Астана +7(7172)727-132, Волгоград (844)278-03-48, Воронеж (473)204-51-73, Екатеринбург (343)384-55-89,
Казань (843)206-01-48, Краснодар (861)203-40-90, Красноярск (391)204-63-61, Москва (495)268-04-70,
Нижний Новгород (831)429-08-12, Новосибирск (383)227-86-73, Ростов-на-Дону (863)308-18-15,
Самара (846)206-03-16, Санкт-Петербург (812)309-46-40, Саратов (845)249-38-78, Уфа (347)229-48-12
Россия, Казахстан и другие страны ТС доставка в любой город
Единый адрес: gpm@nt-rt.ru
Веб-сайт: http://gmp.nt-rt.ru

ПРОТОКОЛ MODBUS

Описание протокола

Протокол MODBUS

1 ОБЩИЕ СВЕДЕНИЯ

Для обмена данными в сети нужны, как минимум, два устройства. Одно из них - главное устройство MASTER (в дальнейшем будем называть его ЗАКАЗЧИК), которое может начать обмен данными, отправив в сеть пакет с инструкциями, а другое - подчиненное устройство SLAVE (в дальнейшем будем называть его ИСПОЛНИТЕЛЬ), которое обрабатывает принятые инструкции.. Порядок обмена данными в сети называется протоколом обмена.

Протокол необходимая часть работы системы. Он определяет как ЗАКАЗЧИК и ИСПОЛНИТЕЛЬ устанавливают и прерывают контакт, как идентифицируются отправитель и получатель, каким образом происходит обмен сообщениями, как обнаруживаются ошибки. Протокол управляет циклом запроса и ответа, который происходит между устройствами ЗАКАЗЧИК и ИСПОЛНИТЕЛЬ.

Протокол подразумевает, что в сети один ЗАКАЗЧИК и до 247 ИСПОЛНИТЕЛЕЙ. Хотя протокол и поддерживает до 247 ИСПОЛНИТЕЛЕЙ, драйвер двухпроводной линии RS-485 обычно поддерживает 32 ИСПОЛНИТЕЛЯ. Каждому ИСПОЛНИТЕЛЮ присвоен уникальный адрес устройства в диапазоне от 1 до 247.

Только ЗАКАЗЧИК может инициировать транзакцию. Транзакции бывают либо типа запрос/ответ (адресуется только один ИСПОЛНИТЕЛЬ), либо широковещательные - без ответа (адресуются все ИСПОЛНИТЕЛИ). Транзакция содержит один кадр запроса и один кадр ответа, либо один кадр широковещательного запроса.

Некоторые характеристики протокола Modbus фиксированы. К ним относятся формат кадра, последовательность кадров, обработка ошибок и исключительных ситуаций, и выполнение функций.

Другие характеристики выбираются пользователем. К ним относятся тип связи, скорость обмена, проверка на четность и число стоповых бит, Эти параметры не могут быть изменены во время работы системы.

При передаче по линиям данных, сообщения помещаются в «конверт». «Конверт» покидает устройство через «порт» и «пересылается» по линиям адресуемому устройству. Протокол Modbus описывает «конверт» в форме кадров сообщений. В сообщении есть *АДРЕС* получателя, *ФУНКЦИЯ*, которую получатель должен выполнить, *ДАННЫЕ*, необходимые для выполнения этой функции, и *КОНТРОЛЬНАЯ СУММА* для контроля достоверности.

Когда сообщение достигает ИСПОЛНИТЕЛЯ, он вскрывает конверт, читает сообщение, и, если не возникло ошибок, выполняет требуемую задачу. Затем ИСПОНИТЕЛЬ помещает в конверт ответное сообщение и посылает его ЗАКАЗЧИКУ. В ответном сообщении есть *АДРЕС* устройства, *ФУНКЦИЯ*, которая была выполнена, *ДАННЫЕ*, полученные в результате выполнения задачи, и *КОНТРОЛЬНАЯ СУММА* для контроля достоверности.

Если сообщение было широковещательным (сообщение для всех ИСПОЛНИТЕЛЕЙ), на что указывает адрес 0, то ответное сообщение не передается.

Обычно ЗАКАЗЧИК посылает следующее сообщение другому ИСПОЛНИТЕЛЮ после приема корректного ответа, либо после истечения времени ожидания ответа (тайм-аута). Все сообщения могут рассматриваться как запросы ЗАКАЗЧИКА, генерирующие ответные сообщения ИСПОЛНИТЕЛЯ. Широковещательные сообщения могут рассматриваться как запросы, не требующие ответных сообщений.

1.1 Режимы передачи

Режим передачи определяет структуру отдельных блоков информации в сообщении и системы счисления, используемую для передачи данных. В системе Modbus существуют два режима

передачи ASCII и RTU (Remote Terminal Unit). Мы используем режим передачи RTU, поэтому будем описывать протокол Modbus-RTU.

В режиме RTU данные передаются непрерывным потоком в виде 8-разрядных двоичных символов.

1.2 Обнаружение ошибок

Существует два типа ошибок, которые могут возникать в системах связи: ошибки передачи и программные или оперативные ошибки. Система Modbus имеет способы определения каждого типа ошибок.

Ошибки связи обычно заключаются в изменении бита или бит сообщения. Например, байт 0001 0100 может измениться на 0001 0110. Ошибки связи выявляются при помощи символа кадра, контроля по четности и избыточным кодированием.

Когда обнаруживается ошибка кадрирования, четности и контрольной суммы, обработка сообщения прекращается. ИСПОЛНИТЕЛЬ не должен генерировать ответное сообщение. Тот же результат будет, если был использован адрес несуществующего ИСПОЛНИТЕЛЯ.

Если возникает ошибка связи, данные сообщения ненадежны. Устройство ИСПОЛНИТЕЛЬ не может с уверенностью определить, что сообщение было адресовано именно ему. Иначе ИСПОЛНИТЕЛЬ может ответить сообщением, которое не является ответом на исходный запрос. Устройство ЗАКАЗЧИК должно программироваться так, чтобы в случае не получения ответного сообщения в течение определенного времени, ЗАКАЗЧИК должен фиксировать ошибку связи. Продолжительность этого времени зависит от скорости обмена, типа сообщения, и времени опроса ИСПОЛНИТЕЛЬ. По истечению этого периода, ЗАКАЗЧИК должен быть запрограммирован на ретрансляцию сообщения.

Для обеспечения качества передачи данных система Modbus обеспечивает несколько уровней обнаружения ошибок. Для обнаружения множественного изменения битов сообщения система использует избыточный контроль: CRC. Обнаружение ошибок с помощью CRC выполняется автоматически.

1.3 Кадровая синхронизация

В режиме RTU началом нового кадра является тишина в сети в течение времени прохождения 3.5 символов (T+T+T+T/2, где T – время прохождения символа при выбранной скорости приёма/передачи данных). ИСПОЛНИТЕЛЬ считает время после прихода символа, и если прошло время, равное периоду следования 3.5 символов, то обрабатывает принятые данные. Следующий принимаемый байт - это адрес устройства в новом сообщении.

Таблица 1

Формат кадра сообщения в режиме RTU

T+T+T+T/2	Адрес	Функция	Данные	Контрольная сумма	T+T+T+T/2
	8 бит	8 бит	N * 8 бит	16 бит	

1.1.1 Поле адреса

Поле адреса следует сразу за началом кадра и состоит из одного 8-разрядного символа. Эти биты указывают адрес устройства, которое должно принять сообщение, посланное ЗАКАЗЧИКОМ. Каждый ИСПОЛНИТЕЛЬ должен иметь уникальный адрес, и только адресуемое устройство может ответить на запрос, который содержит его адрес. В ответном сообщении адрес информирует ЗАКАЗЧИКА, с каким ИСПОЛНИТЕЛЕМ установлена связь. В

широковещательном режиме используется адрес 0. Все ИСПОЛНИТЕЛИ интерпретируют такое сообщение как выполнение определенного действия, но без посылки подтверждения.

1.1.2 Поле функции

Поле кода функции указывает адресуемому ИСПОЛНИТЕЛЮ, какое действие выполнить. Коды функций Modbus специально разработаны для связи ПК и индустриальных коммуникационных систем Modbus.

Старший бит этого поля устанавливается в единицу ИСПОЛНИТЕЛЕМ в случае, если он хочет просигналить ЗАКАЗЧИКУ, что ответное сообщение содержит ошибку. Этот бит остается нулём, если ответное сообщение повторяет запрос или в случае нормального сообщения.

Таблица 2

Коды используемых функций Modbus

Код	Название	Действие
03	READ HOLDING REGISTERS	Получение текущего значения одного или нескольких регистров хранения.
06	FORCE SINGLE REGISTER	Запись нового значения в регистр.
16	FORCE MULTIPLE REGISTERS	Установить новые значения нескольких последовательных регистров.

1.1.3 Поле данных

Поле данных содержит информацию, необходимую ИСПОЛНИТЕЛЮ для выполнения указанной функции, если это запрос, или содержит данные, подготовленные ИСПОЛНИТЕЛЕМ, если это ответ на запрос. Данные передаются старшим байтом вперёд (1→0). Если передаётся 4-байтовое число (2 регистра) с плавающеё запятой, то в каждом из 2-х регистров порядок следования байт тоже старшим байтом вперёд (1→0→3→2).

1.1.4 Поле контрольной суммы

Это поле позволяет ЗАКАЗЧИКУ и ИСПОЛНИТЕЛЮ проверять сообщение на наличие ошибок. Иногда, вследствие электрических помех или других воздействий, сообщение при пересылке от одного устройства к другому может незначительно измениться. Результат проверки контрольной суммы укажет ИСПОЛНИТЕЛЮ или ЗАКАЗЧИКУ реагировать или нет на такое сообщение. Это увеличивает надежность и эффективность систем MODBUS.

В Modbus-RTU применяется циклический код CRC-16 (Cyclic Redundancy Check). Сообщение (только биты данных, без учета старт/стоповых бит и бит четности) рассматриваются как одно последовательное двоичное число, у которого старший значащий бит (MSB) передается первым. Сообщение умножается на X^{16} (сдвигается влево на 16 бит), а затем делится на $X^{16}+X^{15}+X^2+1$, выражаемое как двоичное число (11000000000000101). Целая часть результата игнорируется, а 16-ти битный остаток (предварительно инициализированный единицами для предотвращения случая, когда все сообщение состоит из нулей) добавляется к сообщению как два байта контрольной суммы. Полученное сообщение, включающее CRC, затем в приемнике делится на тот же полином ($X^{16}+X^{15}+X^2+1$). Если ошибок не было, остаток от деления должен получится нулевым. Получатель сообщения должен рассчитать CRC-код и сравнить его с полученным кодом. Вся арифметика выполняется по модулю 2 (без переноса).

1.4 Исключительные ситуации

Коды исключительных ситуаций приведены в таблице. Когда ИСПОЛНИТЕЛЬ обнаруживает одну из этих ошибок, он посылает ответное сообщение ЗАКАЗЧИКУ, содержащее адрес ИСПОЛНИТЕЛЯ, код функции, код ошибки и контрольную сумму. Для указания на то, что ответное сообщение – это уведомление об ошибке, старший бит поля кода функции устанавливается в 1.

Таблица 3

Коды ошибок

Код	Название	Смысл
01	ILLEGAL FUNCTION	Функция в принятом сообщении не поддерживается на данном ИСПОЛНИТЕЛЕ.
02	ILLEGAL DATA ADDRESS	Адрес, указанный в поле данных, является недопустимым для данного ИСПОЛНИТЕЛЯ.
03	ILLEGAL DATA VALUE	Значения в поле данных недопустимы для данного ИСПОЛНИТЕЛЯ.
04	SLAVE DEVICE FAILURE	ИСПОЛНИТЕЛЬ не может записать данные во FRAM память.

2 ФУНКЦИИ

Цель данного раздела - определить общий формат соответствующих команд, доступных программисту системы MODBUS. В разделе описаны формат каждого запросного сообщения, выполняемая функция и формат нормального ответного сообщения.

2.1 Функция 03 (Чтение регистров/Read Holding Registers)

Применяется для чтения двоичного содержания регистров ИСПОЛНИТЕЛЯ.

ЗАПРОС:

Сообщение запроса специфицирует начальный регистр и количество регистров для чтения. Нумерация регистров начинается с 0 (регистры 1-16 нумеруются как 0-15).

Таблица 4.

Запрос на чтение регистров 42-43 ИСПОЛНИТЕЛЯ с адресом 1.

Номер байта	Номер байта в числе	Условное обозначение	Пример	
0	-	Адрес	01	01
1	-	Функция	03	03
2	[1]	Начальный адрес	000B	00
3	[0]			0B
4	[1]	Количество регистров	0002	00
5	[0]			02
6	[1]	Контрольная сумма	B5C9	B5
7	[0]			C9

ОТВЕТ:

Данные регистров в ответе передаются как два байта на регистр. Байты регистров передаются старшим байтом вперёд. Количество регистров передаваемых за одно обращение определяется возможностями ИСПОЛНИТЕЛЯ.

Таблица 5.

Ответ на команду чтение регистров 42-43 ИСПОЛНИТЕЛЯ с адресом 1.

Номер байта	Номер байта в числе	Условное обозначение	Пример	
0	-	Адрес	01	01
1	-	Функция	03	03
2		Счётчик байт	04	
3	[1]	Данные регистр 11	0000	00
4	[0]			00
5	[1]	Данные регистр 12	D20F	D2
6	[0]			0F
7	[1]	Контрольная сумма	E697	E6
8	[0]			97

2.2 Функция 06 (Запись одного регистра/Preset Single Register)

Применяется для записи значения в единичный регистр. При широковещательной передаче на всех ИСПОЛНИТЕЛЯХ устанавливается один и тот же регистр.

Обычно используется для первоначальной установки адреса ИСПОЛНИТЕЛЯ.

ЗАПРОС:

Запрос содержит ссылку на регистр, который необходимо установить и значение, которое надо в него записать.

Таблица 6.

Запрос на запись регистра 00 ИСПОЛНИТЕЛЯ с адресом 1.

Номер байта	Номер байта в числе	Условное обозначение	Пример	
0	-	Адрес	01	01
1	-	Функция	06	06
2	[1]	Адрес регистра	0000	00
3	[0]			00
4	[1]	Данные	0100	01
5	[0]			00
6	[1]	Контрольная сумма	885A	88
7	[0]			5A

ОТВЕТ:

Нормальный ответ повторяет запрос.

Таблица 7.

Ответ на запрос записи регистра 00 ИСПОЛНИТЕЛЯ с адресом 1.

Номер байта	Номер байта в числе	Условное обозначение	Пример	
0	-	Адрес	01	01
1	-	Функция	06	06
2	[1]	Адрес регистра	0000	00
3	[0]			00
4	[1]	Данные	0100	01
5	[0]			00
6	[1]	Контрольная сумма	885A	88
7	[0]			5A

2.3 Функция 16 (Запись в регистры/Preset Multiple Regs)

Применяется для записи значений в последовательность регистров. Запрос указывает регистры для записи, их количество и данные, которые содержатся в поле данных запроса.

Количество регистров записываемых за одно обращение определяется возможностями ИСПОЛНИТЕЛЯ.

ЗАПРОС:

Запрос содержит ссылку на регистр, который необходимо установить и значение, которое надо в него записать.

Таблица 8.

Запрос на запись в регистры с 0 по 2 ИСПОЛНИТЕЛЯ с адресом 1.

Номер байта	Номер байта в числе	Условное обозначение	Пример	
0	-	Адрес	01	01
1	-	Функция	10	10
2	[1]	Начальный адрес	0000	00
3	[0]			00
4	[1]	Количество регистров	0003	00
5	[0]			03
6	-	Счётчик байт	06	06
7	[1]	Данные	0119	01
8	[0]			19
9	[1]	Данные	0405	04
8	[0]			05
10	[1]	Данные	0204	03
11	[0]			04
12	[1]	Контрольная сумма	EB01	EB
13	[0]			01

ОТВЕТ:

Нормальный ответ содержит адрес ИСПОЛНИТЕЛЯ, код функции, начальный адрес, и количество регистров.

Таблица 9.

Ответ на запрос записи регистров 0-2 ИСПОЛНИТЕЛЯ с адресом 1.

Номер байта	Номер байта в числе	Условное обозначение	Пример	
0	-	Адрес	01	01
1	-	Функция	10	10
2	[1]	Начальный адрес	0000	00
3	[0]			00
4	[1]	Количество регистров	0003	00
5	[0]			03
6	[1]	Контрольная сумма	8008	80
7	[0]			08

Для контроля записи регистров можно послать запрос на чтение регистров 0-2 ИСПОЛНИТЕЛЯ с адресом 1: 01 03 00 00 00 03 05 CB и если всё было записано правильно, от ИСПОНИТЕЛЯ придёт ответ: 01 03 06 01 19 04 05 02 04 2C F4.

3 ОПИСАНИЕ РЕГИСТРОВ

3.1 Структура данных

Ниже приведена структура данных, используемая для настройки метекомплекса МК-26. Все параметры структуры доступны для записи и чтения с помощью функций протокола Modbus.

```
typedef struct {
        _U8         object;     // адрес ИСПОЛНИТЕЛЯ
        _U8         algoritm;   // настройка
                                // 0 – стандартный режим работы
                                // 1 – ежеминутная настройка скорости приёма/передачи
//*****************************************************************************
        _U16        id;         //идентификатор
//*****************************************************************************
        _U32        clk;        // время последней записи во FRAM
        _U16        off;        // текущий адрес во FRAM
        _S16        val;        // текущее значение
        _U8         crc;        // контрольная сумма
}
eepromData;
```

Каждая пара байт структуры данных соответствует регистру протокола Modbus. Подробнее соответствие содержимого структуры данных и регистров протокола Modbus будет описано ниже.

3.2 Регистры настройки

Таблица 10.

Номер регистра	Номер байта	Структура	Описание
0	0	algoritm	настройка
	1	object	адрес ИСПОЛНИТЕЛЯ
1	2	id	идентификатор метекомплекса
	3		
2	4	clk	время последней записи во FRAM в секундах с 1 января 1970 года по Гринвичу
	5		
3	6		
	7		
4	8	off	текущий адрес FRAM памяти для записи данных. 14400 регистров (60*24*10*2 – 10 суток)
	9		
5	10	val	Текщее измеренное значение
	11		
…		…	…
16384	32768	FM31256	Начиная с этого адреса доступны 12 регистров часов реального времени
	32769		

3.3 Регистры результатов измерений

Результаты измерений записываются во FRAM память устройства FM31256, начиная с 8 регистра (всего 14400 регистров). Этой памяти хватит для записи измерений за последние 10 суток. Данные размещаются в кольцевом буфере, поэтому для выборки данных используется специальная функция 66. Запрос по структуре как в функции 03

Сообщение запроса специфицирует сколько минут (регистров) назад от текущего момента надо взять в качестве начальной точки и количество минут (регистров) прочитать из памяти. В ИСПОЛНИТЕЛЕ начальный регистр вычисляется.

ОТВЕТ:

Данные регистров в ответе передаются как два байта на регистр. Байты регистров передаются старшим байтом вперёд. Количество регистров передаваемых за одно обращение определяется возможностями ИСПОЛНИТЕЛЯ.

Таблица 11.

Номер байта	Номер байта в числе	Условное обозначение
0	-	Адрес
1	-	Функция
2	-	Состояние
3	[1]	Данные
4	[0]	
…	[1]	…
…	[0]	
…	[1]	Данные
…	[0]	
…	[1]	Контрольная сумма
…	[0]	