



**Описание протокола обмена
электромагнитных расходомеров-счетчиков
«Питерфлоу»
с системой верхнего уровня**

Редакция 1.05



ООО «ТЕРМОТРОНИК»
193318, Россия, Санкт-Петербург, ул. Ворошилова, д.2
Телефон, факс: +7 (812) 326-10-50
Сайт ООО «ТЕРМОТРОНИК»: www.termotronic.ru
Служба технической поддержки: support@termotronic.ru
тел. 8-800-333-10-34

Содержание

1	Введение	4
1.1	Общие сведения	4
1.2	Физический интерфейс и адаптеры	4
1.3	Уровни протокола обмена	4
1.3.1	Логический уровень	4
1.3.2	Коммуникационный уровень	5
	ADU «Modbus RTU»	5
	ADU «Modbus ASCII»	6
	ADU «Modbus TCP»	6
2	Реализованные функции протокола	8
2.1	Функция Modbus 03 (0x03) (Read Holding Registers) и 04 (0x04) (Read Input Registers)	8
2.2	Функция Modbus 06 (0x06) (Write Single Register)	8
2.3	Функция Modbus 16 (0x10) (Write Multiple Registers)	9
2.4	Функция Modbus 72 (0x48) «Расширенная запись и чтение регистров с нумерацией» (Extended Preset And Read Multiple Regs With Numbers)	9
3	Организация доступа к данным	12
3.1	Используемые типы данных и условные обозначения	12
3.2	Регистровый доступ и характеристики параметров	12
3.3	Порядок хранения и передачи байт данных	12
3.4	Чтение/запись текущих и настроечных параметров	13
3.5	Организация архивов и считывание архивных данных	13
3.5.1	Организация журнала событий	14
3.5.2	Организация архивов измерений для приборов, оснащенных RTC	15
4	Карта переменных	18
4.1	Профиль прибора	18
4.2	Авторизация	18
4.3	Характеристики сенсора	19
4.4	Текущие измерения	20
4.5	Импульсные выходы	20
4.6	Диагностика	20
	Приложение 1. Функция расчета контрольной суммы Cyclical Redundancy Checking (CRC16)	21
	Приложение 2. Функция расчета контрольной суммы LRC	22
	Приложение 3. Функции преобразования в ASCII и обратно	23
	Приложение 4. Коды ошибок, возвращаемые прибором	24
	Приложение 5. Функции преобразования в BCD и обратно	25
	Приложение 6. Функция расчета контрольной суммы Crc32	26
	Приложение 7. Система диагностики	27
	Приложение 8. Флаги состояния аппаратуры	28
	Приложение 9. Значения типов выходов	29

История редактирования

- 06.02.2018 создана редакция 1.00;
- 29.11.2018 изменен раздел «Используемые типы данных и условные обозначения»
- 29.11.2018 изменен раздел «Физический интерфейс и адаптеры»;
- 29.11.2018 изменен раздел «Характеристики сенсора»;
- 29.11.2018 изменен раздел «Приложение 7»;
- 29.11.2018 изменен раздел «Приложение 8»;
- 10.08.2020 создана редакция 1.05:
 - исправлены реквизиты организации;

1 Введение

1.1 Общие сведения

Данный документ описывает реализацию протокола обмена данными для электромагнитных расходомеров-счетчиков «Питерфлоу». Расходомер-счетчик «Питерфлоу» (далее Прибор) позволяет получать текущие и архивные параметры, а также предоставляет доступ к чтению и изменению настроечных параметров через коммуникационный интерфейс.

ПРИМЕЧАНИЕ! Данное описание протокола является актуальным для электромагнитных расходомеров-счетчиков «Питерфлоу» с версией встроенного программного обеспечения не менее **3.15**

1.2 Физический интерфейс и адаптеры

Для связи с системой верхнего уровня прибор оснащен последовательным интерфейсом. В зависимости от модели и исполнения могут использоваться интерфейсы различных стандартов.

ПРИМЕЧАНИЕ! Для уточнения характеристик последовательного интерфейса следует обращаться к руководству по эксплуатации на прибор.

В приборе реализован коммуникационный протокол «Modbus RTU» в режиме «ведомый».

Соединение прибора с системой верхнего уровня осуществляется через адаптеры связи, которые являются конвертерами интерфейсов физического уровня и могут выполнять функции преобразователей протоколов. Со стороны системы верхнего уровня они имеют различные физические интерфейсы и реализации коммуникационных протоколов передачи данных.

Существуют следующие типы адаптеров:

- **Ethernet↔Прибор.** Реализует протокол «Modbus TCP»;
- **RS-232↔Прибор.** Реализует протокол «Modbus ASCII»;
- **RS-485↔Прибор.** Реализует протокол «Modbus ASCII»;
- **USB↔Прибор.** Реализует протокол «Modbus RTU»;
- **GSM/GPRS↔Прибор.** Реализация протокола в зависимости от исполнения и режима.

ПРИМЕЧАНИЕ! В некоторых случаях возможно подключение без использования адаптеров связи. Для уточнения характеристик последовательного интерфейса следует обращаться к руководству по эксплуатации на прибор.

ПРИМЕЧАНИЕ! Для уточнения способа подключения, режимов использования и параметров работы адаптеров следует обратиться к документации на адаптеры

1.3 Уровни протокола обмена

Этот раздел содержит краткие сведения из стандарта «Modbus». Более подробную информацию можно получить из документов, размещенных на сайте www.modbus.org:

- Modbus Application Protocol;
- Modbus over serial line;
- Modbus Messaging on TCP/IP Implementation Guide.

1.3.1 Логический уровень

Логический уровень протокола отвечает за способ доступа к данным. Протокол «Modbus» определяет понятие PDU (Protocol Data Unit), независимое от используемого коммуникационного протокола. PDU содержит 2 поля: код функции (длина 1 байт) и данные (длина не более 252 байт).



Рисунок 1: Состав PDU

Подробное описание логического уровня приведено в разделе [Реализованные функции протокола](#).

1.3.2 Коммуникационный уровень

Коммуникационный уровень протокола отвечает за доставку передаваемой информации между двумя совместимыми «Modbus» устройствами. В различных коммуникационных протоколах PDU дополняется полями сетевой адрес, контрольная сумма, заголовок и т.д., образуя при этом, понятие ADU (Application Data Unit). Дополнительные поля требуются для адресации, идентификации и контроля целостности данных.

Независимо от коммуникационного протокола прибор работает только в режиме «ведомый». Это означает, что прибор может выполнить посылку только в ответ на запрос системы верхнего уровня. В случае отсутствия ответа от прибора система верхнего уровня должна выполнить повтор запроса.

ПРИМЕЧАНИЕ! Время ответа прибора не превышает 500 миллисекунд

Для адресации запросов конкретному прибору используется поле «Сетевой адрес», которое может принимать значения в диапазоне 0-247. Прибор реагирует только на запросы, в которых указан или широковещательный адрес 0 или сетевой адрес, совпадающий с сетевым адресом прибора (определяется установками прибора).

ПРИМЕЧАНИЕ! В отклонение от стандарта Modbus, запрос с широковещательным сетевым адресом 0 трактуется прибором как запрос, на который прибор должен дать ответ независимо от собственного сетевого адреса

ADU «Modbus RTU»

В случае использования коммуникационного протокола «Modbus RTU» PDU дополняется полями сетевой адрес и контрольная сумма.

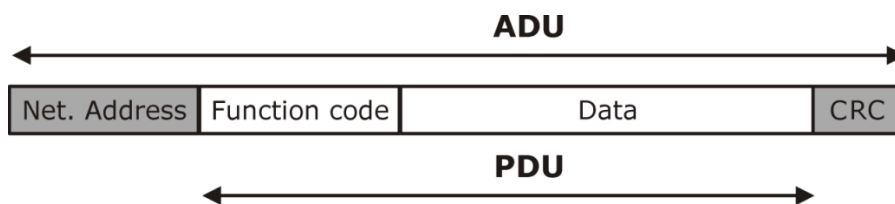


Рисунок 2: ADU для «Modbus RTU»

Сетевой адрес служит для адресации прибора в сети. Контрольная сумма служит для проверки целостности данных. Передающее устройство вычисляет контрольную сумму над всеми полями посылки, и затем результат вычисления добавляет в конец посылки. Принимающее устройство, получив всю посылку, вычисляет контрольную сумму кадра для всех байтов сообщения, исключая байты контрольной суммы. В случае, если принятая и вычисленная контрольные суммы равны, принимается решение о достоверности принятого кадра. В противном случае кадр считается недостоверным. Если прибор получает недостоверный кадр, он его игнорирует и не посылает каких-либо ответных сообщений. Это означает, что система верхнего уровня не получит ответа в течение ожидаемого времени и должна сделать повтор запроса. Если же факт получения недостоверной посылки обнаружен системой верхнего уровня, то она может выполнить повтор запроса.

При передаче используется двоичная система кодирования, начало и конец посылки определяются по длительности «интервала тишины» - времени, в течение которого не происходило передачи. Т.е. кадр сообщения «заклучен» между двумя интервалами тишины. Байты сообщения должны передаваться непрерывным потоком. Длительность интервала тишины по стандарту «Modbus RTU» определяется как 3,5 длительности передачи байта на заданной скорости.

ПРИМЕЧАНИЕ! В отклонение от стандарта Modbus интервал тишины составляет 5 миллисекунд

В качестве функции расчета контрольной суммы в режиме RTU используется Cyclical Redundancy Checking (CRC16). Пример функции расчета CRC16 приведен в [Приложении 1](#), а описание генерации контрольной суммы может быть найдено в документации на сайте www.modbus.org.

ADU «Modbus ASCII»

В случае использования коммуникационного протокола «Modbus ASCII» PDU дополняется полями сетевой адрес и контрольная сумма.

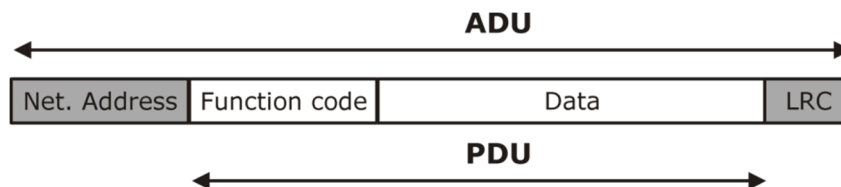


Рисунок 3: ADU для «Modbus ASCII»

Сетевой адрес служит для адресации прибора в сети. Контрольная сумма служит для проверки целостности данных. Передающее устройство вычисляет контрольную сумму над всеми полями посылки, и затем результат вычисления добавляет в конец посылки. Принимающее устройство, получив всю посылку, вычисляет контрольную сумму кадра для всех байтов сообщения, исключая байты контрольной суммы. В случае если принятая и вычисленная контрольные суммы равны, принимается решение о достоверности принятого кадра. В противном случае кадр считается недостоверным. Если прибор получает недостоверный кадр, он его игнорирует и не посылает каких-либо ответных сообщений. Это означает, что система верхнего уровня не получит ответа в течение ожидаемого времени и должна сделать повтор запроса. Если же факт получения недостоверной посылки обнаружен системой верхнего уровня, то она может выполнить повтор запроса.

При передаче исходные двоичные данные кодируются. Начало и конец сообщения помечены специальными маркерами. Началом сообщения всегда является символ двоеточия «:» (0x3A в шестнадцатеричном представлении). Концом сообщения всегда является пара символов «возврат каретки» (CR) и «перевод строки» (LF) (0x0D и 0x0A соответственно в шестнадцатеричном представлении). Каждый байт двоичного исходного сообщения передается в виде пары символов. Например, значение 27 (0x1B в шестнадцатеричном представлении) будет представлено как пара символов '1' (0x31 - символьное представление старших 4-х битов) и 'B' (0x42 - символьное представление младших 4-х битов). Допустимые символы для передачи - это шестнадцатеричные символы 0-9, A-F. В качестве функции расчета контрольной суммы используется Longitudinal Redundancy Checking (LRC). Пример функции расчета LRC приведен в [Приложении 2](#), а описание генерации контрольной суммы может быть найдено в документации на сайте www.modbus.org. Примеры функций перекодировки из двоичного представления в ASCII и из ASCII в двоичное представление приведены в [Приложении 3](#). Над двоичным содержимым буфера передачи сначала выполняется расчет контрольной суммы. Затем двоичные данные вместе с полем контрольной суммы подвергаются преобразованию в ASCII и затем результат дополняется символами начала и конца кадра.

ADU «Modbus TCP»

В случае использования коммуникационного протокола «Modbus TCP» PDU дополняется заголовком MBAP (MODBUS Application Protocol), служащим для идентификации.

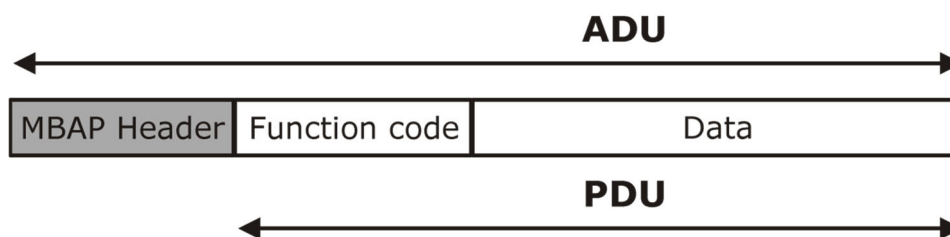


Рисунок 4: ADU для «Modbus TCP»

MBAP Header содержит следующие поля:

- Идентификатор запроса (длина 2 байта). Заполняется системой верхнего уровня. Копируется прибором из запроса при выполнении ответа;
- Идентификатор протокола (длина 2 байта). Должен быть равен 0;
- Длина сообщения в байтах (длина 2 байта). Должна быть равна длине PDU + 1;
- Сетевой адрес прибора (длина 1 байт). Заполняется системой верхнего уровня. При выполнении ответа прибором заполняется в соответствии с его настройкой.

Двухбайтные поля заголовка передаются в порядке сначала старший, затем младший байт.

2 Реализованные функции протокола

2.1 Функция Modbus 03 (0x03) (Read Holding Registers) и 04 (0x04) (Read Input Registers)

Функции предназначены для чтения двоичного содержимого шестнадцатиразрядных регистров прибора. Отличие функции 0x04 от 0x03 в том, что она применяется для чтения параметров, недоступных для записи. В общем виде структура запроса и ответа имеет следующий вид:

PDU запроса	PDU ответа без ошибок	PDU ответа при ошибке															
<table border="1"><thead><tr><th>Поле</th></tr></thead><tbody><tr><td>Функция 0x03/0x04</td></tr><tr><td>Начальный адрес (старший байт)</td></tr><tr><td>Начальный адрес (младший байт)</td></tr><tr><td>Количество регистров (старший байт)</td></tr><tr><td>Количество регистров (младший байт)</td></tr></tbody></table> <p>Поле Data PDU содержит поля:</p> <ul style="list-style-type: none">• «Начальный адрес», указывающий с какого регистра начинать чтение;• «Количество регистров», указывающее, сколько регистров следует прочитать.	Поле	Функция 0x03/0x04	Начальный адрес (старший байт)	Начальный адрес (младший байт)	Количество регистров (старший байт)	Количество регистров (младший байт)	<table border="1"><thead><tr><th>Поле</th></tr></thead><tbody><tr><td>Функция 0x03/0x04</td></tr><tr><td>Количество байт данных в ответе</td></tr><tr><td>1-ый регистр (старший байт)</td></tr><tr><td>1-ый регистр (младший байт)</td></tr><tr><td>Байты регистров 2,3...N</td></tr></tbody></table> <p>В случае успешного выполнения в ответе присутствует содержимое запрошенных регистров. Поле «Количество байт данных в ответе» будет равно количеству запрошенных регистров, умноженному на два. Прочитанное содержимое регистров начинается с байта, следующего за полем «Количество байт данных в ответе».</p>	Поле	Функция 0x03/0x04	Количество байт данных в ответе	1-ый регистр (старший байт)	1-ый регистр (младший байт)	Байты регистров 2,3...N	<table border="1"><thead><tr><th>Поле</th></tr></thead><tbody><tr><td>Функция 0x83/0x84 (уст. старший бит)</td></tr><tr><td>Код ошибки</td></tr></tbody></table> <p>Для информирования ведущего о том, что операция не выполнена или выполнена с ошибкой, прибор устанавливает старший бит поля «Функция» в ответе. Байт, следующий за полем «Функция», будет содержать код ошибки (значения кодов ошибок приведены в Приложении 4).</p>	Поле	Функция 0x83/0x84 (уст. старший бит)	Код ошибки
Поле																	
Функция 0x03/0x04																	
Начальный адрес (старший байт)																	
Начальный адрес (младший байт)																	
Количество регистров (старший байт)																	
Количество регистров (младший байт)																	
Поле																	
Функция 0x03/0x04																	
Количество байт данных в ответе																	
1-ый регистр (старший байт)																	
1-ый регистр (младший байт)																	
Байты регистров 2,3...N																	
Поле																	
Функция 0x83/0x84 (уст. старший бит)																	
Код ошибки																	

2.2 Функция Modbus 06 (0x06) (Write Single Register)

Функция предназначена для записи двоичного содержимого одного шестнадцатиразрядного регистра прибора.

PDU запроса	PDU ответа без ошибок	PDU ответа при ошибке															
<table border="1"><thead><tr><th>Поле</th></tr></thead><tbody><tr><td>Функция 0x06</td></tr><tr><td>Адрес (старший байт)</td></tr><tr><td>Адрес (младший байт)</td></tr><tr><td>Значение (старший байт)</td></tr><tr><td>Значение (младший байт)</td></tr></tbody></table> <p>Поле Data PDU запроса содержит поля:</p> <ul style="list-style-type: none">• «Адрес», указывающий в какой регистр выполняется запись;• Значение записываемого регистра.	Поле	Функция 0x06	Адрес (старший байт)	Адрес (младший байт)	Значение (старший байт)	Значение (младший байт)	<table border="1"><thead><tr><th>Поле</th></tr></thead><tbody><tr><td>Функция 0x06</td></tr><tr><td>Адрес (старший байт)</td></tr><tr><td>Адрес (младший байт)</td></tr><tr><td>Значение (старший байт)</td></tr><tr><td>Значение (младший байт)</td></tr></tbody></table> <p>В случае успешного выполнения PDU ответа содержит копию первых пяти байт PDU запроса.</p>	Поле	Функция 0x06	Адрес (старший байт)	Адрес (младший байт)	Значение (старший байт)	Значение (младший байт)	<table border="1"><thead><tr><th>Поле</th></tr></thead><tbody><tr><td>Функция 0x86 (уст. старший бит)</td></tr><tr><td>Код ошибки</td></tr></tbody></table> <p>Для информирования ведущего о том, что операция не выполнена или выполнена с ошибкой, прибор устанавливает старший бит поля «Функция» в ответе. Байт, следующий за полем «Функция», будет содержать код ошибки (значения кодов ошибок приведены в Приложении 4).</p>	Поле	Функция 0x86 (уст. старший бит)	Код ошибки
Поле																	
Функция 0x06																	
Адрес (старший байт)																	
Адрес (младший байт)																	
Значение (старший байт)																	
Значение (младший байт)																	
Поле																	
Функция 0x06																	
Адрес (старший байт)																	
Адрес (младший байт)																	
Значение (старший байт)																	
Значение (младший байт)																	
Поле																	
Функция 0x86 (уст. старший бит)																	
Код ошибки																	

2.3 Функция Modbus 16 (0x10) (Write Multiple Registers)

Функция предназначена для записи двоичного содержимого шестнадцатиразрядных регистров прибора.

PDU запроса	PDU ответа без ошибок	PDU ответа при ошибке																			
<table border="1"><thead><tr><th>Поле</th></tr></thead><tbody><tr><td>Функция 0x10</td></tr><tr><td>Нач-ый адрес (старший байт)</td></tr><tr><td>Нач-ый адрес (младший байт)</td></tr><tr><td>Кол-во рег-ов (старший байт)</td></tr><tr><td>Кол-во рег-ов (младший байт)</td></tr><tr><td>Кол-во байт для записи</td></tr><tr><td>1-ый регистр (старший байт)</td></tr><tr><td>1-ый регистр (младший байт)</td></tr><tr><td>Байты рег-ов 2,3...N</td></tr></tbody></table>	Поле	Функция 0x10	Нач-ый адрес (старший байт)	Нач-ый адрес (младший байт)	Кол-во рег-ов (старший байт)	Кол-во рег-ов (младший байт)	Кол-во байт для записи	1-ый регистр (старший байт)	1-ый регистр (младший байт)	Байты рег-ов 2,3...N	<table border="1"><thead><tr><th>Поле</th></tr></thead><tbody><tr><td>Функция 0x10</td></tr><tr><td>Нач-ый адрес (старший байт)</td></tr><tr><td>Нач-ый адрес (младший байт)</td></tr><tr><td>Кол-во рег-ов (старший байт)</td></tr><tr><td>Кол-во рег-ов (младший байт)</td></tr></tbody></table> <p>В случае успешного выполнения PDU ответа содержит копию первых пяти байт PDU запроса.</p>	Поле	Функция 0x10	Нач-ый адрес (старший байт)	Нач-ый адрес (младший байт)	Кол-во рег-ов (старший байт)	Кол-во рег-ов (младший байт)	<table border="1"><thead><tr><th>Пол</th></tr></thead><tbody><tr><td>Функция 0x90 (уст. старший бит)</td></tr><tr><td>Код ошибки</td></tr></tbody></table> <p>Для информирования ведущего о том, что операция не выполнена или выполнена с ошибкой, прибор устанавливает старший бит поля «Функция» в ответе. Байт, следующий за полем «Функция», будет содержать код ошибки (значения кодов ошибок приведены в Приложении 4).</p>	Пол	Функция 0x90 (уст. старший бит)	Код ошибки
Поле																					
Функция 0x10																					
Нач-ый адрес (старший байт)																					
Нач-ый адрес (младший байт)																					
Кол-во рег-ов (старший байт)																					
Кол-во рег-ов (младший байт)																					
Кол-во байт для записи																					
1-ый регистр (старший байт)																					
1-ый регистр (младший байт)																					
Байты рег-ов 2,3...N																					
Поле																					
Функция 0x10																					
Нач-ый адрес (старший байт)																					
Нач-ый адрес (младший байт)																					
Кол-во рег-ов (старший байт)																					
Кол-во рег-ов (младший байт)																					
Пол																					
Функция 0x90 (уст. старший бит)																					
Код ошибки																					

Поле Data PDU запроса содержит поля:

- «Начальный адрес», указывающий с какого регистра начинать запись;
- «Количество регистров», указывающее, сколько регистров следует записать;
- «Количество байт для записи»;
- Значения записываемых регистров.

2.4 Функция Modbus 72 (0x48) «Расширенная запись и чтение регистров с нумерацией» (Extended Preset And Read Multiple Regs With Numbers)

Данная функция выполняет последовательную запись и чтение двоичного содержимого шестнадцатиразрядных регистров прибора. Объединяет в себе возможности функций 0x03 и 0x10, позволяя выполнить запись и чтение значений регистров за один запрос. Также есть возможность контролировать порядковый номер запросов и ответов.

ПРИМЕЧАНИЕ! Данная функция является нестандартной функцией Modbus, предназначена для увеличения эффективности и надежности обмена данными. Отличается от стандартной функции Modbus 23 (0x17) Read/Write Multiple registers: а) возможностью работы с буфером данных большего размера (**408** байт); б) возможностью контроля порядкового номера запроса в серии

Часто при получении данных из прибора требуется делать парные запросы: запись регистров, затем чтение регистров. Данная функция была реализована в целях экономии количества запросов при выполнении записи и чтения регистров, связанных по смыслу. Например, при чтении архивов прибора сначала требуется установка типа архива и даты для поиска в архиве, а затем непосредственно чтение данных.

Использование данной функции наиболее оправданно при работе по каналам связи, способным вносить временные задержки в передаче информации (GSM, GPRS модемы и т.п.). В поле запроса передаются адреса и значения записываемых регистров, а также адреса и количество регистров для чтения. При этом прибором сначала выполняется операция записи значений регистров, а затем чтение. Результат выполнения данной функции аналогичен выполнению двух запросов: сначала с функцией 0x10, а затем с функцией 0x03. Однако, если в процессе выполнения операции записи регистров возникла ошибка, то прибор возвращает код ошибки не приступая к чтению регистров.

Функция позволяет выполнить также запись регистров без чтения, или чтение регистров без предварительной записи. В первом случае количество регистров для чтения устанавливается в нуль, а во втором случае в нуль устанавливается количество записываемых регистров и байт.

Запрос содержит поле порядкового номера итерации, которое возвращается прибором в ответе. При отправке каждого следующего запроса программным обеспечением верхнего уровня рекомендуется изменять значение порядкового номера, а при получении ответа контролировать совпадение значений полей порядкового номера в запросе и ответе.

ПРИМЕЧАНИЕ! Контроль порядкового номера запроса/ответа очень полезен тогда, когда может быть нарушена естественная последовательность запросов и ответов. Например, при использовании канала передачи GPRS паузы между отправкой запроса и получением ответа могут иметь очень существенную величину. Не дождавшись ответа на первый запрос (Запрос А), программное обеспечение верхнего уровня может послать повторный запрос (Запрос А). В этот момент после длительной задержки будет получен ответ на первую попытку Запроса А и программное обеспечение пошлет следующий, другой по смыслу запрос (Запрос В). В ответ на это будет получен ответ на вторую попытку первого запроса (Запрос А). Если не контролировать порядковый номер запросов и ответов, то полученный ответ (ответ на Запрос А) может быть интерпретирован как ответ на второй запрос (Запрос В), хотя, по сути, он им не является. Такие нарушения последовательностей могут приводить к ошибкам.

PDU запроса

Поле
Функция 0x48
Нач-ый адрес для чтения (ст.байт)
Нач-ый адрес для чтения (мл.байт)
Кол-во рег-ов для чтения (ст.байт)
Кол-во рег-ов для чтения (мл.байт)
Нач-ый адрес для записи (ст.байт)
Нач-ый адрес для записи (мл.байт)
Кол-во рег-ов для записи (ст.байт)
Кол-во рег-ов для записи (мл.байт)
Кол-во байт для записи (ст.байт)
Кол-во байт для записи (мл.байт)
Номер запроса (ст.байт)
Номер запроса (мл.байт)
1-ый регистр для записи (ст.байт)
1-ый регистр для записи (мл.байт)
Байты рег-ов 2,3...N

Поле Data PDU запроса содержит поля:

- «Начальный адрес для чтения», указывающий с какого регистра начинать чтение;
- «Количество регистров для чтения», указывающее, сколько регистров следует прочитать;
- «Начальный адрес для записи», указывающий с какого регистра начинать запись;
- «Количество регистров для записи», указывающее, сколько регистров следует записать;
- «Количество байт для записи», указывающее, сколько регистров следует записать;
- Номер запроса в серии для идентификации ответа;
- значения записываемых регистров.

PDU ответа без ошибок

Пол
Функция 0x48
Количество байт данных в ответе (ст.байт)
Количество байт данных в ответе (мл.байт)
Номер запроса (ст.байт)
Номер запроса (мл.байт)
1-ый регистр (ст.байт)
1-ый регистр (мл.байт)
Байты рег-ов 2,3...N

В случае успешного выполнения в ответе присутствует содержимое запрошенных регистров. Поле «Количество байт данных в ответе» будет равно количеству запрошенных на чтение регистров, умноженному на два. Прочитанное содержимое регистров начинается с байта, следующего за младшим байтом поля «Количество байт данных в ответе» и заканчивается байтом, предшествующим полю «Контрольная сумма». Поле «Номер запроса» в ответе должен совпадать с порядковым номером в запросе.

PDU ответа при ошибке

Поле
Функция 0xC8 (уст. старший бит)
Код ошибки чтения
Код ошибки записи
Номер запроса (с .байт)
Номер запроса (мл.байт)

Для информирования ведущего о том, что операция не выполнена или выполнена с ошибкой, прибор устанавливает старший бит поля «Функция» в ответе. Байты, следующие за полем «Функция», будут содержать коды ошибки чтения и записи соответственно (значения кодов ошибок приведены в [Приложении 4](#)). Поле «Номер запроса» в ответе должен совпадать с порядковым номером в запросе.

ПРИМЕЧАНИЕ! Если не требуется выполнять чтение, то поле «Количество регистров для чтения» устанавливаются в 0. Если не требуется запись, то поля «Количество регистров для записи» и «Количество байт для записи» устанавливаются в 0 и поле значения записываемых регистров отсутствует.

Ниже, для прибора с сетевым адресом 27, приведен формат PDU для записи 2-х регистров, начиная с 8550-го и чтение 2-х регистров начиная с 28-го:

№ байта в запросе	Наименование поля	Пример значений (hex)
1	Функция	0x48
2	Начальный адрес чтения (старший байт)	0x00
3	Начальный адрес чтения (младший байт)	0x1C
4	Количество регистров для чтения (старший байт)	0x00
5	Количество регистров для чтения (младший байт)	0x02
6	Начальный адрес записи (старший байт)	0x21
7	Начальный адрес записи (младший байт)	0x66
8	Количество регистров для записи (старший байт)	0x00
9	Количество регистров для записи (младший байт)	0x02
10	Количество байт для записи (старший байт)	0x00
11	Количество байт для записи (младший байт)	0x04
12	Порядковый номер запроса (старший байт)	0x00
13	Порядковый номер запроса (младший байт)	0x01
14	1-ый регистр для записи (старший байт)	0x00
15	1-ый регистр для записи (младший байт)	0x00
16	2-ый регистр для записи (старший байт)	0x00
17	2-ый регистр для записи (младший байт)	0x00

В случае успешного выполнения PDU ответа прибора будет иметь следующую структуру:

№ байта в запросе	Наименование поля	Пример значений (hex)
1	Функция	0x48
2	Количество байт данных в ответе (старший байт)	0x00
3	Количество байт данных в ответе (младший байт)	0x04
4	Порядковый номер запроса (старший байт)	0x00
5	Порядковый номер запроса (младший байт)	0x01
6	1-ый регистр (старший байт)	0x00
7	1-ый регистр (младший байт)	0x00
8	2-ый регистр (старший байт)	0x00
9	2-ый регистр (младший байт)	0x00

В случае неуспешного выполнения PDU ответа прибора будет иметь следующую структуру:

№ байта в запросе	Наименование поля	Пример значений (hex)
1	Функция (<i>установлен старший бит</i>)	0xC8
2	Код ошибки чтения	Code
3	Код ошибки записи	Code
4	Порядковый номер запроса (старший байт)	0x00
5	Порядковый номер запроса (младший байт)	0x01

3 Организация доступа к данным

3.1 Используемые типы данных и условные обозначения

- unsigned char – беззнаковое целое число (8 бит);
- signed char – знаковое целое число (8 бит);
- unsigned short – беззнаковое целое число (16 бит);
- signed short – знаковое целое число (16 бит);
- unsigned long – беззнаковое целое число (32 бита);
- signed long – знаковое целое число (32 бита);
- unsigned __int64 – беззнаковое целое число (64 бита);
- signed __int64 – знаковое целое число (64 бита);
- float – вещественное число одинарной точности (32 бита) с плавающей запятой, соответствующее стандарту «IEEE 754»;
- double – вещественное число двойной точности (64 бита) с плавающей запятой, соответствующее стандарту «IEEE 754»;
- bit field – битовое поле. Применяется когда внутри одного или нескольких байт разными битами или группами битов кодируется различная по смыслу информация. Если в пояснениях записано «Бит №0», то имеется ввиду самый младший бит байта; «Бит №1» - бит, следующий по старшинству за самым младшим; ..., «Бит №8» - самый старший бит, «Бит №9» - самый младший бит следующего байта и т.д. Является полным аналогом битовых полей языка программирования СИ;
- time_t - дата/время в формате количества секунд, прошедших с 01.01.1970 00:00:00 (unix time, длина 32 бита);
- date_time - дата/время в формате массива из 6 байт в порядке [год-2000, месяц [1÷12], день [1÷31], час [0÷23], минута [0÷59], секунда [0÷59]];
- [N] – массив из N элементов. Например, запись «unsigned char[4]» обозначает массив из 4-х элементов типа unsigned char;
- R/O – доступ только на чтение;
- R/W – доступ на чтение и запись (безусловный);
- W/O(XXX) – доступ только на запись, где XXX требуемый уровень доступа на запись;
- PASS - доступ на чтение, доступ на запись при предъявлении пароля;
- SERV - доступ на чтение, доступ на запись в сервис-центре;
- RTC - Real Time Clock, часы реального времени;

3.2 Регистровый доступ и характеристики параметров

Доступ к текущим и настроечным параметрам прибора реализован через функции чтения и записи регистров – переменных, имеющих тип шестнадцатиразрядное беззнаковое целое. При организации регистрового доступа делается допущение, что все многообразные структуры данных располагаются в памяти, элементарной ячейкой которой является один шестнадцатиразрядный регистр типа «беззнаковое целое». Физически данные могут находиться в совершенно разных участках памяти прибора и даже в разных типах памяти (оперативная, энергонезависимая и т.д.), но для системы верхнего уровня данные «выглядят» как единое адресное пространство. В этом случае все доступные данные можно представить как массив шестнадцатиразрядных регистров, каждый из которых характеризуется номером в массиве (далее адресом). Каждый параметр прибора может занимать часть регистра, весь регистр целиком или несколько регистров. Таким образом, параметр характеризуется собственным типом и расположением внутри массива регистров.

3.3 Порядок хранения и передачи байт данных

Для чтения и записи регистров в стандарте «Modbus» предусмотрены специальные функции, которые оперируют содержимым шестнадцатиразрядных регистров. Эти функции предполагают, что прибор хранит данные только типа шестнадцатиразрядное беззнаковое целое и ничего не «знают» о тех типах данных, которыми действительно представлены параметры прибора. Таким образом, получается, что в приборе данные хранятся в некоем исходном формате, а передаются

по сети в виде набора шестнадцатиразрядных регистров. При передаче данных, чей размер в исходном формате превышает 16 бит (long, float, double и т.д.), используются несколько последовательных регистров. При этом младшие слова передаются в первую очередь, старшие - в последнюю.

ПРИМЕЧАНИЕ! Для преобразования к порядку байт, естественному для платформы PC, требуется для каждого прочитанного/записываемого регистра изменить порядок байт

Пример размещения данных для типа **long** (MSB-most significant byte, LSB-least significant byte):

B3	B2	B1	B0
MSB			LSB

Регистр	Регистр A0		Регистр A1	
Порядок передачи	первый			последний
Байт	B1	B0(LSB)	B3(MSB)	B2

Пример размещения данных для типа **float**:

B3	B2	B1	B0
SEEEEEEE	EMMMMMMM	MMMMMMMM	MMMMMMMM

Регистр	Регистр A0		Регистр A1	
Порядок передачи	первый			последний
Байт	B1	B0(LSB)	B3(MSB)	B2

Пример размещения данных для типа **double**:

B7	B6	B5-B1	B0
SEEEEEEE	EEEEMMMM	MMMMMMMM	MMMMMMMM

Регистр	Регистр A0		Регистр A1		Регистр A2		Регистр A3	
Порядок передачи	первый							последний
Байт	B1	B0(LSB)	B3	B2	B5	B4	B7(MSB)	B6

3.4 Чтение/запись текущих и настроечных параметров

Чтение/запись текущих и настроечных параметров производится при помощи функций чтения/записи регистров (функции чтения 0x03, 0x04, функций записи 0x06, 0x10, функции 0x48). Распределение переменных в адресном пространстве описано в разделе [Карта переменных](#).

ПРИМЕЧАНИЕ! Перед изменением любого параметра (кроме параметров структуры Авторизация) система верхнего уровня обязана выполнить запись параметра «Данные авторизации» для идентификации времени изменения параметра (см. [Авторизация](#))

3.5 Организация архивов и считывание архивных данных

Приборы в зависимости от типа могут содержать архивы следующих типов:

- Журнал событий;
- Архивы измерений (часовой, суточный, месячный, годовой) с поиском по времени для приборов, оснащённых часами реального времени (RTC).

Прибор предоставляет информацию об архивах каждого типа (глубина архива, степень заполнения, длина архивной записи и т.д.). Для размещения информации об архивах в регистровом пространстве Modbus выделены специальные дескрипторы архивов.

Для доступа к архивам в регистровом пространстве Modbus выделены окна архивов. В окнах архивов может быть размещено до 32 последовательных архивных записей с заданным типом архива, датой архива, номером записи и т.д. Записи отсортированы в порядке возрастания

хронологии, т.е. в первой записи окна хранится наиболее старая запись, в последней записи окна - наиболее новая.

Для позиционирования архива относительно окна архива необходимо сделать предварительную запись информации о требуемых архивных данных (тип архива, дата архива, номер архивной записи и т.д.). Для записи информации об архиве в регистровом пространстве Modbus выделены дескрипторы окон архивов. После установки дескриптора окна первая запись окна архива будет содержать запись с заданными атрибутами. Последующие записи окна архива будут содержать последующие записи архива по мере возрастания хронологии.

ПРИМЕЧАНИЕ! Операции чтения и записи регистров архива выполняются с помощью функций регистрового доступа Modbus (0x03, 0x04, 0x10, 0x48)

При считывании архива рекомендуется следующая последовательность:

- 1) прочитать дескриптор архива (определить параметры имеющегося архива);
- 2) произвести запись в дескриптор окна архива (установить позицию архива относительно окна архива);
- 3) произвести одно или несколько последовательных чтений из окна архива (получить одну или несколько архивных записей);
- 4) повторить п. 2 и 3 при необходимости

ПРИМЕЧАНИЕ! Реализованный механизм позволяет минимизировать операции записи регистров и в ряде случаев свести получение архива к последовательному считыванию содержимого окна архива. При этом с помощью одного запроса Modbus можно прочитать сразу несколько архивных записей

Таким образом, для считывания любого архива имеются следующие регистры пространства Modbus:

Пространство регистров Modbus

Адрес Modbus	Дескриптор архива
0xXXXX	Содержит глубину архива, степень заполнения, длину архивной записи и т.д.

Адрес Modbus	Дескриптор окна архива
0xYYYY	Устанавливает атрибуты первой записи для чтения (тип, дата, номер и т.д.)

Адрес Modbus	Окно архива	
0xZZZZ + (0)	Архивная запись N (с атрибутами из дескриптора окна)	Самая старая зап.
0xZZZZ + (x*1)*	Архивная запись N + 1
0xZZZZ + (x*2)*	Архивная запись N + 2
.....
0xZZZZ + (x*31)*	Архивная запись N + 31	Самая новая зап.

* - x - размер одной записи окна (в количестве регистров)

ПРИМЕЧАНИЕ! В случае отсутствия записи в архиве соответствующая запись окна архива заполняется нулями

Конкретные адреса, смещения, размер и тип полей описаны далее в соответствующих разделах.

3.5.1 Организация журнала событий

Журнал событий представляет собой нестираемый архив, записи в который заносятся при возникновении различных событий (сброс архива измерений, изменение настроечных параметров и т.д.). Более подробные сведения о работе журнала могут быть получены из документации на прибор. Доступ к записям журнала осуществляется по индексу (номеру) записи. Это означает, что для чтения записей журнала необходимо предварительно записать индекс первой желаемой записи в дескриптор окна журнала.

Список адресов регистров:

Наименование	Адрес Modbus	Доступ
Дескриптор журнала	10040	R/O

Дескриптор окна журнала	11000	RW
Окно журнала (запись 0)	12000 + (0 * 32)	R/O
Окно журнала (запись 1)	12000 + (1 * 32)	R/O
.....
Окно журнала (запись 31)	12000 + (31 * 32)	R/O

Дескриптор журнала представлен структурой данных:

Наименование	Тип	Доступ	Примечание
Версия структуры	unsigned short	R/O	Не используется
Общее количество записей	unsigned long	R/O	Глубина журнала
Резерв	unsigned short	R/O	Не используется
Сколько записей занято	unsigned long	R/O	Заполнение архива
Резерв	unsigned short	R/O	Не используется
Длина записи	unsigned short	R/O	В количестве байт

Дескриптор окна журнала представлен структурой данных:

Наименование	Тип	Доступ	Примечание
Индекс записи	unsigned long	RW	Значение в диапазоне (0-"глубина журнала-1"). Позиционирует начало окна архива
Резерв	unsigned short	RW	Не используется
Тип архива	unsigned short	RW	Должен быть равен 5 для журнала

Каждая запись окна журнала представлена структурой данных:

Наименование	Тип	Доступ	Примечание
Тип архива	unsigned char	R/O	Равен 5 для журнала
Версия структуры	unsigned char	R/O	Не используется
Идентификатор записи	unsigned long	R/O	Значение в диапазоне (1-"глубина журнала"). Каждая последующая запись имеет идентификатор на 1 больший, чем предыдущая запись
Тип события	unsigned char	R/O	0-нет события; 1-Стирание кольцевого архива. Нарастивается нестираемый счётчик стираний архива исходно равный 0, его значение сохраняется; 2-Изменение параметра; 3 - переполнение интегратора
Тип значения	unsigned char	R/O	0 – нет значения; 1 – unsigned char; 2 – signed char; 3 – unsigned short; 4 – signed short; 5 – unsigned long; 6 – signed long; 7 - float 8 – unsigned __int64; 9 – signed __int64; 10 – double; 17 – date_time.
Время наработки (минут)	unsigned long	R/O	Время наработки на момент создания записи
Время внешнее	time_t	R/O	Метка времени от внешнего компьютера
Дата/время RTC	date_time	R/O	Дата/время создания архивной записи (при наличии RTC)
Адрес	unsigned short	R/O	Modbus адрес измененного параметра
Старое значение	unsigned char[8]	R/O	Содержит значение типа, определенного в «Тип значения»
Новое значение	unsigned char[8]	R/O	Содержит значение типа, определенного в «Тип значения»
ID ключа	unsigned long	R/O	Идентификатор ключа доступа

3.5.2 Организация архивов измерений для приборов, оснащенных RTC

Прибор может содержать архивы измерений следующих типов:

- часовой архив;
- суточный архив;
- месячный архив;
- годовой архив.

Архивы организованы как кольцевые структуры, каждый архив имеет определенную глубину. При заполнении архива на полную глубину он «закольцовывается», что приводит к тому, что на место самых старых архивных записей записываются новые. Информация о ведении архива доступна для считывания. Более подробные сведения о работе архивов измерений могут быть получены из документации на прибор. Доступ к архивным записям осуществляется по типу архива и дате создания архивной записи. Это означает, что для чтения архивных записей необходимо предварительно записать тип архива и дату в дескриптор окна архива измерений.

Список адресов регистров:

Наименование	Адрес Modbus	Доступ
Дескриптор часового архива	10008	R/O
Дескриптор суточного архива	10016	R/O
Дескриптор месячного архива	10024	R/O
Дескриптор годового архива	10032	R/O
Дескриптор окна архива измерений	11010	RW
Окно архива измерений (запись 0)	14000 + (0 * 40)	R/O
Окно архива измерений (запись 1)	14000 + (1 * 40)	R/O
.....
Окно архива измерений (запись 31)	14000 + (31 * 40)	R/O

Дескриптор архива измерений представлен структурой данных:

Наименование	Тип	Доступ	Примечание
Версия структуры	unsigned short	R/O	Не используется
Дата начала архива	date_time	R/O	Дата/время самой старой записи архива
Дата окончания архива	date_time	R/O	Дата/время самой новой записи архива
Длина записи	unsigned short	R/O	В количестве байт

ПРИМЕЧАНИЕ! В случае отсутствия записей в архиве поля «Дата начала архива» и «Дата окончания архива» будут равны 0

Дескриптор окна архива измерений представлен структурой данных:

Наименование	Тип	Доступ	Примечание
Дата/время	date_time	RW	Значение в диапазоне «Дата начала архива»-«Дата окончания архива» или 0. Позиционирует начало окна архива
Тип архива	unsigned short	RW	1 – часовой архив; 2 – суточный архив; 3 – месячный архив; 4 – годового архив;

ПРИМЕЧАНИЕ! В случае записи в дескриптор окна архива всех нулей первая запись окна архива будет содержать последнюю имеющуюся архивную запись (самую новую)

Каждая запись окна архива измерений представлена структурой данных:

Наименование	Тип	Доступ	Примечание
Тип архива	unsigned char	R/O	1 – часовой архив; 2 – суточный архив; 3 – месячный архив; 4 – годовой архив;
Версия структуры	unsigned char	R/O	Не используется
Дата/время	date_time	R/O	Дата/время. Все поля 0, если данные отсутствуют
Время наработки (минут)	unsigned long	R/O	Время наработки на момент создания записи
Интеграл V+ (м³)	double	R/O	Интеграл объема в прямом направлении
Интеграл V- (м³)	double	R/O	Интеграл объема в обратном направлении
Флаги событий	unsigned long	R/O	Набор битовых флагов в соответствии с Приложением 7
Время отсут.счета (минут)	unsigned long	R/O	Время отсутствия счета, диагностика A24
Миним. расход (м³/ч)	float	R/O	Минимальный расход на интервале
Максим. расход (м³/ч)	float	R/O	Максимальный расход на интервале
Напряжение питания (В)	float	R/O	Напряжение питания процессора
Температура (°C)	float	R/O	Температура индуктора
Остат. ёмкость батареи (А*ч)	float	R/O	Остаточная ёмкость батареи
Сопротивление среды (кОм)	float	R/O	Сопротивление среды
Флаги состояния аппаратуры	unsigned long	R/O	Набор битовых флагов в соответствии с Приложением 8
Ток индуктора (мА)	float	R/O	Ток индуктора

Формат записей одинаков для всех типов архивов измерений.

4 Карта переменных

4.1 Профиль прибора

Название	Адрес	Тип	Доступ	Примечание
Тип прибора	0	unsigned short	R/O	Принимает значения: <ul style="list-style-type: none">• 0x1701 - Питерфлоу РС;• 0x1707 - Питерфлоу К;• 0x1708 - Питерфлоу СВ;• 0x1709 - Питерфлоу М
Версия программы	1	unsigned short	R/O	старший байт версия, младший-редакция. Индикация версии ПО производится в виде XX.YY, где XX и YY - старший и младший байт параметра с адресом 1
Порядок байт и слов	3	unsigned short	R/O	<i>Не используется</i>
CRC прошивки	4	unsigned short	R/O	Индицировать в шестнадцатеричном виде
Регистр состояния прибора	6	unsigned short	R/O	Содержит информацию о наличии на плате часов реального времени. Установленный младший бит говорит о наличии часов на плате
Калибр. коэффициент А	7	float	R/O	
Калибр. коэффициент В	9	float	R/O	
Версия программы (метрологически незначимая часть)	11	unsigned short	R/O	При значении не равно 0 индикация версии ПО производится в виде XX.YY.ZZ, где XX и YY - старший и младший байт параметра с адресом 1, ZZ - значение параметра с адресом 11
Производитель	50	unsigned char[40]	R/O	Текстовая строка "TERMOTRONIC"
Модель прибора	70	unsigned char[40]	R/O	Текстовая строка наименования модели прибора
Сетевой адрес	440	unsigned short	PASS	
Серийный номер	570	unsigned long	R/O	

4.2 Авторизация

Структура данных «Авторизация» предназначена для управления уровнем доступа к изменению параметров.

Название	Адрес	Тип	Доступ	Примечание
Текущий уровень доступа	200	unsigned short	R/W	Служит для определения текущего уровня доступа к изменению параметров. Значение представлено битовой маской: бит 0- PASS; бит 1-SERV. Чем выше номер бита, тем выше предоставляемый уровень доступа. Запись любого значения сбрасывает доступ в 0.
Данные авторизации	201	unsigned char[20]	R/W	Байты 0...15 - строковый пароль; байты 16...19 метка времени в формате time_t.
Пароль пользователя	220	unsigned char[8]	W/O(PASS)	Служит для установки пользовательского пароля. Параметр доступен только на запись при уровне доступа «пароль» (предъявление текущего пароля) или при более высоком уровне доступа

ПРИМЕЧАНИЕ! Перед изменением любого параметра (кроме параметров структуры Авторизация) система верхнего уровня **обязана** выполнить запись параметра «Данные авторизации», содержащего допустимые значения. Значение даты/времени, передаваемые в байтах 16...19 сохраняются в записях журнала событий

ПРИМЕЧАНИЕ! Получение уровня доступа SERV не рассматривается в этом документе.

При необходимости изменения параметра рекомендуется следующая последовательность:

- чтение текущего уровня доступа и сравнение с требуемым уровнем доступа изменяемого параметра;
- при недостаточном уровне доступа следует попытаться повысить уровень доступа:

- Для получения уровня доступа PASS требуется выполнить запись поля «Данные авторизации» так, что байты 0...15 содержат значение пароля (байты с номерами более длины пароля заполняются нулями), а байты 16...19 содержат локальное время системы верхнего уровня в формате time_t;
- при достаточном уровне доступа следует:
 - произвести запись поля «Данные авторизации» так, что байты 0...15 содержат все нули, а байты 16...19 содержат временную метку локального времени системы верхнего уровня в формате time_t;
 - произвести запись значения желаемого параметра

ПРИМЕЧАНИЕ! Поле «Данные авторизации» должно быть записано единым массивом (20 байт)

4.3 Характеристики сенсора

Название	Адрес	Тип	Доступ	Примечание
Калибр. коэффициент А	7	float	R/O	
Калибр. коэффициент В	8	float	R/O	
Количество знаков после запятой для индикации калибр. коэффициента А	40	unsigned short	R/O	
Количество знаков после запятой для индикации калибр. коэффициента В	41	unsigned short	R/O	
Количество знаков после запятой для индикации интегралов	42	unsigned short	R/O	
Напряжение питания электроники (В)	134	float	R/O	
Предел коррекции часов реального времени (сек.)	410	unsigned short	R/O	
Коррекция часов реального времени (сек.)	411	signed short	PASS	
Период измерения автономного прибора (сек.)	450	unsigned short	R/O	Только для Питерфлой СВ
Диаметр условный (мм)	540	unsigned short	R/O	
Класс расходомера	550	unsigned short	R/O	Если старший байт равен 0, то отображается значение младшего байта (символы 'A', 'B', 'C', ''). Если старший байт не равен 0, то отображается «Кxxx», где «xxx» - десятичное значение старшего байта.
Максимальный расход Q3 (м³/ч)	554	float	R/O	
Уровень отсечки (м³/ч)	580	float	SERV	Предельное значение 1/100 макс. расхода
Часы реального времени	590	unsigned char[6]	SERV	Сек., мин., час, день, месяц, год в формате BCD
Часы реального времени	595	date time	SERV	
Постоянная автофильтра	610	unsigned short	SERV	Постоянная автофильтра в циклах замера
Фильтрация с ограничением скорости (%)	612	unsigned short	PASS	По умолчанию 2%/отсчёт, 0 отключает фильтр
Емкость применяемой батареи (А*ч)	683	float	R/O	Только для приборов с автономным питанием
Остаточная емкость батареи (А*ч)	685	float	R/O	Только для приборов с автономным питанием
Порог предупреждения батареи Low Battery, мВ	700	unsigned short	PASS	Только для приборов с автономным питанием
Порог отказа батареи (мВ)	702	unsigned short	R/O	Только для приборов с автономным питанием
Порог предупреждения батареи Low Battery (% первоначальной емкости батареи)	704	unsigned short	PASS	Только для Питерфлой СВ

4.4 Текущие измерения

Название	Адрес	Тип	Доступ	Примечание
Часы реального времени	10500	date time	R/O	
Время наработки (мин.)	10503	unsigned long	R/O	
Интеграл V+ (м³)	10505	double	R/O	Интеграл объема в прямом направлении
Интеграл V- (м³)	10509	double	R/O	Интеграл объема в обратном направлении
Флаги событий	10513		R/O	Набор битовых флагов в соответствии с Приложением 7
Время наработки с ошибкой (мин.)	10515	unsigned long	R/O	
Расход (м³/ч)	10517	float	R/O	
Код АЦП	10519	float	R/O	
Напряжение питания (В)	10521	float	R/O	
Температура индуктора (°С)	10523	float	R/O	
Остаточная емкость батареи (А*ч)	10525	float	R/O	
Сопротивление среды (кОм)	10527	float		
Флаги состояния	10529	unsigned long		Набор битовых флагов в соответствии с Приложением 8
Ток индуктора (мА)	10531	float		

4.5 Импульсные выходы

Название	Адрес	Тип	Доступ	Примечание
Уставка компаратора F1 (м³/ч)	420	float	PASS	
Уставка компаратора F2 (м³/ч)	430	float	PASS	
Тип выхода F1	500	unsigned short	SERV	Значение в соответствии с Приложением 9
Тип выхода F2	510	unsigned short	SERV	Значение в соответствии с Приложением 9
Вес импульса (л./имп.)	520	float	SERV	

4.6 Диагностика

Название	Адрес	Тип	Доступ	Примечание
Событие «Нет расхода» (мин.)	690	unsigned short	PASS	Задаёт порог в минутах: 0-выкл; максимум - 100 минут
Событие «Разрыв» (%)	692	unsigned short	PASS	Порог расхода в % от максимума. 0-выкл. Период фиксированный 30 минут
Событие «Утечка»	694	unsigned short	PASS	Порог расхода в тысячных долях от максимума. 0-выкл., максим. - 10
Событие «Утечка» (сут.)	696	unsigned short	PASS	Порог времени в сутках. Максимум 7
Обнулять показания расхода при магните	720	unsigned short	PASS	0 - не обнулять, 1 - обнулять
Флаги событий (необработанные)	900	unsigned long	R/O	Набор битовых флагов в соответствии с Приложением 7
Флаги событий	902	unsigned long	R/O	Набор битовых флагов в соответствии с Приложением 7
Маска OFF	920	unsigned long	PASS	Запрещает события. Константа в программе, может быть частично перепрограммирована пользователем
События выхода F1	935	unsigned long	PASS	Битовая маска событий выхода F1 в соответствии с Приложением 7
События выхода F2	938	unsigned long	PASS	Битовая маска событий выхода F2 в соответствии с Приложением 7

Приложение 1. Функция расчета контрольной суммы Cyclic Redundancy Checking (CRC16)

Расчет контрольной суммы кадра RTU может выполняться по следующему алгоритму (текст на языке программирования СИ):

```
WORD Crc16(BYTE *Data, ULONG size) {
    union
    {
        {BYTE b[2]; unsigned short w;} Sum;
        char shift_cnt;
        BYTE *ptrByte;
        ULONG byte_cnt = size;
    };

    ptrByte=Data;
    Sum.w=0xffffU;
    for(; byte_cnt>0; byte_cnt--) {
        Sum.w=(unsigned short) ((Sum.w/256U)*256U+((Sum.w%256U)^(*ptrByte++)));
        for(shift_cnt=0; shift_cnt<8; shift_cnt++) {
            if((Sum.w&0x1)==1)
                Sum.w=(unsigned short) ((Sum.w>>1)^0xa001U);
            else
                Sum.w>>=1;
        }
        ptrByte++;
    }
    return Sum.w;
}
```

Пусть какое-либо сообщение, имеющего длину N, записано в массиве Data[N+2] типа unsigned char. Тогда для этого сообщения контрольную сумму следует формировать следующим образом:

```
WORD CheckSumm = Crc16(Data, N);
Data[N] = CheckSumm;
Data[N+1] = CheckSumm>>8;
```

Приложение 2. Функция расчета контрольной суммы LRC

Пример функции расчета контрольной суммы кадра на языке СИ:

```
unsigned char Lrc(unsigned char * pSrc, int length) {  
    unsigned char locLrc=0;  
    for(int i=0;i<length;i++)  
        locLrc += *(pSrc+i);  
    return locLrc = ~locLrc + 1;  
}
```

где:

- pSrc – указатель на буфер, содержащий сообщение;
- length – количество байт данных, для которых требуется произвести подсчет LRC.

Приложение 3. Функции преобразования в ASCII и обратно

Ниже приведены примеры на языке СИ функций преобразования из ASCII формата в двоичный и обратного преобразования из двоичного в ASCII.

```
const unsigned char CharToBin[23]={
0,1,2,3,4,5,6,7,8,9,0,0,0,0,0,0,0,0,10,11,12,13,14,15};
const char BinToChar[16]={
'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};

char TwoASCIIToBIN(char *cptr, unsigned char *bptr) {
char ca,i;
unsigned char cb;

cb=0;
for(i=0; i<2; i++) {
ca=*cptr++;
cb<<=4;
if((ca >= '0') && (ca <= '9') || (ca >= 'A') && (ca <= 'F'))
cb|=CharToBin[ca-0x30];
else
return(0);
}
*bptr=cb;
return(1);
}
```

где:

- cptr – указатель на буфер, содержащий символы ASCII;
- bptr - указатель на буфер, куда записываются двоичные байты.

```
void BINtoTwoASCII(unsigned char *bptr, char *cptr) {
unsigned char cb;

cb=*bptr;
*cptr=BinToChar[(cb>>4) & 0x0F];
cptr++;
*cptr=BinToChar[cb & 0x0F];
}
```

где:

- bptr – указатель на буфер, содержащий двоичные байты;
- cprt - указатель на буфер, куда записываются символы ASCII.

Приложение 4. Коды ошибок, возвращаемые прибором

- 0 - нет ошибок;
- 1 - недопустимая функция;
- 2 - недопустимый адрес в запросе;
- 3 - недопустимые значения в поле данных запроса;
- 4 – непоправимая ошибка возникла во время выполнения операции;
- 5 – подтверждение выполнения операции;
- 6 - запрос не может быть обработан сейчас, необходимо повторить запрос позднее;

Приложение 5. Функции преобразования в BCD и обратно

```
unsigned char BinByteToBcdByte(unsigned char Src) {
    unsigned char res;
    res = (unsigned char) ( ((Src/10)<<4) | (Src % 10) );
    return res;
}

unsigned char BcdByteToBinByte(unsigned char Src) {
    unsigned char res;
    res = (unsigned char) ( ((Src>>4) * 10) + (Src & 0x0f) );
    return res;
}
```

Приложение 6. Функция расчета контрольной суммы Crc32

```
#define CRC32_Pre()      0xFFFFFFFF
#define CRC32_Pnom()    0xEDB88320
#define CRC32_XOR()     0xFFFFFFFF
#define CRC32_Check(_C)      (_C == 0xDEBB20E3)

unsigned char CRC32(void * pBuf, unsigned long len) {
    unsigned long  j;
    unsigned long  _CRC;

    _CRC = CRC32_Pre();
    for (j=0; j < len; j++) {
        _CRC = CRC32_Calc_Byte(*((BYTE*) (pBuf)+j), _CRC);
    }

    if (CRC32_Check(_CRC)) {
        return 1;
    } else {
        return 0;
    }
}

unsigned long CRC32_Calc_Byte(unsigned char _D, unsigned long _C) {
    unsigned char  _i = 8;
    _C ^= (_D & 0xFF);
    do {
        if (_C & 1) { _C=(_C>>1)^CRC32_Pnom(); } else { _C>>=1; }
    } while (--_i);
    return (_C);
}
```

Приложение 7. Система диагностики

Список флагов событий:

Бит	Описание
0	Загрязнение электродов
1	Запись в защищенный журнал невозможна по причине заполнения. Изменение параметров невозможно.
2	Нет воды, останов счёта.
3	Неисправность аппаратуры
4	Магнит, может быть отключен пользователем. Также можно остановить счёт
5	Короткое замыкание электродов
6	Перепополнение АЦП
7	Расчетная ошибка. Программная ошибка
8	Заданы неверные значения калибровки
9	Установлено аппаратное разрешение записи, необходимое при калибровке.
10	Ошибка стабилизатора тока.
11	Перепополнение частотного выхода
12	Расход больше максимума- Q3
13	RESERVED
14	Утечка - вода протекала через счетчик, не останавливаясь ни на один целый час в течении заданного времени. Можно задать порог срабатывания в процентах от максимума, Диапазон 0-100%, 0 отключает, 1% по умолчанию
15	Разрыв - расход воды имеет большое значение непрерывно в течении получаса, Можно задать порог срабатывания в процентах от максимума. Диапазон 0-100%, 0 отключает, 10% по умолчанию
16	Останов протока воды — расход меньше отсечки непрерывно в течение заданного времени. Диапазон 0-256 минут, 0 отключает
17	Большой шум электродов
18	Только для батарейных приборов. Low battery –сообщает за два месяца (уставка в % емкости) до момента, когда следует заменить батарейку или если напряжение питания падает ниже 3,0 вольт.
19	Только для батарейных приборов. Bad battery возникает при падении напряжения питания ниже 2,7 вольт и останавливает работу
20	Обрыв индуктора
21	КЗ индуктора
22	Ошибка тока
23	RESERVED
24	Счёт не вёлся, расход обнуляется (во время фатальных ошибок, перезапуска, не было воды)
25	Изменился защищённый журнал
26	Только для батарейных приборов. Внешнее питание
27	Перезапуск
28	Часы RTC (при наличии) не идут
29	RESERVED
30	RESERVED
31	RESERVED

Приложение 8. Флаги состояния аппаратуры

Флаги состояния прибора представлены битовой маской:

- бит 0 - отсутствие батарейки;
- бит 1 - питание от 5В;
- бит 2 - питание от 12В;
- бит 3 - питание от батареи.

Значения остальных битов могут быть представлены в шестнадцатеричном виде.

Приложение 9. Значения типов выходов

Список значений типов выходов:

№	Назначение	Значение
0	оба направления потока, импульсы 0 (ключ замкнут)	96
1	оба направления потока, импульсы 1 (ключ разомкнут)	224
2	прямое направление потока, импульсы 0	32
3	прямое направление потока, импульсы 1	160
4	обратное направление потока, импульсы 0	64
5	обратное направление потока, импульсы 1	192
6	компаратор: выход=0 если расход больше уставки	1
7	компаратор: выход=1 если расход больше уставки	129
8	диагностика: выход=0 при событиях (по ИЛИ)	2
9	диагностика: выход=1 при событиях (по ИЛИ)	130
10	телеметрия (передача данных в цифровой форме)	3