

Программа обслуживания УрТ

Описание программы

Листов 13

Аннотация

Документ "Описание программы" предназначен для использования в качестве руководства по применению программы "АСК".

В настоящем документе содержится описание программы и сведения, необходимые для эксплуатации программы:

- приводятся общие сведения (назначение, основные функции, входные и выходные данные и т.д.);
- описываются параметры для настройки программы.

1 ОБЩИЕ СВЕДЕНИЯ

Программа "АСК" управляет процессом сбора данных от уровнемеров тензометрических (УрТ) и осуществляет представление на экране результатов.

Связь с УрТ осуществляется по двухпроводной физической линии.
Программа написана на языке Borland C++. Компилятор "C++ Builder Standard" версия 5.03.



2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПРОГРАММЫ

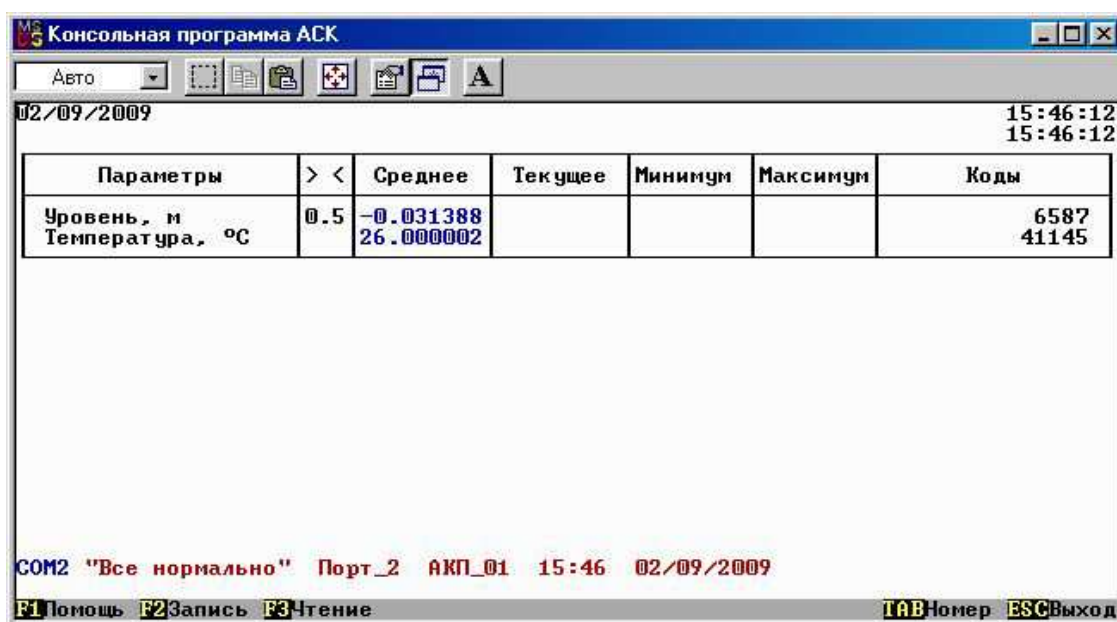
Программа "АСК" предназначена для управления процессом сбора данных от УрТ по RS-485 или по USB через конвертор.

3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1 Общая схема работы программы

Интерактивная программа "АСК" работает под управлением оператора, который с помощью горячих клавиш и различных подсказок сам выбирает нужные ему действия.

Отказ от выбранного вида работы всегда осуществляется нажатием клавиши Esc.
В программе реализовано консольное окно, в котором размещается таблица с измеренной информацией.



Для управления в программе используются следующие клавиши:

- F1 - помощь. Вывод на экран файла помощи;
- F2 - загрузка файла с настройками в УрТ. Имя файла формируется из номера/идентификатора датчика и расширения ".xmx". Файл должен находиться в директории "UPLOAD". Номер датчика вводится с клавиатуры;
- F3 - чтение настроек из УрТ и запись их в файл в директорию "DOWNLOAD" - аналоговые. Имя файла формируется из номера/идентификатора датчика и расширения ".xmx".;
- TAB - чтение номера/идентификатора УрТ ;
- DEL – переключение на калибровку токового выхода;
- ESC - выход.

3.2 Оперативный просмотр информации

Таблица на экране содержит измерительную информацию и имеет следующую структуру:

- Параметры - наименование измеряемого параметра;
- > < - максимально возможное отклонение текущего измерения от его среднего значения;
- Среднее - скользящее среднее за 10 минут;
- Текущее - среднее измеренное значение (обычно за 5 сек);
- Минимум – минимальное среди всех измеренных значений, из которых вычисляется среднее значение;
- Максимум – максимальное среди всех измеренных значений, из которых вычисляется среднее значение;
- Коды АЦП – измеренная информация до преобразования в физическую величину.

4 ВХОДНЫЕ ДАННЫЕ

4.1 Описание файловой среды функционирования программы

- ❑ ACK.INI. Используется для настройки программы на конкретные условия применения. В этом файле описаны встроенные измерительные платы, коммуникационные порты ввода/вывода, периоды записи результатов в базу данных;
- ❑ ACK.PAR. Используется для полного описания измеряемых параметров.
- ❑ ACK.PSE. Используется для настройки временных задержек и других вспомогательных значений в программе.

4.2 Инициализация УРТ

При запуске программа ACK считывает файл инициализации УРТ ACK.INI, в котором находятся значения параметров, определяющих порядок работы.

Для настройки УРТ используются следующие параметры:

- Номер - номер объекта. Объектом может быть встроенная измерительная плата, измерительный контроллер, подключенный по каналу связи, дополнительный УРТ и т.д. Номер объекта уникален и определяет набор измеряемых параметров или протокол обмена. Все строки файла с одним и тем же номером относятся к одному объекту, и вся получаемая информация собирается в одном буфере.
- Период - интервал, через который происходит передача данных в ЦДП;
- Порт - состоит из двух частей:
 - 1) тип взаимодействия с объектом:
 - ✓ "п", "N" - порты используются для подключения по линии с протоколом обмена MODBUS-RTU описанным ниже;
 - 2) коммуникационный порт 1 - Com1;
- Протокол - состоит из 4-х частей:
 - 1) скорость передачи данных (19200);
 - 2) длина посылки (8);
 - 3) количество стоповых бит (1);
 - 4) контроль бита четности (0-нет контроля).
- Дополнение – вспомогательный параметр, который используется в зависимости от выбранного типа взаимодействия.

4.3 Инициализация измеряемых параметров и поправок

При запуске программы считывается файл ACK.PAR, содержащий следующие параметры настройки:

- Наименование - наименование параметра или прочерк.
- Единицы измерения - единицы измерения параметра.
- Условное имя - сокращенное имя или формула для представления в телеграмме или прочерк.
- Код параметра - латинский символ, которым кодируется параметр внутри программы и в базе данных. Если условное имя - прочерк, то код используется вместе с номером для представления параметра в телеграмме.
- Номер параметра - порядковый номер параметра с одинаковым кодом.
- Тип – как измеряется параметр. "А" – аналоговый параметр.
- ИСО – интервал скользящего осреднения в секундах.

- Формат вывода - формат представления измеренного значения в телеграмме или в таблице в форме, принятой в языке программирования C ($\%n.df$ - где n - общее число символов вместе с десятичной точкой, d - число цифр после десятичной точки).
- Левая граница - левая граница измеряемого диапазона, выход за которую говорит о неправильной работе аппаратуры.
- Правая граница - правая граница измеряемого диапазона, выход за которую так же говорит о неправильной работе аппаратуры.
- Погрешность – возможные отклонения мгновенных значений от средней величины. Если описывается модуль скорости ветра, то это значение определяет смещение левой границы в сторону правой границы для вычисления диапазона измерения модуля скорости ветра, в котором можно измерять и направление ветра, т.е. если сложить левую границу и погрешность получится левая граница для отбраковки направления ветра.
- Коэффициенты полинома – начальное смещение аргумента и коэффициенты аппроксимирующего полинома до 9 степени через пробел ($(/x_0/, c_0, c_1, c_2$ и т.д. в формуле $y = c_0 + c_1*(x-x_0) + c_2*(x-x_0)^2 + \dots$). По умолчанию $x_0=0$ и вводится в случае необходимости после слэша “/” перед коэффициентом c_0 . Если степень полинома 0, т.е. указан один коэффициент c_0 , результат преобразования будет - правая граница, если измеренное значение больше или равно c_0 плюс погрешность - левая граница, если измеренное значение меньше или равно c_0 минус погрешность и среднее арифметическое левой и правой границы в противном случае.

4.4 Инициализация временных задержек

При запуске программы считывается файл ACK.PSE, содержащий следующие параметры настройки:

- идентификационный номер УРТ.
- разрешение расчета среднеквадратического отклонения;
- пауза после получения сигнала АТС "занято" до следующего набора номера;
- пауза после сбоя в линии связи перед следующим набором номера;
- время ожидания телеграммы после установления связи с УРТ через телефонный HAYES-модем после сигнала CONNECT;
- ожидание установления связи с УРТ после набора номера - время ожидания ответной телеграммы;
- ожидание сигнала CTS от модема;
- ожидание радиосвязи - ожидание ответной телеграммы через УС-100;
- время снятия сигнала DTR для отключения от телефонной линии - "положить трубку";
- ожидание сообщения "OK" от HAYES-модема;
- время для идентификации заикливания очереди;
- время, отводимое программе на обработку события;
- таймаут для сброса максимума в тиках. После прихода очередного запроса программа выдерживает заданный интервал и начинает определение нового максимума;
- интервал для блокирования очереди, ответчиком при приходе RING от HAYES-модема;
- общее время звукового сопровождения при выходе за границы диапазона или по тревоге в тиках;
- длительность звука/паузы при воспроизведении звукового сигнала в тиках;
- частота звука при выходе за границы диапазона в герцах (0 - нет звука);

- частота звука при объявлении тревоги в герцах (0 - нет звука);
- интервал, в секундах, через который повторяется аварийное сообщение;
- интервал, в секундах, через который гаснет экран;
- сдвиг запроса по времени в секундах;
- режим резервирования первичной информации в поддиректории SAFE;
- количество попыток установить связь с ЦДП;
- временной интервал в тиках, в течение которого усредняются данные при измерении частоты через цепи коммуникационного порта;
- ошибка в процентах (не используется);
- время калибровки в секундах. Если время калибровки равно 0, в программе !INFO.EXE вместо калибровочных коэффициентов в таблице будет выводиться измеряемое напряжение;
- время на снятие контроля вскрытия в секундах;
- интервал скользящего осреднения в секундах;
- процент хороших значений;
- запретить заполнять файл протокола;
- временной интервал сторожевого таймера;
- установка тестового режима для всех измеряемых параметров;
- интервал в микросекундах для удаления искаженных импульсов.

5 ПРОТОКОЛ ОБМЕНА

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 ИСПОЛНИТЕЛЬ 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.

5.1 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.

5.2 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 ИСПОЛНИТЕЛЯ. 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.

5.3 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	

5.1.1 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.

5.1.2 Поле функции

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

Старший бит этого поля устанавливается в единицу ИСПОЛНИТЕЛЕМ в случае, если он хочет просигнализировать 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.

5.1.3 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 плавающей запятой, то в каждом из 2-х регистров порядок следования байт тоже старшим байтом вперёд (1→0→3→2).

5.1.4 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 получиться нулевым. Получатель сообщения должен рассчитать CRC-код и сравнить его с полученным кодом. Вся арифметика выполняется по модулю 2 (без переноса).

5.4 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 FLASH память.

6 ФУНКЦИИ ПРОТОКОЛА

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.

6.1 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 4.

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

6.2 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 5.

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 5.

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

6.3 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 5.

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	ЕВ01	ЕВ
13	[0]			01

ОТВЕТ:

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

Таблица 5.

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 СВ и если всё было записано правильно, от ИСПОЛНИТЕЛЯ придёт ответ: 01 03 06 01 19 04 05 02 04 2С F4.

1 Общие сведения	2
2 Функциональное назначение программы.....	2
3 Описание логической структуры	2
3.1 Общая схема работы программы	2
3.2 Оперативный просмотр информации.....	3
4 Входные данные	4
4.1 Описание файловой среды функционирования программы	4
4.2 Инициализация УРТ	4
4.3 Инициализация измеряемых параметров и поправок.....	4
4.4 Инициализация временных задержек	5
5 Протокол обмена.....	6
5.1 Режимы передачи	7
5.2 Обнаружение ошибок	7
5.3 Кадровая синхронизация	8
5.1.1 Поле адреса	8
5.1.2 Поле функции	8
5.1.3 Поле данных	8
5.1.4 Поле контрольной суммы.....	9
5.4 Исключительные ситуации	9
6 Функции протокола	10
6.1 Функция 03 (Чтение регистров/Read Holding Registers)	10
6.2 Функция 06 (Запись одного регистра/Preset Single Register)	10
6.3 Функция 16 (Запись в регистры/Preset Multiple Regs).....	12