

С улыбкой о серьёзном

Протокол ОВЕН для «чайников»

Фёдор РАЗАРЁНОВ,
зам. начальника отдела новых разработок ОВЕН

Подчас не только «чайникам», но и специалистам бывает трудно разобраться с новым протоколом. Одной из причин этого является сухое описание, требующее наличия у читающего серьёзного уровня начальной подготовки. В статье, предлагаемой вашему вниманию, понятным языком рассказывается о протоколе обмена ОВЕН для сети RS-485.

Прежде чем рассматривать протокол ОВЕН для сети RS-485, попробуем определить, что же такое протокол обмена вообще. Протокол – это задокументированные правила, по которым производится доставка чего-то из одного места в другое. Как правило, речь идёт о доставке информации от одного электронного устройства к другому, но мы упростим задачу и рассмотрим понятие протокола на примере доставки грузов.

Пример

Бабушка-слониха, проживающая в кенийском заповеднике, расположенном на африканском континенте, хочет послать своему внуку гроздь бананов в зоопарк Алабамы, находящийся в Северной Америке. Давайте посмотрим, что будет происходить с бананами на пути из



Африки в Северную Америку.

Прежде всего, бабушка-слониха упакует бананы в коробку, напишет на ней адрес «США, в штат Алабама, зоопарк, слонёнку» и отнесёт посылку на почту. На почте соберут все посылки, которые идут в Алабаму, в один большой ящик, и переправят в порт. В порту все большие ящики упакуют в морской контейнер. На корабле доставят контейнер в Америку, разгрузят и развезут по адресатам.

Теперь детально рассмотрим этот пример применительно к протоколу ОВЕН. И вы увидите, что процесс пересылки бананов с одного континента на другой и процесс доставки информации по электронным сетям имеют много общего.

В процессе перевозки с материка на материк наши бананы были последовательно упакованы в три коробки: посылку, большой ящик и морской контейнер. Также и сетевой протокол состоит из нескольких уровней. Упакованные данные от нижележащего уровня передаются на вышерасположенный (рис. 1), как и в примере с бананами, когда в морской контейнер загружали большие ящики, а не посылки или бананы. Поэтому при реализации каждого слоя протокола не надо заботиться о том, что находится в нижних слоях, подобно тому, как грузчикам в порту не нужно знать, что лежит в каждой посылке.

Так что первое правило, которое необходимо усвоить – любой протокол имеет послойную структуру и реализовывать его надо именно по слоям (по уровням). Причём не так важно, сколько слоёв имеет протокол. В примере с пересылкой бананов их три, а в классической модели сетевого протокола – семь. Протокол ОВЕН также имеет многослойную структуру: она показана на рис. 2. Протокол ОВЕН имеет четыре слоя.

Комментарий

Некоторые читатели могут возразить, что существуют простые одноуровневые протоколы, которые прекрасно работают, а многоуровневость вредна. Да, такие протоколы существуют, но они пригодны только для решения самых простых задач. Например, если на корабле надо доставить единственную гроздь бананов, то можно её вовсе не упаковывать в ящик и затем в контейнер, а переслать в единственной упаковке, то есть просто в посылке. Но когда следует передать много информации разным адресатам, то это уже неэффективно и может привести к большим сложностям при написании программ верхнего уровня. Ведь вполне очевидно, что неразумно скидывать в один контейнер содержимое тысячи посылок без их предварительной упаковки. Даже если наша гроздь бананов не будет раздавлена, то потом, при разгрузке, придётся приложить массу усилий, чтобы понять, кому она была предназначена.

Уровни сетевого протокола:

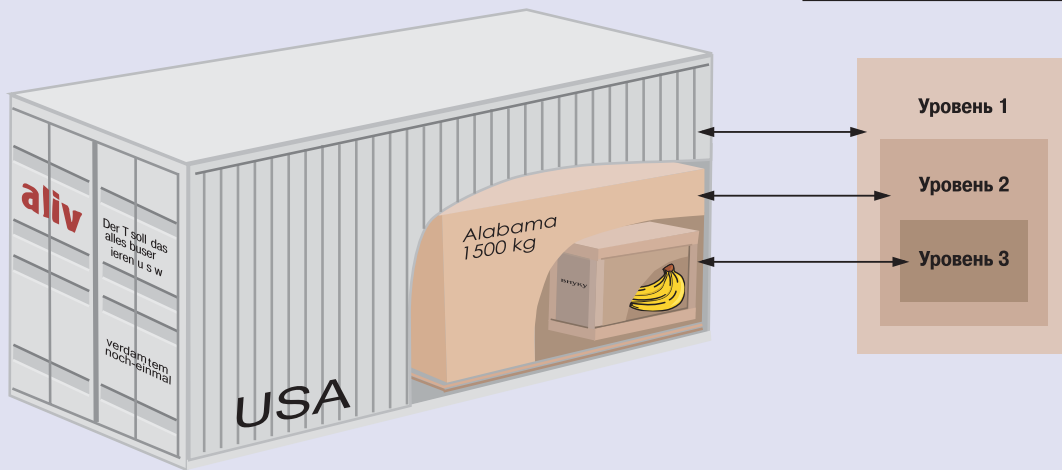


Рис. 1. Уровни упаковки данных

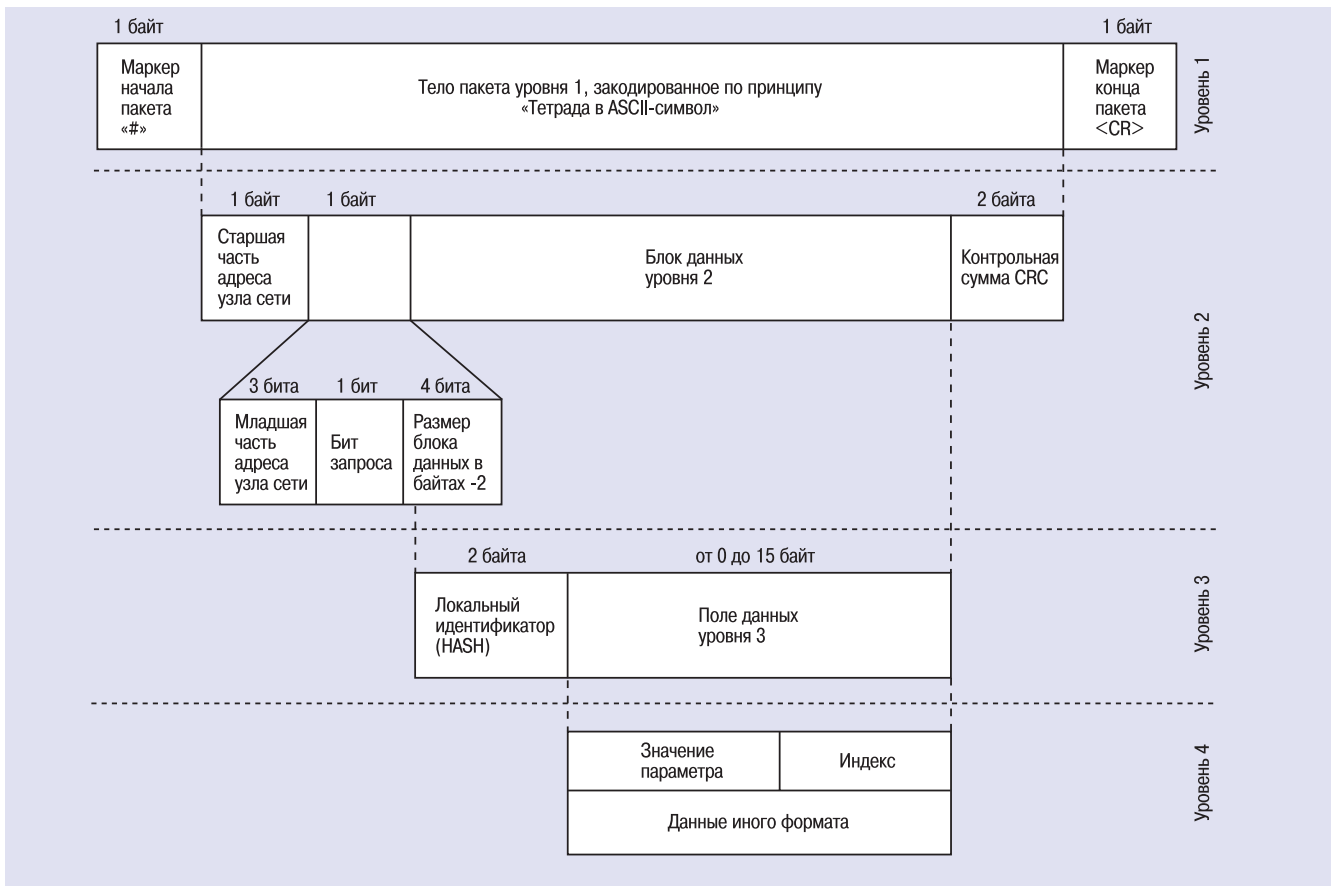


Рис. 2. Структура протокола OBEH

Уровень 1

Уровень 1 отвечает за непосредственную передачу байт в сети RS. Как правило, это сеть RS-485, но может быть и RS-232, или канал с иным физическим интерфейсом. Уровень 1 не знает, что он передаёт, его задача закодировать и переслать в сеть набор байт от уровня 2. В протоколе OBEH реализована передача в ASCII-формате, при котором один байт кодируется двумя симво-

лами из ASCII-таблицы. В начале и в конце пакета уровень 1 ставит служебные символы: символ «#» (код 0x23) и символ «<CR>» (символ перевода каретки, код 0x0D).

Самое простое преобразование – это преобразование по таблице, приведённой на рис. 3. В ней показано, с помощью какого символа кодируется одна тетрада (4 бита) каждого байта.

Предположим, уровень 2 просит передать в сеть три байта: 0xA1 0x45 0x5D. Для уровня 1 не важно, что они означают, его задача – передать их в сеть. Согласно таблице первая тетрада первого байта (0xA) кодируется буквой «Q», вторая тетрада (0x1) кодируется буквой «H» и так далее. В начале и конце пакета ставятся служебные символы. И конечный набор символов, который уйдёт в сеть, будет выглядеть так: #QHKLLT<CR>. При приёме необходимо откинуть служебные символы и выполнить обратное преобразование.

Такой способ кодирования ведёт к удвоению информации, передающейся по сети, и повышает надёжность протокола, так как пакеты с некорректными символами при приёме будут забракованы. В настоящий момент идёт внедрение протокола OBEH-RTU в ряд приборов. В этом протоколе данный уровень будет упразднён, и байты будут передаваться в сеть без перекодирования.



Продолжим упражнение с примером. Уровень 2 для протокола доставки бананов – это большой ящик, на котором написано «в штат Алабама», и уточнение от почтовой службы – «вес 1500 кг». Слова «штат Алабама» соответствуют адресу прибора, уточнение «вес 1500 кг» указывает на длину пакета, предлог «в» определяет направление передачи, то есть бит запроса. Этой информации вполне доста-

0x0 = 0000	->	'G' = 0x47
0x1 = 0001	->	'H' = 0x48
0x2 = 0010	->	'I' = 0x49
0x3 = 0011	->	'J' = 0x4a
0x4 = 0100	->	'K' = 0x4b
0x5 = 0101	->	'L' = 0x4c
0x6 = 0110	->	'M' = 0x4d
0x7 = 0111	->	'N' = 0x4e
0x8 = 1000	->	'O' = 0x4f
0x9 = 1001	->	'P' = 0x50
0xA = 1010	->	'Q' = 0x51
0xB = 1011	->	'R' = 0x52
0xC = 1100	->	'S' = 0x53
0xD = 1101	->	'T' = 0x54
0xE = 1110	->	'U' = 0x55
0xF = 1111	->	'V' = 0x56

Рис. 3. Кодирование тетрад

точно для понимания того, что делать с этим ящиком, и он будет успешно доставлен на центральный почтамт Алабамы.

Уровень 2

Уровень 2 протокола ОВЕН можно сравнить с большим ящиком в примере с пересылкой бананов. Все пакеты, у которых адрес будет совпадать с базовым адресом прибора, будут обрабатываться именно этим прибором, а остальными будут проигнорированы. Адрес состоит из 8 или 11 бит и стоит в самом начале набора байтов уровня 2.

Далее на уровне 2 стоит бит, показывающий тип пакета – запрос это или ответ, поскольку данные могут запрашиваться у прибора, а могут передаваться на прибор. После бита запроса стоят 4 бита, определяющие размер поля данных уровня 2. Но сейчас не принципиально, что в них находится, для уровня 2 это просто набор байтов.

В конце набора байтов уровня 2 ставится контрольная сумма, позволяющая проконтролировать корректность принятого пакета. Алгоритм вычисления контрольной суммы подробно описан в спецификации протокола ОВЕН. Для вычисления контрольной суммы используются все байты пакета на уровне 2, за исключением самой контрольной суммы.

Уровень 3

На этом уровне производится разбор всех пакетов, приходящих в прибор. Перед этим на уровне 2 было проверено совпадение адреса и откинута все чужие пакеты. На уровне 3 содержится закодированное имя параметра – 16-битный локальный идентификатор. Имя параметра для приборов ОВЕН состоит из четырёх символов, между которыми могут ставиться точки. Это удобный формат представления, так как оперировать с символическими именами человеку привычнее. И поэтому на передних панелях приборов, в программах конфигураторов и в руководствах параметры представлены именно символическими именами. Но передавать символические имена в сеть неудобно, так как для передачи четырёх символов с четырьмя возможными точками потребуется немало места – 8 байт. Это слишком большие накладные расходы, поэтому более экономично символическое имя свернуть в двухбайтный идентификатор. Алгоритм такой свертки называется Hash. И по сети передаётся не имя, а именно свёрнутый локальный идентификатор.

Для того чтобы не реализовывать алгоритм свертки, некоторые программы сразу оперируют с Hash-идентификаторами. Для всех приборов компании ОВЕН есть списки параметров, в которых напротив имени указан его Hash (параметры не именуется, а просто нумеруются 16-битным номером).

После идентификатора параметра стоят данные уровня 3. Это может быть сразу значение параметра, или сложные составные данные (об этом расскажем на уровне 4), или может вообще ничего не стоять. Например, если мы запрашиваем значение параметра, то в поле данных уровня 3 не будет ничего, данные будут только в ответе на этот запрос.



Путь посылки с бананами продолжается. Она уже прибыла в штат Алабама и на почтамте была определена для пересылки в зоопарк. Это стало возможным благодаря тому, что бабушка-слониха указала на посылке пункт назначения – «Зоопарк». Пока посылка лежала в большом ящике, эта информация была скрыта, но это не имело значения при транспортировке посылки с одного материка на другой. Но когда большой ящик распаковали, адрес на посылке опять обрёл актуальность.

Локальный идентификатор (или соответствующее имя параметра) можно сравнить с адресом на посылке. Он также важен на этом этапе доставки данных уже внутри прибора, так как определяет, с каким параметром идёт работа.

Уровень 4

Это самый последний уровень, на нём идёт разбор изначально посылаемых значений для последующего присвоения их параметру, имя которого было указано в Hash-идентификаторе. В поле данных третьего уровня может сразу лежать значение параметра, но могут быть составные данные, содержащие линейный индекс.



Напомним адрес, который бабушка-слониха написала на посылке – «США, в штат Алабама, зоопарк, слонёнку». При ближайшем рассмотрении выясняется, что информации здесь недостаточно. Бабушка по рассеянности забыла, что в зоопарке Алабамы живёт также внучатый племянник её знакомой из соседнего заповедника. Но для служителя зоопарка оба слонёнка равны, и по такой надписи он никак не смог бы определить, кому из них отдать бананы. Если бы в зоопарке был всего один слонёнок – вопросов бы не возникло, но это не так. Хорошо, что бабушка-слониха в последний момент спохватилась и вложила внутрь посылки записку, в которой написала «слонёнку из вольера № 2». И увидев такую записку, служитель зоопарка сразу понял, кому отдать бананы.

В протоколе ОВЕН ситуация выглядит аналогично. Если по сети надо задать уставку в одноканальном приборе-регуляторе – то проблем не возникает, поскольку она в приборе единственная. Восьмиканальный прибор-регулятор имеет, соответственно, восемь уставок, и чтобы информация по сети была отправлена корректно, после значения уставки передаётся номер регулятора, к которому она относится. Этот номер называется линейным индексом, который может обозначать номер блока в приборе, но может нести и другую информацию. Это определяется при проектировании конкретного прибора, и информация об этом содержится в его описании.

В программном регуляторе ТРМ151 существует довольно много уставок. Прибор может выполнить 12 программ по 10 шагов каждая, и на каждом из этих шагов задаётся своя уставка. Соответственно, линейный индекс, который содержит номер программы и номер шага, будет изменяться от 0 до 119.

Комментарий

Формат параметра никак не определяется протоколом. Протокол ОВЕН позволяет передавать данные любого типа. Поэтому, чтобы понять друг друга, принимающая и передающая сторона должны знать, в каком формате представлены данные. Информация для каждого конкретного прибора содержится в списках параметров, предоставленных в описаниях. Сам протокол не преобразует формат передаваемых данных, это не его задача. Подобно тому, когда вы звоните в Америку и говорите в трубку по-русски, на другом конце слышна русская, а не английская речь, хотя Америка – англоязычная страна. Протоколы передачи данных телефонных компаний не будут переводить вашу речь на английский, так как их задача другая – передать голос без искажений.

Особенности протокола ОВЕН

Одна из особенностей протокола ОВЕН – это передача оперативных параметров. Все параметры прибора принципиально можно разбить на две группы: оперативные и конфигурационные. Оперативные параметры – это параметры, отражающие текущее состояние системы управления. Это измеренные значения, вычисленные данные, команды и другая информация, которая оперативно меняется в процессе работы. Конфигурационные параметры – это параметры, определяющие настройки прибора. Значения конфигурационных параметров хранятся в энергонезависимой памяти и являются неизменными. В процессе передачи протокол ОВЕН не разделяет оперативные или конфигурацион-

ные параметры, но есть некоторые тонкости в организации индексации этих параметров.



Вернёмся к нашему примеру с пересылкой бананов. Как мы помним, бабушка-слониха не указала в адресе номер вольера, а сделала это только в записке, которую вложила внутрь посылки. Но можно было написать нужный номер вольера в адресе. В почтовой службе Алабамы должны быть зарегистрированы два адреса: «Зоопарк, вольер № 1» и «Зоопарк, вольер № 2».

Также и в протоколе ОВЕН индекс можно передать в адресе прибора. В этом случае если нам надо опросить результаты измерения с восьмиканального модуля МВА8, то можно представить МВА8 как восемь независимых одноканальных измерителей, каждый из которых имеет в сети собственный адрес. Адресное пространство восьмиканального МВА8 расширяется, и он занимает подряд восемь адресов в сети. И после этого опрашивать каждый одноканальный прибор по своему адресу, не заботясь об индексации внутри данных, также как служителю зоопарка не надо искать получателя посылки с бананами, если почтовая служба сразу доставит её в нужный вольер.

Важно

Для протокола ОВЕН действует правило, по которому индексация оперативных параметров производится в адресе прибора. Для конфигурационных параметров линейный индекс передается рядом со значением параметра в поле данных уровня 4.

Для передачи оперативных параметров каждый многоканальный прибор занимает в адресном пространстве количество адресов, равных количеству его каналов. И адреса эти получают

суммированием номера канала (считая с нуля) и базового адреса прибора. При такой индексации идёт повышенный расход сетевых адресов, поэтому в протокол заложена возможность работы с 11-битными адресами, расширяющими доступное адресное пространство. Но если приборов мало, и не требуется опрос большого количества оперативных параметров, то вполне можно обойтись 8-битным адресом.

Комментарий

У читателя могут возникнуть вопросы, зачем нужны два способа передачи индекса, почему так сложно? Отвечаем. Способ передачи индекса в адресе упрощает работу, так как не надо отделять линейный индекс в поле данных уровня 4. В этом случае уровень 4 вообще вырождается и может не реализовываться. Это удобно для пользователей, которые пишут свои программы, опрашивающие только несколько оперативных параметров. Но для конфигурирования приборов такой способ индексации не годится. Как мы говорили ранее, при передаче конфигурационных параметров индекс может быть довольно большой (как в уставках ТРМ151) и его нельзя передавать в адресе, поскольку в этом случае может не хватить адресного пространства.

Задача этой статьи – дать общее понятие о протоколе ОВЕН и популярно изложить основные принципы его работы. Мы не ставили перед собой цель описать весь протокол, поэтому у читателя могут остаться вопросы. Для ответа на них необходимо обратиться к официальному описанию протокола, которое в свободном доступе имеется на сайте www.owen.ru. Также для ещё более простой реализации протокола ОВЕН мы рекомендуем воспользоваться стандартным набором библиотек Win DLL, содержащих все четыре уровня протокола. Библиотеки распространяются бесплатно через веб-сайт компании ОВЕН. ■

Календарь выставок, в которых участвует компания ОВЕН в первом полугодии 2007 года

Название выставки	Дата проведения	Город	Место проведения
Мир климата 2007	13–16 марта	Москва	ВК «Крокус Экспо»
SHK 2007	28–31 мая	Москва	ВК «Экспоцентр» на Красной Пресне, м. Улица 1905 года или м. Деловой центр
Электро 2007	13–16 июня	Москва	ВК «Экспоцентр» на Красной Пресне, м. Улица 1905 года или м. Деловой центр
Нефть и газ 2007	26–29 июня	Москва	ВК «Экспоцентр» на Красной Пресне, м. Улица 1905 года или м. Деловой центр