной информации как бит данных D0, а информация об отказе, если таковая требуется, всегда определяется состоянием бита данных D1.

Битами параметров определены постоянные функции, которые могут быть и нереализованными. Код конфигурации ввода-вывода и идентификационный код представлены в шестнадцатеричной форме, и никакое ведомое устройство с другим профилем не может иметь такую же комбинацию этих кодов.

Бит данных	Тип	Назначение
D0	I	Сигнал включения датчика
D1	I	Индикатор аварийного состояния
D2	I	Индикатор готовности
D3	O	Разрешение самотестирования
Бит параметров		
P0		Переключение рабочей частоты
P1		Инвертирование сигнала включения
P2		Диапазон срабатывания (у индуктивных датчиков приближения)
		Задержка импульсов (у оптических датчиков)
P3		Специальная функция
Применение битов с D1 по D3 и с P1 по P3 является опцией (дополнительной возможностью).		
I/O-Code:	Hex 1	Шестнадцатеричный код конфигурации ввода-вывода
ID-Code:	Hex 1	Шестнадцатеричный идентификационный код

Рис. 10.5. Пример фрагмента профиля интеллектуального датчика с функцией контроля работоспособности

После введения спецификации 2.1 для передачи оцифрованных аналоговых величин был разработан сложный профиль (S-7.1), который предназначен для передачи данных длиной более 4 битов. Так как, прежде всего речь идет об использовании этого профиля для передачи аналоговых величин, то он был назван аналоговым профилем. Передача аналоговой величины выполняется по определенной команде от ведущего устройства. Ведомое устройство инициализирует внутренний регистр-фиксатор данных с новым значением и посылает биты статуса назад (рис. 10.6). Каждый новый запрос требует передачи следующих трех битов цифрового кода аналоговой величины, таким образом за 8 циклов передается максимально возможное значение аналоговой величины, регламентированное профилем, быстродействием канала (600 бит/с) и количеством циклов. Допускается прерывать передачу и тем самым передавать аналоговые величины с меньшим разрешением.



Условные обозначения: S1...S3 – биты статуса; D1...D18 – биты данных; V – знак; U – переполнение; X – резервные биты

Рис. 10.6. Передача значений оцифрованного аналогового сигнала

10.4. Модуляция

Взаимодействие между оборудованием в сетях AS-интерфейса осуществляется по специальным незранированным кабелям. По этим же кабелям осуществляется и питание самих устройств. Для уверенной передачи данных AS-интерфейс использует на физическом уровне ВОР особый метод модуляции – Alternating Puls Modulation (APM, рис. 10.7).

Биты передаваемой последовательности кодируются манчестерским кодом (когда каждое изменение передаваемого сигнала приводит к фазовой инверсии сигнала). При этом происходит формирование тока передачи, который в линии AS-интерфейса благодаря имеющейся распределенной индуктивности создает дифференциальные уровни напряжения. Каждое увеличение тока передачи ведет к появлению отрицательного, а понижение – положительного импульса напряжения. Приемники AS-интерфейса обнаруживают эти сигналы напряжений и преобразуют в исходную последовательность бит, соответствующую исходной.

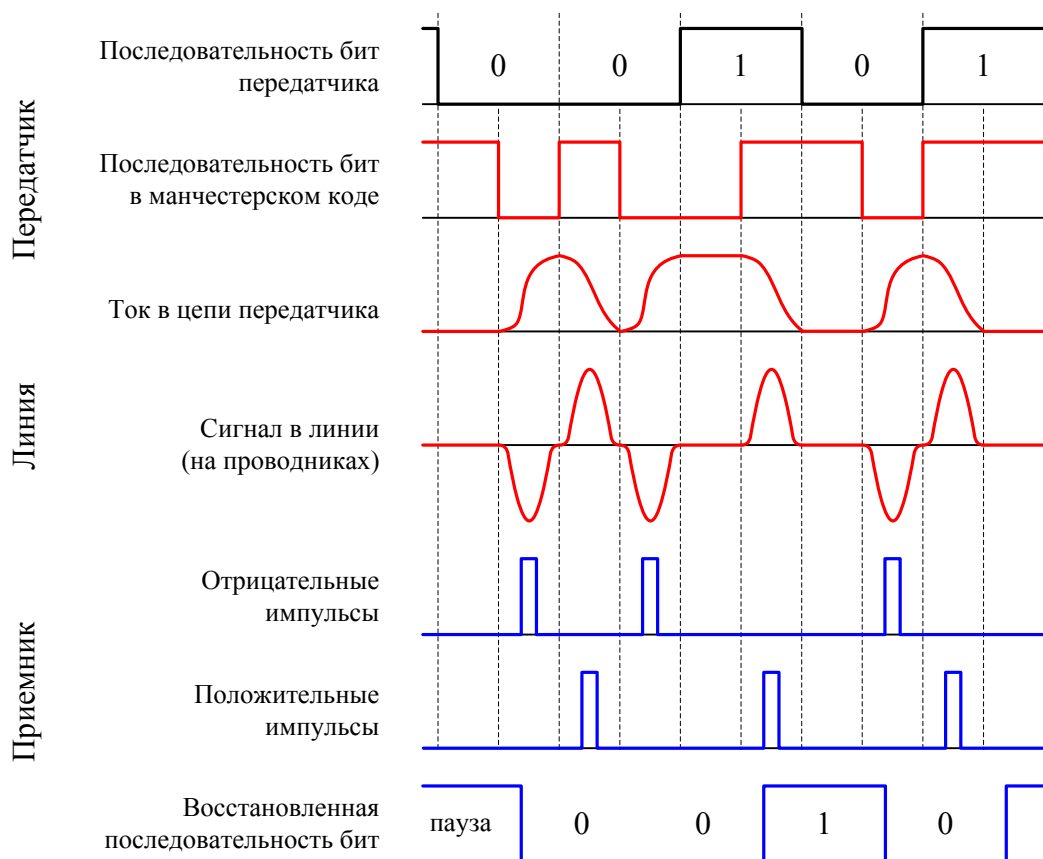


Рис. 10.7. Альтернативная импульсная модуляция AS-интерфейса

10.5. Полевые устройства

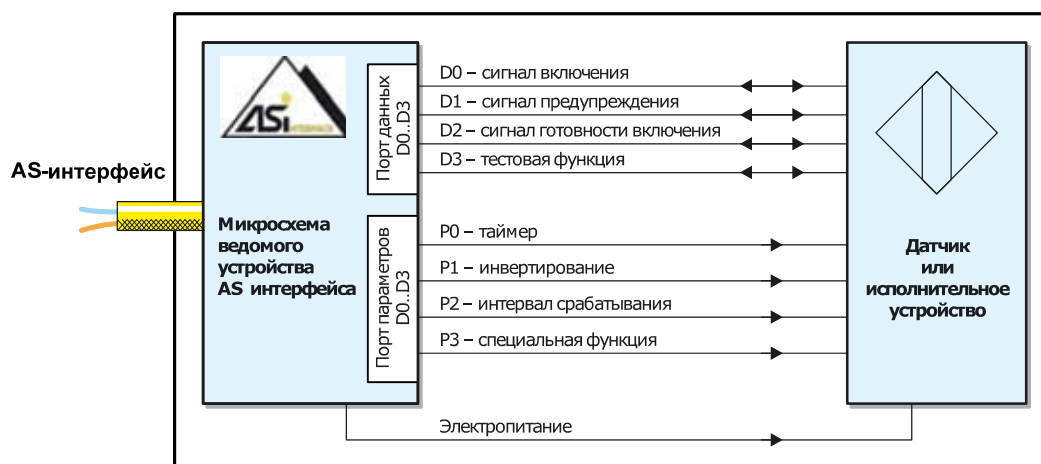
Электронная часть ведомых устройств AS-интерфейса должна быть компактной и дешевой, так как она фактически встраивается непосредственно в датчик или исполнительное устройство. Это возможно только при использовании специализированных микросхем с высокой степенью интеграции. Через интегральную микросхему ведомого устройства двоичные датчики и исполнительные устройства подключаются к сети AS-интерфейса. Специализированные микросхемы (ASIC) обеспечивают полевое оборудование питанием от сети, распознают переданную от ведущего устройства информацию и посылают в ответ собственные данные.

Соответственно возможно два варианта использования таких микросхем в сетях AS-i:

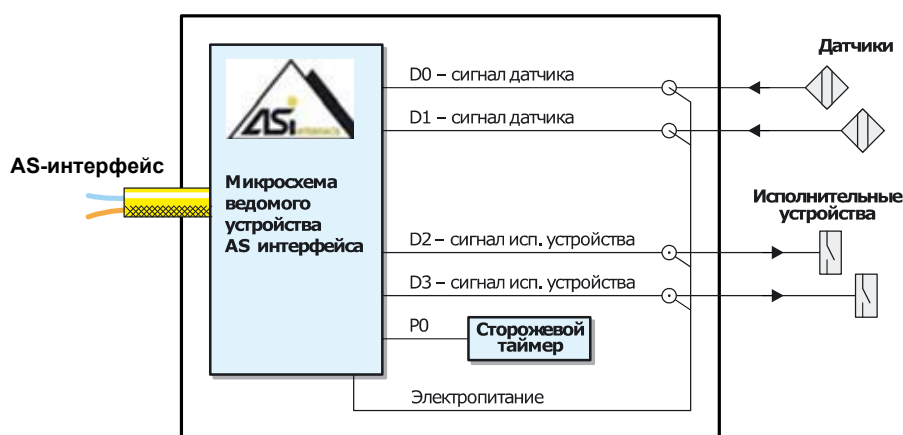
- чип ведомого устройства AS-интерфейса встраивается прямо в датчик или исполнительное устройство, в результате чего получается устройство с интегрированным AS-интерфейсом (рис. 10.8, а);
- чип ведомого устройства AS-интерфейса встраивается в коммуникационный модуль таким образом, что к нему можно подключать обыкновенный датчик или исполнительное устройство, которые характеризуются как устройства с внешним AS-интерфейсом (рис. 10.8, б).

В каждом цикле базовой спецификации передаются 4 бита данных от ведущего устройства последовательно к каждому ведомому и обратно. Необходимые для этого порты данных каждой микросхемы можно конфигурировать отдельно как входные, выходные или двунаправленные порты. Конфигурация портов ведомых устройств устанавливается в соответствии с так называемой конфигурацией ввода-вывода. По команде «Write Parameter» ведомое устройство получает от ведущего 4 бита данных, соответствующих значению параметра. С их помощью можно управлять особыми функциями ведомого устройства. Установка кодов параметров производится ациклично, причем в одном цикле AS-интерфейса она может быть выполнена только для одного ведомого устройства.

Прежде чем технологическое оборудование с системой управления на базе AS-интерфейса будет запущено в эксплуатацию, необходимо для всех ведомых устройств предусмотреть и установить адреса. Для устройств, имеющих встроенную микросхему AS-интерфейса, адрес записывается в EEPROM микросхемы. Адресация выполняется с помощью специальных команд ведущего устройства через линию AS-интерфейса. При выполнении адресации на линии может находиться только одно ведомое устройство с нулевым адресом. Программирование адреса может автоматически проходить непосредственно через ведущее устройство в режиме ввода сети в эксплуатацию посредством функции «Address Exchange» («Изменить адрес»).



а)



б)

Рис. 10.8. Варианты построения полевых устройств:
а – с интегрированным интерфейсом, б – с внешним интерфейсом

В качестве другой возможности можно использовать специальный прибор адресации, который позволяет установить адрес в ручном режиме, с помощью его клавиатуры можно инициировать считывание адреса подключенного ведомого устройства и отображение его на индикаторе. Это особенно важно для реализации режима проверки и диагностики сетевых устройств. Принцип работы заключается в том, что всем адресам ведомых устройств отсылаются телеграммы с запросом «Read I/O Configuration» («Считать конфигурацию ввода-вывода») до тех пор, пока искомое ведомое устройство не ответит, после чего его адрес можно изменить с клавиатуры и задействовать ту же функцию «Address Exchange» («Изменить адрес»).

Электропитание полевых устройств AS-интерфейса должно осуществляться от специализированных источников питания, так как AS-интерфейс предполагает совместное использование линий связи, как для передачи информации, так и для передачи энергии, обеспечивающей функционирование полевого оборудования. По этой причине для питания сетевых ком-

понентов AS-интерфейса не могут использоваться обычные стабилизированные блоки питания.

Блоки питания AS-интерфейса – это импульсные блоки питания с выходным напряжением ≈ 30 В, отличающиеся высокой стабильностью выходного напряжения и низким уровнем пульсаций, которые формируют напряжение питания электроники ведущего и ведомых сетевых устройств, а также всех подключенных датчиков. В зависимости от модификации номинальный выходной ток одного блока питания может лежать в пределах от 2.4 до 8 А. Источник должен быть оснащен защитами от длительного короткого замыкания и перегрузок или же в составе сети AS-интерфейса могут использоваться специализированные модули контроля коротких замыканий.

Цепи нагрузки (выходов ведомых устройств), как правило, получают питание от вспомогательного блока питания. Для подключения этих цепей используется отдельный кабель (например, черный профилированный кабель AS-интерфейса).

На рис. 10.9 схематически показано подключение источника питания к двухпроводной симметричной линии передачи данных.

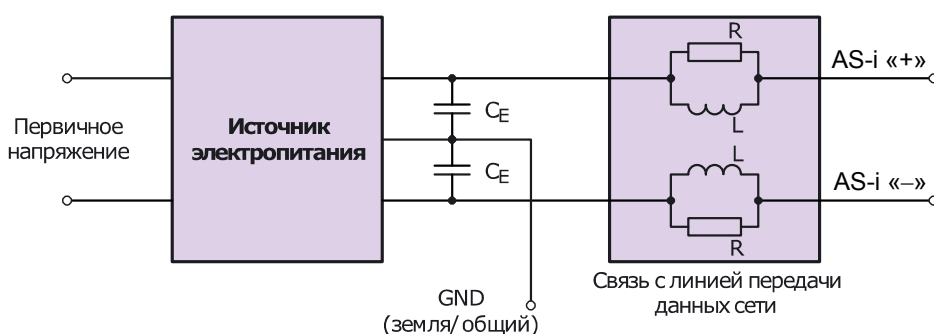


Рис. 10.9. Схема подключения источника электропитания к линии передачи данных

Схема связи с линией передачи данных, выполненная по рациональному способу в одном корпусе с источником питания, состоит из двух индуктивностей, каждая по 50 мкГн, и двух параллельно включенных сопротивлений по 39 Ом. Эти RL-цепочки служат для того, чтобы токовые импульсы, которые производит передатчик AS-интерфейса, посредством дифференцирования были преобразованы в импульсы напряжения. В силу того, что AS-интерфейс представляет собой симметричную незаземленную систему для оптимизации защиты от помех, возникающих вследствие перекрестных наводок, необходимо по возможности соблюдать симметричное построение двухпроводной линии AS-интерфейса. Для решения этой задачи служат обе емкости C_E . Только в указанной на рис. 10.9 точке GDN между этими емкостями допускается подключение приборной «земли».

С целью увеличения протяженности линии связи в структуре сети AS-интерфейса могут использоваться повторители сигналов и сигналов. Без

дополнительных компонентов длина одного сегмента сети не может превышать 100 м. За счет каскадного включения повторителей и удлинителей длина сети может быть увеличена до 600 м.

Повторители сигналов являются активными устройствами и позволяют увеличить протяженность магистрального участка AS-интерфейса вплоть до 300 м. Удлинители являются пассивными сетевыми компонентами и позволяют увеличивать максимальную длину сегмента AS-интерфейса со 100 до 200 м, а общую протяженность сети – до 600 м.

Из-за того что объединенные повторителем участки сети гальванически развязаны, с обеих сторон повторителя должны подключаться источники питания AS-интерфейса.

Ограничение на число допустимых в сети повторителей связано с требованием организации заданного быстродействия обмена в сети.

10.6. Физические характеристики

Техника монтажа является одним из главных факторов в обеспечении надежности эксплуатации любых промышленных сетей. Именно на этапе монтажа выявляется большинство проблем. Количество дискретных датчиков на современном технологическом объекте зачастую исчисляется сотнями штук. Обычно они распределены на объекте в разных пространственных зонах, то есть имеет место децентрализация. Кроме того, предъявляются жесткие требования к обеспечению защиты датчиков и исполнительных устройств от электромагнитных воздействий, разные степени защиты от климатических и других внешних воздействий.

Физические характеристики AS-интерфейса учитывают все эти обстоятельства. Монтаж систем на основе AS-интерфейса отличается исключительной простотой. Все станции AS-интерфейса объединяются в единую систему с помощью двужильного кабеля, имеющего оболочку трапециевидального сечения с кодировочным выступом, исключающим возможность возникновения ошибок при монтаже. Подключение сетевых компонентов производится методом прокалывания изоляции кабеля. Это позволяет ускорить монтаж, производить подключение сетевых устройств в любой точке кабеля, осуществлять быструю замену сетевых компонентов.

Для специальных применений может использоваться кабель с оболочкой круглого сечения.

Кабели AS-интерфейса имеют несколько модификаций, отличающихся материалом изоляции (EPDM – резиновая, TPE – полихлорвиниловая и PUR – полиуретановая) и предназначенных для эксплуатации в различных условиях.

В зависимости от своего функционального назначения кабели AS-интерфейса имеют различный цвет оболочки:

- *Желтый кабель* используется для передачи данных, питания электроники ведущего и ведомых сетевых устройств, а также всех подключенных датчиков.
- *Черный кабель* используется для подключения вспомогательных источников питания напряжением ≈ 24 В, используемых для питания исполнительных устройств, а также компонентов, работающих в распределенных системах автоматики безопасности и противоаварийной защиты.
- *Красный кабель* используется для подключения вспомогательных источников питания напряжением ~ 230 В, используемых для питания исполнительных устройств (например, в фидерах нагрузки).

Желтый плоский кабель стал своего рода рыночным знаком AS-интерфейса. Он имеет строго определенную геометрическую форму сечения в виде трапеции с выступом (рис.10.10), который обеспечивает однозначное положение кабеля в соединительных модулях и, как следствие, исключает возможность переполюсовки двухпроводной линии. Поперечное сечение кабеля в форме трапеции облегчает прижим и создает уплотнение в местах ввода кабеля в модули, переходные устройства, соединители и т.д., тем самым обеспечивая герметичность, соответствующую степени защиты IP67.

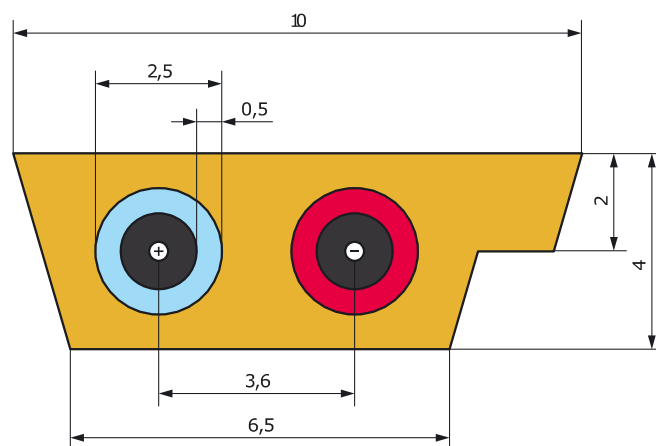


Рис. 10.10. Форма и размеры поперечного сечения плоского кабеля AS-интерфейса

Площадь поперечного сечения каждого проводника, равная 1.5 мм^2 , установлена стандартом. При таком сечении гарантируется питающий ток 2 А. Таким образом, при длине кабеля 100 м в случае подключения 31 ведомого устройства на равных расстояниях друг от друга при условии потребления каждым устройством не более 65 мА общее падение напряжения не превышает 3 В, что соответствует допустимому отклонению питающего напряжения.

Черный кабель имеет аналогичные размеры.

Как альтернатива плоскому кабелю допускается применение круглого кабеля типа H05VV-F2×1,5. Для подключения такого кабеля используются клеммы, а в качестве уплотнительных устройств – герметичные PG-соединители.

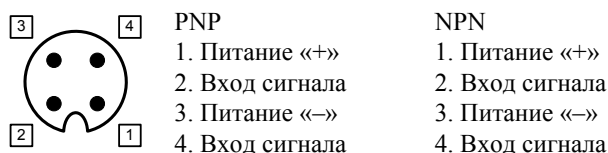
Так как топология сети AS-интерфейса может быть достаточно сложной (например, дерево), то для адаптации сети к требованиям решаемой задачи могут применяться распределители, ответвители и адаптеры.

Распределители AS-интерфейса выполняют функции узловых устройств, позволяющих выполнять разделение профилированного кабеля AS-интерфейса по нескольким направлениям.

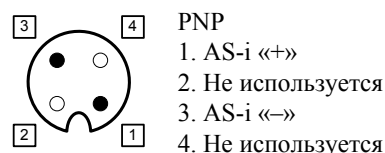
Многие устройства AS-интерфейса подключаются к сети круглым соединительным кабелем через встроенные гнезда M12. Переход с профилированного кабеля AS-интерфейса на круглый кабель выполняется через ответвители или переходные адаптеры M12. По способам подключения к профилированным кабелям AS-интерфейса ответвители M12 аналогичны распределителям. В зависимости от модификации через корпус устройства пропускается только желтый или желтый и черный профилированный кабель.

На рис. 10.11 показано стандартизованное расположение выводов четырехполюсных соединителей M12 в соответствии с IEC 947-5-2, приложение D.

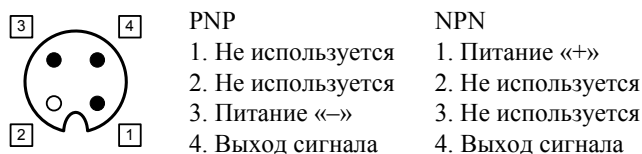
Стандартный вход (розетка)



Разделитель AS-i (розетка)



Стандартный выход (розетка)



Дополнительное питание (вилка)

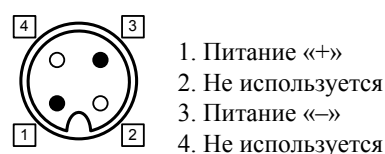


Рис. 10.11. Стандартизованное расположение контактов на разъемах M12 модулей AS-интерфейса

Контрольные вопросы

1. В чем особенности и преимущества кабеля AS-интерфейса? Каким еще способом возможно подключение устройств к сети?
2. Сколько ведущих и ведомых устройств может быть в одном сегменте сети AS-интерфейса, и какая скорость передачи данных в этой сети?

3. Какова топология сети AS-интерфейса.
4. Какова максимальная длина сегмента сети AS-интерфейса и как можно увеличить ее протяженность?
5. Как осуществляется межсетевой обмен данными, и на какие уровни модели ВОС можно передать данные AS-интерфейса?
6. Каким способом осуществляется кодирование данных на физическом уровне?
7. Поясните структуру протокола AS-интерфейса. Чем объясняется длина пакетов и каково назначение бит в них?
8. Перечислите базовые сетевые компоненты, используемые для построения сетей AS-интерфейса.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кириллов, Н.П. Признаки класса и определение понятия «технические системы» / Н.П. Кириллов // Авиакосмическое приборостроение. – 2009. – № 8. – С. 32–38.
2. Сахнюк, А.А. Промышленные сети. / А.А. Сахнюк, А.М. Литвин // ПИКАД. – 2004. – №2. – С. 6–8.
3. Данилушкин, И.А. Аппаратные средства и программное обеспечение систем промышленной автоматизации: учебное пособие / И.А. Данилушкин. – Самара: Самар. гос. техн. ун-т. – 2005. – 168 с.
4. Денисенко, В.В. Компьютерное управление технологическим процессом, экспериментом, оборудованием / В.В. Денисенко. – М.: Горячая линия – Телеком. – 2009. – 608 с.
5. Таненбаум, Э. Компьютерные сети. / Э. Таненбаум. – 4-е изд. – СПб.: Питер. – 2003. – 992 с.
6. Парк, Дж. Передача данных в системах контроля и управления: Практическое руководство / Дж. Парк, С. Маккей, Э. Райт. – М.: Группа ИДТ. – 2007. – 480 с.
7. Столлингс, В. Беспроводные линии связи и сети / В. Столлингс. – М.: Вильямс. – 2003. – 640 с.
8. Гук, М. Интерфейсы ПК. Справочник / М. Гук. – СПб.: Питер. – 1999. – 416 с.
9. Шевкопляс, Б.В. Особенности управления потоками данных в интерфейсах семейства RS. / Б.В. Шевкопляс // Схемотехника. – 2004. – №9. – С. 34–36 ; №10. – С. 34–36.
10. Звонарев, Е. Интерфейсы RS-485 дальнего радиуса действия. / Е. Звонарев, И. Фурман // Новости электроники. – 2007. – №6. – С. 3–6.
11. Kugelstadt, Thomas. The RS-485 Design Guide. Texas Instruments. October 2016.

12. Кангин, В.В. Аппаратные и программные средства систем управления. Промышленные сети и контроллеры: учебное пособие / В.В. Кангин. – М.: БИНОМ. Лаборатория знаний. – 2010. – 418 с.
13. Борисов, А.М. Основы построения промышленных сетей автоматики: учебное пособие / А.М. Борисов. – Челябинск: Изд. Центр ЮУрГУ. – 2012. – 108 с.
14. Modbus Protocol Specification [Электронный ресурс] // Modbus Organization, Inc. URL: <http://modbus.org/specs.php> (дата обращения: 02.05.2019)
15. Промислові мережі та інтеграційні технології в автоматизованих системах: Навчальний посібник / О.М. Пупена, І.В. Ельперін, Н.М. Луцька, А.П. Ладанюк. – К.: Ліра-К. – 2011. – 552 с.
16. Локотков, А.А. Интерфейсы последовательной передачи данных. Стандарты EIA RS-422A/RS-485. / А.А. Локотков // Современные технологии автоматизации. – 1997. – №3. – С. 110–119.
17. Кругляк, К.В. Промышленные сети: цели и средства / К.В. Кругляк // Современные технологии автоматизации. – 2002. – №4. – С. 6–17.
18. Current Loop. Application Note № CLAN1495. – B&B Electronics. – 1995. – 13 p.
19. Каталог серийных кабелей и специальных конструкций [Электронный ресурс] / Каталог №2, 2019 // М: Кабельный завод Спецкабель. 2019. URL: <http://spetskabel.ru/f/1/global/pricelist/spetskabel-catalog.pdf> (дата обращения: 7.05.2019).
20. Highway Addressable Remote Transducer Protocol [Электронный ресурс] / Wikipedia, the free encyclopedia. 2019. URL: https://en.wikipedia.org/wiki/Highway_Addressable_Remote_Transducer_Protocol (дата обращения: 17.05.2019).
21. Emerson Proves Advancements in EDDL (Electronic Device Description Language) Technology [Электронный ресурс] // Automation.com. 2005. URL: <https://www.automation.com/automation-news/industry/emerson-proves-advancements-in-eddl-electronic-device-description-language-technology> (дата обращения: 10.05.2019).
22. About FieldComm Group | FieldComm Group // The FieldComm Group. URL: <https://fieldcommgroup.org/about> (дата обращения: 7.05.2019).
23. HART SPECIFICATIONS | FieldComm Group // The FieldComm Group. URL: <https://www.fieldcommgroup.org/hart-specifications> (дата обращения: 7.05.2019).
24. HART–протокол первичной связи. Технический обзор. // Fisher Rosemount. URL: <http://www.hartcomm.org>. (дата обращения: 12.12.2014).
25. Саинский, И.В. Принципы построения АСУТП с использованием HART протокола [Электронный ресурс] / И.В. Саинский // Челябинск: ЭлМетро. 2016. URL: https://www.elmetro.ru/netcat_files/userfiles/

company/_article/postroenie_ASUTP_HART.pdf (дата обращения: 28.04.2019).

26. Саинский, И.В. Технический обзор HART-протокола / И.В. Саинский, Н. Дерябина // Челябинск: ПГ Метран, 2003.
27. Денисенко, В.В. HART-протокол: общие сведения и принципы построения сетей на его основе / В.В. Денисенко // Современные технологии автоматизации. – 2010. – Вып. 3. – С. 94–101.
28. Саинский, И.В. HART протокол – перспективное решение для российских систем управления технологическими процессами / И.В. Саинский, Л.В. Ушаков // Датчики и системы. – 2000. – Вып. 11–12. – С. 15–17.
29. Фетисов, В.С. Протокол информационного обмена HART в измерительных и управляющих системах: учебное пособие / В.С. Фетисов. – Уфа: ФОТОН, 2011. – 88 с.
30. HART communication protocol. Application guide HCF LIT 39. [Электронный ресурс] // HART Communication Foundation. 2013. URL: https://www.fieldcommgroup.org/sites/default/files/technologies/hart/ApplicationGuide_r7.1.pdf (дата обращения: 15.04.2019).
31. HART Master Protocol Manual. [Электронный ресурс] // Bristol Babcock Inc. 2000. URL: <http://www.emerson.com/documents/automation/133286.pdf> (дата обращения: 19.04.2019).
32. Корнова, Т.Л. HART-протокол и другие коммуникационные технологии, применяемые в России. / Т.Л. Корнова // Автоматизация в промышленности. – 2004. – №8. – С. 34–39.
33. Тойвонен, Д. Диагностики и параметризация HART-устройств без головной боли. / Д. Тойвонен // Control Engineering Россия. – 2016. – №2 (62). – С. 32–33.
34. AS-Interface [Электронный ресурс] / Wikipedia, the free encyclopedia. 2019. URL: <https://ru.wikipedia.org/wiki/AS-Interface> (дата обращения: 17.05.2019).
35. Половинкин, В. Основные понятия и базовые компоненты AS-Интерфейса. / В. Половинкин // Современные технологии автоматизации. – 2002. – №4. – С. 18–29.
36. AS-interface: das Actuator-Sensor-Interface fur die Automation / hrsg. Von Werner R. Kriesel. – Munchen; Wien: Hanser. Buch, 1999.
37. Kriesel W., Heimbold N., Telschow D. Bus Technologien fur die Automation. — Heidelberg: Huthing, 2000.

Учебное издание

Барбасова Татьяна Александровна,
Канашев Евгений Александрович

ПРОМЫШЛЕННЫЕ СЕТИ И СИСТЕМЫ СВЯЗИ

Учебное пособие

Техн. редактор *А.В. Миних*

Издательский центр Южно-Уральского государственного университета

Подписано в печать 16.01.2020. Формат 60×84 1/16. Печать цифровая.
Усл. печ. л. 8,60. Тираж 40 экз. Заказ 10/303.

Отпечатано в типографии Издательского центра ЮУрГУ.
454080, г. Челябинск, проспект Ленина, 76.