

## МУЛЬТИПЛЕКСНЫЕ СИСТЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ В СИСТЕМАХ ЭЛЕКТРОННОГО УПРАВЛЕНИЯ ДИЗЕЛЯМИ

В.М. Пилявский, асп.

Санкт-Петербургский государственный университет водных коммуникаций

За последние 20 лет значительно возросла сложность электрооборудования для управления двигателем. Сегодня разработка и изготовление жгута проводов является проблемой из-за его размеров и веса. В современном двигателе может быть более 500 отдельных проводов. Помимо увеличения размеров и веса, большое число проводов и соединителей снижает надежность. Растет число систем двигателя, имеющих автотронное управление. Эти системы в той или иной степени связаны друг с другом. Выходные сигналы некоторых датчиков могут использоваться несколькими электронными системами. Можно использовать один управляющий блок для управления большим числом систем. Но сегодня и в ближайшем будущем это экономически нецелесообразно. В настоящее время претворяется в жизнь другое техническое решение, когда контроллеры отдельных электронных блоков управления (ЭБУ) связываются друг с другом коммуникационной шиной для обмена данными. Датчики и исполнительные механизмы, подключенные к этой шине через специальные согласующие устройства, становятся доступными для всех ЭБУ. По этому принципу строится локальная вычислительная сеть (ЛВС) двигателя с электронным управлением.

### Мультиплексные системы

Термин «мультиплексный» широко используется в автомобильной промышленности. Обычно его относят к последовательным каналам передачи данных между различными электронными устройствами. Несколько проводов, по которым передаются управляющие сигналы, заменяются шиной для обмена данными. Необходимость уменьшения количества проводов в электропроводке двигателя — одна из причин разработки мультиплексных систем.

Другая причина — необходимость объединения в ЛВС контроллеров различных ЭБУ для эффективной работы и диагностики. На рис. 1 в качестве примера показана традиционная схема электропроводки и мультиплексная система.

Мультиплексные системы значительно отличаются от обычных, при этом принципиальные отличия состоят в следующем.

➤ В обычных системах электропроводки информация и питание передаются по одним и тем же проводам. В мультиплексных системах сигналы и электропитание разделены.

➤ В мультиплексных системах управляющие ключи непосредственно не включают и не выключают электропитание нагрузок.

➤ В некоторых случаях электронная схема узла должна постоянно считывать состояние управляющего ключа, даже когда большая часть электрооборудования обесточена.

По схеме, приведенной на рис. 1, может быть реализовано электропитание корпусных потребителей. Электропроводка упрощается за счет приема и передачи различных сигналов между узлами по одной и той же шине (проводу). При обычной схеме проводки для реализации каждой функции требуется отдельный проводник. Через узлы осуществляется доступ к сети. Узел, как правило, содержит микропроцессор,

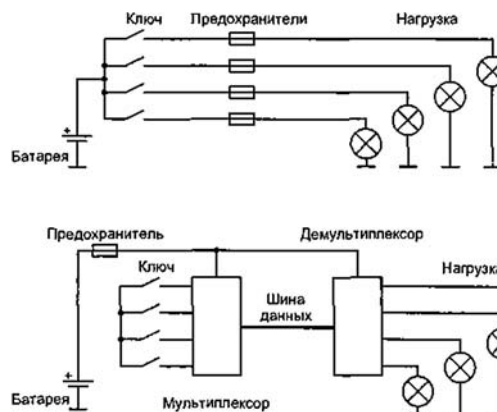


Рис. 1. Обычное и мультиплексное подключение нагрузок

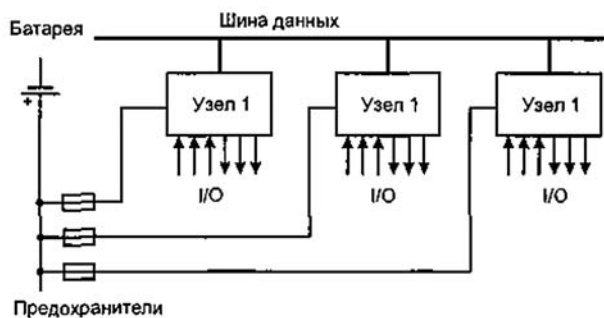


Рис. 2. Структура мультиплексной системы

подключенный к коммутационной шине, и электронные цепи, управляющие работой датчиков и исполнительных механизмов, подключенных к узлу. Коммуникационная шина на современном двигателе чаще всего представляет собой витую пару проводов, хотя возможны и другие варианты.

На рис. 2 в общем виде показана мультиплексная система. Ко входам узлов могут подключаться любые датчики, к выходам — исполнительные устройства.

#### Локальные вычислительные сети

Мультиплексные системы делят на три класса:

**Класс А.** Системы, в которых электропроводка упрощается за счет использования коммуникационной шины. По этой шине между узлами передаются сигналы, которые проходят по отдельным проводам с обычной электропроводкой. Узлы, являющиеся частью мультиплексной системы, при обычной электропроводке отсутствуют.

**Класс В.** Системы, в которых между узлами передаются информационные данные (обычно значения параметров), чем достигается устранение избыточности датчиков и иных элементов по сравнению с обычной схемой электропроводки. В этом случае узлы существуют и в обычной системе в виде несвязанных элементов.

**Класс С.** Системы с высокой скоростью обмена данными, осуществляющая управление в реальном времени, например, двигателем и т. д.

Системы класса А используются для включения/выключения различных нагрузок, скорость передачи данных по шине не более 10 Кбит/с.

В системах класса В осуществляется обмен информацией между подсистемами, когда требуется скорость передачи данных 100–250 Кбит/с. В настоящее время системы класса В используются чаще других. Задачи систем класса В можно решить, применив шину CAN (controller area network — локальная сеть контроллеров), но на сегодняшний день это экономически нецелесообразно.

В системах класса С осуществляется распределенное управление в реальном масштабе времени, скорость обмена данными около 1 Мбит/с. Шина CAN является стандартом для мультиплексных систем класса С. Крупнейшие производители электронного оборудования (Bosch, Delco) и комплектующих (Intel, Motorola) поддерживают CAN.

По мере значительного усложнения бортовой электроники мультиплексные системы, выполненные по классам А и В, становятся неоптимальными. Лучшим техническим решением является использование гибридной локальной сети, где датчики и исполнительные механизмы

через канал класса А подключены к бортовому компьютеру, а интерфейс компьютера (дисплей и органы управления) подключены к компьютеру через канал класса В, мультиплексоры интегрированы в датчики и исполнительные механизмы. Обмен данными проводится по одному проводу, дополнительных узлов нет, улучшена диагностика системы за счет введения в компоненты электроники. Такая конфигурация позволит вводить в систему управления дополнительные датчики и исполнительные устройства.

#### Эталонные модели взаимодействия систем

Для определения задач, решаемых сложной технической системой, а также для выделения главных характеристик и параметров, которыми она должна обладать, создаются общие модели таких систем. Общая модель вычислительной сети определяет характеристики сети в целом и характеристики и функции входящих в нее основных компонентов.

Многообразие вариантов вычислительных сетей и сетевых программных продуктов создало проблему объединения сетей различных архитектур. Для ее решения была разработана эталонная модель архитектуры открытых систем (рис. 3).

Открытая система — это система, взаимодействующая с другими системами в соответствии с принятыми стандартами.

Эталонная модель архитектуры открытых систем является моделью взаимодействия открытых систем (моделью ВОС) и служит базой для производителей при разработке совместимого сетевого оборудования. Эта модель не является неким физическим телом, отдельные элементы которого можно осязать. Эталонная модель представляет собой самые общие рекомендации для построения стандартов совместимых сетевых программных продуктов. Эти рекомендации должны быть реализованы как в аппаратуре, так и в программных средствах вычислительных сетей. В настоящее время модель взаимодействия открытых систем (модель ВОС) является наиболее популярной сетевой архитектурной моделью. Модель рассматривает общие функции, а не специальные решения, поэтому не все реальные сети абсолютно точно ей соответствуют. Модель взаимодействия открытых систем состоит из семи уровней (см. рис. 3).

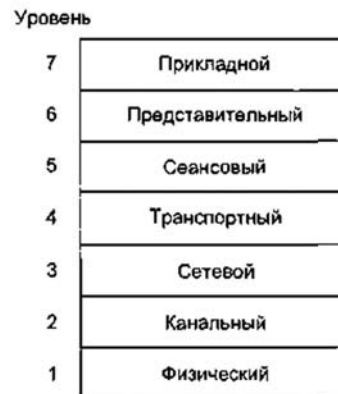


Рис. 3. Эталонная модель архитектуры открытых систем

Седьмой уровень — прикладной — обеспечивает поддержку прикладных процессов конечных пользователей. Этот уровень определяет круг прикладных задач, реализуемых в данной вычислительной сети.

Шестой уровень — представительный — определяет синтаксис данных в модели, т. е. представление данных. Он гарантирует представление информации в кодах и форматах, принятых в данной системе. В некоторых системах этот уровень может быть объединен с прикладным.

Пятый уровень — сеансовый — реализует установку и поддержку сеанса связи между двумя абонентами через коммуникационную сеть. Он позволяет производить обмен информацией в режиме, определенном прикладной программой, или предоставляет возможность выбора режима обмена. Сеансовый уровень поддерживает и завершает сеанс связи.

Три верхних уровня объединяются под общим названием — процесс или прикладной процесс. Эти уровни определяют функциональные особенности вычислительной сети как прикладной системы.

Четвертый уровень — транспортный — обеспечивает интерфейс между процессами и сетью. Он устанавливает логические каналы между процессами и обеспечивает передачу по этим каналам информационных пакетов, которыми обмениваются процессы. Пакет — группа байтов, передаваемых абонентами сети друг другу. Логические каналы, устанавливаемые транспортным уровнем, называются транспортными каналами.

Третий уровень — сетевой — определяет интерфейс оконечного оборудования пользователя с сетью коммутации пакетов. Он также отвечает за маршрутизацию пакетов в коммуникационной сети и за связь между сетями — реализует межсетевое взаимодействие.

Второй уровень — канальный — реализует процесс передачи информации по информационному каналу. Информационный канал — логический канал, он устанавливается между двумя ЭВМ, соединенными физическим каналом. Канальный уровень обеспечивает управление потоком данных в виде кадров, в которые упаковываются информационные пакеты, обнаруживает ошибки передачи и реализует алгоритм восстановления информации в случае обнаружения сбоев или потерь данных. Кадр имеет структуру: информация об источнике данных, информация о приемнике, тип кадра, данные, контрольная информация.

Первый уровень — физический — выполняет все необходимые процедуры в канале связи. Его основная задача — управление аппаратурой

передачи данных и подключенным к ней каналом связи.

При передаче информации от прикладного процесса в сеть происходит ее обработка уровнями модели ВОС. Смысл этой обработки заключается в том, что каждый уровень добавляет к информации процесса свой заголовок — служебную информацию, которая необходима для адресации сообщений и для некоторых контрольных функций. Канальный уровень кроме заголовка добавляет еще и концевик — контрольную последовательность, которая используется для проверки правильности приема сообщения из коммуникационной сети.

Физический уровень заголовка не добавляет. Сообщение, обрамленное заголовками и концевиком, уходит в коммуникационную сеть и поступает на абонентские ЭВМ вычислительной сети. Каждая абонентская ЭВМ, принявшая сообщение, дешифрирует адреса и определяет, предназначено ли ей данное сообщение.

При этом в абонентской ЭВМ происходит обратный процесс — чтение и отсечение заголовков уровнями модели ВОС. Каждый уровень реагирует только на свой заголовок. Заголовки верхних уровней нижними уровнями не воспринимаются и не изменяются — они прозрачны для нижних уровней.

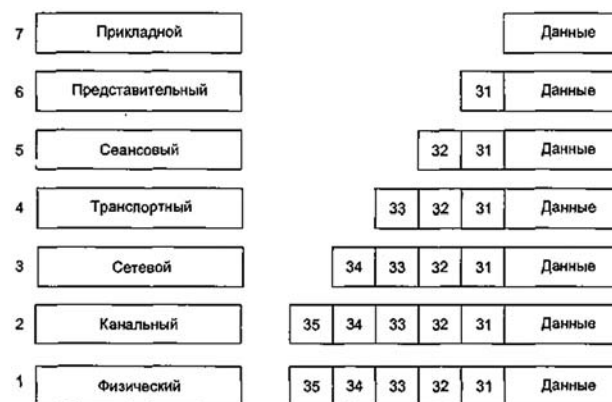


Рис. 4. Обработка сообщений уровнями модели ВОС

В процессе развития и совершенствования любой системы возникает потребность изменять ее отдельные компоненты. Иногда это вызывает необходимость изменять и другие компоненты, что существенно усложняет и затрудняет процесс модернизации системы. В таком случае проявляются преимущества семиуровневой модели ВОС. Если между уровнями определены однозначно интерфейсы, то изменение одного из уровней не влечет за собой необходимости внесения изменений в другие уровни. Таким образом, существует относительная независимость уровней друг от друга.

Функции, описываемые уровнями модели, должны быть реализованы либо в аппаратуре, либо в виде программ.

Функции физического уровня всегда реализуются в аппаратуре. Это адаптеры, мультиплексоры передачи данных, сетевые платы и т. п.

Функции остальных уровней реализуются в виде программных модулей — драйверов.

#### Протоколы компьютерных сетей

При обмене информацией в компьютерной сети каждый уровень модели ВОС реагирует на свой заголовок. Иными словами, происходит взаимодействие между одноименными уровнями модели в различных абонентских ЭВМ. Такое взаимодействие должно выполняться по определенным правилам — протоколам.

Протокол — это не программа. Правила и последовательность выполнения действий при обмене информацией, определенные протоколом, должны быть реализованы в программе. Обычно функции протоколов различных уровней реализуются в драйверах для различных вычислительных сетей.

В соответствии с семиуровневой структурой модели ВОС можно говорить о необходимости существования протоколов для каждого уровня.

Концепция открытых систем предусматривает разработку стандартов для протоколов различных уровней. Легче всего поддаются стандартизации протоколы трех нижних уровней модели архитектуры открытых систем, так как они определяют действия и процедуры, характерные для вычислительных сетей любого класса.

Труднее всего стандартизовать протоколы верхних уровней (особенно прикладного) из-за множественности прикладных задач и в ряде случаев их уникальности. Если по типам структур, методам доступа к физической передающей среде, используемым сетевым технологиям и некоторым другим особенностям можно насчитать примерно десяток различных моделей вычислительных сетей, то по их функциональному назначению пределов не существует.

Проще всего представить особенности сетевых протоколов на примере протоколов канального уровня, которые делятся на две основные группы: байт-ориентированные и бит-ориентированные.

Байт-ориентированный протокол обеспечивает передачу сообщения по информационному каналу в виде последовательности байтов. Кроме информационных байтов в канал передаются также управляющие и служебные байты. Такой

тип протокола удобен для ЭВМ, так как она ориентирована на обработку данных, представленных в виде двоичных байтов. Для коммуникационной среды байт-ориентированный протокол менее удобен, так как разделение информационного потока в канале на байты требует использования дополнительных сигналов, что в конечном счете снижает пропускную способность канала связи.

Бит-ориентированный протокол предусматривает передачу информации в виде потока битов, не разделяемых на байты. Поэтому для разделения кадров используются специальные последовательности — флаги. В начале кадра ставится флаг открывающий, а в конце — флаг закрывающий. Бит-ориентированный протокол удобен относительно коммуникационной среды, так как канал связи как раз и ориентирован на передачу последовательности битов.

В то же время для ЭВМ бит-ориентированный протокол не очень удобен, потому что из поступающей последовательности битов приходится выделять байты для последующей обработки сообщения. Впрочем, учитывая быстрое действие ЭВМ, можно считать, что эта операция не окажет существенного влияния на ее производительность.

Таким образом, рассмотренные в настоящей работе принципы построения и алгоритмы функционирования мультиплексных систем передачи информации позволяют надежно решать задачи управления системами современных дизельных двигателей любой степени сложности. При этом конструкция рассмотренных систем отличается небольшими габаритами и легко вписывается в архитектуру (компоновку) современного дизельного двигателя.

#### Литература

1. Юмт В.Е. Электронные системы управления ДВС и методы их диагностирования. — М., 2007.
2. Дием Л.В. Модели и алгоритмы технического диагностирования судовых ДВС в процессе эксплуатации. — СПб., 2006.
3. Шадрин А.Б. Измерительно-управляющие микросети на транспорте. — СПб., 2005.
4. Тузов Л.В., Шадрин А.Б. Сетевое обучение через теле-автотронику в жестких условиях эксплуатации транспорта. — СПб., 2007.
5. Шадрин А.Б. Сетевая мехатроника ДВС. — СПб., 2006.