Применение SWITCH-технологии при разработке прикладного программного обеспечения для микроконтроллеров.

Часть 6. Реализация протокола Modbus

В предыдущих статьях цикла мы подробно рассмотрели ряд аспектов, связанных с применением SWITCH-технологии в микроконтроллерных устройствах. Были рассмотрены такие важные программные структуры, как механизм передачи сообщений, локальные и глобальные таймеры. Сейчас мы можем перейти к рассмотрению ряда распространенных алгоритмов на базе SWITCH-технологии, находящих применение в современных микроконтроллерных устройствах. В качестве первого примера рассмотрим реализацию широко распространенного в системах промышленной автоматики протокола Modbus.

Владимир ТАТАРЧЕВСКИЙ
arktur04@mail.ru

Краткое введение в протокол Modbus

Протокол Modbus получил широкое распространение в современных системах автоматики. Спецификация протокола не определяет тип физического уровня сети передачи данных, оставляя выбор за разработчиком. На практике широкое распространение получила связка, состоящая из сети RS-485 в качестве физической среды передачи данных и протокола Modbus в качестве логического уровня сети. Такое решение позволяет использовать Modbus в самом широком спектре микроконтроллерных устройств, начиная с самых простых, таких как различные интеллектуальные датчики, и заканчивая сложными распределенными системами на базе программируемых логических контроллеров.

Протокол построен по схеме «ведущий–ведомый» (master–slave). В системе выделяется одно ведущее устройство (мастер), которое инициализирует любую транзакцию в сети. Все остальные устройства являются ведомыми и выполняют команды мастера или передают информацию в ответ на запрос мастера. Протокол предусматривает два типа адресации: индивидуальную, когда сообщение адресуется одному ведомому устройству и широковещательную (broadcast messages), при которой сообщение адресуется всем устройствам сети. При индивидуальной адресации ведомое устройство возвращает мастеру ответное сообщение, при широковещательной адресации ответные сообщения не посылаются.

Сообщения, посылаемые мастером, имеют следующую структуру: адрес ведомого устройства (или код широковещательного сообщения), код, определяющий действия ведомого устройства, данные и контрольная сумма. Ответное сообщение состоит из поля, подтверждающего выполнение действия, данных и контрольной суммы. Если при приеме сообщения от мастера произошла ошибка, либо ведомое устройство по каким-либо причинам не может выполнить запрашиваемое действие, ведомое устройство посылает мастеру сообщение об ошибке.

Существует также расширенная версия протокола, называемая Modbus Plus. Протокол Modbus Plus формирует одноранговую сеть, в которой любое устройство может инициализировать транзакцию и, таким образом, каждое устройство может выступать в роли как ведущего, так и ведомого в разных транзакциях. Однако мы не будем рассматривать Modbus Plus в рамках данной статьи.

Сеть Modbus может работать в одном из двух режимов: RTU и ASCII. В режиме RTU информация передается «как есть», в двоичном коде. В режиме ASCII информация передается в текстовом виде как последовательность символов '0'–'9', 'A'–'F' в ASCII-кодировке.

Выбор режима RTU или ASCII, а также настройки сети, такие как скорость передачи, бит четности и т. п., выбираются пользователем при конфигурировании контроллеров и должны быть одинаковы для всех устройств в сети.

Рассмотрим оба режима протокола Modbus.

Режим ASCII

Главным преимуществом данного режима является то, что символы могут передаваться с интервалом вплоть до одной секунды без возникновения ошибки передачи данных. Недостатком данного режима является его сниженная (более чем в два раза) информационная пропускная способность по сравнению с режимом RTU при равной скорости физической линии. Передача символа осуществляется в следующем формате: 1 стартовый бит, 7 бит данных (младший разряд передается первым), 1 бит контроля четности (или нечетности; при отсутствии контро-

Таблица 1. Передача символа ASCII-кода

№ бита	0	1–7	8	9
С контролем четности	Стартовый бит	7 бит данных	Бит контроля четности	Стоповый бит
Без контроля четности	Стартовый бит	7 бит данных	Стоповый бит	Стоповый бит

Таблица 2. Структура сообщения в формате ASCII

Начало	Адрес	Код функции	Данные	Контрольная сумма	Конец
':' 3Ah	2 символа	2 символа	n символов	2 символа	CRLF 0D0Ah

Таблица 3. Передача символа ASCII-кода

№ бита	0	1–8	9	10
С контролем четности	Стартовый бит	8 бит данных	Бит контроля четности	Стоповый бит
Без контроля четности	Стартовый бит	8 бит данных	Стоповый бит	Стоповый бит

ля четности данный бит отсутствует), 1 стоповый бит, если контроль четности присутствует, 2 стоповых бита при отсутствии контроля четности (табл. 1).

Сообщение в данном режиме начинается с символа двоеточия (код 3Ah), заканчивается последовательностью символов «возврат каретки», «перевод строки» (CRLF, код 0D0Ah). Между передачей символов возможны интервалы времени до 1 секунды. При превышении тайм-аута принимающее устройство фиксирует ошибку передачи. Структура сообщения приведена в таблице 2.

В некоторых ранних реализациях протокола сообщение заканчивается контрольной суммой, без последовательности CRLF. Таким образом, принимающее устройство должно выждать как минимум 1 секунду после приема контрольной суммы, и если последовательность CRLF не получена, сообщение считается успешно принятым.

Режим RTU

Передача символа в режиме RTU состоит из следующих этапов: 1 стартовый бит, 8 бит данных (младший разряд передается первым), 1 бит контроля четности (нечетности), при отсутствии контроля четности данный бит отсутствует, 1 стоповый бит, если контроль четности присутствует, 2 стоповых бита при отсутствии контроля четности (табл. 3).

Сообщение представляет собой последовательность символов, передаваемых непрерывно, без пауз. При возникновении паузы длительностью более 1,5 t, где t — время передачи одного символа при заданной скорости передачи, принимающее устройство фиксирует ошибку в приеме сообщения и начинает прием нового сообщения после паузы. Структура сообщения приведена в таблице 4.

Таблица 4. Структура сообщения в формате RTU

Адрес	Код функции	Данные	Контрольная сумма
1 символ	1 символ	n символов	2 символа

Сообщение предваряется и заканчивается паузой не менее 3,5t при типичном значении 4t.

Адресация в протоколе Modbus

Значение адреса ведомого устройства находится в диапазоне 1–247. Широковещательные сообщения передаются с адресом 0.

Код команды Modbus

Код команды может принимать значение в диапазоне 1–255. Когда ведомое устройство отвечает на запрос мастера, оно использует код команды для индикации правильности выполнения команды. Если операция выполнена успешно, ведомое устройство просто повторяет код команды. Если во время выполнения операции по какой-либо причине произошла ошибка или операцию выполнить невозможно, ведомое устройство устанавливает старший бит команды в 1.

Поле данных в протоколе Modbus

Содержание и структура поля данных зависит от конкретной команды, и для ряда команд может вообще отсутствовать. Структуры данных для каждой конкретной команды описаны в спецификации протокола [1] и в данной статье не рассматриваются.

Контрольная сумма сообщения в протоколе Modbus

Контрольная сумма для режимов ASCII и RTU рассчитывается по разным алгоритмам. Алгоритм подсчета контрольной суммы для режима ASCII носит название LRC (Longitudinal Redundancy Check), результатом его работы являются 2 ASCII-символа, а для режима RTU контрольная сумма вычисляется по алгоритму CRC (Cyclical Redundancy Check), результатом работы которого является 16-битное число. Оба алгоритма подробно описываются в руководстве [1] и здесь не приводятся.

Реализация протокола Modbus

Рассмотрим реализацию протокола Modbus на основе SWITCH-технологии. Следует отметить, что преимуществом реализации алгоритма обмена данными в виде конечного автомата с применением SWITCH-технологии является возможность работы прикладной программы одновременно с приемом

Таблица 5. Структура буфера данных для передачи

№ поля	Наименование	Назначение поля	Размер поля, байт
1	Addr	Адрес ведомого устройства	1
2	Command	Код команды	1
3	Num_bytes	Количество байт данных	1
4	Data[i]	Данные	Определяется полем 3

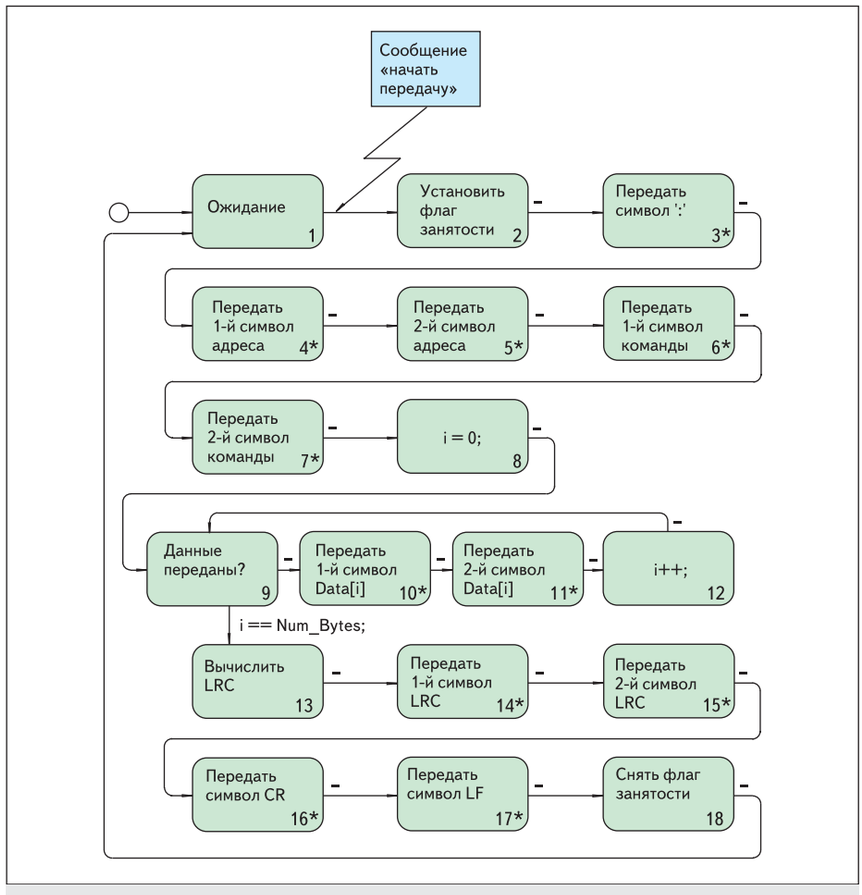

Рис. 1. Передача сообщения в режиме ASCII

и передачей сообщений, что сводит к сокращению до минимума задержки в работе прикладной программы на время приема или передачи сообщения, в особенности в режиме ASCII. Минимизация задержек при обмене данными имеет особенно важное значение при построении систем реального времени, таких как системы управления технологическими процессами.

Реализация передачи сообщен ия мастером в режиме ASCII

При передаче сообщения мастеров в режиме ASCII программа контроллера подготавливает данные, содержащие адрес устройства, код команды, данные команды и размер поля данных в байтах, и записывает их в выделенный буфер, имеющий структуру, которая представлена в таблице 5.

Указатель на буфер передается автомату Modbus в качестве параметра сообщения передачи. Во время передачи автомат Modbus устанавливает специальный флаг занятости, который должна проверять прикладная программа перед передачей. В случае, если флаг установлен, прикладная программа должна ждать его освобождения. Для преодоления данного ограничения возможно введение автомата, реализующего очередь сообщений и служащего промежуточным слоем между прикладной программой и автоматом Modbus. Граф автомата передачи сообщения в режиме ASCII изображен на рис. 1. Из рисунка видно, что символы передаются по одному, в различных состояниях, при этом прикладная программа (также написанная по SWITCH-технологии) может выполняться одновременно с передачей сообщения. Состояния, номера которых отмечены звездочкой, являются сложными состояниями (суперсостояниями), и имеют для данной граф-схемы одну структуру, показанную на рис. 2. Смысл введения суперсостояний в данном случае заключается в том, что UART может не успеть передать предыдущий символ, поэтому перед передачей очередного символа осуществляется проверка занятости UART. В силу свойств SWITCH-технологии остановки прикладной программы при этом не происходит. Единственным ограничением на работу цикла прикладной программы является то, что длительность одной итерации данного цикла не может превышать 1 секунды в соответствии со спецификацией протокола Modbus.

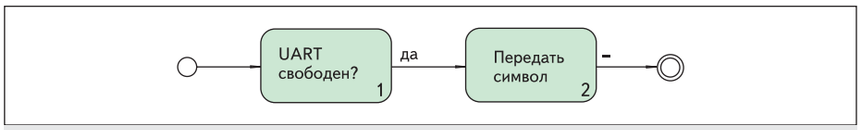

Рис. 2. Передача символа

В следующей статье мы продолжим рассмотрение реализации протокола Modbus на основе SWITCH-технологии.

Автор выражает глубокую благодарность Анатолию Абрамовичу Шалыто за ценные замечания и редактирование статьи.

Литература

1. Modicon Modbus Protocol Reference Guide. MODICON, Inc., Industrial Automation Systems, 1996.