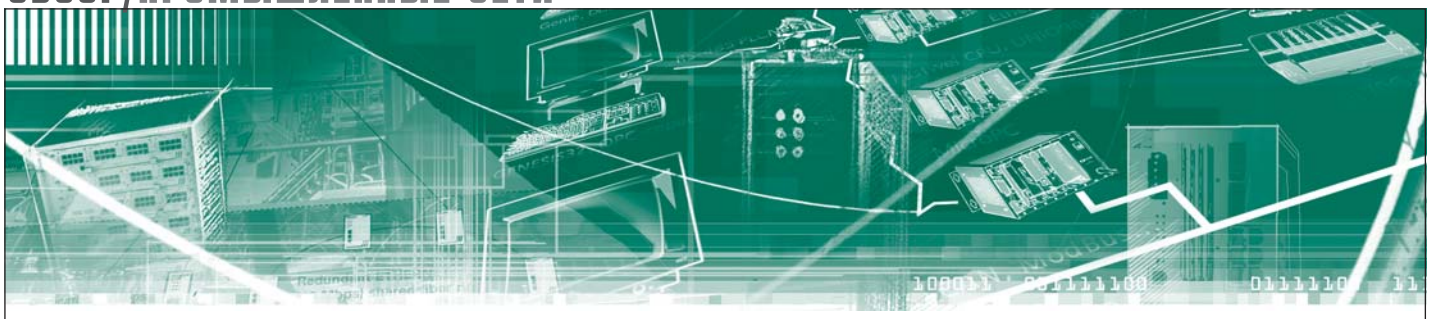

Джордж Томас

Введение в протокол Modbus

Часть 1

В статье даётся описание протокола Modbus: режимы ASCII и RTU, формат и структурирование сообщений, таблица распределения регистров и коды функций.

Настоящий материал является первой из двух статей, посвящённых Modbus, в которой рассматривается сам протокол. Вторая статья будет посвящена двум реализациям — Modbus Serial и Modbus TCP, благодаря которым Modbus продолжает оставаться столь популярным протоколом.

Что общего между технологиями ARCNET®, Ethernet и Modbus? Все они были разработаны в 70-х годах прошлого века и до сих пор широко используются. Конечно, с течением времени они развивались, но основы их функционирования остались неизменными. Ведь от добра добра не ищут.

Между этими тремя технологиями имеется одно основополагающее различие. И ARCNET®, и Ethernet представляют собой каналы передачи данных и стандарты физического уровня без протокола, в то время как Modbus является протоколом, который может работать на нескольких типах каналов связи и физических уровнях. Задуманный первоначально как интерфейс «точка-точка» между собственными устройствами компании Modicon, данный протокол нашел применение в многоточечных и равноранговых сетях, таких как TCP/IP. Теперь область применения этого протокола уже не ограничивается оборудованием фирмы Modicon.

Коммуникационный протокол Modicon

Modbus был предложен в 1979 году компанией Modicon — лидером тогда только зарождавшегося рынка программируемых логических контроллеров (ПЛК). Он должен был служить протоколом реализации внутренних коммуникаций «точка-точка» между ПЛК Modicon и панелью программирования, предназначенной для ввода программ в этот ПЛК. После ряда поглощений Modicon сейчас входит в состав компании AEG Schneider Automation, которой принадлежат бренды Modicon, Square D и Telemecanique. Могло бы сложиться впечатление, что протокол Modbus давно забыт, но это не так, вы и сейчас найдете его в заголовках на сайте Modbus-IDA http://www.modbus.org. Протокол продолжает процветать, так как он достаточно прост для понимания и многие инженеры получили первый опыт работы именно с протоколом Modbus. Кроме того, он построен по принципу открытой системы, и пользоваться им можно бесплатно. Далее, область применения этого протокола не ограничивается только промышленной автоматизацией. Modbus можно встретить и во многих других областях, включая системы автоматизации зданий.

Литературу по протоколу Modbus, в том числе руководство Modicon Modbus Reference Guide, выпущенное в июне 1996 г., можно найти на сайте Modbus-IDA.

Если вы обратитесь к исходным текстам документации по Modbus, то заметите множество упоминаний продукции компании Modicon. Только позднее группа Modbus-IDA разработала общие стандарты, призванные облегчить внедрение данного протокола. Кроме упомянутого руководства, были выпущены еще три: Modbus Application Protocol Specification, Modbus over Serial Line Specification and Implementation Guide и Modbus Messaging on TCP/IP Implementation Guide. Все эти документы можно получить бесплатно. Протокол Modbus предназначен для использования в сетевых структурах нескольких разновидностей, в том числе в разработанной компанией Modicon одноранговой сети Modbus Plus. Прежде чем перейти к описанию более современных реализаций, которым будет посвящена вторая статья, рассмотрим сам протокол.

Первоначальная реализация протокола компанией Modicon

Интересно заметить, что первоначально компания Modicon не применяла протокол Modbus в многоточечных сетях, но вместо этого использовала соединения «точка-точка» с ПЛК по интерфейсу EIA-232C (RS-232C). Modbus представляет собой протокол, построенный по принципу master-slave (ведущий-ведомый). Следует отметить, что термины «master» и «slave» употребляются до сих пор. Modbus допускает

наличие в структуре только одного ведущего устройства и от 1 до 247 ведомых. В качестве ведомого устройства обычно выступает ПЛК Modicon с интерфейсом EIA-232C. Роль ведущего устройства обычно играет либо панель программирования, либо главный компьютер. Следовательно, если одному главному компьютеру необходимо обмениваться информацией с четырьмя ПЛК, то нужно, чтобы этот компьютер располагал четырьмя последовательными портами. Это приводит к необходимости иметь систему с топологией звезды. Допустимая длина интерфейсного кабеля EIA-232C сравнительно невелика, поэтому если надо обеспечить связь на большие расстояния, то требуются модемы. Многоточечные сети с 2-проводными и 4-проводными интерфейсами EIA-485 (RS-485) появились значительно позднее.

По правилам протокола Modbus передача сообщения может инициироваться только ведущим устройством, ведомые этого делать не могут. Поэтому если ведомое устройство отмечает такое событие, как «остановились насосы подачи воды в систему охлаждения атомного реактора», то оно не может проинформировать об этом ведущее устройство до тех пор, пока ведущее не пришлет ему запрос о том, как идут дела. Идеология протокола такова, что ведущему устройству адрес не присваивается, а ведомые пронумерованы от 1 до 247.

Адрес «0» зарезервирован в качестве адреса широковещательной передачи сообщений, предназначенных всем ведомым устройствам. Такое сообщение получают все ведомые устройства, но ответ на него не предусмотрен.

Обмен сообщениями по принципу «запрос-ответ»

Сообщения-команды, исходящие от ведущего устройства, именуются запросами, а ответные сообщения, присылаемые ведомым устройством, ответами. Формат сообщения на рис. 1 показывает упрощенную структуру как запросов, так и ответов.

Ведущее устройство не имеет адреса вообще, поэтому в поле адреса всегда указывается номер ведомого устройства. Если это запрос, то он направляется ведомому устройству с указанным адресом. Если сообщение является ответом, то оно поступает от ведомого устройства с проставленным в этом поле адресом. Сообщение-запрос всегда содержит тот или иной код функции, например, код 03 — это функция «Чтение регистров хранения». В этом случае ведущее устройство должно указать диапазон номеров подлежащих считыванию регистров. Ведомое устройство отвечает на такой запрос сообщением, содержащим запрошенные данные, с учётом указанного в запросе диапазона регистров. Формат сообщения одинаков для всех кодов функций, но содержимое поля данных, естественно, для разных кодов будет различным. В последнем поле каждого сообщения помещается код ошибки, формируемый устройством-отправителем, так что устройство-получатель может проверить целостность пришедшего сообщения.

Адрес устройства
Код функции
Данные
Контрольная сумма

Рис. 1. Упрощённый формат сообщения в протоколе Modbus

Описанный сценарий предполагает успешный обмен «запрос-ответ». Если же ведомое устройство хочет сообщить об ошибке или об исключительной ситуации (исключении), то оно модифицирует поле кода функции, устанавливая в старший значащий разряд кода значение «1». В поле данных помещается информация, описывающая исключение. Но и при этом ведущее устройство может выделить тот код функции, который оно посылало ведомому устройству в своем запросе.

Следует отметить, что ведущее устройство может перейти к отправке следующего запроса тому же самому или другому ведомому устройству только по завершении предыдущего цикла «запрос-ответ». Это отличает данный протокол от других, таких как DeviceNet, которые могут отправлять команду, предназначенную сразу нескольким ведомым устройствам, и затем переходить в режим ожидания ответов. При этом никакого определённого порядка очерёдности ответов не существует. В протокол Modbus возможность широковещательной передачи запросов по нескольким адресам не заложена, поэтому имеет место определённая потеря времени, так как при каждом запросе со стороны ведущего устройства ведомое устройство должно не только получить сам запрос, но также его обработать и на него ответить, и после этого ведущее устройство может перейти к следующему циклу обмена.

Режимы ASCII и RTU

До сих пор всё просто, но дальше идут определённые сложности, поскольку протокол Modbus рассчитан на два режима последовательной передачи данных. Один именуется ASCII (American Standard Code for Information Interchange), а второй — режимом RTU (Remote Terminal Unit). Термин RTU ведет происхождение от SCADA-систем (Supervisor Control and Data Acqusition), в которых ведущее устройство, именуемое CTU (Central Terminal Unit), обменивается информацией с несколькими удалёнными устройствами (RTU), находящимися от него на определённых расстояниях. Такая конфигурация подобна первоначальной реализации Modicon, где одно CTU обменивалось информацией с несколькими RTU в системах с топологией «звезда». Применение режимов ASCII и RTU никак не связано с топологией, но для каждого режима определена структура кадров сообщений и их синхронизация. В процессе передачи по каналам последовательной связи оба режима предусматривают асинхронную передачу, при которой имеется заранее определённая структура кадра и символы пересылаются последовательно — по одному в каждый момент.

Рисунок 2 иллюстрирует отправку символа при использовании асинхронной последовательной передачи данных. Каждый символ передаётся как последовательность битов, причем время, затрачиваемое на передачу одного бита, обратно пропорционально скорости передачи данных (бод). Например, при скорости 9600 бод время передачи 1 бита равно 104,1 мкс. Когда информация не передаётся, говорят о

Стартовый бит	1	2	3	4	5	6	7	Бит чётности	Стоповый бит	
Стартовый бит	1	2	3	4	5	6	7	Стоповый бит	Стоповый бит	
Стартовый бит	1	2	3	4	5	6	7	8	Бит чётности	Стоповый бит
Стартовый бит	1	2	3	4	5	6	7	8	Стоповый бит	Стоповый бит

Рис. 2. Структура кадра для 7-битового режима ASCII и 8-битового режима RTU с битом чётности или без него

маркерном (marking) состоянии линии связи. Противоположное ему состояние именуется заполненным (spacing). Когда линия переходит в заполненное состояние для побитовой передачи данных, каждому символу предшествует стартовый бит, а в конце идёт один стоповый бит или больше, после этого линия возвращается в маркерное состояние.

В промежутке между стартовым и стоповым битами осуществляется передача 7 (режим ASCII) или 8 (режим RTU) битов, составляющих символ, причём первым посылается младший бит (LSB). После символа идёт либо бит чётности, либо еще один стоповый бит. При этом пользователь имеет возможность выбирать один из трёх вариантов: контроль на чётность, или на нечётность, либо отсутствие контроля. В режиме ASCII передача одного символа требует передачи 10 битов, а в режиме RTU — 11. При асинхронной связи символы могут пересылаться либо вплотную, либо с временным интервалом между ними. Последовательности символов, образующих сообщения, имеют различные структуры в зависимости от режима — ASCII или RTU.

Структура кадра сообщения в режиме ASCII

Семибитовый код ASCII был разработан в начале 60-х годов прошлого века как универсальный код для отображения символов английского языка для телетайпов, таких как, например, Teletype Model ASR-33. Когда на смену электромеханическим телетайпам стали приходить устройства с мониторами на ЭЛТ, стандарт ASCII был сохранён, что облегчило процесс перехода на новую технику. ASCII является принятым в США стандартом для представления символов английского языка и управляющих символов, например, CR (возврат каретки) и LF (перевод строки). Наименования этих символов сохранились со времён электромеханических телетайпов. Причиной того, что перед символом LF всегда передаётся символ CR, была необходимость дать время каретке телетайпа переместиться из конца в начало строки. Команда LF предназначалась для перемещения бумаги по вертикали, но поскольку данная операция занимает меньше времени, чем перемещение каретки, то она посылается в последовательности второй.

Электромеханический телетайп не имел буфера данных, поэтому если вместо последовательности CRLF была передана последовательность LFCR, то печать могла начаться с середины строки, а не с её левого края, поскольку возможна ситуация, что каретка не успела вернуться в начало строки до момента получения очередного подлежащего печати символа. Последовательность команд CRLF была очень важной, когда речь шла о телетайпах, работавших на скорости 10 символов/с, но когда на смену им пришли ЭЛТ-терминалы, ситуация изменилась. Для протокола Modbus в режиме ASCII последовательность CRLF сейчас просто указывает на конец кадра. Преимуществом данного режима является то, что если в качестве ведомого устройства включить ЭЛТ-терминал, то можно увидеть на экране понятный человеку отлично отформатированный код, который послан ведущим устройством на экран ЭЛТ-терминала.

На рис. 3 показана структура сообщения Modbus в режиме ASCII. Его начало обозначается символом «:», а конец — последовательностью CRLF (два символа ASCII). Любой символ ASCII представляется 7 битами. Все остальные символы во всех остальных полях фрейма должны быть либо цифрами от 0 до 9, либо буквами от A до F, так как предполагается, что данные представляются в шестнадцатеричном формате, но отображаются в виде символов ASCII. Например, код функции 03 будет отображаться двумя ASCII-символами — «0» и «3». То же самое относится и к содержимому поля данных. Одним из преимуществ режима ASCII является то, что он не предъявляет особо жёстких требований к синхронизации. Допускается временной промежуток между символами до 1 с — только по истечении его генерируется сообщение о превышении лимита времени. Так что квалифицированная машинистка может имитировать работу ведущего устройства, печатая на экране ЭЛТ-терминала строку символов, передаваемую ведомому устройству, и наблюдая на экране ответное сообщение последнего.

Начало	Адрес устройства	Код функции	Данные	Контрольная сумма (LRC)	Конец
1 символ (:)	2 символа	2 символа	n символов	2 символа	2 символа (CRLF)

Рис. 3. Структура кадра сообщения Modbus ASCII

Структура кадра сообщения в режиме RTU

При работе в режиме RTU синхронизация имеет более важное значение, чем в режиме ASCII. В этом варианте специальный начальный символ отсутствует. Вместо этого кадр сообщения начинается с маркерного интервала, длительность которого равна времени передачи четырёх символов. После истечения этого интервала передаётся адрес устройства, затем код функции и собственно данные. Имеются и другие отличия от кадра сообщения в режиме ASCII, как это показано на рис. 4. Вместо контрольной суммы LRC (Longitudinal Redundancy Check — продольный контроль по избыточности) в режиме RTU используется контрольная сумма CRC (Cyclic Redundancy Check — циклический контроль по избыточности). Конец кадра отмечается маркерным интервалом, равным времени передачи четырёх символов.

RTU-сообщения должны посылаться в виде непрерывного потока, и появление значительного временного «зазора» между смежными символами рассматривается как прерывание в передаче сообщения. В отличие от режима ASCII сообщения в режиме RTU не поддаются считыванию человеком.

Однако сообщения в этом режиме весьма компактны и более эффективны с точки зрения их передачи. Поэтому режим RTU является более популярным.

Таблица распределения регистров Modbus

Прежде чем перейти к рассмотрению кодов функций, необходимо описать таблицу распределения регистров Modbus, показанную на рис. 5,

Начало	Адрес устройства	Код функции	Данные	Контрольная сумма CRC	Конец
Интервал, равный времени передачи 4 символов	8 бит	8 бит	n×8 бит	16 бит	Интервал, равный времени передачи 4 символов

Рис. 4. Структура кадра сообщения Modbus RTU

Адреса регистров	Описание
00001 – 10000	Дискретные выходы (чтение/запись)
10001 – 20000	Дискретные входы (чтение)
30001 – 40000	16-битовые аналоговые входы (чтение)
40001 – 50000	Регистры хранения (чтение/запись)

Рис. 5. Таблица распределения регистров Modbus

поскольку коды некоторых функций привязаны к конкретным диапазонам регистров. Первые по времени создания ПЛК имели дело, главным образом, с дискретными входными и дискретными выходными сигналами. При этом каждый дискретный вход и дискретный выход представлен в таблице распределения регистров 1 битом. Для ПЛК компании Modicon дискретные выходы начинаются с адреса (ячейки) 00001, а дискретные входы — с адреса 10001. Каждому из них требуется 1 бит памяти. Содержимое входных регистров (в терминологии первых контроллеров они именовались contacts) можно только читать, в то время как содержимое выходных регистров (в терминологии первых ПЛК они именовались coils) можно и читать, и записывать.

По мере увеличения сложности ПЛК появились средства работы с аналоговыми входами-выходами и средства выполнения вычислений. Регистры аналоговых входов и выходов являются 16-разрядными. Их адреса начинаются с 30001 — это адрес первого аналогового входа (только чтение, например, для ввода сигналов от барабанных переключателей). С адреса 40001 начинается диапазон универсальных регистров (чтение и запись), которые могут служить также и аналоговыми выходами. В зависимости от фирмы-изготовителя ПЛК эти регистры могут быть внутренними регистрами, аналоговыми входами, аналоговыми выходами и даже дискретными входами и выходами. Однако не все функциональные коды работают с адресами этих регистров.

Коды функций

Коды функций определены в документации Modicon Modbus Reference Guide и Modbus Application Protocol Specification. Поскольку в этих документах имеются разночтения по наименованиям функций и количеству функциональных кодов, то рекомендуется пользоваться вторым из них. Несмотря на то что кодам функций отведён диапазон от 1 до 127, в качестве предназначенных для общего пользования кодов определены примерно 20 кодов. В этот же диапазон входят коды, назначение которых определяется каждым отдельным пользователем. Следует иметь в виду, что многие Modbus-устройства поддерживают только небольшие подмножества имеющихся кодов. Чтобы понять, как происходит работа с входами-выходами, мы рассмотрим только те функциональные коды, которые обеспечивают доступ к однобитовым и 16-битовым данным. Перечень рассматриваемых кодов приведён на рис. 6.

Здесь пора разъяснить один из наиболее запутанных аспектов протокола Modbus, касающийся ссылок на адреса входов-выходов. При разработке протокола компания Modicon приняла решение нумеровать физические координаты точек, входящих в тот или иной диапазон, начиная с 0, а не с 1. Дискретный выход 1 адресуется в сообщении как ячейка 0000, а не 00001. Дискретный вход 1 адресуется как ячейка 0000, а не 10001. То же самое относится и к регистру хранения 1, которому поставлена в соответствие ячейка 40001, — его адрес записывается в виде 0000. Код функции всегда ассоциирован с конкретным диапазоном адресов входов-выходов, и поэтому для однозначной идентификации координат точки достаточно указать величину её смещения относительно базового адреса.

Код	1 или 16 бит	Описание	Диапазон адресов входов-выходов
1	1	Read coils Чтение текущего состояния (ON/OFF) дискретных выходов	00001 – 10000
2	1	Read contacts Чтение текущего состояния (ON/OFF) дискретных входов	10001 – 20000
5	1	Write a single coil Изменение состояния дискретного выхода в ON или OFF	00001 – 10000
15	1	Write multiple coils Изменение состояния (ON/OFF) нескольких дискретных выходов	00001 – 10000
3	16	Read holding registers Чтение регистров хранения	40001 – 50000
4	16	Read input registers Чтение входных регистров	30001 – 40000
6	16	Write single register Запись одного регистра	40001 – 50000
16	16	Write multiple registers Запись нескольких регистров	40001 – 50000
22	16	Mask write register Маскированная запись регистра	40001 – 50000
23	16	Read/write multiple registers Чтение/запись нескольких регистров	40001 – 50000
24	16	Read FIFO queue Чтение содержимого очереди FIFO	40001 – 50000

Рис. 6. Коды функций, обеспечивающие доступ к данным

Как видно из рис. 6, одноразрядные коды функций относятся к дискретным входам и выходам, а 16-разрядные — к входным регистрам и регистрам хранения. При этом к входным регистрам применима только функция чтения, а к выходным — как чтения, так и записи. Заметим также, что имеется чёткая привязка каждого кода функции к определённому диапазону адресов входов-выходов. Например, функциональный код 06 (запись одного регистра) относится только к диапазону адресов 40001 — 50000 и ни к какому другому. Следовательно, при описании сообщения можно указывать не абсолютный адрес входа или выхода, а задать величину смещения относительно базового адреса. Таким образом, в данном случае вместо адреса 40001 мы указываем просто 0000.

Смещение записывается в виде 16-разрядного слова и при просмотре реального Modbus-сообщения отображается соответствующим шестнадцатеричным числом, а в таблице распределения регистров Modbus все адреса даются в виде десятичных чисел. Поэтому регистру 40016 соответствует число 0x000F, которое является шестнадцатеричным представлением разности 40016 — 40001. На первый взгляд всё кажется запутанным, но на самом деле это важно только для специалистов, которые занимаются написанием драйверов для систем с протоколом Modbus. ●

Автор — президент компании Contemporary Controls