



Описание протокола обмена тепловычислителя «ТВ7» с системой верхнего уровня

Редакция 7.04



ООО «THERMOTRONIK»
193318, Россия, Санкт-Петербург, ул. Ворошилова, д.2
Телефон, факс: +7 (812) 326-10-50
Сайт ООО «THERMOTRONIK»: 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	Физические интерфейсы и адаптеры	4
1.3	Уровни протокола обмена	5
1.3.1	Логический уровень протокола	5
1.3.2	Коммуникационный уровень протокола	5
1.3.2.1	ADU «Modbus RTU»	6
1.3.2.2	ADU «Modbus ASCII»	7
1.3.2.3	ADU «Modbus TCP»	7
1.3.2.4	ADU «PPP»	8
2	Реализованные функции протокола	9
2.1	Функции Modbus 03 (0x03) (Read Holding Registers) и 04 (0x04) (Read Input Registers)	9
2.2	Функция Modbus 16 (0x10) (Write Multiple Registers)	9
2.3	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	13
3.1	Используемые типы данных и условные обозначения	13
3.2	Регистровый доступ и характеристики параметров	14
3.3	Порядок хранения и передачи байт данных	14
3.4	Существующие типы значений и терминология	14
3.5	Считывание текущих данных	15
3.6	Организация архивов и считывание архивных данных	15
3.6.1	Считывание данных синхронных архивов	16
3.6.2	Считывание данных асинхронных архивов	16
3.7	Рекомендуемая последовательность считывания данных	16
4	Карта переменных	17
4.1	Информация об устройстве	18
4.2	Настройки	18
4.2.1	Информация о настройках	18
4.2.2	Настройки защищенные	18
4.2.2.1	Системные настройки	19
4.2.2.2	Настройки труб	20
4.2.2.3	Настройки тепловых вводов	21
4.2.2.4	Настройки дополнительного импульсного входа	23
4.2.2.5	Настройки дискретных входов	23
4.2.3	Настройки незащищенные	23
4.3	Архивы	23
4.3.1	Информация о датах начала/конца архивов	23
4.3.2	Информация об индексах архивов параметров	25
4.3.3	Тип читаемых данных	25
4.3.4	Архивная запись (часовой, суточный, месячный архивы)	27
4.3.5	Архивная запись (итоговый архив)	30
4.3.6	Информация об асинхронных архивах	31
4.3.7	Архивные записи асинхронных архивов	32
4.3.7.1	Интерпретация архива изменений базы данных	33
4.3.7.2	Интерпретация архива событий	35
4.3.7.3	Интерпретация диагностического архива	35
4.4	Текущие значения	36
4.4.1	Текущие итоговые значения	36
4.4.2	Текущие мгновенные значения	38
4.5	Сервисная команда	40
4.6	Структура технологического изменения параметров	41
4.7	Телеметрия	41
	Приложение 1. Функция расчета контрольной суммы Cyclical Redundancy Checking (CRC16)	44
	Приложение 2. Функция расчета контрольной суммы LRC	45
	Приложение 3. Функции преобразования в ASCII и обратно	46
	Приложение 4. Коды ошибок, возвращаемые прибором	47
	Приложение 5. Описание структуры файла tvb	48
	Приложение 6. Реализация обмена данными по интерфейсу USB	50

История редактирования

- 20.04.2011 создана редакция 1;
- 19.05.2011 создана редакция 2;
- 14.07.2011 создана редакция 3;
- 08.01.2012 создана редакция 4;
- 06.02.2012 создана редакция 5;
- 24.05.2012 создана редакция 6:
 - Дополнен пункт «Структура технологического изменения параметров»; Исправлено описание списка событий в текущей и архивной записях;
- 07.12.2012 создана редакция 6.03:
 - Исправлена ошибка в п. 6.12 «Текущие итоговые значения» в списке адресов даты/времени;
- 24.09.2014 создана редакция 6.07:
 - Добавлено приложение «Описание структуры файла tvb» ;
- 31.10.2014 создана редакция 6.08:
 - Добавлено описание параметра «Перевод часов» ;
 - 18.12.2015 описаны события «Изменение настроек» и «Калибровки разрешены» в архивных записях;
 - 14.01.2016 в разделе «Интерпретация диагностического архива» добавлено упоминание о бите 7 – ВВ;
 - 24.01.2017 в разделе «Информация о приборе» изменен список кодировки модели прибора;
 - 04.04.2017 в п.5 добавлено перечисление используемых единиц измерения;
 - 04.04.2017 в п.6.1 добавлена расшифровка номеров моделей приборов;
 - 25.10.2017 в п. «Настройки тепловых вводов» изменена размерность поля ФРТ;
- 07.08.2018 создана редакция 7.00:
- 15.10.2018 п. 3.1 дополнен информацией о множительных коэффициентах перевода систем единиц СИ и МКС;
- 30.10.2018 п.4 «Карта переменных» дополнен обозначением системы нумерации труб;
- 01.03.2019 Изменен комментарий о номере версии для поля Единицы измерения в разделе Незащищенные настройки;
- 21.04.2020 Создана редакция 7.03:
 - 21.04.2020 Добавлена расшифровка сокращения Ндв;
 - 21.04.2020 Добавлено описание параметра «Контр.Р» в пункте «Настройки тепловых вводов»;
 - 21.04.2020 Добавлен пункт «Настройки дискретных входов»;
- 10.08.2020 Создана редакция 7.04:
 - 10.08.2020 исправлены реквизиты организации;

1 Введение

1.1 Общие сведения

Данный документ описывает реализацию протокола обмена данными для тепловычислителя «ТВ7». Тепловычислитель «ТВ7» (далее Прибор) позволяет получать текущие и архивные параметры, а также предоставляет доступ к чтению и изменению настроечных параметров через коммуникационные интерфейсы.

ПРИМЕЧАНИЕ! Из-за разницы в функционале версии встроенного программного обеспечения ПВ \geq 2.20 имеют существенные отличия в распределении адресного пространства переменных в сравнении с ПВ $<$ 2.20. При работе с приборами, имеющими ПВ \geq 2.20 возможно чтение параметров, размещенных по адресам для ПВ $<$ 2.20. В этом случае, в целях сохранения максимальной совместимости, в процессе чтения производится преобразование структур данных ПВ \geq 2.20 к структурам ПВ $<$ 2.20, что может приводить к потерям значимой информации

1.2 Физические интерфейсы и адаптеры

Прибор оснащен интерфейсами:

- USB, класс HID device;
- COM1, последовательный интерфейс RS-232 (с гальванической развязкой);
- COM2, опциональный сменный интерфейс, реализуемый одним из встроенных адаптеров:
 - RS-485 (с гальванической развязкой);
 - Ethernet (с гальванической развязкой).

Настройка параметров передачи данных последовательных интерфейсов осуществляется через меню прибора в разделе «СЕРВИС ▶ КОММУНИКАЦИИ».

Интерфейс COM1 поддерживает:

- физические параметры передачи данных:
 - скорость 1200, 2400, 4800, 9600 бит/с.;
 - 8 бит данных;
 - 1 стоп-бит;
 - нет контроля четности.
- логические протоколы передачи данных:
 - Modbus RTU;
 - Modbus ASCII;
 - PPP.

ПРИМЕЧАНИЕ! При использовании интерфейса RS-232 внешнее устройство должно обеспечивать высокий уровень сигнала RTS для питания выходных оптронов адаптера

Интерфейс COM2 поддерживает:

- физические параметры передачи данных:
 - скорость 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит/с.;
 - 8 бит данных;
 - 1 стоп-бит;
 - нет контроля четности.
- логические протоколы передачи данных:
 - Modbus RTU;
 - Modbus ASCII;
 - Modbus TCP (только с адаптером Ethernet);
 - PPP.

ПРИМЕЧАНИЕ! При использовании интерфейса COM2 прибор должен быть запитан от внешнего сетевого блока питания. При работе от встроенной батареи интерфейс COM2 не работает

верхнего уровня. В случае отсутствия ответа от прибора система верхнего уровня должна выполнить повтор запроса.

ПРИМЕЧАНИЕ! Ожидаемое время получения ответа от ведомого и количество повторов запросов, после которого можно констатировать факт отсутствия качественной связи с прибором, зависит от скорости передачи и используемого физического канала связи. Так, время выполнения цикла «запрос ответ» при прямом соединении по RS-232/485 и при использовании проводного или GPRS модема будет кардинально отличаться. При прямом соединении по интерфейсам RS-232, RS-485 или USB время реакции прибора не превышает 500 мс

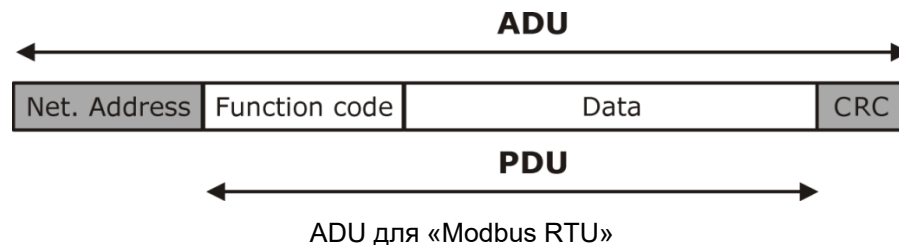
Для адресации запросов конкретному прибору используется поле «Сетевой адрес», которое может принимать значения в диапазоне 0-255. Прибор реагирует только на достоверные запросы, в которых указан или широковещательный адрес 0 или сетевой адрес, совпадающий с сетевым адресом прибора (определяется установками 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. Использование широковещательных адресов допускается только при подключении типа «точка-точка»

Возможно использование трех стандартных режимов передачи данных: RTU (Remote Terminal Unit), ASCII (American Standard Code for Information Interchange), TCP (Transmission Control Protocol) и одного нестандартного: PPP (Point to Point Protocol). Пользователь выбирает необходимый режим путем подбора оборудования и конфигурирования прибора наряду с такими параметрами как: скорость обмена, сетевой адрес и т.д. Выбор используемого режима обуславливается характеристиками физического канала связи, особенностями других устройств, подключенных на шину данных, требованиями к быстродействию и т.д. Все устройства, объединенные в сеть, должны использовать одинаковый режим передачи данных.

1.3.2.1 ADU «Modbus RTU»

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



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 может сделать повтор запроса. Если система верхнего уровня обнаруживает недостоверный ответ, 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 «заключен» 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:

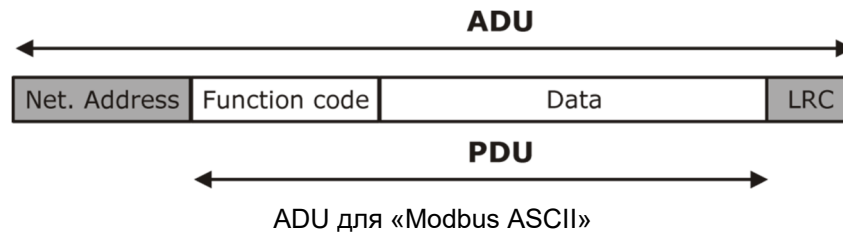
- 7,8 мс для скоростей 9600 бит/с и выше;
- 15,6 мс для скорости 4800 бит/с;
- 31,2 мс для скорости 2400 бит/с;
- 62,5 мс для скорости 1200 бит/с

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.

ПРИМЕЧАНИЕ! Режим RTU является быстродействующим и экономным с точки зрения пропускной способности канала связи, однако накладывает определенные ограничения на параметры передачи данных. Так, например, при связи через оборудование, способное вносить паузы в передаваемый пакет данных (например, модемы), связь может оказаться невозможной

1.3.2.2 ADU «Modbus ASCII»

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



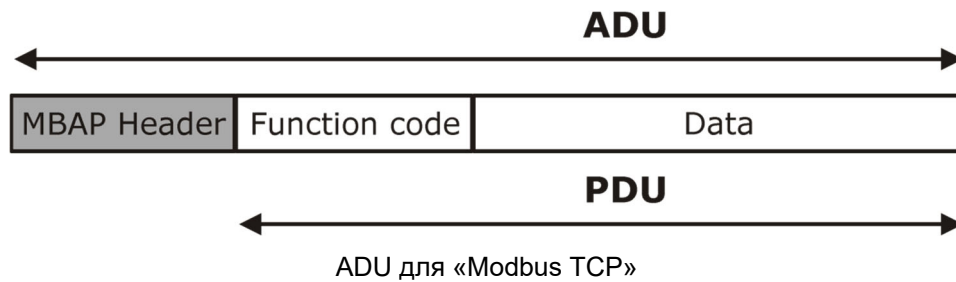
Сетевой адрес служит для адресации прибора в сети. Контрольная сумма служит для проверки целостности данных. Передающее устройство вычисляет контрольную сумму над всеми полями посылки, и затем результат вычисления добавляет в конец посылки. Принимающее устройство, получив всю посылку, вычисляет контрольную 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 может сделать повтор запроса. Если система верхнего уровня обнаруживает недостоверный ответ, 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.

ПРИМЕЧАНИЕ! Главным преимуществом режима ASCII является то, что пауза между передачей символов сообщения может достигать значительной величины без обнаружения ошибки. Это позволяет обмениваться данными с прибором по интерфейсам, в которых невозможно соблюсти временные соотношения в передаваемом потоке данных. Однако этот режим является менее экономным по сравнению с RTU, т.к. каждый байт данных требует передачи двух ASCII символов

1.3.2.3 ADU «Modbus TCP»

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



MBAP Header содержит следующие поля:

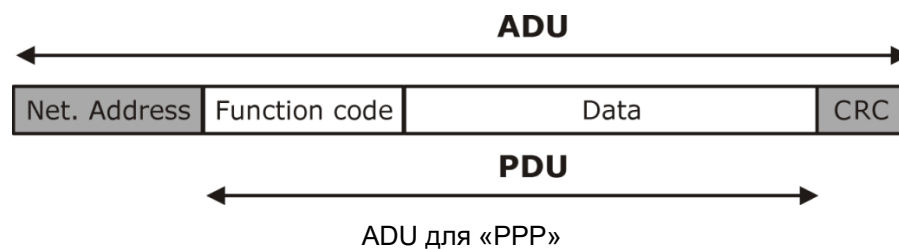
- Идентификатор запроса (длина 2 байта). Заполняется системой верхнего уровня. Копируется прибором из запроса при выполнении ответа;
- Идентификатор протокола (длина 2 байта). Должен быть равен 0;
- Длина сообщения в байтах (длина 2 байта). Должна быть равна длине PDU + 1;
- Сетевой адрес прибора (длина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

1.3.2.4 ADU «PPP»

Режим передачи данных PPP является нестандартным и позволяет реализовать обмен данными с более высокой помехозащищенностью в сравнении с использованием стандартных режимов.

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



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 может сделать повтор запроса. Если система верхнего уровня обнаруживает недостоверный ответ, то она может выполнить повтор запроса.

Используется двоичная система кодирования, начало и конец посылки помечены специальными маркерами. Началом сообщения всегда является символ с кодом 0x7E (в шестнадцатеричном представлении). Концом сообщения всегда является символ с кодом 0x7F (в шестнадцатеричном представлении). Если внутри сообщения встречаются символы начала сообщения, конца сообщения, символы с кодом 0x7D (в шестнадцатеричном представлении) или символы с кодом менее 0x20 (в шестнадцатеричном представлении), то такие символы заменяются парой символов: символ с кодом 0x7D и символ из сообщения с инвертированным 6-ым битом (над символом надо выполнить побитовую операцию исключения или со значением 0x20).

ПРИМЕЧАНИЕ! Режим PPP сочетает в себе быстрдействие режима RTU с устойчивостью режима ASCII к временным искажениям посылаемых сообщений. Использование CRC16 делает обмен данными более устойчивым в сравнении с режимом ASCII, где используется LRC.

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.

2 Реализованные функции протокола

2.1 Функции 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>	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>	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>	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																	
<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. 	<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>	<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>															

2.2 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-в 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-в 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>	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>	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-в 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																					
<p>Поле Data PDU запроса содержит поля:</p> <ul style="list-style-type: none"> • «Начальный адрес», указывающий, с какого регистра начинать запись; • «Количество регистров», указывающее, сколько регистров следует записать; • «Количество байт для записи»; • Значения записываемых регистров. 	<p>В случае успешного выполнения PDU ответа содержит копию первых пяти байт PDU запроса.</p>	<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>																			

2.3 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: 1) возможностью работы с буфером данных большего размера; 2) 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. Программному обеспечению верхнего уровня при отправке каждого следующего запроса 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 прочесть;
- «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
Кол-во байт данных в ответе (ст.байт)
Кол-во 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 последним байтом PDU. Поле «Номер запроса» в ответе должно совпадать с порядковым номером в запросе.

PDU ответа при ошибке

Поле
Функция 0xC8 (уст. старший бит)
Код ошибки чтения
Код ошибки записи
Номер запроса (ст.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 должно совпадать с порядковым номером в запросе.

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

Ниже 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-ой регистр для записи (старший байт)	0x00
17	2-ой 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-ой регистр (старший байт)	0x00
9	2-ой 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 (установлен старший бит)	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;
- [N] – массив из N элементов. Например, запись «unsigned char[4]» обозначает массив из 4-х элементов типа unsigned char;
- RO – доступ только на чтение (read only);
- R/W – доступ на чтение и запись (read/write);
- R/W* – доступ на чтение и запись (read/write). Запись разрешена только при особых условиях, которые будут специально оговариваться;
- (!) – размер для битового поля указан в количестве бит;
- АЧ – архив часовой;
- АС – архив суточный;
- АМ – архив месячных;
- АИ – архив итоговый;
- АИБД – архив изменений базы данных;
- ААС – архив административных событий;
- АД – архив диагностик;
- ТВ – тепловой ввод;
- ДП – дополнительный параметр;
- ПВ – версия встроенного программного обеспечения;
- АВ – версия аппаратного обеспечения;
- БД – база данных настроечных параметров;
- Nтр – номер трубы, изменяющийся в диапазоне 0÷6 для ПВ≥2.20 и 0÷5 для ПВ<2.20;
- Nтв – номер теплового ввода, изменяющийся в диапазоне 0÷2 для ПВ≥2.20 и 0÷1 для ПВ<2.20;
- Nдв – номер дискретного входа, изменяющийся в диапазоне 0÷3 для ПВ≥2.20, отсутствует для ПВ<2.20;
- Ng – номер канала измерения расхода, изменяющийся в диапазоне 0÷6;
- Нарх – номер архива, изменяющийся в диапазоне 0-2, где 0-АИБД, 1-ААС, 2-АД;
- НС – нештатная ситуация;
- ДВ – двухпозиционный вход;
- V – объем;
- P – давление;
- t – температура;
- Q – тепло;
- h – энтальпия;
- G_o – объемный расход;
- G_m – массовый расход;
- Ф – тепловой поток;
- P_x – давление холодной воды;
- t_x – температура холодной воды;
- h_x – энтальпия холодной воды;
- dt – разность температур;
- t_{нв} – температура наружного воздуха;

ПРИМЕЧАНИЕ! Значения настроечных, текущих, архивных параметров всегда передаются в системе единиц СИ, независимо от системы единиц отображения, установленной в приборе (объем (м³), масса (т), давление (МПа), энтальпия (ГДж/т), тепло (ГДж)). Для перевода значений в систему единиц МКС следует учитывать, что 1Гкал = 4.1868 ГДж, 1 МПа = 10.1972 кгс/см².

3.2 Регистровый доступ и характеристики параметров

Доступ к 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 Существующие типы значений и терминология

Параметры прибора можно разделить на следующие типы:

- Текущие;
- Архивные;
- Настройки.

Текущие данные дополнительно терминологически подразделяются на:

- Текущие мгновенные;
- Текущие итоговые.

Под Текущими мгновенными данными понимаются измеренные или вычисленные параметры, такие как температуры, давления, расходы теплоносителя, тепловой поток, разность температур и т.д.

Под Текущими итоговыми данными понимаются интегральные параметры теплотребления от момента сброса архива прибора до текущего момента. Т.е. это данные, представляющие «нарастающий итог» параметров теплотребления (объемы, массы, тепло, разность масс, времена нормальной работы и отсутствия счета и т.д.).

ПРИМЕЧАНИЕ! Дискретность обновления текущих параметров зависит от режима работы прибора

Архивные данные можно разделить на 2 группы по дисциплине архивирования:

- синхронные архивы. Это архивы часовой, суточный, месячный, итоговый. Формирование архивных записей происходит регулярно и с периодичностью, определяемой типом архива. Архивы часовой, суточный, месячный имеют интервалы архивирования соответственно час, сутки и месяц, и содержат данные о теплотреблении на этом интервале (температуры и давления - средние на интервале, объемы; массы, тепло - потребленные за час, сутки и месяц соответственно). Архив итоговый имеет интервал архивирования одни сутки и хранит только интегральные параметры теплотребления от момента сброса архива прибора до момента архивирования (объемы, массы, тепло за все время работы от сброса архива до момента архивирования). По сути, архивная запись итогового архива является заархивированным значением Текущих итогов на момент архивирования.

Время архивирования определяется настроечным параметром «Час отчета».
Дата архивирования определяется настроечным параметром «Дата отчета».

ПРИМЕЧАНИЕ! Например: часовая архивная запись с хронологической меткой «10 часов» хранит параметры теплотребления, зафиксированные на интервале времени с 10:00:00 до 11:00:00.

ПРИМЕЧАНИЕ! Например: если дата отчета задана 25, а час отчета 23, то формирование суточных и итоговых записей будет происходить в 00:00:00. Каждая суточная и итоговая запись будут иметь хронологическую метку ДД.ММ.ГГ 23ч. Формирование месячных записей будет происходить в 00:00:00 26-го числа каждого месяца и архивная запись будет иметь хронологическую метку 25.ММ.ГГ 23ч.

- асинхронные архивы. Это архив изменений базы данных, архив событий, диагностический архив. Формирование архивных записей происходит асинхронно по времени, только при наступлении определенных событий (изменения базы данных пользователем, выполнение различных сервисных операций, таких как сброса архива и т.д.).

3.5 Считывание текущих данных

Считывание текущих (мгновенных и итоговых) значений производится при помощи операции чтение регистров. Адрес стартового регистра для чтения определяется тем, какие именно, текущие мгновенные или текущие итоговые данные, читаются (см. «[Текущие мгновенные значения](#)» и «[Текущие итоговые значения](#)»).

3.6 Организация архивов и считывание архивных данных

Синхронные и асинхронные архивы организованы как кольцевые индексированные структуры. Архив каждого типа имеет определенную глубину. При заполнении архива на полную глубину архив «закольцовывается», что приводит к тому, что на место самых старых архивных записей заносятся новые значения. Информация о ведении архива доступна для считывания. Т.о., система верхнего

уровня может знать: насколько заполнен архив, закольцован ли он, максимальную глубину архива, размер архивной записи и т.д.

3.6.1 Считывание данных синхронных архивов

Доступ к чтению архивных записей производится через указание прибору даты/времени архивной записи и типа читаемого архива. Чтение архивной записи выглядит следующим образом:

- Операция записи регистров, «[Тип читаемых данных](#)» с указанием дата/времени архивной записи (с точностью до часа для часового архива, с точностью до суток для суточного и итогового архивов, с точностью до месяца для месячного архива) и типа архива, который надо прочитать;
- Операция чтения регистров. Адрес стартового регистра для чтения зависит от типа читаемого архива (см. «[Архивная запись \(часовой, суточный и месячный архивы\)](#)» и «[Архивная запись \(итоговый архив\)](#)»). Количество регистров соответствует длине архивной записи, или длине фрагмента архивной записи. За один прием можно прочитать содержимое только одной архивной записи.

3.6.2 Считывание данных асинхронных архивов

Из-за того, что в группе асинхронных архивов новые записи могут появляться в произвольный момент времени, доступ к чтению архивных записей реализован через указание индекса архивной записи, которую требуется прочитать. В этом случае, система верхнего уровня читает информацию о ведении архива (см. «[Информация об асинхронных архивах](#)»), определяет, насколько он заполнен, и читает архив запись за записью. Для этого она делает запрос на запись регистров, в котором указывает индекс той записи, начиная с которой требуется получить данные. Затем делает запрос на чтение регистров. За один запрос можно прочитать содержимое от 1 до 16 соседних записей асинхронного архива. Чтение 32-х архивных записей, начиная с самой первой, будет выглядеть таким образом:

- Операция записи регистров, «Тип читаемых данных» с указанием индекса записи 0 (см. «[Тип читаемых данных](#)»). В ответ прибор присылает подтверждение записи;
- Операция чтения регистров. Адрес стартового регистра для чтения зависит от типа архива (см. «[Архивные записи асинхронных архивов](#)»). Количество читаемых регистров соответствует 16-ти записям, исходя из размера архивной записи. В ответ прибор присылает содержимое архивных записей с индексами 0-15;
- Операция записи регистров, «Тип читаемых данных» с указанием индекса записи 16 (см. «[Тип читаемых данных](#)»). В ответ прибор присылает подтверждение записи;
- Операция чтения регистров. Адрес стартового регистра для чтения зависит от типа архива (см. «[Архивные записи асинхронных архивов](#)»). Количество читаемых регистров соответствует 16-ти записям, исходя из размера архивной записи. В ответ прибор присылает содержимое архивных записей с индексами 16-31.

3.7 Рекомендуемая последовательность считывания данных

В большинстве случаев задача считывания данных сводится к чтению текущих показаний прибора и получению архивов. Для формирования отчетных ведомостей или интерпретации данных может потребоваться информация о приборе и/или его настройках. Исходя из этого, можно рекомендовать следующую последовательность запросов:

- Чтение информации об устройстве. Выполняется контроль типа подключенного прибора, версии его аппаратного и программного обеспечения, серийного номера и т.д.;
- Чтение настроек прибора;
- Чтение информации о синхронных и асинхронных архивах, интервал дат архивов;
- Чтение текущих значений;
- Чтение архивов.

4 Карта переменных

ПРИМЕЧАНИЕ! Из-за разницы в функционале версии встроенного программного обеспечения ПВ≥2.20 имеют существенные отличия в распределении адресного пространства переменных в сравнении с ПВ<2.20. При работе с приборами, имеющими ПВ≥2.20 возможно чтение параметров, размещенных по адресам для ПВ<2.20. В этом случае, в целях сохранения максимальной совместимости, в процессе чтения производится преобразование структур данных ПВ≥2.20 к структурам ПВ<2.20, что может приводить к потерям значимой информации

ПРИМЕЧАНИЕ! Вычислитель хранит настройки двух баз данных (БД1 и БД2). В таблице параметров указываются адреса с учетом нахождения в БД1 или БД2. Запись вида XXXX[YYYY] означает, что параметр имеет адрес XXXX в БД1 и YYYY в БД2

ПРИМЕЧАНИЕ! В таблице адресов параметров запись вида «---» означает, что параметр отсутствует

ПРИМЕЧАНИЕ! В таблице адресов параметров запись вида Nтр означает номер трубы и изменяется в диапазоне 0+6 для ПВ≥2.20 и 0+5 для ПВ<2.20. Система обозначений труб в зависимости от модели приведена ниже

ПРИМЕЧАНИЕ! В таблице адресов параметров запись вида Nтв означает номер теплового ввода и изменяется в диапазоне 0+2 для ПВ≥2.20 и 0+1 для ПВ<2.20

ПРИМЕЧАНИЕ! В таблице адресов параметров запись вида Ng означает номер канала измерения расхода, изменяющийся в диапазоне 0+6

ПРИМЕЧАНИЕ! В таблице адресов параметров запись вида Nарх означает номер архива, изменяющийся в диапазоне 0-2, где 0-АИБД, 1-ААС, 2-АД

Для моделей 01, 03, 04, 04.1, 01М, 03М, 04М, 04.1М принята следующая система нумерации труб:

Тепловой ввод	Тепловой ввод 1			Тепловой ввод 2		
	Труба 1	Труба 2	Труба 3	Труба 1	Труба 2	Труба 3
Nтр	0	1	2	3	4	5

При работе с приборами, имеющими ПВ≥2.20, труба с Nтр=6 должна игнорироваться.

Для модели 05 принята следующая система нумерации труб:

Тепловой ввод	Тепловой ввод 1			Тепловой ввод 2		Тепловой ввод 3	
	Труба 1	Труба 2	Труба 3	Труба 1	Труба 2	Труба 1	Труба 2
Nтр	0	1	2	3	4	5	6

4.1 Информация об устройстве

Параметр	Адрес	Тип	Доступ	Примечание
Тип устройства	0	unsigned short	R/O	Принимает значения: • 0x1702 - тепловычислитель ТВ7
Программная версия	1	unsigned short	R/O	Старший байт содержит номер версии, младший – номер редакции
Аппаратная версия	2	unsigned short	R/O	Старший байт содержит номер версии, младший – номер редакции
Контрольная сумма ПО	3	unsigned short	R/O	
Модель	4	unsigned short	R/O	Младший байт принимает значения: • 0 - модель 01(01M); • 2 - модель 03(03M); • 3 - модель 04(04M); • 4 - модель 04.1(04.1M); • 5 - модель 05; Старший байт не используется.
Серийный номер	5	unsigned long	R/O	

ПРИМЕЧАНИЕ! При отображении номера модели следует добавлять символ «М», если аппаратная версия $\geq 0x0500$

4.2 Настройки

Настройки разделяются на 2 большие группы: защищенные кнопкой «Доступ» и настройки, доступные для изменения всегда.

4.2.1 Информация о настройках

Структура предназначена для операций с такой информацией, как контрольные суммы настроек, номер выбранной активной базы данных и т.д.

Параметр	Адрес	Тип	Доступ	Примечание
Контрольная сумма защищенных настроек	2669	unsigned short	R/W*	При чтении настроек по адресам ПВ<2.20
Активная БД	2670	unsigned short	R/O	Младший байт. Принимает значения: • 0–активна БД1; • 2–активна БД2; Старший байт не используется.
Контрольная сумма калибровок	2671	unsigned short	R/O	
Контрольная сумма защищенных настроек	2672	unsigned short	R/W*	При чтении настроек по адресам ПВ \geq 2.20
Контрольная сумма незащищенных настроек	2673	unsigned short	R/O	

4.2.2 Настройки защищенные

Доступ к изменению настроек открывается после однократного нажатия кнопки «Доступ» на вычислителе. При изменении настроек прибора система верхнего уровня должна выполнить следующие действия:

- подать сервисную команду «старт записи настроек» (см. «[Сервисная команда](#)»);
- рассчитать контрольную сумму записываемых настроек и записать ее в структуру «Информация о настройках» (см. «[Информация о настройках](#)»);
- записать содержимое настроек;
- подать сервисную команду «окончание записи настроек» (см. «[Сервисная команда](#)»);

В момент получения сервисной команды «окончание записи настроек» прибор считает контрольную сумму записанных настроек и сравнивает полученное значение со значением контрольной суммы, записанной системой верхнего уровня. При совпадении значений прибор сохраняет настройки в энергонезависимую память и начинает работать в соответствии с «новыми» настройками.

4.2.2.1 Системные настройки

Параметр	ПВ	Адрес	Тип	Доступ	Примечание
Час/День отчета	≥2.20	5000	unsigned short	R/W*	Час - младший байт, значения 0 - 23 День - старший байт, значения 1 - 31
	<2.20	105			
БД1 с: день/месяц	≥2.20	5001	unsigned short	R/W*	БД1/2 с: день/месяц/год/час: дата и время автоматического переключения активной базы данных. Поле «год» игнорируется. День - младший байт, значения 1-31 Месяц - старший байт, значения 1-12
	<2.20	106			
БД1 с: год/час	≥2.20	5002	unsigned short	R/W*	Год - младший байт, не исп-ся Час - старший байт, 0-23 См. поле «БД1 с: день/месяц»
	<2.20	107			
БД2 с: день/месяц	≥2.20	5003	unsigned short	R/W*	День - младший байт, значения 1-31 Месяц - старший байт, значения 1-12
	<2.20	108			
БД2 с: год/час	≥2.20	5004	unsigned short	R/W*	Год - младший байт, не исп-ся Час - старший байт, 0-23 См. поле «БД2 с: день/месяц»
	<2.20	109			
Использование БД2	≥2.20	5005	bit field	R/W*	Бит 0. Принимает значения: • 0 - не используется; • 1 - используется
	<2.20	110			
Способ переключ. БД	≥2.20	5005	bit field	R/W*	Биты 1-2. Принимает значения: • 0—вручную; • 1—автоматически по дате
	<2.20	110			
Доступ к переключ. БД с клавиат.	≥2.20	5005	bit field	R/W*	Биты 3-5. Принимает значения: • 0—запрет; • 1—с паролем; • 2—при нажатой кнопке «Доступ»
	<2.20	110			
Доступ к переключ. БД по сети	≥2.20	5005	bit field	R/W*	Биты 6-8. Принимает значения: • 0—запрет; • 1—с паролем
	<2.20	110			
Резерв	≥2.20	5005	bit field	R/W*	Биты 9-15
	<2.20	110			
Ед. измерения	≥2.20	---	bit field	R/W*	Бит 0. Принимает значения: • 0—СИ; • 1—МКС Только в ПВ<2.00
	<2.20	111			
Тип термодатчиков	≥2.20	5006	bit field	R/W*	Биты 1-4. Принимает значения: • 0—100П α=0.00391 (ω=1.391); • 1—500П α=0.00391 (ω=1.391); • 2—Pt100 α=0.00385 (ω=1.385); • 3—Pt500 α=0.00385 (ω=1.385)
	<2.20	111			
Резерв	≥2.20	5006	bit field	R/W*	Бит 5
	<2.20	111			
Перевод часов	≥2.20	5006	bit field	R/W*	Бит 6. Принимает значения: • 0-нет; • 1-да
	<2.20	111			
Инверсия дисплея	≥2.20	5006	bit field	R/W*	Бит 7. Принимает значения: • 0-нет; • 1-да
	<2.20	111			
Резерв	≥2.20	5006	bit field	R/W*	Биты 8-31
	<2.20	111			

Примечание:

При работе с приборами, имеющими ПВ≥2.20 возможно чтение параметров, размещенных по адресам для ПВ<2.20. В этом случае, в целях сохранения максимальной совместимости, в процессе чтения производится преобразование структур данных ПВ≥2.20 к структурам ПВ<2.20, что может приводить к потерям значимой информации

4.2.2.2 *Настройки труб*

Параметр	ПВ	Адрес	Тип	Доступ	Примечание
Контроль V	≥2.20	5008[5161]+(Nтр*18)+0	bit field	R/W*	Биты 0-2. Принимает значения: • 0–нет; • 1–Без подст.; • 2–С подст.; • 3–С подст. и контр. U; • 4–Счет отменен
	<2.20	113[233]+(Nтр*17)+0			
Контроль BC	≥2.20	5008[5161]+(Nтр*18)+0	bit field	R/W*	Биты 3-4. Принимает значения: • 0–нет; • 1–Сеть (общий); • 2-Индивид
	<2.20	113[233]+(Nтр*17)+0			
Резерв/Датчик P	≥2.20	---	bit field	R/W*	Бит 5. Принимает значения: • 0–нет; • 1–есть
	<2.20	113[233]+(Nтр*17)+0			
Тип BC	≥2.20	5008[5161]+(Nтр*18)+0	bit field	R/W*	Биты 6-8. Принимает значения: • 0–механический; • 1–электронный; • 2-телеметрия (только ПВ≥2.20)
	<2.20	113[233]+(Nтр*17)+0			
Датчик P	≥2.20	5008[5161]+(Nтр*18)+0	bit field	R/W*	Биты 9-10. Принимает значения: • 0–нет; • 1–есть; • 2–есть, не исп.
	<2.20	---			
Напр. потока при телеметрии	≥2.20	5008[5161]+(Nтр*18)+0	bit field	R/W*	Биты 11-12. Принимает значения: • 0–прямой; • 1–обратный; • 2–реверс
	<2.20	---			
Резерв	≥2.20	5008[5161]+(Nтр*18)+00	bit field	R/W*	Биты 13-15
	<2.20	---			
tдог.	≥2.20	5008[5161]+(Nтр*18)+01	float	R/W*	В диапазоне 0–175 °С
	<2.20	113[233]+(Nтр*17)+01			
Pдог.	≥2.20	5008[5161]+(Nтр*18)+03	float	R/W*	В диапазоне 0–1,667 МПа (17 кгс/см ²)
	<2.20	113[233]+(Nтр*17)+03			
Pп	≥2.20	5008[5161]+(Nтр*18)+05	float	R/W*	В диапазоне ±0-9.99 (м)
	<2.20	113[233]+(Nтр*17)+05			
Pв	≥2.20	5008[5161]+(Nтр*18)+07	float	R/W*	В диапазоне 0–1,667 МПа (17 кгс/см ²)
	<2.20	113[233]+(Nтр*17)+07			
Вес имп.	≥2.20	5008[5161]+(Nтр*18)+09	float	R/W*	В диапазоне 0–9999.9999 (л)
	<2.20	113[233]+(Nтр*17)+09			
Vmin	≥2.20	5008[5161]+(Nтр*18)+11	float	R/W*	В диапазоне 0-99999,9 (м ³)
	<2.20	113[233]+(Nтр*17)+11			
Vmax	≥2.20	5008[5161]+(Nтр*18)+13	float	R/W*	В диапазоне 0-99999,9 (м ³)
	<2.20	113[233]+(Nтр*17)+13			
Vдог	≥2.20	5008[5161]+(Nтр*18)+15	float	R/W*	В диапазоне 0-99999,9 (м ³)
	<2.20	113[233]+(Nтр*17)+15			
Учет пустой трубы	≥2.20	5008[5161]+(Nтр*18)+17	bit field	R/W*	Бит 0. Принимает значения: • 0–нет; • 1–да
	<2.20	---			
Вход пустой трубы	≥2.20	5008[5161]+(Nтр*18)+17	bit field	R/W*	Биты 1-3. Принимает значения: • 0–не исп.; • 1-4 (номер входа)
	<2.20	---			
Резерв	≥2.20	5008[5161]+(Nтр*18)+17	bit field	R/W*	Бит 4
	<2.20	---			
Вход реверса	≥2.20	5008[5161]+(Nтр*18)+17	bit field	R/W*	Биты 5-7. Принимает значения: • 0–не исп.; • 1-4 (номер входа)
	<2.20	---			
Резерв	≥2.20	5008[5161]+(Nтр*18)+17	bit field	R/W*	Биты 8-15
	<2.20	---			

Примечание:

- запись вида XXXX[YYYY] означает, что параметр имеет адрес XXXX в БД1 и YYYY в БД2
- запись вида «---» означает, что параметр отсутствует
- Nтр означает номер трубы и изменяется в диапазоне 0÷6 для ПВ≥2.20 и 0÷5 для ПВ<2.20
- при работе с приборами, имеющими ПВ≥2.20 возможно чтение параметров, размещенных по адресам для ПВ<2.20. В этом случае, в целях сохранения максимальной совместимости, в процессе чтения производится преобразование структур данных ПВ≥2.20 к структурам ПВ<2.20, что может приводить к потерям значимой информации

4.2.2.3 Настройки тепловых вводов

Параметр	ПВ	Адрес	Тип	Доступ	Примечание
СИ	≥2.20	5134[5287]+(NТВ*9)+0	bit field	R/W*	Биты 0-7. Принимает значения: • 0—тепловой ввод не исп-ся; • 1—14;
	<2.20	215[335]+(NТВ*9)+0			
Исп. QТВ	≥2.20	5134[5287]+(NТВ*9)+0	bit field	R/W*	Биты 8-9. Принимает значения: • 0—есть; • 1—нет; • 2—сумма; • 3—разность Только для ПВ≥2.00
	<2.20	215[335]+(NТВ*9)+0			
Резерв	≥2.20	5134[5287]+(NТВ*9)+0	bit field	R/W*	Биты 10-15
	<2.20	215[335]+(NТВ*9)+0			
dMmax	≥2.20	5134[5287]+(NТВ*9)+1	float	R/W*	В диапазоне 0-4(%)
	<2.20	215[335]+(NТВ*9)+1			
txдог	≥2.20	5134[5287]+(NТВ*9)+3	float	R/W*	В диапазоне 0–99 (°C)
	<2.20	215[335]+(NТВ*9)+3			
Pxдог	≥2.20	5134[5287]+(NТВ*9)+5	float	R/W*	В диапазоне 0–1,667 МПа (17 кгс/см2)
	<2.20	215[335]+(NТВ*9)+5			
КТЗ	≥2.20	5134[5287]+(NТВ*9)+7	bit field	R/W*	Биты 0-2. Принимает значения: • 0 – труба 3 не используется; • 1 – измерение V в трубе 3; • 2 – измерение t в трубе 3; • 3 – измерение V и t в трубе 3
	<2.20	215[335]+(NТВ*9)+7			
ФРТ	≥2.20	5134[5287]+(NТВ*9)+7	bit field	R/W*	Биты 3-6. Принимает значения: • 0 - Q12=M1(h1-h2) • 1 - Q12=M1(h1-h2)+dM(h2-hx) • 2 - Q12=M2(h1-h2) • 3 - Q12=M2(h1-h2)+dM(h1-hx) • 4 - Q12=M1(h1-hx) • 5 - Q12=M2(h2-hx) • 6 - Q12=M1(h1-hx)+M2(h2-hx) • 7 - Q12=(M1*h1)-(M2*h2)-(M3*hx) • 8 - Q12=(M1*h1)-(M2*h2)-(M3*h3) • 9 - Q12=M1(h1-h2)+dM(h2-h3) • 10 - Q12=M2(h1-h2)+dM(h1-h3)
	<2.20	215[335]+(NТВ*9)+7			
Контроль t	≥2.20	5134[5287]+(NТВ*9)+7	bit field	R/W*	Биты 7-8. Принимает значения: • 0 – с подст.; • 1 – счет отмен.
	<2.20	215[335]+(NТВ*9)+7			
Контроль dM	≥2.20	5134[5287]+(NТВ*9)+7	bit field	R/W*	Биты 9-11. Принимает значения: • 0 – нет; • 1 – без подст 1.; • 2 – без подст 2.; • 3 – с подст. 1; • 4 – с подст. 2;
	<2.20	215[335]+(NТВ*9)+7			
Контроль Q	≥2.20	5134[5287]+(NТВ*9)+7	bit field	R/W*	Биты 12-14. Принимает значения: • 0 – нет; • 1 – без подст.; • 2 – с подст.; • 3 – счет отменен
	<2.20	215[335]+(NТВ*9)+7			
Контроль dt	≥2.20	5134[5287]+(NТВ*9)+7	bit field	R/W*	Биты 15-17. Принимает значения: • 0 – без подст.; • 1 – с подст.; • 2 – счет отменен; • 3 – нет (только для ПВ≥2.20);
	<2.20	215[335]+(NТВ*9)+7			
dtmin	≥2.20	5134[5287]+(NТВ*9)+7	bit field	R/W*	Биты 18-20. Принимает значения: • 0 – 2(°C); • 1 – 3(°C)
	<2.20	215[335]+(NТВ*9)+7			
Исп.tx	≥2.20	5134[5287]+(NТВ*9)+7	bit field	R/W*	Биты 21-24. Принимает значения для ПВ≥2.20: • 0 – hx не используется; • 1 – договорная; • 2 – Изм. на X8; • 3 – Изм. на X9; • 4 – Изм. на X3:A; • 5 – Изм. на X3:B; Принимает значения для ПВ<2.20: • 0 – hx не используется; • 1 – договорная; • 2 – измеряется в данном ТВ; • 3 – измеряется в другом ТВ
	<2.20	215[335]+(NТВ*9)+7			

Исп.тнв	≥2.20	5134[5287]+(Nтв*9)+7	bit field	R/W*	Биты 25-28. Принимает значения для ПВ≥2.20: <ul style="list-style-type: none"> • 0 – не используется; • 2 – Изм. на X8; • 3 – Изм. на X9; • 4 – Изм. на X3:A; • 5 – Изм. на X3:B; Принимает значения для ПВ<2.20: <ul style="list-style-type: none"> • 0 – не используется; • 1 – измеряется
	<2.20	215[335]+(Nтв*9)+7			
Контр.R	≥2.20	5134[5287]+(Nтв*9)+7	bit field	R/W*	Биты 29-31. Принимает значения: <ul style="list-style-type: none"> • 0 – нет; • 1 – Без подст. • 2 – С подст. • 3 – Счет отм. Только для ПВ≥2.20
	<2.20	215[335]+(Nтв*9)+7			

Примечание:

- запись вида XXXX[YYYY] означает, что параметр имеет адрес XXXX в БД1 и YYYY в БД2
- запись вида «---» означает, что параметр отсутствует
- Nтв означает номер теплового ввода и изменяется в диапазоне 0÷2 для ПВ≥2.20 и 0÷1 для ПВ<2.20
- при работе с приборами, имеющими ПВ≥2.20 возможно чтение параметров, размещенных по адресам для ПВ<2.20. В этом случае, в целях сохранения максимальной совместимости, в процессе чтения производится преобразование структур данных ПВ≥2.20 к структурам ПВ<2.20, что может приводить к потерям значимой информации

4.2.2.4 Настройки дополнительного импульсного входа

Параметр	ПВ	Адрес	Тип	Доступ	Примечание
Способ использ.	≥2.20	5314	bit field	R/W*	Биты 0-2. Принимает значения: • 0—нет; • 1—контроль сети; • 2—счет импульсов; • 3—сигнализация
	<2.20	353			
Уровень сигнала	≥2.20	5314	bit field	R/W*	Бит 3. Принимает значения: • 0—нормально разомкнутый; • 1—нормально замкнутый Только способа исп. «Сигнализация»
	<2.20	353			
Единиц. измер.	≥2.20	5314	bit field	R/W*	Биты 4-7. Принимает значения: • 0—м ³ ; • 1—кВт*ч Только способа исп. «Счет имп.»
	<2.20	353			
Резерв	≥2.20	5314	bit field	R/W*	Биты 8-15
	<2.20	353			
Вес имп./Тподтв	≥2.20	5315	float	R/W*	Для способа исп. «Счет имп.» в диапазоне 0-9999.9999 Для способа исп. «Сигнализация» в диапазоне 0.1-9.9 (с.)
	<2.20	354			
Знач. ЧН	≥2.20	5317	float	R/W*	Для способа исп. «Счет имп.» в диапазоне 0-99999.9
	<2.20	356			
Знач. ЧВ	≥2.20	5319	float	R/W*	Для способа исп. «Счет имп.» в диапазоне 0-99999.9
	<2.20	358			

4.2.2.5 Настройки дискретных входов

Параметр	ПВ	Адрес	Тип	Доступ	Примечание
Тип датчика	≥2.20	5321+(Ндв*2)	bit field	R/W*	Бит 0. Принимает значения: • 0—НР; • 1—НЗ;
Время подтв.	≥2.20	5322+(Ндв*2)	unsigned short	R/W*	В диапазоне 0+99 (с.)

4.2.3 Настройки незащищенные

Структура предназначена для хранения такой информации как код организации, номер договора, сетевой адрес. При изменении настроек прибора система верхнего уровня должна выполнить следующие действия:

- рассчитать контрольную сумму записываемых настроек и записать ее в структуру «Информация о настройках» (см. «Информация о настройках»);
- записать содержимое настроек;

Параметр	Адрес	Тип	Доступ	Примечание
Сетевой адрес	2153	bit field	R/W	Биты 0-7. В диапазоне 0-255
Ед. измерения	2153	bit field	R/W	Бит 8. Принимает значения: • 0—СИ; • 1—МКС Только в ПВ≥2.00
Резерв	2153	bit field	R/W	Биты 9-15
Код организации	2154	char[10]	R/W	Допустимы только цифровые символы ('0'-'9'). Последний символ последовательности, задающий длину строки, должен иметь значение двоичного нуля
Договор	2159	char[10]	R/W	Допустимы только цифровые символы ('0'-'9'). Последний символ последовательности, задающий длину строки, должен иметь значение двоичного нуля

4.3 Архивы

4.3.1 Информация о датах начала/конца архивов

Структура предназначена для чтения дат начала/окончания архивов параметров.

Параметр	Адрес	Тип	Доступ	Примечание
Дата начала АЧ,	2676	unsigned short	R/O	Младший байт, день: 1-31

день/месяц				Старший байт, месяц: 1-12
Дата начала АЧ, год/час	2677	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
Дата начала АЧ, минута/секунда	2678	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
Дата начала АС, день/месяц	2679	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
Дата начала АС, год/час	2680	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
Дата начала АС, минута/ секунда	2681	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
Дата начала АМ, день/месяц	2682	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
Дата начала АМ, год/час	2683	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
Дата начала АМ, минута/ секунда	2684	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
Дата начала АИ, день/месяц	2685	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
Дата начала АИ, год/час	2686	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
Дата начала АИ, минута/ секунда	2687	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
Дата окончания АЧ, день/месяц	2688	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
Дата окончания АЧ, год/час	2689	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
Дата окончания АЧ, минута/ секунда	2690	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
Дата окончания АС, день/месяц	2691	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
Дата окончания АС, год/час	2692	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
Дата окончания АС, минута/ секунда	2693	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
Дата окончания АМ, день/месяц	2694	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
Дата окончания АМ, год/час	2695	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
Дата окончания АМ, минута/ секунда	2696	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
Дата окончания АИ, день/месяц	2697	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
Дата окончания АИ, год/час	2698	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
Дата окончания АИ, минута/ секунда	2699	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
Дата сброса архива, день/месяц	2700	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
Дата сброса архива, год/час	2701	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
Дата сброса архива, минута/ секунда	2702	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59

ПРИМЕЧАНИЕ! Значение 255 во всех полях даты/времени для архива определённого типа говорят о том, что архив пуст

4.3.2 Информация об индексах архивов параметров

Структура предназначена для чтения служебной информации о ведении архивов.

Параметр	Адрес	Тип	Доступ	Примечание
Дата сброса архива, день/месяц	2708	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
Дата сброса архива, год/час	2709	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
Дата сброса архива, минута/ секунда	2710	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
Макс.кол-во записей (АЧ)	2711	unsigned short	R/O	Максимальное количество записей для архива данного типа
Индекс записи (АЧ)	2712	unsigned short	R/O	В диапазоне 0-«Макс. количество записей», Означает номер записи, в который будет произведено сохранение архивной информации в следующий раз
Размер записи, байт (АЧ)	2713	unsigned short	R/O	Размер записи, (байт)
Признак закольц. (АЧ)	2714	bit field	R/O	Бит 0. Принимает значения: • 0–архив еще не заполнен; • 1– архив заполнен и сохранение новой записи вызывает стирание самой старой записи
Резерв	2714	bit field	R/O	Биты 1-15
Макс.кол-во записей (АС)	2715	unsigned short	R/O	Максимальное количество записей для архива данного типа
Индекс записи (АС)	2716	unsigned short	R/O	В диапазоне 0-«Макс. количество записей», Означает номер записи, в который будет произведено сохранение архивной информации в следующий раз
Размер записи, байт (АС)	2717	unsigned short	R/O	Размер записи, (байт)
Признак закольц. (АС)	2718	bit field	R/O	Бит 0. Принимает значения: • 0–архив еще не заполнен; • 1– архив заполнен и сохранение новой записи вызывает стирание самой старой записи
Резерв	2718	bit field	R/O	Биты 1-15
Макс.кол-во записей (АМ)	2719	unsigned short	R/O	Максимальное количество записей для архива данного типа
Индекс записи (АМ)	2720	unsigned short	R/O	В диапазоне 0-«Макс. количество записей», Означает номер записи, в который будет произведено сохранение архивной информации в следующий раз
Размер записи, байт (АМ)	2721	unsigned short	R/O	Размер записи, (байт)
Признак закольц. (АМ)	2722	bit field	R/O	Бит 0. Принимает значения: • 0–архив еще не заполнен; • 1– архив заполнен и сохранение новой записи вызывает стирание самой старой записи
Резерв	2722	bit field	R/O	Биты 1-15
Макс.кол-во записей (АИ)	2723	unsigned short	R/O	Максимальное количество записей для архива данного типа
Индекс записи (АИ)	2724	unsigned short	R/O	В диапазоне 0-«Макс. количество записей», Означает номер записи, в который будет произведено сохранение архивной информации в следующий раз
Размер записи, байт (АИ)	2725	unsigned short	R/O	Размер записи, (байт)
Признак закольц. (АИ)	2726	bit field	R/O	Бит 0. Принимает значения: • 0–архив еще не заполнен; • 1– архив заполнен и сохранение новой записи вызывает стирание самой старой записи
Резерв	2726	bit field	R/O	Биты 1-15

4.3.3 Тип читаемых данных

Некоторые данные можно получить из прибора только выполнив специальную последовательность запросов. Например, это касается чтения архивов. При помощи операции записи регистров система верхнего уровня сообщает прибору, какие именно данные она собирается читать. Например, «архив часовой за 21.04.2011 14:00:00». Далее система верхнего уровня читает интересующие ее данные из адресного пространства, соответствующего записанному типу данных (чтение регистров). В момент получения запроса на чтение прибор готовит данные, буферизирует их, и передает системе верхнего уровня. Т.о. чтение архивных данных выглядит как несколько последовательных серий операций «запись регистров, чтение регистров».

ПРИМЕЧАНИЕ! Структура «Тип читаемых данных» должна использоваться системой верхнего уровня при чтении архива параметров, архивов изменений базы данных, административных событий и диагностик

Параметр	Адрес	Тип	Доступ	Примечание
Дата: день, месяц	99	unsigned short	R/W	Младший байт, день: 1-31 Старший байт, месяц: 1-12
Дата: год-2000, час	100	unsigned short	R/W	Младший байт, год минус 2000 Старший байт, час: 0-23
Время: минута, секунда	101	unsigned short	R/W	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
Тип архива	102	unsigned short	R/W	Принимает значения: <ul style="list-style-type: none"> • 0 - часовой архив; • 1 - суточный архив; • 2 - месячный архив; • 3 - итоговый архив;
Номер записи	103	unsigned short	R/W	Означает номер записи для чтения. При чтении асинхронных архивов изменяется от нуля до «максимальное количество записей в архиве - 1» (см. «Архивные записи асинхронных архивов»). Также данное поле используется для технологического чтения flash-памяти
Резерв	104	unsigned short	R/W	Не используется

ПРИМЕЧАНИЕ! При чтении часового, суточного, месячного, и итогового архива система верхнего уровня должна установить поля «Дата:...», «Время:...», а также «Тип архива» в допустимые значения (необходимо выполнить операцию записи регистров). Поле «Номер записи» прибором игнорируется. После этого нужно выполнить чтение регистров по адресу, зависящему от выбранного типа архива (см. «Архивная запись (часовой, суточный, месячный архивы)» и «Архивная запись (итоговый архив)»).

ПРИМЕЧАНИЕ! При чтении архива изменения базы данных, архива административных событий или диагностического архива система верхнего уровня должна установить только поле «Номер записи» в допустимое для данного типа архива значение (см. «Архивные записи асинхронных архивов») (необходимо выполнить операцию записи регистров). Остальные поля прибором игнорируются. После этого нужно выполнить чтение регистров по адресу, зависящему от выбранного типа архива (см. «Архивные записи асинхронных архивов»).

4.3.4 Архивная запись (часовой, суточный, месячный архивы)

Структура предназначена для чтения значений архивных параметров из часового, суточного и месячного архивов. Для получения архивной записи необходимо сообщить прибору за какую дату и из какого типа архива необходимо извлекать данные (см. «[Тип читаемых данных](#)»).

Параметр	ПВ	Адрес	Тип	Доступ	Примечание
Дата: день/месяц	≥2.20	7048	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
	<2.20	2740			
Дата: год/час	≥2.20	7049	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
	<2.20	2741			
Температура	≥2.20	7050+(Nтр*8)+00	float	R/O	(°C)
	<2.20	2742+(Nтр*8)+00			
Давление	≥2.20	7050+(Nтр*8)+02	float	R/O	(МПа)
	<2.20	2742+(Nтр*8)+02			
Объем	≥2.20	7050+(Nтр*8)+04	float	R/O	(м³)
	<2.20	2742+(Nтр*8)+04			
Масса	≥2.20	7050+(Nтр*8)+06	float	R/O	(т)
	<2.20	2742+(Nтр*8)+06			
tнв	≥2.20	7106+(Nтв*18)+00	float	R/O	(°C)
	<2.20	2790+(Nтв*18)+00			
tx	≥2.20	7106+(Nтв*18)+02	float	R/O	(°C)
	<2.20	2790+(Nтв*18)+02			
Pх	≥2.20	7106+(Nтв*18)+04	float	R/O	(МПа)
	<2.20	2790+(Nтв*18)+04			
dt	≥2.20	7106+(Nтв*18)+06	float	R/O	(°C)
	<2.20	2790+(Nтв*18)+06			
dM	≥2.20	7106+(Nтв*18)+08	float	R/O	(т)
	<2.20	2790+(Nтв*18)+08			
Qтв	≥2.20	7106+(Nтв*18)+10	float	R/O	(ГДж)
	<2.20	2790+(Nтв*18)+10			
Q12	≥2.20	7106+(Nтв*18)+12	float	R/O	(ГДж)
	<2.20	2790+(Nтв*18)+12			
Qг	≥2.20	7106+(Nтв*18)+14	float	R/O	(ГДж)
	<2.20	2790+(Nтв*18)+14			
ВНР	≥2.20	7106+(Nтв*18)+16	unsigned short	R/O	(ч)
	<2.20	2790+(Nтв*18)+16			
ВОС	≥2.20	7106+(Nтв*18)+17	unsigned short	R/O	(ч)
	<2.20	2790+(Nтв*18)+17			
ДП	≥2.20	7160	float	R/O	
	<2.20	2826			
НС по Трубе 1/2	≥2.20	7162	unsigned short	R/O	Младший байт, труба 1 Старший байт, труба 2 Битовая маска. Принимает значения: <ul style="list-style-type: none"> • 0-НС отсутствует; • 1-НС присутствует Порядок бит: <ul style="list-style-type: none"> • Бит 0 – НС t<min; • Бит 1 – НС t>max; • Бит 2 – НС неиспр. датчика t; • Бит 3 – НС P<min; • Бит 4 – НС P>max; • Бит 5 – НС V<min; • Бит 6 – НС V>max; • Бит 7 – НС неиспр. или отсут. питания ВС
	<2.20	2828			
НС по Трубе 3/4	≥2.20	7163	unsigned short	R/O	Младший байт, труба 3 Старший байт, труба 4 См. «НС по Трубе 1/2»
	<2.20	2829			
НС по Трубе 5/6	≥2.20	7164	unsigned short	R/O	Младший байт, труба 5 Старший байт, труба 6 См. «НС по Трубе 1/2»
	<2.20	2830			
НС по Трубе 7	≥2.20	7165	unsigned short	R/O	Младший байт, труба 7 Старший байт, не исп-ся См. «НС по Трубе 1/2»
	<2.20	---			
НС по ТВ1	≥2.20	7166	unsigned short	R/O	Битовая маска. Принимает значения: <ul style="list-style-type: none"> • 0-НС отсутствует; • 1-НС присутствует Порядок бит: <ul style="list-style-type: none"> • Бит 0 – НС по dt; • Бит 1 – НС по dM; • Бит 2 – НС по Qтв; • Бит 3 – НС tx<min;

	<2.20	2831			<ul style="list-style-type: none"> • Бит 4 – HC tx>max; • Бит 5 – HC неиспр. датчика tx; • Бит 6 – HC тнв<min; • Бит 7 – HC тнв>max; • Бит 8 – HC неиспр. датчика тнв • Бит 9 – HC по Q12; • Бит 10 – HC по Qg; • Бит 11 – HC по Pх<min; • Бит 12 – HC по Pх>max
HC по TB2	≥2.20	7167	unsigned short	R/O	См. «HC по TB1»
	<2.20	2832			
HC по TB3	≥2.20	7168	unsigned short	R/O	См. «HC по TB1»
	<2.20	---			
HC по ДП	≥2.20	7169	unsigned short	R/O	Младший байт, Наличие HC Битовая маска. Принимает значения: <ul style="list-style-type: none"> • 0-HC отсутствует; • 1-HC присутствует Порядок бит: <ul style="list-style-type: none"> • 0 – HC по ДП<min; • 1 – HC по ДП>max Старший байт, не исп-ся
	<2.20	2833			
Признаки событий	≥2.20	7170	unsigned short	R/O	Битовая маска. Принимает значения: <ul style="list-style-type: none"> • Бит 0 – нажатие кнопки «Доступ»; • Бит 1 – отсутствие сетевого питания; • Бит 2 – низкое напряжение батареи; • Бит 3 – перезапуск процессора; • Бит 4 – ускоренный режим; • Бит 5 – ресурс батареи близок к окончанию; • Бит 6 – перевод часов; • Бит 7 – включение подсветки дисплея; • Бит 8 – нажатие клавиатуры; • Бит 9 – обращение по сети; • Бит 10 – сигнализация; • Бит 11 – изменение настроек (было изменение настроек, приводящее к изменению контрольной суммы); • Бит 12 – калибровки разрешены; • Бит 13 – использование SD карты • Бит 14 – использование USB • Бит 15 – депассивация батареи (технологическая операция поддержания батареи в рабочем состоянии).
	<2.20	2834			
Резерв	≥2.20	7171	unsigned short	R/O	
	<2.20	2835			
Длит. работы по сети	≥2.20	7172	unsigned short	R/O	(мин.)
	<2.20	2836			
Длит. работы дисплея	≥2.20	7173	unsigned short	R/O	(мин.)
	<2.20	2837			
Длит. отсут. сет. питания	≥2.20	7174	unsigned short	R/O	(мин.)
	<2.20	2838			
Активная БД	≥2.20	7175+(Nтв*2)+00	bit field	R/O	Биты 0-7. Принимает значения: <ul style="list-style-type: none"> • 0-активна БД1; • 1-активна БД2
	<2.20	2839+(Nтв*2)+00			
СИ ТВ	≥2.20	7175+(Nтв*2)+00	bit field	R/O	Биты 8-15. значение конфигурационного параметра СИ
	<2.20	2839+(Nтв*2)+00			
КТЗ ТВ	≥2.20	7175+(Nтв*2)+01	bit field	R/O	Биты 0-7. значение конфигурационного параметра КТЗ
	<2.20	2839+(Nтв*2)+01			
ФРТ ТВ	≥2.20	7175+(Nтв*2)+01	bit field	R/O	Биты 8-15. значение конфигурационного параметра ФРТ
	<2.20	2839+(Nтв*2)+01			
Апп. неиспр	≥2.20	7181	unsigned short	R/O	Битовая маска. Принимает значения: <ul style="list-style-type: none"> • Бит 0 – «АЦП1»; • Бит 1 – «АЦП2»; • Бит 2 – «EEPROM»; • Бит 3 – «EEPROMw»; • Бит 4 – «Пит. дат.»; • Бит 5 – «RTC»; • Бит 6 – «CLK»; Только для ПВ≥2.00
	<2.20	2843			

КСН	≥2.20	7182	unsigned short	R/O	Только для ПВ≥2.00
	<2.20	2844			
Исп. tx	≥2.20	7183+(Nтв*1)+00	bit field	R/O	Биты 0-7. Значение конфигурационного параметра «Исп.tx» Только для ПВ≥2.00
	<2.20	2845+(Nтв*1)+00			
Исп. тнв	≥2.20	7183+(Nтв*1)+00	bit field	R/O	Биты 8-15. Значение конфигурационного параметра «Исп.тнв» Только для ПВ≥2.00
	<2.20	2845+(Nтв*1)+00			
Признаки пустой трубы	≥2.20	7186	bit field	R/O	Биты 0-7. Битовая маска. Принимает значения: • 0-Ситуация отсутствует; • 1-Ситуация присутствует Номер бита соответствует номеру трубы
	<2.20	---			
Признаки реверса	≥2.20	7186	bit field	R/O	Биты 8-15. Битовая маска. Принимает значения: • 0-Ситуация отсутствует; • 1-Ситуация присутствует Номер бита соответствует номеру трубы
	<2.20	---			
Событие реверса	≥2.20	7187	bit field	R/O	Биты 0-7. Битовая маска. Принимает значения: • 0-Ситуация отсутствует; • 1-Ситуация присутствует Номер бита соответствует номеру трубы
	<2.20	---			
Резерв	≥2.20	7187	bit field	R/O	Биты 8-15.
	<2.20	2848			
Qсум	≥2.20	7188	float	R/O	(ГДж), Только для ПВ≥2.00
	<2.20	2849			
Исп.Qсум	≥2.20	7190	unsigned short	R/O	Битовая маска. • Биты 0-1 - значение по ТВ1 • Биты 2-3 - значение по ТВ2 • Биты 4-5 - значение по ТВ3 Хранит значение конфигурационного параметра «Исп.Qсум» Только для ПВ≥2.00
	<2.20	2847			

Примечание:

- запись вида «---» означает, что параметр отсутствует
- Nтр означает номер трубы и изменяется в диапазоне 0÷6 для ПВ≥2.20 и 0÷5 для ПВ<2.20
- Nтв означает номер теплового ввода и изменяется в диапазоне 0÷2 для ПВ≥2.20 и 0÷1 для ПВ<2.20
- при работе с приборами, имеющими ПВ≥2.20 возможно чтение параметров, размещенных по адресам для ПВ<2.20. В этом случае, в целях сохранения максимальной совместимости, в процессе чтения производится преобразование структур данных ПВ≥2.20 к структурам ПВ<2.20, что может приводить к потерям значимой информации

4.3.5 Архивная запись (итоговый архив)

Структура предназначена для чтения значений архивных параметров из часового, суточного и месячного архивов. Для получения архивной записи необходимо сообщить прибору за какую дату и из архива какого типа необходимо извлекать данные (см. «[Тип читаемых данных](#)»).

Параметр	ПВ	Адрес	Тип	Доступ	Примечание
Дата: день/месяц	≥2.20	7304	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
	<2.20	2868			
Дата: год/час	≥2.20	7305	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
	<2.20	2869			
Объем	≥2.20	7306+(Nтр*8)+0	double	R/O	(м ³)
	<2.20	2870+(Nтр*8)+0			
Масса	≥2.20	7306+(Nтр*8)+4	double	R/O	(т)
	<2.20	2870+(Nтр*8)+4			
dM	≥2.20	7362+(NТВ*23)+0	double	R/O	(т)
	<2.20	2918+(NТВ*23)+0			
Qтв	≥2.20	7362+(NТВ*23)+4	double	R/O	(ГДж)
	<2.20	2918+(NТВ*23)+4			
Q12	≥2.20	7362+(NТВ*23)+8	double	R/O	(ГДж)
	<2.20	2918+(NТВ*23)+8			
Qг	≥2.20	7362+(NТВ*23)+12	double	R/O	(ГДж)
	<2.20	2918+(NТВ*23)+12			
ВНР	≥2.20	7362+(NТВ*23)+16	unsigned short	R/O	(ч)
	<2.20	2918+(NТВ*23)+16			
ВОС	≥2.20	7362+(NТВ*23)+17	unsigned short	R/O	(ч)
	<2.20	2918+(NТВ*23)+17			
TVmin	≥2.20	7362+(NТВ*23)+18	unsigned short	R/O	(ч)
	<2.20	2918+(NТВ*23)+18			
TVmax	≥2.20	7362+(NТВ*23)+19	unsigned short	R/O	(ч)
	<2.20	2918+(NТВ*23)+19			
Tdt	≥2.20	7362+(NТВ*23)+20	unsigned short	R/O	(ч)
	<2.20	2918+(NТВ*23)+20			
Тбез пит.	≥2.20	7362+(NТВ*23)+21	unsigned short	R/O	(ч)
	<2.20	2918+(NТВ*23)+21			
Тнеиспр.	≥2.20	7362+(NТВ*23)+22	unsigned short	R/O	(ч)
	<2.20	2918+(NТВ*23)+22			
ДП	≥2.20	7431	double	R/O	
	<2.20	2964			
Длит. работы по сети	≥2.20	7435	unsigned long	R/O	(мин.)
	<2.20	2968			
Длит. работы дисплея	≥2.20	7437	unsigned long	R/O	(мин.)
	<2.20	2970			
Длит. отсут. сет. питания	≥2.20	7439	unsigned long	R/O	(мин.)
	<2.20	2972			
Активная БД	≥2.20	7441+(NТВ*2)+0	bit field	R/O	Биты 0-7. Принимает значения: • 0-активна БД1; • 1-активна БД2
	<2.20	2974+(NТВ*2)+0			
СИ ТВ	≥2.20	7441+(NТВ*2)+0	bit field	R/O	Биты 8-15. значение конфигурационного параметра СИ
	<2.20	2974+(NТВ*2)+0			
КТЗ ТВ	≥2.20	7441+(NТВ*2)+1	bit field	R/O	Биты 0-7. значение конфигурационного параметра КТЗ
	<2.20	2974+(NТВ*2)+1			
ФРТ ТВ	≥2.20	7441+(NТВ*2)+1	bit field	R/O	Биты 8-15. значение конфигурационного параметра ФРТ
	<2.20	2974+(NТВ*2)+1			
Признаки событий		7447	unsigned short		Битовая маска. Принимает значения: • Бит 0 – нажатие кнопки «Доступ»; • Бит 1 – отсутствие сетевого питания; • Бит 2 – низкое напряжение батареи; • Бит 3 – перезапуск процессора; • Бит 4 – ускоренный режим; • Бит 5 – ресурс батареи близок к окончанию; • Бит 6 – перевод часов; • Бит 7 – включение подсветки дисплея;

		2978			<ul style="list-style-type: none"> • Бит 8 – нажатие клавиатуры; • Бит 9 – обращение по сети; • Бит 10 – сигнализация; • Бит 11 – изменение настроек (было изменение настроек, приводящее к изменению контрольной суммы); • Бит 12 – калибровки разрешены; • Бит 13 – использование SD карты; • Бит 14 – использование USB; • Бит 15 – депассивация батареи (технологическая операция поддержания батареи в рабочем состоянии).
KCH		7448 2979	unsigned short		Только для ПВ≥2.00
Исп. tx	≥2.20	7449+(Nтв*1)+0	bit field	R/O	Биты 0-7. Значение конфигурационного параметра «Исп.tx» Только для ПВ≥2.00
	<2.20	2980+(Nтв*1)+0			
Исп. тнв	≥2.20	7449+(Nтв*1)+0	bit field	R/O	Биты 8-15. Значение конфигурационного параметра «Исп.тнв» Только для ПВ≥2.00
	<2.20	2980+(Nтв*1)+0			
Исп. Qсум	≥2.20	7452	unsigned short	R/O	Битовая маска. <ul style="list-style-type: none"> • Биты 0-1 - значение по TB1 • Биты 2-3 - значение по TB2 • Биты 4-5 - значение по TB3 Хранит значение конфигурационного параметра «Исп.Qсум» Только для ПВ≥2.00
	<2.20	2982			
Резерв	≥2.20	7453	unsigned short	R/O	Только для ПВ≥2.00
	<2.20	2983			
Тпустой трубы	≥2.20	7454+(Nтв*2)+0	unsigned short	R/O	(ч). Только для ПВ≥2.00
	<2.20	---			
Треверса	≥2.20	7454+(Nтв*2)+1	unsigned short	R/O	(ч). Только для ПВ≥2.00
	<2.20	---			
Qсум	≥2.20	7460	double	R/O	(ГДж). Только для ПВ≥2.00
	<2.20	2984			
Примечание:					
<ul style="list-style-type: none"> - запись вида «---» означает, что параметр отсутствует - Nтр означает номер трубы и изменяется в диапазоне 0÷6 для ПВ≥2.20 и 0÷5 для ПВ<2.20 - Nтв означает номер теплового ввода и изменяется в диапазоне 0÷2 для ПВ≥2.20 и 0÷1 для ПВ<2.20 - при работе с приборами, имеющими ПВ≥2.20 возможно чтение параметров, размещенных по адресам для ПВ<2.20. В этом случае, в целях сохранения максимальной совместимости, в процессе чтения производится преобразование структур данных ПВ≥2.20 к структурам ПВ<2.20, что может приводить к потерям значимой информации 					

4.3.6 Информация об асинхронных архивах

Структура предназначена для чтения информации об архиве изменений базы данных, архиве событий и диагностическом архиве.

Параметр	Адрес	Тип	Доступ	Примечание
Макс. кол-во записей	2996+(Nарх*4)+0	unsigned short	R/O	Максимальное количество записей для архива данного типа
Индекс записи	2996+(Nарх*4)+1	unsigned short	R/O	В диапазоне 0-«Макс. количество записей». Означает номер записи, в который будет произведено сохранение архивной информации в следующий раз
Размер записи, байт	2996+(Nарх*4)+2	unsigned short	R/O	Размер записи, (байт). Для ПВ≥2.20 к полученному значению надо прибавить 4
Признак закольц.	2996+(Nарх*4)+3	bit field	R/O	Бит 0. Принимает значения: <ul style="list-style-type: none"> • 0 – архив еще не заполнен; • 1 – архив заполнен и сохранение новой записи вызывает стирание самой старой записи
Резерв	2996+(Nарх*4)+3	bit field	R/O	Биты 1-15
Примечание:				
- Nарх – номер архива, изменяющийся в диапазоне 0-2, где 0-АИБД, 1-ААС, 2-АД				

4.3.7 Архивные записи асинхронных архивов

Структура предназначена для чтения записей архива изменений базы данных, архива событий и диагностического архива. Возможно чтение сразу нескольких архивных записей. В этом случае система верхнего уровня указывает индекс первой желаемой записи (см. [Тип читаемых данных](#)) и запрашивает количество регистров, соответствующее 1, 2, 3...N записям.

Параметр	ПВ	Адрес	Тип	Доступ	Примечание
АИБД Дата: день/месяц	≥2.20	8236	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
	<2.20	3028			
АИБД Дата: год/час	≥2.20	8237	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
	<2.20	3029			
АИБД Дата: мин/сек	≥2.20	8237	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
	<2.20	3030			
АИБД Событие/Параметр	≥2.20	8239	bit field	R/O	Биты 0-8. Условный код события или наименование параметра, вызвавшего создание архивной записи
	<2.20	3031			
АИБД № ТВ/Трубы	≥2.20	8239	bit field	R/O	Биты 9-11. Контекстно означает номер теплового ввода или трубы, в которой произошло изменение параметра (только для АИБД)
	<2.20	3031			
АИБД Резерв	≥2.20	8239	bit field	R/O	Биты 12-14
	<2.20	3031			
АИБД БД	≥2.20	8239	bit field	R/O	Бит 15. Номер базы данных, в которой произошло изменение параметра (только для АИБД). Принимает значения: • 0-активна БД1; • 1-активна БД2
	<2.20	3031			
АИБД Старое значение	≥2.20	8240	unsigned char[4]	R/O	Может быть представлено различными типами данных в зависимости от измененного параметра
	<2.20	3032			
АИБД Новое значение	≥2.20	8242	unsigned char[4]	R/O	Может быть представлено различными типами данных в зависимости от измененного параметра
	<2.20	3034			
АИБД Общее кол-во записей	≥2.20	8244	unsigned long	R/O	Общее кол-во записей, произведенное в архиве
	<2.20	---			
ААС Дата: день/месяц	≥2.20	8396	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
	<2.20	3156			
ААС Дата: год/час	≥2.20	8397	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
	<2.20	3157			
ААС Дата: мин/сек	≥2.20	8398	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
	<2.20	3158			
ААС Событие/Параметр	≥2.20	8399	bit field	R/O	Биты 0-8. Условный код события или наименование параметра, вызвавшего создание архивной записи
	<2.20	3159			
Резерв	≥2.20	8399	bit field	R/O	Биты 9-15.
	<2.20	3159			
ААС Старое значение	≥2.20	8400	unsigned char[4]	R/O	Может быть представлено различными типами данных в зависимости от измененного параметра
	<2.20	3160			
ААС Новое значение	≥2.20	8402	unsigned char[4]	R/O	Может быть представлено различными типами данных в зависимости от измененного параметра
	<2.20	3162			
ААС Общее кол-во записей	≥2.20	8404	unsigned long	R/O	Общее кол-во записей, произведенное в архиве
	<2.20	---			
АД Дата: день/месяц	≥2.20	8556	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
	<2.20	3284			
АД Дата: год/час	≥2.20	8557	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
	<2.20	3285			
АД Дата: мин/сек	≥2.20	8558	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
	<2.20	3286			
АД Событие/Параметр	≥2.20	8559	bit field	R/O	Биты 0-8. Условный код события или наименование параметра, вызвавшего создание архивной записи
	<2.20	3287			
Резерв	≥2.20	8559	bit field	R/O	Биты 9-15.
	<2.20	3287			

АД Старое значение	≥2.20	8560	unsigned char[4]	R/O	Может быть представлено различными типами данных в зависимости от измененного параметра
	<2.20	3288			
АД Новое значение	≥2.20	8562	unsigned char[4]	R/O	Может быть представлено различными типами данных в зависимости от измененного параметра
	<2.20	3290			
АД Общее кол-во записей	≥2.20	8264	unsigned long	R/O	Общее кол-во записей, произведенное в архиве
	<2.20	---			
Примечание:					
<ul style="list-style-type: none"> - запись вида «---» означает, что параметр отсутствует - Нарх – номер архива, изменяющийся в диапазоне 0-2, где 0-АИБД, 1-ААС, 2-АД - при работе с приборами, имеющими ПВ≥2.20 возможно чтение параметров, размещенных по адресам для ПВ<2.20. В этом случае, в целях сохранения максимальной совместимости, в процессе чтения производится преобразование структур данных ПВ≥2.20 к структурам ПВ<2.20, что может приводить к потерям значимой информации 					

4.3.7.1 Интерпретация архива изменений базы данных

Архив изменений базы данных содержит информацию о действиях пользователя по изменению значений параметров настроечной базы данных прибора. Каждая архивная запись может содержать метку времени, код измененного параметра, номер базы данных, номер трубы/ТВ, старое и новое значения. При интерпретации содержимого базы данных следует учитывать, что некоторые из параметров являются общими (не входят ни в БД1 ни в БД2, см. столбец «БД» таблицы). Или могут принадлежать одной из БД и при этом относиться к какой-либо трубе или какому-либо ТВ. Если в столбце «№ ТВ/трубы» указано «труба», значит, параметр принадлежит трубе. В этом случае поле архивной записи «№ТВ/Трубы» содержит номер трубы в диапазоне 0-6 для ПВ≥2.20 и 0-5 для ПВ<2.20. Если в столбце «№ ТВ/трубы» указано «ТВ», значит, параметр принадлежит тепловому вводу. Значение 0 в поле «№ТВ/Трубы» кодирует ТВ1, значение 1 – ТВ2, значение 3 - ТВ3 (только для ПВ≥2.20). Значение 0 поля «БД» кодирует БД1, значение 1 – БД2.

№	Событие	№ ТВ/Тр.	БД	Тип значений	Список значений	Примечание
0	Исп. БД2	-	-	unsigned char	0-нет, 1-да	
1	БД1<->БД2	-	-	unsigned char	0-вручную, 1-авто по дате	
2	С клав.	-	-	unsigned char	0-запрет, 1-с паролем, 2-с Доступом	
3	С ПК	-	-	unsigned char	0-запрет, 1-с паролем	
4	БД1 с	-	-	unsigned char[4]	Дата	ДД.ММ.ГГ ЧЧ
5	БД2 с	-	-	unsigned char[4]	Дата	ДД.ММ.ГГ ЧЧ
6	Резерв 1	-	-	---	---	
7	Резерв 2	-	-	---	---	
8	Резерв 3	-	-	---	---	
9	Резерв 4	-	-	---	---	
10	Резерв 5	-	-	---	---	
11	Резерв 6	-	-	---	---	
12	Резерв 7	-	-	---	---	
13	Резерв 8	-	-	---	---	
14	Резерв 9	-	-	---	---	
15	Резерв 10	-	-	---	---	
16	Час отчета	-	-	unsigned char	0-23	
17	Дата отчета	-	-	unsigned char	1-31	
18	Сет. адрес	-	-	unsigned char	0-255	
19	Код. орг.	-	-	---	Старое и новое зн. не формируются	
20	Договор	-	-	---	Старое и новое зн. не формируются	
21	Сист. единиц	-	-	unsigned char	0-СИ, 1-МКС	
22	Термопреобр.	-	-	unsigned char	0–100П, 1–500П, 2-Рt100, 3-Рt500	
23	Перев. часов	-	-	unsigned char	0-Выкл, 1-Вкл	
24	Время	-	-	unsigned char[3]	Время	ЧЧ:ММ:СС
25	Пароль	-	-	---	Старое и новое зн. не формируются	
26	Дата	-	-	unsigned char[3]	Дата	ДД.ММ.ГГ
27	Резерв 11	-	-	---	---	
28	Резерв 12	-	-	---	---	
29	Резерв 13	-	-	---	---	
30	Резерв 14	-	-	---	---	
31	Резерв 15	-	-	---	---	
32	Резерв 16	-	-	---	---	
33	Резерв 17	-	-	---	---	
34	Резерв 18	-	-	---	---	

№	Событие	№ ТВ/Тр.	БД	Тип значений	Список значений	Примечание
35	Резерв 19	-	-	---	---	
36	Резерв 20	-	-	---	---	
37	Контр.V	труба	+	unsigned char	0-нет,1-без подст.,2-Подст.,3-Подст и контр U,4-счет отменен	
38	Контр. BC	труба	+	unsigned char	0-нет, 1-сеть(общ.),2-Резерв	
39	Датчик P	труба	+	unsigned char	0-нет, 1-да	
40	Тип BC	труба	+	unsigned char	0-Механич.,1-Электрон.	
41	Рдог	труба	+	float		
42	тдог	труба	+	float		
43	Рп	труба	+	float		
44	Рв	труба	+	float		
45	Вес импульса	труба	+	float		
46	Vmin	труба	+	float		
47	Vmax	труба	+	float		
48	Vдог	труба	+	float		
49	Вес имп.(телем)	труба	+	float		
50	Учет потока	труба	+	unsigned char	0-прямой, 1-обратный, 2-реверс	
51	Контроль ПТ	труба	+	unsigned char	0-нет, 1 есть	
52	Вход ПТ	труба	+	unsigned char	0-нет, 1-4 (ДВ1-ДВ4)	
53	Резерв 25	-	-	---	---	
54	Вход R	труба	+	unsigned char	0-нет, 1-4 (ДВ1-ДВ4)	
55	Резерв 27	-	-	---	---	
56	Резерв 28	-	-	---	---	
57	Резерв 29	-	-	---	---	
58	Резерв 30	-	-	---	---	
59	СИ	ТВ	+	unsigned char		
60	dMmax	ТВ	+	float		
61	тхд	ТВ	+	float		
62	Рхд	ТВ	+	float		
63	КТЗ	ТВ	+	unsigned char		
64	ФРТ	ТВ	+	unsigned char		
65	Контр.t	ТВ	+	unsigned char	0-с подст.,1-счет отмен.	
66	Контр.dM	ТВ	+	unsigned char	0-Нет,1-Без подст.1,2-Без подст.2,3-С подст.1,4-С подст.2	
67	Контр.Q	ТВ	+	unsigned char	0-Нет, 1-Без подст., 2-С подст.	
68	Контр.dt	ТВ	+	unsigned char	0-без подст., 1-с подст., 2-счет отмен.	
69	dtmin	ТВ	+	unsigned char	0-2°C, 1-3°C	
70	Исп.тх	ТВ	+	unsigned char	0-Не исп., 1-догов., 2-изм.в данном ТВ, 3-изм. в другом ТВ	
71	Исп.Рх	ТВ	+	unsigned char	0-Не изм., 1-Изм.	
72	Контр.R	ТВ	+	unsigned char	0-нет, 1-есть	
73	Исп.Qсум	ТВ	+	unsigned char	0-есть, 1-нет, 2-сумма, 3-разность	
74	Резерв 33	-	-	---	---	
75	Резерв 34	-	-	---	---	
76	Резерв 35	-	-	---	---	
77	Резерв 36	-	-	---	---	
78	Резерв 37	-	-	---	---	
79	Резерв 38	-	-	---	---	
80	Резерв 39	-	-	---	---	
81	Резерв 30	-	-	---	---	
82	Назнач. ДП	-	-	unsigned char	0-нет, 1-контр.сети, 2-счет имп., 3-сигн-ия	
83	Уровень	-	-	unsigned char	0-НР, 1-НЗ	
84	Ед.изм.	-	-	unsigned char	0-мЗ, 1кВт*ч	
85	Вес имп.	-	-	float		
86	ДПmin	-	-	float		
87	ДПmax	-	-	float		
88	Т подт.	-	-	float		
89	Резерв 41	-	-	---	---	
90	Резерв 42	-	-	---	---	
91	Резерв 43	-	-	---	---	
92	Резерв 44	-	-	---	---	
93	Резерв 45	-	-	---	---	
94	Резерв 46	-	-	---	---	
95	Резерв 47	-	-	---	---	
96	Резерв 48	-	-	---	---	
97	Резерв 49	-	-	---	---	
98	Резерв 40	-	-	---	---	
99	ДВ Уровень	ДВ	-	unsigned char	0-НР, 1-НЗ	
100	ДВ Тподтв.	ДВ	-	unsigned char		

4.3.7.2 Интерпретация архива событий

Архив событий предназначен для фиксации таких событий как сброс архива, переключение активной БД, переход на летнее/зимнее время и т.д.. Для событий, у которых в графе ID ключа установлен знак «+», поле «старое значение» содержит идентификатор специализированного аппаратного обеспечения, выполнившего авторизацию. ID имеет тип unsigned long.

№	Событие	ID ключа	Примечание
0	Запрет калибр. (ПК)	+	
1	Запись настроек (ПК)		
2	Уст. СОМ'ов на умолч.		
3	Сброс архива		
4	Резерв		
5	Калибр.т(ПК)	+	
6	Калибр.Р(ПК)	+	
7	Калибр.т		
8	Калибр.Р		
9	Поверка (Старт, ПК)		
10	Поверка (Стоп, ПК)		
11	Поверка (Старт)		
12	Поверка (Стоп)		
13	Уск. режим (старт)		
14	Уск. режим (стоп)		
15	Доступ запр.		
16	Доступ разр.		
17	Калибр запр.		
18	Калибр.разр.		
19	Уст. акт. БД1		
20	Уст. акт. БД2		
21	Уст. акт. БД1 (авто)		
22	Уст. акт. БД2 (авто)		
23	Уст. акт. БД1 (ПК)		
24	Уст. акт. БД2 (ПК)		
25	Запись незащ. настр. (ПК)		
26	Запись дан. производит.	+	
27	Запись дан. серв.-центра	+	
28	Авто перевод часов		
29	Очистка архивов	+	
30	Замена батареи		
31	Запись алармов		

4.3.7.3 Интерпретация диагностического архива

Архив событий предназначен для фиксации таких событий как рестарт процессора, отключения внешнего питания и т.д..

№	Событие	Примечание
0	Рестарт	Поле «старое значение» содержит причину перезапуска (тип unsigned long).
1	Инициализация	
2	Вкл. 220В	
3	Откл. 220В	
4	LB	
5	BR	
7	BB	

4.4 Текущие значения

Структура предназначена для чтения текущих итоговых параметров (интегралы параметров теплотребления от сброса архива до текущего момента).

4.4.1 Текущие итоговые значения

Параметр	ПВ	Адрес	Тип	Доступ	Примечание
Дата: день/месяц	≥2.20	7560	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
	<2.20	3412			
Дата: год/час	≥2.20	7561	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
	<2.20	3413			
Дата: минута/ секунда	≥2.20	7562	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
	<2.20	3414			
V	≥2.20	7563+(Nтр*8)+0	double	R/O	(м³)
	<2.20	3415+(Nтр*8)+0			
M	≥2.20	7563+(Nтр*8)+4	double	R/O	(т)
	<2.20	3415+(Nтр*8)+4			
dM	≥2.20	7619+(NТВ*23)+0	double	R/O	(т)
	<2.20	3463+(NТВ*23)+0			
Qтв	≥2.20	7619+(NТВ*23)+4	double	R/O	(ГДж)
	<2.20	3463+(NТВ*23)+4			
Q12	≥2.20	7619+(NТВ*23)+8	double	R/O	(ГДж)
	<2.20	3463+(NТВ*23)+8			
Qг	≥2.20	7619+(NТВ*23)+12	double	R/O	(ГДж)
	<2.20	3463+(NТВ*23)+12			
BHP	≥2.20	7619+(NТВ*23)+16	unsigned short	R/O	(ч)
	<2.20	3463+(NТВ*23)+16			
BOC	≥2.20	7619+(NТВ*23)+17	unsigned short	R/O	(ч)
	<2.20	3463+(NТВ*23)+17			
TVmin	≥2.20	7619+(NТВ*23)+18	unsigned short	R/O	(ч)
	<2.20	3463+(NТВ*23)+18			
TVmax	≥2.20	7619+(NТВ*23)+19	unsigned short	R/O	(ч)
	<2.20	3463+(NТВ*23)+19			
Tdt	≥2.20	7619+(NТВ*23)+20	unsigned short	R/O	(ч)
	<2.20	3463+(NТВ*23)+20			
Тбез.пит.	≥2.20	7619+(NТВ*23)+21	unsigned short	R/O	(ч)
	<2.20	3463+(NТВ*23)+21			
Ттнеиспр.	≥2.20	7619+(NТВ*23)+22	unsigned short	R/O	(ч)
	<2.20	3463+(NТВ*23)+22			
ДП	≥2.20	7688	double	R/O	
	<2.20	3509			
Длит. работы по сети	≥2.20	7692	unsigned long	R/O	(мин.)
	<2.20	3513			
Длит. работы дисплея	≥2.20	7694	unsigned long	R/O	(мин.)
	<2.20	3515			
Длит. отсут. сет. питания	≥2.20	7696	unsigned long	R/O	(мин.)
	<2.20	3517			
Активная БД	≥2.20	7698+(NТВ*2)+0	bit field	R/O	Биты 0-7. Принимает значения: • 0-активна БД1; • 1-активна БД2
	<2.20	3519+(NТВ*2)+0			
СИ	≥2.20	7698+(NТВ*2)+0	bit field	R/O	Биты 8-15. значение конфигурационного параметра СИ
	<2.20	3519+(NТВ*2)+0			
КТЗ	≥2.20	7698+(NТВ*2)+1	bit field	R/O	Биты 0-7. значение конфигурационного параметра КТЗ
	<2.20	3519+(NТВ*2)+1			
ФРТ	≥2.20	7698+(NТВ*2)+1	bit field	R/O	Биты 8-15. значение конфигурационного параметра ФРТ
	<2.20	3519+(NТВ*2)+1			
Признаки событий	≥2.20	7704	unsigned short	R/O	Битовая маска. Принимает значения: • Бит 0 – нажатие кнопки «Доступ»; • Бит 1 – отсутствие сетевого питания; • Бит 2 – низкое напряжение батареи; • Бит 3 – перезапуск процессора; • Бит 4 – ускоренный режим; • Бит 5 – ресурс батареи близок к окончанию; • Бит 6 – перевод часов; • Бит 7 – включение подсветки дисплея; • Бит 8 – нажатие клавиатуры; • Бит 9 – обращение по сети; • Бит 10 – сигнализация;
	<2.20	3523			

					<ul style="list-style-type: none"> • Бит 11 – изменение настроек (было изменение настроек, приводящее к изменению контрольной суммы); • Бит 12 – калибровки разрешены; • Бит 13 – использование SD карты; • Бит 14 – использование USB; • Бит 15 – депассивация батареи (технологическая операция поддержания батареи в рабочем состоянии).
КСН	≥2.20	7705	unsigned short	R/O	Только для ПВ≥2.00
	<2.20	3524			
Исп. tx	≥2.20	7706+(НТВ*1)+0	bit field	R/O	Биты 0-7. Значение конфигурационного параметра «Исп.tx» Только для ПВ≥2.00
	<2.20	3525+(НТВ*1)+0			
Исп. тнв	≥2.20	7706+(НТВ*1)+0	bit field	R/O	Биты 8-15. Значение конфигурационного параметра «Исп.тнв» Только для ПВ≥2.00
	<2.20	3525+(НТВ*1)+0			
Исп. Qсум	≥2.20	7709	unsigned short	R/O	Битовая маска. <ul style="list-style-type: none"> • Биты 0-1 - значение по ТВ1 • Биты 2-3 - значение по ТВ2 • Биты 4-5 - значение по ТВ3 Хранит значение конфигурационного параметра «Исп. Qсум» Только для ПВ≥2.00
	<2.20	3527			
Резерв	≥2.20	7710	unsigned short	R/O	Только для ПВ≥2.00
	<2.20	3528			
Тпустой трубы	≥2.20	7711+(НТВ*2)+0	unsigned short	R/O	(ч). Только для ПВ≥2.00
	<2.20	---			
Треверса	≥2.20	7711+(НТВ*2)+1	unsigned short	R/O	(ч). Только для ПВ≥2.00
	<2.20	---			
Qсум	≥2.20	7717	double	R/O	(ГДж). Только для ПВ≥2.00
	<2.20	3529			

Примечание:

- запись вида «---» означает, что параметр отсутствует
- Нтр означает номер трубы и изменяется в диапазоне 0÷6 для ПВ≥2.20 и 0÷5 для ПВ<2.20
- Нтв означает номер теплового ввода и изменяется в диапазоне 0÷2 для ПВ≥2.20 и 0÷1 для ПВ<2.20
- при работе с приборами, имеющими ПВ≥2.20 возможно чтение параметров, размещенных по адресам для ПВ<2.20. В этом случае, в целях сохранения максимальной совместимости, в процессе чтения производится преобразование структур данных ПВ≥2.20 к структурам ПВ<2.20, что может приводить к потерям значимой информации

4.4.2 Текущие мгновенные значения

Параметр	ПВ	Адрес	Тип	Доступ	Примечание
Дата: день/месяц	≥2.20	7816	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
	<2.20	3540			
Дата: год/час	≥2.20	7817	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
	<2.20	3541			
Дата: минута/секунда	≥2.20	7818	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
	<2.20	3542			
t	≥2.20	7819+(Nтр*2)+0	float	R/O	(°C)
	<2.20	3543+(Nтр*2)+0			
P	≥2.20	7833+(Nтр*2)+0	float	R/O	(МПа)
	<2.20	3555+(Nтр*2)+0			
Go	≥2.20	7847+(Nтр*2)+0	float	R/O	(м³)
	<2.20	3567+(Nтр*2)+0			
Gм	≥2.20	7861+(Nтр*2)+0	float	R/O	(т/ч)
	<2.20	3579+(Nтр*2)+0			
Ф	≥2.20	7875+(Nтр*2)+0	float	R/O	(ГДж/ч)
	<2.20	3591+(Nтр*2)+0			
h	≥2.20	7889+(Nтр*2)+0	float	R/O	(ГДж/т)
	<2.20	3603+(Nтр*2)+0			
Фтв	≥2.20	7903+(Nтв*2)+0	float	R/O	(ГДж/ч)
	<2.20	3615+(Nтв*2)+0			
hx	≥2.20	7909+(Nтв*2)+0	float	R/O	(ГДж/т)
	<2.20	3619+(Nтв*2)+0			
ДП	≥2.20	7915	float	R/O	
	<2.20	3623			
НС по трубе	≥2.20	7917	unsigned char[7]	R/O	НС по трубам 1-7(6 для ПВ<2.20) Каждый байт представляет битовую маску НС по одной трубе. Принимает значения: • 0-НС отсутствует; • 1-НС присутствует Порядок бит: • Бит 0 – НС t<min; • Бит 1 – НС t>max; • Бит 2 – НС неиспр. датчика t; • Бит 3 – НС P<min; • Бит 4 – НС P>max; • Бит 5 – НС V<min; • Бит 6 – НС V>max; • Бит 7 – НС неиспр. или отсут. питания ВС
	<2.20	3625	unsigned char[6]		
ДВ	≥2.20	7920	bit field	R/O	Биты 8-15. Битовая маска состояний двухпозиционных входов. Порядок бит: • Бит 8 – ДВ1; • Бит 9 – ДВ2; • Бит 10 – ДВ3; • Бит 11 – ДВ4; • Биты 12-15 – не исп-ся;
	<2.20	---			
НС по ТВ	≥2.20	7921	unsigned short[3]	R/O	НС по ТВ 1-3(2 для ПВ<2.20) Каждый байт представляет битовую маску НС по одному ТВ. Принимает значения: • 0-НС отсутствует; • 1-НС присутствует Порядок бит: • Бит 0 – НС по dt; • Бит 1 – НС по dM; • Бит 2 – НС по Qтв; • Бит 3 – НС tx<min; • Бит 4 – НС tx>max; • Бит 5 – НС неиспр. датчика tx; • Бит 6 – НС тнв<min; • Бит 7 – НС тнв>max; • Бит 8 – НС неиспр. датчика тнв • Бит 9 – НС по Q12; • Бит 10 – НС по Qг; • Бит 11 – НС по Pх<min; • Бит 12 – НС по Pх>max
	<2.20	3628	unsigned short[2]		

НС по ДП	≥2.20	7924	bit field	R/O	Биты 0-7, Наличие НС Битовая маска. Принимает значения: • 0-НС отсутствует; • 1-НС присутствует Порядок бит: • 0 – НС по ДП<min; • 1 – НС по ДП>max Старший байт, не исп-ся
	<2.20	3630			
Резерв	≥2.20	7924	bit field	R/O	Биты 8-15
	<2.20	3630			
Признаки событий	≥2.20	7925	unsigned short	R/O	Битовая маска. Принимает значения: • Бит 0 – нажатие кнопки «Доступ»; • Бит 1 – отсутствие сетевого питания; • Бит 2 – низкое напряжение батареи; • Бит 3 – перезапуск процессора; • Бит 4 – ускоренный режим; • Бит 5 – ресурс батареи близок к окончанию; • Бит 6 – перевод часов; • Бит 7 – включение подсветки дисплея; • Бит 8 – нажатие клавиатуры; • Бит 9 – обращение по сети; • Бит 10 – сигнализация; • Бит 11 – изменение настроек (было изменение настроек, приводящее к изменению контрольной суммы); • Бит 12 – калибровки разрешены; • Бит 13 – использование SD карты; • Бит 14 – использование USB; • Бит 15 – депассивация батареи (технологическая операция поддержания батареи в рабочем состоянии).
	<2.20	3631			
Резерв	≥2.20	7926	unsigned short	R/O	
	<2.20	3632			
tx	≥2.20	$7927+(N_{ТВ} \cdot 2)+0$	float	R/O	(°C)
	<2.20	$3633+(N_{ТВ} \cdot 2)+0$			
Px	≥2.20	$7933+(N_{ТВ} \cdot 2)+0$	float	R/O	(МПа)
	<2.20	$3637+(N_{ТВ} \cdot 2)+0$			
dt	≥2.20	$7939+(N_{ТВ} \cdot 2)+0$	float	R/O	(°C)
	<2.20	$3641+(N_{ТВ} \cdot 2)+0$			
tнв	≥2.20	$7945+(N_{ТВ} \cdot 2)+0$	float	R/O	(°C)
	<2.20	$3645+(N_{ТВ} \cdot 2)+0$			
Признаки пустой трубы	≥2.20	7951	bit field	R/O	Биты 0-7. Битовая маска. Номер бита означает номер трубы. Принимает значения: • 0-отсутствует; • 1- присутствует
	<2.20	---			
Признаки реверса по трубам	≥2.20	7951	bit field	R/O	Биты 8-15. Битовая маска. Номер бита означает номер трубы. Принимает значения: • 0-отсутствует; • 1- присутствует
	<2.20	---			
Ситуация реверса по трубам	≥2.20	7952	bit field	R/O	Биты 0-7. Битовая маска. Номер бита означает номер трубы. Принимает значения: • 0-отсутствует; • 1- присутствует
	<2.20	---			
Резерв	≥2.20	7952	bit field	R/O	
	<2.20	---			
Активная БД	≥2.20	7953	bit field	R/O	Бит 0. Принимает значения: • 0-активна БД1; • 1- активна БД2
	<2.20	3649			
Резерв	≥2.20	7953	bit field	R/O	Биты 1-15
	<2.20	3649			
Примечание:					
- запись вида «---» означает, что параметр отсутствует					
- Nтр означает номер трубы и изменяется в диапазоне 0÷6 для ПВ≥2.20 и 0÷5 для ПВ<2.20					
- Nтв означает номер теплового ввода и изменяется в диапазоне 0÷2 для ПВ≥2.20 и 0÷1 для ПВ<2.20					

- при работе с приборами, имеющими ПВ \geq 2.20 возможно чтение параметров, размещенных по адресам для ПВ $<$ 2.20. В этом случае, в целях сохранения максимальной совместимости, в процессе чтения производится преобразование структур данных ПВ \geq 2.20 к структурам ПВ $<$ 2.20, что может приводить к потерям значимой информации

4.5 Сервисная команда

Структура предназначена для выполнения сервисных операций. Для всех команд, кроме команд установки активной базы данных, изменение структуры возможно только при нажатой кнопке «Доступ».

Параметр	Адрес	Тип	Доступ	Примечание
Команда	2665	unsigned short	R/W*	Принимает значения: <ul style="list-style-type: none"> • 1–старт записи настроек; • 2–окончание записи настроек; • 7-установить активной БД1; • 8-установить активной БД2; • 7-установить дату время; Только в ПВ \geq 2.00
Дата: день, месяц	2666	unsigned short	R/W*	Младший байт, день: 1-31 Старший байт, месяц: 1-12 Используется только с командой «Установить дату время»
Дата: год-2000, час	2667	unsigned short	R/W*	Младший байт, год минус 2000 Старший байт, час: 0-23 Используется только с командой «Установить дату время»
Время: минута, секунда	2668	unsigned short	R/W*	Младший байт, минута: 0-59 Старший байт, секунда: 0-59 Используется только с командой «Установить дату время»

До выполнения команд «Установить активной БД» необходимо произвести запись пароля смены активной базы данных по адресу структуры «Тип читаемых данных». Пароль представляет собой массив из 5-ти цифровых символов с завершающим двоичным нулем (6-ой по счету байт). Если длина установленного пароля менее 5-ти символов, то оставшиеся символы заполняются символом пробел (код 0x20). Например, пароль «123» должен быть представлен в виде массива байт: 0x31, 0x32, 0x33, 0x20, 0x20, 0x00.

4.6 Структура технологического изменения параметров

Структура предназначена для изменения различных параметров вычислителя. Изменение параметра выполняется при помощи операции записи регистров. При этом указывается условный номер изменяемого параметра и новое значение параметра. Уровень доступа к изменяемому параметру определяется типом параметра.

Параметр	Адрес	Тип	Доступ	Примечание
Номер параметра	3796	unsigned short	R/W*	Принимает значения: <ul style="list-style-type: none"> • 0 – коррекция времени. Поле «Значение параметра» содержит величину коррекции времени. Допустимый интервал ± 900 секунд. Если текущее время прибора находится в пределах ЧЧ:20:00 \div ЧЧ:40:00, то коррекция времени производится немедленно. В противном случае, коррекция времени выполняется после следующего достижения времени ЧЧ:20:00; • 1 – тип использования дополнительного импульсного входа. Значение в соответствии с настроечными параметрами; • 2, 3, 4-значение txd для ТВ1, ТВ2 или ТВ3 соответственно. До изменения значений необходимо произвести запись пароля смены активной базы данных по адресу структуры «Тип читаемых данных». Пароль представляет собой массив из 5-ти цифровых символов с завершающим двоичным нулем (6-ой по счету байт). Если длина установленного пароля менее 5-ти символов, то оставшиеся символы заполняются символом пробел (код 0x20). Например, пароль «123» должен быть представлен в виде массива байт: 0x31, 0x32, 0x33, 0x20, 0x20, 0x00.
Значение параметра	3797	float	R/W*	
Резерв	3799	unsigned char[4]	R/W*	

4.7 Телеметрия

Структура предназначена для чтения информации, получаемой прибором по цифровому каналу от расходомеров Питерфлоу.

ПРИМЕЧАНИЕ! Функционал телеметрии реализуется только в ПВ \geq 2.20

Параметр	Адрес	Тип	Доступ	Примечание
Резерв	4213+(Ng*49)+00	unsigned long	R/O	
Резерв	4213+(Ng*49)+02	unsigned long	R/O	
Диагностика	4213+(Ng*49)+04	unsigned long	R/O	<p>Битовая маска событий в порядке возрастания номера битов:</p> <ul style="list-style-type: none"> • E0 • A1 • E2 • F3 • E4 • E5 • E6 • A7 • F8 • A9 • A10 • E11 • E12 • E13 • A14 • A15 • A16 • A17 • A18 • F19 • F20 • F21 • A22 • A23 • A24 • A25 • A26 • A27 • F28 • A29 <p>См. документацию на расходомер</p>
Серийный номер	4213+(Ng*49)+06	unsigned long	R/O	Серийный номер расходомера
Время раб.	4213+(Ng*49)+08	unsigned long	R/O	(мин.)
Время раб. с ош.	4213+(Ng*49)+10	unsigned long	R/O	(мин.)
F1	4213+(Ng*49)+12	unsigned short	R/O	<p>Принимает значения:</p> <ul style="list-style-type: none"> • 96 оба направления потока, импульсы 0 (ключ замкнут) • 224 оба направления потока, импульсы 1 (ключ разомкнут) • 32 прямое направление потока, импульсы 0 • 160 прямое направление потока, импульсы 1 • 64 обратное направление потока, импульсы 0 • 192 обратное направление потока, импульсы 1 • 1 компаратор: выход=0 если расход больше уставки • 129 компаратор: выход=1 если расход больше уставки • 2 диагностика: выход=0 при событиях (по ИЛИ) • 130 диагностика: выход=1 при событиях (по ИЛИ) • 3 телеметрия (передача данных в цифровой форме)
F2	4213+(Ng*49)+13	unsigned short	R/O	См. F1
ПО	4213+(Ng*49)+14	unsigned short	R/O	Старший байт - версия; Младший байт - редакция.
CRC прошивки	4213+(Ng*49)+15	unsigned short	R/O	Индексировать в шестнадцатеричном виде
ДУ	4213+(Ng*49)+16	unsigned short	R/O	(мм)
Gmax	4213+(Ng*49)+17	unsigned short	R/O	(м ³ /ч)
Класс	4213+(Ng*49)+18	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 «Kxxx», где «xxx» - десятичное значение старшего байта.
Тип прибора	4213+(Ng*49)+19	unsigned short	R/O	Принимает значения:

				<ul style="list-style-type: none"> • 0x1701 - Питерфлоу РС; • 0x1707 - Питерфлоу К; • 0x1708 - Питерфлоу СВ; • 0x1709 - Питерфлоу М
Резерв	4213+(Ng*49)+20	unsigned short	R/O	
Параметры индик.	4213+(Ng*49)+21	bit field	R/O	Биты 0-3 - кол-во знаков после запятой для «Кэф.А» Биты 4-7 - кол-во знаков после запятой для «Кэф.В» Биты 8-11 - кол-во знаков после запятой для «V+» и «V-» Бит 15: 0 - нет модуля архивации, 1- есть модуль архивации
V+	4213+(Ng*49)+22	double	R/O	(м ³)
V-	4213+(Ng*49)+26	double	R/O	(м ³)
Вес имп.	4213+(Ng*49)+30	float	R/O	(л)
Расход	4213+(Ng*49)+32	float	R/O	(м ³ /ч)
Кэф.А	4213+(Ng*49)+34	float	R/O	
Кэф.В	4213+(Ng*49)+36	float	R/O	
Сопр.воды	4213+(Ng*49)+38	float	R/O	(КОМ)
Дата/время:день/месяц	4213+(Ng*49)+40	unsigned short	R/O	Младший байт, день: 1-31 Старший байт, месяц: 1-12
Дата/время:год/час	4213+(Ng*49)+41	unsigned short	R/O	Младший байт, год минус 2000 Старший байт, час: 0-23
Дата/время: минута/секунда	4213+(Ng*49)+42	unsigned short	R/O	Младший байт, минута: 0-59 Старший байт, секунда: 0-59
Ожидание	4213+(Ng*49)+43	unsigned short	R/O	Время с прихода последней достоверной телеграммы (с)
Кол-во телеграмм	4213+(Ng*49)+44	unsigned long	R/O	Общее кол-во принятых телеграмм
Кол-во недостоверных телеграмм	4213+(Ng*49)+46	unsigned long	R/O	Общее кол-во недостоверных телеграмм
Нет связи	4213+(Ng*49)+48	bit field	R/O	Биты 0-7. Принимает значения: 0 - связь с расходомером есть; 1 - связи с расходомером нет;
Резерв	4213+(Ng*49)+48	bit field	R/O	Биты 8-15
Примечание:				
- Ng означает номер канала измерения расхода				

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;

bprt - 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 (не используется);
 - 6 - запрос не может быть обработан сейчас, необходимо повторить запрос позднее;
 - 7 – (не используется);
 - 8 - (не используется);
 - 9 – устройство не готово;
 - 10 - при чтении запрошено больше регистров, чем допустимо;
 - 11 - сделана попытка записи большего количества регистров, чем допустимо;
 - 12 - недопустимый начальный адрес;
 - 13 - недопустимый конечный адрес (начальный адрес + количество регистров);
 - 14 - адрес доступен только для чтения;
 - 15 - доступ запрещен (защита, например кнопка ДОСТУП);
 - 16 - другая ошибка при обработке запроса;
-
- 130 – Ошибка выполнения;
 - 132 - Указанная дата вне диапазона архива прибора;
 - 133 - Нет данных за указанную дату;

Приложение 5. Описание структуры файла tvb

Файл данных формата tvb может быть получен следующими способами:

- путем считывания данных из ТВ7 на устройство передачи данных «USB-ППД»;
- путем считывания данных из ТВ7 на планшетный компьютер с ПО производства «ЗАО Термотроник» для ОС Android;
- путем считывания данных из ТВ7 на SD карту.
- Файл состоит из множества последовательно расположенных сегментов данных. Каждый сегмент данных содержит:
- описатель данных;
- один или несколько однотипных элементов данных.

Чтение и интерпретация содержимого файла должна выполняться в последовательности:

1. Чтение из файла описателя данных фиксированной длины;
2. Чтение из файла элементов данных. Заключается в чтении из файла переменного количества байт, в зависимости от длины и количества элементов данных, указанных в описателе данных. Полученные данные интерпретируются в соответствии с типом данных, указанным в описателе. В зависимости от типа данных принимается решения о продолжении чтения (переход к выполнению п.1) или окончании операции.

Описатель данных имеет следующую структуру:

Поле	Тип	Размер (байт)	Примечание
Тип	целое	4	Определяет тип хранимых данных
Длина	целое	4	Определяет длину одного элемента данных (байт)
Количество	целое	4	Определяет количество элементов данных (*)
Доп. информация	целое	4	Описывается контекстно в зависимости от поля «тип»

(*) – для сегмента с полем «Тип» равным 0 поле «Количество» анализировать не нужно (количество равно 1).

Если иное не оговорено специально, то каждый элемент данных содержит структуру одного из типов, описанных выше (например, структуру «Информация об устройстве», «Архивная запись» и т.д.). Если поле «Длина» содержит значение, большее, чем суммарная длина структуры, описанная в соответствующем разделе, то для каждого элемента данных необходимо прочитать из файла количество байт, равное значению поля «Длина», а интерпретировать следует только поля, указанные в описании соответствующей структуры данных. Например, в описателе указано, что элемент данных представлен структурой «Информация об устройстве», длина 38 байт. Описание структуры «Информация об устройстве» содержит описание элементов суммарной длиной 14 байт. В этом случае из файла необходимо прочитать 38 байт, а интерпретировать только первые 14 из них.

Поле «Тип» кодирует следующие типы элементов данных:

- 0 – начало файла. Элемент данных должен содержать строку «START TV7 DATA FILE\r\n», где символы «\r\n» - это символы с кодами 0x0D и 0x0A соответственно. Поле «Количество» анализировать не нужно;
- 1 – элемент данных представлен структурой «Информация об устройстве»;
- 2 – элемент данных представлен структурой «Информация о настройках»;
- 3 – элемент данных представлен структурой «Настройки защищенные» для ПВ<2.20;
- 4 – Элемент данных содержит значение контрольной суммы незащищенных настроек. Остальные байты анализировать не нужно;
- 5 – элемент данных представлен структурой «Настройки незащищенные»;
- 6 – анализировать не нужно;
- 7 - элемент данных представлен структурой «Информация о датах начала/конца архивов»;
- 8 - элемент данных представлен структурой «Информация об индексах архивов параметров»;
- 9 - элемент данных представлен структурой «Информация об асинхронных архивах»;
- 10 - элемент данных представлен структурой «Текущие значения» для ПВ<2.20;
- 11 - элемент данных представлен структурой «Текущие итоговые значения» для ПВ<2.20;
- 12, 13, 14 - элементы данных представлены структурой «Архивная запись (часовой, суточный, месячный архивы)» для ПВ<2.20. Тип 12 - записи представляют часовой архив, 13 – суточный архив, 14 - месячный архив;
- 15 - элемент данных представлен структурой «Архивная запись (итоговый архив)» для ПВ<2.20;
- 16, 17, 18 - элементы данных представлены структурой «Архивные записи асинхронных архивов». Тип 16 - записи представляют архив изменений базы данных, 17 - архив событий, 18 - диагностический архив. Поле «Доп.информация» содержит индекс архивной записи в приборе, из которой была считана первая архивная запись данного сегмента данных;
- 19 – элемент данных представлен структурой «Настройки защищенные» для ПВ≥2.20;
- 20 - элемент данных представлен структурой «Текущие значения» для ПВ≥2.20;

- 21 - элемент данных представлен структурой «Текущие итоговые значения» для $PВ \geq 2.20$;
- 22, 23, 24 - элементы данных представлены структурой «Архивная запись (часовой, суточный, месячный архивы)» для $PВ \geq 2.20$. Тип 22 - записи представляют часовой архив, 23 – суточный архив, 24 - месячный архив;
- 25 - элемент данных представлен структурой «Архивная запись (итоговый архив)» для $PВ \geq 2.20$;
- 26, 27, 28 - элементы данных представлены структурой «Архивные записи асинхронных архивов». Тип 26 - записи представляют архив изменений базы данных, 27 - архив событий, 28 - диагностический архив. Поле «Доп.информация» содержит индекс архивной записи в приборе, из которой была считана первая архивная запись данного сегмента данных;
- 65535 - файла. Элемент данных должен содержать строку «END TV7 DATA FILE\r\n», где символы «\r\n» - это символы с кодами 0x0D и 0x0A соответственно. Поле «Количество» анализировать не нужно. Данный элемент является признаком окончания файла и служит сигналом к прекращению чтения файла;

Приложение 6. Реализация обмена данными по интерфейсу USB

Отправляемый и принимаемый буфер при работе по USB всегда имеет длину 64 байта. Из них полезных данных 61 байт. При необходимости отправить большее количество байт, необходимо делать несколько последовательных посылок. При передаче нескольких посылок в направлении прибора между последовательными посылками должна выдерживаться пауза не менее 10 мс. Каждая посылка содержит:

- Заголовок длиной 2 байта: первый байт всегда равен 63; второй байт содержит длину секции пользовательских данных.
- Секцию пользовательских данных длиной 62 байта. Первый байт содержит счетчик полезных байт "ByteCnt". ByteCnt может принимать значения 0-61. Остальные байты используются для передачи «полезной» информации.

Если программа хочет отправить количество байт, кратное 61, то после отправки полезных данных она также обязана отправить дополнительный кадр с ByteCnt равным нулю. Принимающая программа принимает решение, что все данные приняты если ByteCnt имеет значение менее 61.

Исходные тексты примера работы с прибором по интерфейсу USB приведены ниже (проект компилировался в среде разработки Visual Studio 6.0 при предустановленном пакете Windows Driver Kit версии 7600.16385.1).

Листинг файла HIDDevice.h

```
// HIDDevice.h: interface for the CHIDDevice class.
//
//
////////////////////////////////////

#if !defined(AFX_HIDDEVICE_H_9C9236F9_16D8_4B6A_B3D1_416C79E14191__INCLUDED_)
#define AFX_HIDDEVICE_H_9C9236F9_16D8_4B6A_B3D1_416C79E14191__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

extern "C" {
#pragma warning (disable: 4201)
#include "DDK\\HIDSDI.h"
#pragma warning (default: 4201)
}

////////////////////////////////////
////////////////////////////////////
#define USB_VENDOR      0x2047          //TI MSP430 vendor ID
#define USB_PRODUCT     0x0301          //ID for this product (HID MSP430)
////////////////////////////////////
////////////////////////////////////

#define CONST_REPORT_ID 63

//!! Defines the maximum length of a serial number
#define SERNUM_LEN 40
//!! Defines the maximum number of physical devices
#define MAX_PHYS_DEVICES 6

#define HID_DEV_CONNECTABLE_MAX 128
#define HID_DEV_INTERFACE_CNT 128

//!! HID Device return codes
//!! HID action/transfer was successful
#define HID_DEVICE_SUCCESS          0x00
//!! HID device was not found
#define HID_DEVICE_NOT_FOUND        0x01
//!! HID device is not opened
#define HID_DEVICE_NOT_OPENED       0x02
//!! HID device is allready opened
#define HID_DEVICE_ALREADY_OPENED   0x03
//!! Timeout occurs during transfer
#define HID_DEVICE_TRANSFER_TIMEOUT 0x04
//!! HID transfer failed
#define HID_DEVICE_TRANSFER_FAILED   0x05

//!! Invalid handle
#define HID_DEVICE_HANDLE_ERROR      0x06
```

```
//! Buffer overflow
#define HID_DEVICE_BUFFER_OVERFLOW      0x07
//! User interrupt
#define HID_DEVICE_USER_INTERRUPT      0x08
//! Unknown error
#define HID_DEVICE_UNKNOWN_ERROR      0xFF

//*****
//
//! A structure that tracks the number of serial numbers
//
//*****
typedef struct
{
    char serialNum[SERNUM_LEN];
    char devicePath[MAX_PATH];
    BYTE deviceStatus;
}TTrackSerialNumbers;

typedef struct
{
    TTrackSerialNumbers m_serialNumList[MAX_PHYS_DEVICES];
    WORD m_Cnt;
}TTrackSerialNumbersAr;

//*****
//
//! Device information structure.
//
//*****
typedef struct
{
    //! Handle for hid device
    HANDLE hndHidDevice;
    //! Indicator if device is opened
    BOOL bDeviceOpen;
    //! Timeout for GetReport requests
    UINT uGetReportTimeout;
    //! Timeout for SetReport requests
    UINT uSetReportTimeout;
    //! Asynchronous I/O structure
    OVERLAPPED oRead;
    //! Asynchronous I/O structure
    OVERLAPPED oWrite;
    //! Maximum length of InReport's
    WORD wInReportBufferLength;
    //! Maximum length of OutReport's
    WORD wOutReportBufferLength;
}THidDevice;

#define MAX_IN_BUF_SIZE 256

BYTE HID_ReadFile (THidDevice * pHidDevice, BYTE * pBuffer, WORD bufferSize, WORD * pnBytesRead, ULONG msTout);
BYTE HID_WriteFile(THidDevice * pHidDevice, BYTE* buffer, DWORD bufferSize);
```

```

void HID_Init      (THidDevice * pHidDevice);
void HID_UnInit   (THidDevice * pHidDevice);

BYTE HID_Open (THidDevice * pHidDevice, WORD vid, WORD pid, TTrackSerialNumbers * pTrackSerialNumbers);
BYTE HID_Close(THidDevice * pHidDevice);

HANDLE HID_OpenDeviceByVidPid(char* devicePath, WORD vid, WORD pid, DWORD * err);
BYTE HID_GetHidDevicePath(WORD VID, WORD PID, DWORD index, char* devicePath);
DWORD HID_GetSerNums(WORD vid, WORD pid, TTrackSerialNumbersAr * pSerialNumListAr);

#endif // !defined(AFX_HIDDEVICE_H__9C9236F9_16D8_4B6A_B3D1_416C79E14191__INCLUDED_)

```

Листинг файла HIDDevice.cpp

```

// HIDDevice.cpp: implementation of the CHIDDevice class.
//
//
#include "stdafx.h"
#include "HIDDevice.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

void MyAddLogString(CString str, int wait);
void MyOnCharsRecieved(BYTE * pBuf, int len);
void MyOnCharsTransmitted(BYTE * pBuf, int len);

//
// Construction/Destruction
//

DWORD HID_GetSerNums(WORD vid, WORD pid, TTrackSerialNumbersAr * pSerialNumListAr)
{
    wchar_t HID_Serial_Number[SERNUM_LEN];
    DWORD i=0, error=0;

    //
    for(i= 0;i<MAX_PHYS_DEVICES;i++) {
        memset(&pSerialNumListAr->m_serialNumList[i].serialNum, 1, sizeof(pSerialNumListAr->m_serialNumList[i].serialNum));
        pSerialNumListAr->m_serialNumList[i].deviceStatus = HID_DEVICE_NOT_FOUND;
    }
    pSerialNumListAr->m_Cnt = 0;
    //

```



```

for(i=0;i<HID_DEV_CONNECTABLE_MAX;i++) {
    if(HID_GetHidDevicePath(vid, pid, i, pSerialNumListAr->m_serialNumList[pSerialNumListAr->m_Cnt].devicePath) == HID_DEVICE_SUCCESS) {
        HANDLE hHidDeviceHandle = HID_OpenDeviceByVidPid(pSerialNumListAr->m_serialNumList[pSerialNumListAr->m_Cnt].devicePath, vid, pid, &error);

        if( (hHidDeviceHandle != INVALID_HANDLE_VALUE) || (error == ERROR_SHARING_VIOLATION) ) {
            if(HidD_GetSerialNumberString(hHidDeviceHandle,HID_Serial_Number,sizeof(HID_Serial_Number)))
            {
                CloseHandle(hHidDeviceHandle);

                if(pSerialNumListAr->m_Cnt < MAX_PHYS_DEVICES) {
                    pSerialNumListAr->m_serialNumList[pSerialNumListAr->m_Cnt].deviceStatus = HID_DEVICE_SUCCESS;
                    strcpy(pSerialNumListAr->m_serialNumList[pSerialNumListAr->m_Cnt].serialNum, CString(HID_Serial_Number));
                    pSerialNumListAr->m_Cnt++;
                } else {
                    break;
                }
            }
        }
    }
}

return pSerialNumListAr->m_Cnt;
}

BYTE HID_GetHidDevicePath(WORD vid, WORD pid, DWORD index, char *devicePath)
{
    BYTE status = HID_DEVICE_UNKNOWN_ERROR;

    GUID          hidGuid;
    HDEVINFO      hHidDeviceInfo = NULL;

    HidD_GetHidGuid(&hidGuid);

    hHidDeviceInfo = SetupDiGetClassDevs(&hidGuid, NULL, NULL, DIGCF_PRESENT | DIGCF_DEVICEINTERFACE);

    if(hHidDeviceInfo != NULL) {
        SP_DEVICE_INTERFACE_DATA hidDeviceInterfaceData;
        hidDeviceInterfaceData.cbSize = sizeof(hidDeviceInterfaceData);

        if(index < HID_DEV_CONNECTABLE_MAX) {
            BOOL hidResult = hidResult = SetupDiEnumDeviceInterfaces(hHidDeviceInfo, 0, &hidGuid, index, &hidDeviceInterfaceData);

            if(hidResult) {
                BOOL detailResult;
                DWORD length, required;
                PSP_DEVICE_INTERFACE_DETAIL_DATA hidDeviceInterfaceDetailData;

                SetupDiGetDeviceInterfaceDetail(hHidDeviceInfo, &hidDeviceInterfaceData, NULL, 0, &length, NULL);
                hidDeviceInterfaceDetailData = (PSP_DEVICE_INTERFACE_DETAIL_DATA)malloc(length);

                if(hidDeviceInterfaceDetailData) {
                    char strVidPid[40];
                    char strTmp[40];
                }
            }
        }
    }
}

```

```

        hidDeviceInterfaceDetailData->cbSize = sizeof(SP_DEVICE_INTERFACE_DETAIL_DATA);
        detailResult = SetupDiGetDeviceInterfaceDetail(hHidDeviceInfo, &hidDeviceInterfaceData, hidDeviceInterfaceDetailData, length,
&required, NULL);

        strcpy(strVidPid, "vid_");
        sprintf(strTmp, "%04x", vid);
        strcat(strVidPid, strTmp);

        strcat(strVidPid, "&pid_");
        sprintf(strTmp, "%04x", pid);
        strcat(strVidPid, strTmp);

        if(detailResult && ( strstr(hidDeviceInterfaceDetailData->DevicePath, strVidPid))) {
            strcpy(devicePath, hidDeviceInterfaceDetailData->DevicePath);
            status = HID_DEVICE_SUCCESS;
        }

        free(hidDeviceInterfaceDetailData);
    }
}

return status;
}

HANDLE HID_OpenDeviceByVidPid(char *devicePath, WORD vid, WORD pid, DWORD *err)
{
    HANDLE hHidDeviceHandle = INVALID_HANDLE_VALUE;
    int count = 0;

    while(count<2) {
        if(count == 0) {
            hHidDeviceHandle = CreateFile(devicePath, GENERIC_READ | GENERIC_WRITE, 0, 0, OPEN_EXISTING, FILE_FLAG_OVERLAPPED, 0);
        } else {
            hHidDeviceHandle = CreateFile(devicePath, GENERIC_READ | GENERIC_WRITE, FILE_SHARE_READ | FILE_SHARE_WRITE, NULL, OPEN_EXISTING,
FILE_FLAG_OVERLAPPED, NULL);
        }

        *err = GetLastError();

        if(*err == ERROR_SHARING_VIOLATION) {
            count ++;
        } else {
            break;
        }
    }

    if(hHidDeviceHandle != INVALID_HANDLE_VALUE) {
        HIDD_ATTRIBUTES hidDeviceAttributes;

        if(HidD_GetAttributes(hHidDeviceHandle, &hidDeviceAttributes)) {
            if( (hidDeviceAttributes.VendorID != vid) || (hidDeviceAttributes.ProductID != pid) ) {
                CloseHandle(hHidDeviceHandle);
            }
        }
    }
}

```

```

        hHidDeviceHandle = INVALID_HANDLE_VALUE;
    }
} else {
    CloseHandle(hHidDeviceHandle);
    hHidDeviceHandle = INVALID_HANDLE_VALUE;
}
}
return hHidDeviceHandle;
}
void HID_Init(THidDevice *pstrHidDevice)
{
    memset(pstrHidDevice, 0, sizeof(*pstrHidDevice));
    pstrHidDevice->hndHidDevice = NULL;
    pstrHidDevice->bDeviceOpen = FALSE;
    pstrHidDevice->uGetReportTimeout = 1500;
    pstrHidDevice->uSetReportTimeout = 1500;

    pstrHidDevice->oRead.hEvent = CreateEvent(NULL, FALSE, FALSE, NULL);
    pstrHidDevice->oWrite.hEvent = CreateEvent(NULL, FALSE, FALSE, NULL);
    pstrHidDevice->wInReportBufferLength = 0;
    pstrHidDevice->wOutReportBufferLength = 0;
}
void HID_UnInit(THidDevice *pstrHidDevice)
{
    if(pstrHidDevice->oRead.hEvent != NULL) {
        ResetEvent(pstrHidDevice->oRead.hEvent);
        CloseHandle(pstrHidDevice->oRead.hEvent);
    }
    if(pstrHidDevice->oWrite.hEvent != NULL) {
        ResetEvent(pstrHidDevice->oWrite.hEvent);
        CloseHandle(pstrHidDevice->oWrite.hEvent);
    }
}
BYTE HID_Close(THidDevice * pHidDevice)
{
    BYTE status = HID_DEVICE_SUCCESS;
    if(pHidDevice->bDeviceOpen == TRUE) {
        if( (pHidDevice->hndHidDevice != INVALID_HANDLE_VALUE) && (pHidDevice->hndHidDevice != NULL) ) {
            CloseHandle(pHidDevice->hndHidDevice);
            pHidDevice->hndHidDevice = INVALID_HANDLE_VALUE;
            pHidDevice->bDeviceOpen = FALSE;
        } else {
            status = HID_DEVICE_HANDLE_ERROR;
        }
    } else {
        status = HID_DEVICE_NOT_OPENED;
    }
    return status;
}
BYTE HID_Open(THidDevice * pHidDevice, WORD vid, WORD pid, TTrackSerialNumbers * pTrackSerialNumbers)
{
    HANDLE hHidDeviceHandle;
    wchar_t HID_Serial_Number[SERNUM_LEN];
    PHIDP_PREPARSED_DATA preparsedData;

```

```
OSVERSIONINFO osVer;
DWORD error = 0;
CString str;
HID_Close(pHidDevice);

if(pTrackSerialNumbers->deviceStatus != HID_DEVICE_SUCCESS) {
    MyAddLogString( "pTrackSerialNumbers->deviceStatus != HID_DEVICE_SUCCESS!", false );
    return HID_DEVICE_NOT_OPENED;
}

hHidDeviceHandle = HID_OpenDeviceByVidPid(pTrackSerialNumbers->devicePath, vid, pid, &error);

if(hHidDeviceHandle == INVALID_HANDLE_VALUE) {
    MyAddLogString( "hHidDeviceHandle == INVALID_HANDLE_VALUE", false );
    return HID_DEVICE_ALREADY_OPENED;
}

HidD_GetSerialNumberString(hHidDeviceHandle, HID_Serial_Number, sizeof(HID_Serial_Number));

if(strcmp(pTrackSerialNumbers->serialNum, CString(HID_Serial_Number)) != 0) {
    MyAddLogString( "strcmp(pTrackSerialNumbers->serialNum, CString(HID_Serial_Number)) != 0", false );
    return HID_DEVICE_ALREADY_OPENED;
}

pHidDevice->bDeviceOpen = TRUE;
pHidDevice->hndHidDevice = hHidDeviceHandle;

if(HidD_GetPreparsedData(hHidDeviceHandle, &preparsedData) {
    HIDP_CAPS capabilities;

    if(HidP_GetCaps(preparedData, &capabilities) {
        if(capabilities.InputReportByteLength) {
            pHidDevice->wInReportBufferLength = capabilities.InputReportByteLength;

            if (pHidDevice->wInReportBufferLength > MAX_IN_BUF_SIZE)
                pHidDevice->wInReportBufferLength = MAX_IN_BUF_SIZE;
        }

        if(capabilities.OutputReportByteLength) {
            pHidDevice->wOutReportBufferLength = capabilities.OutputReportByteLength;
        }
    }
    osVer.dwOSVersionInfoSize = sizeof(OSVERSIONINFO);
    GetVersionEx(&osVer);
    // Only set the max report requests if we are using 2K or later
    if((osVer.dwPlatformId == 2) && (osVer.dwMajorVersion == 5)) {
        if(osVer.dwMinorVersion >= 1) {
            // XP or later supports 512 input reports
            HidD_SetNumInputBuffers(pHidDevice->hndHidDevice, 512);
        } else if(osVer.dwMinorVersion == 0) {
            // 2K supports 200 input reports
            HidD_SetNumInputBuffers(pHidDevice->hndHidDevice, 200);
        }
    }
}
```

```

        HidD_FreePreparedData(preparedData);
        return HID_DEVICE_SUCCESS;
    }

    MyAddLogString( "return HID_DEVICE_ALREADY_OPENED", false );
    return HID_DEVICE_ALREADY_OPENED;
}

BYTE HID_WriteFile(THidDevice * pHidDevice, BYTE* buffer, DWORD bufferSize)
{
    BYTE status      = HID_DEVICE_SUCCESS, quit = 0;
    int  stillToSend = bufferSize;
    BYTE bytesCount  = 0;
    BYTE ReportId    = CONST_REPORT_ID;// ONLY 63ReportID is available!
    BYTE report[256];
    DWORD error = 0, fRes = 0;

    while((stillToSend >= 0) && (quit == 0) && (status == HID_DEVICE_SUCCESS)) {
        memset(report, 0, sizeof(report));

        if(stillToSend >= (ReportId-2)) {
            bytesCount = (BYTE)(ReportId-2);
        } else {
            bytesCount = (BYTE)stillToSend;
            quit++;
        }

        //fill header
        report[0] = ReportId;//CONST_REPORT_ID
        report[1] = (BYTE)(bytesCount + 1);//(BYTE)(ReportId-1);//Always transmit 63 bytes
        report[2] = bytesCount;//Data segment length

        //Copy data
        memcpy(&report[3], &buffer[bufferSize - stillToSend], bytesCount);

        stillToSend = (stillToSend - bytesCount);

        // Check to see that the device is opened
        if(pHidDevice->bDeviceOpen) {
            DWORD bytesWritten = 0;

            // Try to write the file
            if(!WriteFile(pHidDevice->hndHidDevice, report, pHidDevice->wOutReportBufferLength, &bytesWritten, &pHidDevice->oWrite)) {

                error = GetLastError();

                if(error == ERROR_IO_PENDING) {
                    fRes = WaitForSingleObject(pHidDevice->oWrite.hEvent, pHidDevice->uSetReportTimeout);
                    switch(fRes)
                    {
                        {
                            case WAIT_OBJECT_0:
                                {
                                    GetOverlappedResult(pHidDevice->hndHidDevice, &pHidDevice->oWrite, &bytesWritten, FALSE);
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
        break;
    case WAIT_TIMEOUT:
    {
        CancelIo(pHidDevice->hndHidDevice);
        status = HID_DEVICE_TRANSFER_TIMEOUT;
        break;
    }
    break;
    default:
    {
        CancelIo(pHidDevice->hndHidDevice);
        status = HID_DEVICE_TRANSFER_FAILED;
        break;
    }
    break;
}
} else {
    CancelIo(pHidDevice->hndHidDevice);
    status = HID_DEVICE_TRANSFER_FAILED;
    break;
}
} else {
    status = HID_DEVICE_SUCCESS;
}
if(status == HID_DEVICE_SUCCESS) {
    if(status == HID_DEVICE_SUCCESS) {
        MyOnCharsTransmitted(report, pHidDevice->wOutReportBufferLength);
    }

    if(quit == 0)
        Sleep(10);
}
} else {
    status = HID_DEVICE_NOT_OPENED;
}
}
return status;
}
BYTE HID_ReadFile(THidDevice * pHidDevice, BYTE * pBuffer, WORD bufferSize, WORD * pnBytesRead, ULONG msTout)
{
    BYTE status = HID_DEVICE_SUCCESS;
    DWORD bytesRead = 0, error = 0, fRes = 0;
    BYTE ReportId = CONST_REPORT_ID; // ONLY 63ReportID is available!
    ULONG startTime = GetTickCount();
    BYTE inBuffer[MAX_IN_BUF_SIZE];

    memset(pBuffer, 0, bufferSize);
    (*pnBytesRead) = 0;

#ifdef _DEBUG
    //msTout = 0xffffffff;
#endif
}

```

```
while(status == HID_DEVICE_SUCCESS) {
    bytesRead = 0;

    // Check to see that the device is opened
    if(pHidDevice->bDeviceOpen == false) {
        status = HID_DEVICE_NOT_OPENED;
        break;
    }

    memset(inBuffer, 0, sizeof(inBuffer));

    if(!ReadFile(pHidDevice->hndHidDevice, inBuffer, pHidDevice->wInReportBufferLength, &bytesRead, &pHidDevice->oRead)) {

        error = GetLastError();

        if(error == ERROR_IO_PENDING) {
            fRes = WaitForSingleObject(pHidDevice->oRead.hEvent, pHidDevice->uGetReportTimeout);

            if( (GetTickCount() - startTime) > (msTout) ) {
                CancelIo(pHidDevice->hndHidDevice);
                status = HID_DEVICE_TRANSFER_TIMEOUT;
                break;
            }

            switch(fRes)
            {
            case WAIT_OBJECT_0:
                {
                    GetOverlappedResult(pHidDevice->hndHidDevice, &pHidDevice->oRead, &bytesRead, FALSE);
                }
                break;

            case WAIT_TIMEOUT:
                {
                    CancelIo(pHidDevice->hndHidDevice);
                    bytesRead = 0;
                }
                break;

            default:
                {
                    CancelIo(pHidDevice->hndHidDevice);
                    status = HID_DEVICE_TRANSFER_FAILED;
                }
                break;
            }
        } else {
            status = HID_DEVICE_TRANSFER_FAILED;
            break;
        }
    }

    //////////////////////////////////////
}
```

```

        if(status == HID_DEVICE_SUCCESS) {
            if(bytesRead > 0)
                MyOnCharsRecieved(inBuffer, bytesRead);
        }
        //////////////////////////////////////

        if( (status == HID_DEVICE_SUCCESS) && (bytesRead >= 2) ) {
            if(((pnBytesRead) + inBuffer[2]) > bufferSize) {
                status = HID_DEVICE_BUFFER_OVERFLOW;
                break;
            }

            if(inBuffer[0] == CONST_REPORT_ID) {
                memcpy(&pBuffer[pnBytesRead], &inBuffer[3], inBuffer[2]);

                (pnBytesRead) = (WORD)((pnBytesRead) + inBuffer[2]);

                if(inBuffer[2] < (ReportId-2)) {
                    status = HID_DEVICE_SUCCESS;
                    break;
                }
            }
        }
    };

    return status;
}

```

Листинг файла TestUSB.cpp

```

// TestUsb.cpp : Defines the entry point for the console application.

#include "stdafx.h"
#include "TestUsb.h"
#include "HidDevice.h"
#include "Service.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// The one and only application object

CWinApp theApp;

using namespace std;

int _tmain(int argc, TCHAR* argv[], TCHAR* envp[])
{
    int nRetCode = 0;
    // initialize MFC and print and error on failure
    if (!AfxWinInit(::GetModuleHandle(NULL), NULL, ::GetCommandLine(), 0))

```



```
{
    // TODO: change error code to suit your needs
    cerr << _T("Fatal Error: MFC initialization failed") << endl;
    nRetCode = 1;
}
else
{
    THidDevice hidDevice;
    TTrackSerialNumbersAr serialNumListAr;
    BYTE outbuffer[] = {0x00, 0x48, 0x00, 0x00, 0x00, 0x13, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x93, 0xB9};
    BYTE inbuffer[300];
    WORD nBytesRead = 0, ret;

    memset(&serialNumListAr, 0, sizeof(serialNumListAr));
    HID_Init(&hidDevice);

    if (HID_GetSerNums(USB_VENDOR, USB_PRODUCT, &serialNumListAr) == 0 )
    {
        MyAddLogString("Device not found!", true);
        return 0;
    }

    if (HID_Open(&hidDevice, USB_VENDOR, USB_PRODUCT, &serialNumListAr.m_serialNumList[0]) != HID_DEVICE_SUCCESS )
    {
        MyAddLogString("Fail to open USB connection!", true);
        return 0;
    }

    MyAddLogString("Start write binary buffer to USB:", false);
    MyOnCharsTransmitted(outbuffer, sizeof(outbuffer));

    if (HID_WriteFile(&hidDevice, outbuffer, sizeof(outbuffer)) != HID_DEVICE_SUCCESS)
    {
        MyAddLogString("Fail to write data to USB connection!", true);
        return 0;
    }

    ret = HID_ReadFile(&hidDevice, inbuffer, (WORD)(sizeof(inbuffer)), &nBytesRead, 1000);

    switch(ret)
    {
    case HID_DEVICE_SUCCESS:
        MyAddLogString("Binary buffer from USB:", false);
        MyOnCharsRecieved(inbuffer, nBytesRead);
        MyAddLogString("Read OK!", true);
        break;

    case HID_DEVICE_USER_INTERRUPT:
        MyAddLogString("Read interrupted!", true);
        break;

    case HID_DEVICE_TRANSFER_TIMEOUT:
        MyAddLogString("Read timeout!", true);
        break;
    }
}
```

```
        default:
            MyAddLogString("Read error!", true);
            break;
    }

    HID_Close(&hidDevice);
}
return nRetCode;
```