

Описание протокола обмена тепловычислителя ТВ7 с системой верхнего уровня

Содержание

1	Обц	цие сведения	4
2		имы передачи данных	
	2.1	Поддерживаемые режимы передачи данных	
	2.2	Режим RTU	
	2.3	Режим ASCII	6
	2.4	Режим РРР	7
3	Орг	анизация доступа к данным	8
	3.1	Регистровый доступ и характеристики параметров	
	3.2	Порядок хранения и передачи байт данных	
4	Pea	лизованные функции протокола	10
	4.1	Структура поля «Информационная часть» в посылке	
	4.2	Стандартная функция Modbus 03 (0x03 hex) «Чтение регистров» (Read Holding Registers)	10
	4.3	Стандартная функция Modbus 16 (0х10 hex) «Запись регистров» (Preset Multiple Regs)	11
	4.4	Нестандартная функция 72 (0х48 hex) «Расширенная запись и чтение регистров с	
		цией» (Extended Preset And Read Multiple Regs With Numbers)	
5		сание параметров, доступных по сети	
	5.1	Используемые типы данных и условные обозначения	
	5.2	Существующие типы значений и терминология	
	5.3	Считывание мгновенных данных	
	5.4	Организация архивов и считывание архивных данных	
	5.4.1 5.4.2	о = — да = «р = «р =	
	5.4.2	Рекомендуемая последовательность считывания данных	
6		та переменных	
U	6.1	Информация об устройстве	
	6.2	Тип читаемых данных	
	6.3	Настройки вычислителя	
	6.3.1	·	
	6.3.2		
	6.4	Сервисная команда	25
	6.5	Информация о настройках	
	6.6	Информация о датах начала/конца архивов	
	6.7	Информация об индексах архивов параметров	
	6.8	Архивная запись (часовой, суточный, месячный архивы)	
	6.8.1 6.8.2		
	6.9	Архивная запись (итоговый архив)	
	6.9.1	·	33
	6.9.2		
	6.10	Информация об асинхронных архивах	
	6.11	Архивные записи асинхронных архивов	
	6.11.	1 Интерпретация архива изменений базы данных	36
	6.11.		
	6.11.	1 1 ' ''	
	6.12	Текущие итоговые значения	
	6.12.		
	6.12.	2 Значения по тепловым вводам (текущие итоги)	
	6.13 6.14	Текущие значения Структура технологического изменения параметров	.41 ለገ
п		ение 1. Функция расчета контрольной суммы Cyclical Redundancy Checking	. 40
			44
		ение 2. Функция расчета контрольной суммы Longitudinal Redundancy Checkin	
(L	.KC)		44
	-	ение 3. Функции преобразования из ASCII формата в двоичный и обратного	
		зования из двоичного в ASCII	
П	рилож	ение 4. Коды ошибок, возвращаемые прибором	46
	•	ение 5. Описание структуры файла tvb	
		ение 6. Реализация обмена данными по интерфейсу USB	
-	•	- L. L	_

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».

1 Общие сведения

Вычислитель количества теплоты ТВ7 (далее Прибор) позволяет получать текущие и архивные параметры, а также предоставляет доступ к чтению и изменению настроечных параметров.

Прибор имеет интерфейс USB device, RS-232, а также по заказу может дополнительно оснащаться интерфейсом Ethernet или GPRS модемом в различных комбинациях.

Последовательные интерфейсы поддерживают следующие параметры: скорость 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит/сек., 8 бит данных, 1 стоп-бит, нет контроля четности.

Прибор реализует подмножество стандартных функций протокола Modbus, предполагающего использование принципа ведущий-ведомый (Master-Slave). Это означает, что ведомый (Slave) может послать сообщение по сети только в ответ на запрос ведущего (Master). Вычислители всегда являются ведомыми. В качестве ведущего может выступать система верхнего уровня (персональный компьютер со специализированным программным обеспечением (ПО) или другое устройство, поддерживающее протокол Modbus и способное выступать в роли ведущего). При наличии нескольких устройств, объединенных в сеть, допускается только один ведущий. Каждое ведомое устройство в сети должно иметь уникальный сетевой адрес, позволяющий его идентифицировать. Благодаря сетевому адресу ведущий может делать персональные посылки к конкретному ведомому устройству в сети. Кроме этого вычислитель поддерживает нестандартные функции, являющиеся расширением функций протокола Modbus. Использование нестандартных функций возможно также только при соблюдении принципа ведущий-ведомый.

ПРИМЕЧАНИЕ!	При пропадании сетевого питания и настройке на скорость 9600 бит/с
	и выше принудительно устанавливается скорость 9600 бит/с.

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

Каждая сетевая посылка содержит три обязательных части:

- Поле «Сетевой адрес»;
- Поле «Информационная часть»;
- Поле «Контрольная сумма».

Длина посылки не может превышать 256 байт для стандартных функций Modbus, и 300 байт для нестандартных функций.

«Информационная часть» заключена между полями «Сетевой адрес» и «Контрольная сумма».

Сетевой	Информационная часть	Контрольная
адрес	информационная часть	сумма

<u>Поле «Сетевой адрес»</u> необходимо для адресации посылки конкретному прибору. Допустимые значения сетевого адреса находятся в диапазоне от 0 до 255.

<u>Поле «Информационная часть»</u> зависит от назначения посылки и будет описано более подробно ниже.

<u>Поле «Контрольная сумма»</u> необходимо для проверки целостности посылки. Передающее устройство вычисляет контрольную сумму над полями «Сетевой адрес» и «Информационная часть» 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 ведомое устройство получает недостоверный кадр, оно его игнорирует и не посылает каких-либо ответных сообщений. Это означает, что ведущий не получит ответа в течение ожидаемого времени и должен сделать повтор запроса. Если же факт получения недостоверной посылки обнаружен ведущим, то ведущее устройство должно выполнить повтор запроса.

ПРИМЕЧАНИЕ!

Стандарт Modbus предполагает наличие широковещательных посылок. Широковещательной посылкой является запрос с сетевым адресом, равным нулю. По стандарту **Modbus** ведомое устройство не должно отвечать на широковещательную посылку. Однако, для упрощения процесса конфигурирования приборов, в реализации протокола сделано отступление ОТ стандарта. Ha широковещательную посылку ответит любой прибор. Следствием этого является то, что при объединении нескольких приборов в сеть ведущим устройством не должны использоваться широковещательные посылки, т.к. может возникнуть ситуация, когда одновременно несколько устройств перейдут в состояние недопустимым. Использование передачи. что является широковещательных посылок возможно только при связи типа «точка-точка».

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.

Таким образом, для того, чтобы прибор ответил на запрос необходимо, чтобы кадр был достоверным и сетевой адрес в запросе совпадал с сетевым адресом прибора (либо сетевой адрес в запросе должен быть равным нулю).

2 Режимы передачи данных

2.1 Поддерживаемые режимы передачи данных

Возможно использование двух стандартных режимов последовательной передачи данных: RTU (Remote Terminal Unit) и ASCII (American Standard Code for Information Interchange) и одного нестандартного: PPP (Point to Point Protocol). Пользователь выбирает необходимый режим во время конфигурирования прибора наряду с такими параметрами как скорость обмена, сетевой адрес и т.д. Выбор используемого режима обусловлен 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.

2.2 Режим RTU

В режиме RTU для каждого байта сообщения используется двоичная система кодирования, а 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. Однако при определенных условиях выдержать такие параметры передачи невозможно. По этой причине в приборах интервал тишины сделан равным:

- 7,8 мс для скоростей 9600 бит/сек. и выше;
- 15,6 мс для скорости 4800 бит/сек.;
- 31.2 мс для скорости 2400 бит/сек.;
- 62,5 мс для скорости 1200 бит/сек.

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.

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.

2.3 Режим ASCII

В режиме ASCII каждый двоичный байт данных передается как 2 ASCII символа, а начало и конец сообщения помечены специальными маркерами. Началом сообщения всегда является символ двоеточия ':' (0х3А в шестнадцатеричном представлении). Концом сообщения всегда является пара символов «возврат каретки» (СR) и «перевод строки» (LF) (0х0D и 0х0А соответственно в шестнадцатеричном представлении). Двоичный байт данных преобразуется в пару символов. Например, значение 27 (0х1В в шестнадцатеричном представлении) будет представлено как пара символов '1' (0х31 - символьное представление старших 4-х битов) и 'В' (0х42 - символьное представление младших 4-х битов). Допустимые символы для передачи - это шестнадцатеричные символы 0-9. А-F.

В качестве функции расчета контрольной суммы в режиме ASCII 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 символа начала и конца кадра.

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.

2.4 Режим РРР

В режиме РРР для каждого байта сообщения используется двоичная система кодирования, а 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 исключающего или с значением 0x20).

В качестве функции расчета контрольной суммы в режиме PPP 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.

Режим PPP сочетает в себе быстродействие режима RTU с устойчивостью режима ASCII к временным искажениям посылаемых сообщений.

3 Организация доступа к данным

3.1 Регистровый доступ и характеристики параметров

Все данные, доступные по сети, имеют строго определенные типы, выбор которых диктуется смыслом и назначением параметров. В зависимости от сущности параметра это могут быть различные типы данных, такие как символы, восьми-, шестнадцати-, тридцатидвухразрядные знаковые и беззнаковые целые, биты, вещественные числа, массивы, структуры и т.д. Так, например, аналоговые параметры удобно представлять вещественными числами с плавающей запятой. Наличие или отсутствие каких-либо событий может быть представлено одним битом. Некоторые настроечные параметры, такие как идентификатор прибора, могут быть представлены массивом символов.

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 памяти, элементарной ячейкой которой является В 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. Параметры прибора могут иметь различные уровни доступа в зависимости от их сущности и назначения. Например, текущие измеренные прибором параметры могут быть доступны только на чтение, но изменить их невозможно. А настроечные параметры могут быть доступны для записи всегда или только при условии разрешенного доступа (разрешение доступа в приборах может быть реализовано различными способами). Таким образом, параметр, кроме собственного типа и расположения внутри массива регистров, будет также характеризоваться уровнем доступа (только чтение; чтение/запись; чтение/запись при условии).

3.2 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. Т.о., для преобразования к порядку байт, естественному для платформы РС, 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 А0		Реги	стр А1
Порядок передачи	первый			последний
Байт	B1	B0(LSB)	B3	B2(MSB)

Пример размещения данных для типа **float**:

B3 B2		B1	B0	
SEEEEEE	EMMMMMMM	MMMMMMM	MMMMMMM	

Регистр	Регистр А0		Регист	p A1
Порядок передачи	первый			последний
Байт	B1	B0(LSB)	B3	B2(MSB)

Пример размещения данных для типа double:

B7	B6	B5-B1	B0	
SEEEEEE	EEEEMMMM	MMMMMMMM	MMMMMMMM	

Регистр	Регис	стр А0	Регист	гр А1	Регис	тр А2	Реги	стр АЗ
Порядок передачи	первый							последний
Байт	B1	B0(LSB)	В3	B2	B5	B4	B7(MSB)	В6

4 Реализованные функции протокола

4.1 Структура поля «Информационная часть» в посылке

Поле «Информационная часть» всегда содержит две обязательных компоненты:

- Поле «Код функции», указывающее прибору, какую именно операцию он должен выполнить;
- Поле «Данные», которое зависит от кода функции и может иметь различное смысловое наполнение в зависимости от используемой функции.

Сетево	й	Информационная часть	Контрольная		
адрес	Функция	Данные	сумма		

4.2 Стандартная функция Modbus 03 (0x03 hex) «Чтение регистров» (Read Holding Registers)

Функция предназначена для чтения двоичного содержимого шестнадцатиразрядных регистров прибора. В общем виде структура запроса и ответа имеет следующий вид:

Запрос:

-		Начальный	Начальный	Количество	Количество	
Сетевой	Функция	адрес	адрес	регистров	регистров	Контрольная
адрес	0x03	(старший	(младший	(старший	(младший	сумма
-		байт)	байт)	байт)	байт)	-

«Информационная часть» запроса содержит поля «Начальный адрес», указывающий с какого регистра начинать чтение, и «Количество регистров», указывающее, сколько регистров следует прочитать.

Ответ в случае выполнения без ошибок:

Сетевой	Функниа			1-ый регистр	Байты	Контрольная
	0х03	байт данных	(старший	(младший	регистров	сумма
адрес	0,03	в ответе	байт)	байт)	2,3N	Cylviivia

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» и заканчивается байтом, предшествующим полю «Контрольная сумма».

Ответ при возникновении ошибки:

Сетевой	Функция 0х83 (установлен	К од онимбии	Контрольная	
адрес	старший бит)	Код ошибки	сумма	l

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).

Ниже, для прибора с сетевым адресом 27, приведен формат запроса 18-ти регистров, начиная с 806-го:

№ байта в запросе	Наименование поля	Пример значений (hex)
1	Сетевой адрес	0x1B
2	Функция	0x03
3	Начальный адрес (старший байт)	0x03
4	Начальный адрес (младший байт)	0x26
5	Количество регистров (старший байт)	0x00
6	Количество регистров (младший байт)	0x12
7 (8)	Контрольная сумма	LRC (CRC16)

В случае успешного выполнения ответ прибора будет иметь следующую структуру:

№ байта в	Наименование поля	Пример значений (hex)
запросе		
1	Сетевой адрес	0x1B
2	Функция	0x03
3	Количество байт данных в ответе	0x24
4	1-ый регистр (старший байт)	0x01
5	1-ый регистр (младший байт)	0x01
	•••	
38	18-ый регистр (старший байт)	0x01
39	18-ый регистр (младший байт)	0x01
40 (41)	Контрольная сумма	LRC (CRC16)

В случае неуспешного выполнения ответ прибора будет иметь следующую структуру:

№ байта в	Наименование поля	Пример значений (hex)
запросе		
1	Сетевой адрес	0x1B
2	Функция (установлен старший бит)	0x83
3	Код ошибки	Code
4 (5)	Контрольная сумма	LRC (CRC16)

Пример запроса и ответа при успешном выполнении в режиме RTU (в шестнадцатеричном представлении):

Запрос: 1В 03 03 26 00 12 26 72

Пример запроса и ответа при успешном выполнении в режиме ASCII (в шестнадцатеричном представлении):

Запрос: 3A 31 42 30 33 30 33 32 36 30 30 31 32 41 37 0D 0A

Пример запроса и ответа при успешном выполнении в режиме РРР (в шестнадцатеричном представлении):

Запрос: 7E 7D 3B 7D 23 7D 23 26 7D 20 7D 32 26 72 7F

OTBET: 7E 7D 3B 7D 23 24 7D 20 7D 20

4.3 Стандартная функция Modbus 16 (0х10 hex) «Запись регистров» (Preset Multiple Regs)

Функция предназначена для записи двоичного содержимого шестнадцатиразрядных регистров прибора.

Запрос:

Сетево й адрес	, ,	адрес (старши й	` ий	рег-ов (старши й	рег-ов (младш ий	кол-во байт для	1-ый регистр (старши й байт)	(младш ий	Байты рег-ов 2,3N	Контр. сумма
		байт)	байт)	байт)	байт)	записи	и байт)	байт)		

«Информационная часть» запроса содержит поля «Начальный адрес», указывающий с какого регистра начинать запись, «Количество регистров»,

указывающее, сколько регистров следует записать, «Количество байт для записи» и непосредственно значения записываемых регистров.

Ответ в случае выполнения без ошибок:

Сетевой	Функция	Нач-ый	Нач-ый адрес	Кол-во рег-ов	Кол-во рег-ов	Контрольная	
	Функция 0х10	адрес	(младший	(старший	(младший		
адрес	0.00	(старший байт)	байт)	байт)	байт)	сумма	ĺ

В случае успешного выполнения ответ содержит копию первых шести байт запроса и поле контрольной суммы.

Ответ при возникновении ошибки:

Сетевой адрес	Функция 0х90 (установлен старший бит)	Код ошибки	Контрольная сумма
	'		

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).

Ниже, для прибора с сетевым адресом 27, приведен формат записи 4-х регистров, начиная с 28-го:

начиная с 2	D - 10.	
№ байта в	Наименование поля	Пример значений (hex)
запросе		
1	Сетевой адрес	0x1B
2	Функция	0x10
3	Начальный адрес (старший байт)	0x00
4	Начальный адрес (младший байт)	0x1C
5	Количество регистров (старший байт)	0x00
6	Количество регистров (младший байт)	0x04
7	Количество байт для записи	0x08
8	1-ый регистр (старший байт)	0x00
9	1-ый регистр (младший байт)	0x09
10	2-ый регистр (старший байт)	0x06
11	2-ый регистр (младший байт)	0x1B
12	3-ый регистр (старший байт)	0x06
13	3-ый регистр (младший байт)	0x01
14	4-ый регистр (старший байт)	0xFF
15	4-ый регистр (младший байт)	0xCF
16 (17)	Контрольная сумма	LRC (CRC16)

В случае успешного выполнения ответ прибора будет иметь следующую структуру:

№ байта в	Наименование поля	Пример значений (hex)
запросе		
1	Сетевой адрес	0x1B
2	Функция	0x10
3	Начальный адрес (старший байт)	0x00
4	Начальный адрес (младший байт)	0x1C
5	Количество регистров (старший байт)	0x00
6	Количество регистров (младший байт)	0x04
7(8)	Контрольная сумма	LRC (CRC16)

В случае неуспешного выполнения ответ прибора будет иметь следующую структуру:

№ байта в	Наименование поля	Пример значений (hex)
запросе		
1	Сетевой адрес	0x1B
2	Функция (установлен старший бит)	0x90
3	Код ошибки	Code
4 (5)	Контрольная сумма	LRC (CRC16)

Пример запроса и ответа в случае ошибки в режиме RTU (в шестнадцатеричном представлении):

```
Запрос: 1B 10 00 1C 00 04 08 00 09 06 1B 06 01 FF CF 50 16 Ответ : 1B 90 0E EC 03
```

Пример запроса и ответа в случае ошибки в режиме ASCII (в шестнадцатеричном представлении):

```
Запрос: 3A 31 42 31 30 30 30 31 43 30 30 30 34 30 38 30 30 30 39 30 36 31 42 30 36 30 31 46 46 43 46 41 45 0D 0A ОТВЕТ: 3A 31 42 39 30 30 45 34 37 0D 0A
```

Пример запроса и ответа в случае ошибки в режиме РРР (в шестнадцатеричном представлении):

```
Запрос: 7E 7D 3B 7D 30 7D 20 7D 3C 7D 20 7D 24 7D 28 7D 20 7D 29 7D 26 7D 3B 7D 26 7D 21 FF CF 50 7D 36 7F
Ответ : 7E 7D 3B 90 7D 2E EC 7D 23 7F
```

4.4 Нестандартная функция 72 (0х48 hex) «Расширенная запись и чтение регистров с нумерацией» (Extended Preset And Read Multiple Regs With Numbers)

Данная функция является нестандартной и 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.

Очень 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 0х10, а затем с функцией 0х03. 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.

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, программное обеспечение верхнего уровня может послать повторный запрос. В этот момент после длительной задержки будет получен ответ на первую попытку запроса и программное обеспечение пошлет следующий, другой по смыслу запрос. В ответ на это будет получен ответ на вторую попытку первого запроса. Если не контролировать порядковый номер запросов и ответов, то полученный ответ может быть интерпретирован как ответ на второй запрос, хотя, по сути, он им не является. Такие нарушения последовательностей могут приводить к ошибкам.

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

Запрос:

Сет.			RaL	RcH	RcL	МаН	WaL	WcH	WcL	WcbH	WcbL	NumH	NumL	1-ый регистр (старш байт)	1-ый регистр (младш байт)	Байты рег-ов 2,3N	Контр
------	--	--	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------------------------------------	------------------------------------	-------------------------	-------

«Информационная часть» запроса содержит поля:

- RaH старший байт начального адреса для чтения;
- RaL младший байт начального адреса для чтения;
- RcH старший байт количества регистров для чтения;
- RcL младший байт количества регистров для чтения;
- WaH старший байт начального адреса для записи;
- WaL младший байт начального адреса для записи;
- WcH старший байт количества регистров для записи;
- WcL младший байт количества регистров для записи;
- WcbH старший байт счетчика количества записываемых байт;
- WcbL младший байт счетчика количества записываемых байт;
- NumH старший байт номера запроса;
- NumL младший байт номера запроса;
- значения записываемых регистров.

Если не требуется выполнять чтение, то поля RcH и RcL устанавливаются в 0. Если не требуется запись, то поля WcH, WcL, WcbH, WcbL устанавливаются в 0 и поле «значение записываемых параметров» отсутствует.

Ответ в случае выполнения без ошибок:

Сет.	Функц. 0х48	байт данных в ответе (старший	1 1	NumH	NumL (из запроса)	1-ый регистр (старши й байт)	1-ый регистр (младши й байт)	Байты регистро в 2,3N	Контр. сумма
		байт)	байт)						

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

Ответ при возникновении ошибки:

Сетевой	Функция 0хС8	Код	Код	NumH	NumL	Контр.
адрес	(установлен	ошибки	ошибки	(из запроса)	(из запроса)	сумма
' "	старший бит)	чтения	записи	' '	` ' '	,

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). Порядковый номер в ответе (байты NumH и NumL) должен совпадать с порядковым номером в запросе.

Ниже, для прибора с сетевым адресом 27, приведен формат записи 2-х регистров,

начиная с 8550-го и чтение 2-х регистров начиная с 28-го:

№ байта в	Наименование поля	Пример значений
запросе		(hex)
1	Сетевой адрес	0x1B
2	Функция	0x48
3	Начальный адрес чтения (старший байт)	0x00
4	Начальный адрес чтения (младший байт)	0x1C
5	Количество регистров для чтения (старший байт)	0x00
6	Количество регистров для чтения (младший байт)	0x02
7	Начальный адрес записи(старший байт)	0x21
8	Начальный адрес записи (младший байт)	0x66
9	Количество регистров для записи (старший байт)	0x00
10	Количество регистров для записи (младший байт)	0x02
11	Количество байт для записи (старший байт)	0x00
12	Количество байт для записи (младший байт)	0x04
13	Порядковый номер запроса (старший байт)	0x00
14	Порядковый номер запроса (младший байт)	0x01
15	1-ый регистр для записи (старший байт)	0x00
16	1-ый регистр для записи (младший байт)	0x00
17	2-ый регистр для записи (старший байт)	0x00
18	2-ый регистр для записи (младший байт)	0x00
19 (20)	Контрольная сумма	LRC (CRC16)

В случае успешного выполнения ответ прибора будет иметь следующую структуру:

№ байта в	Наименование поля	Пример значений
запросе		(hex)
1	Сетевой адрес	0x1B
2	Функция	0x48
3	Количество байт данных в ответе (старший байт)	0x00
4	Количество байт данных в ответе (младший байт)	0x04
5	Порядковый номер запроса (старший байт)	0x00
6	Порядковый номер запроса (младший байт)	0x01
7	1-ый регистр (старший байт)	0x00
8	1-ый регистр (младший байт)	0x00
9	2-ый регистр (старший байт)	0x00
10	2-ый регистр (младший байт)	0x00
11(12)	Контрольная сумма	LRC (CRC16)

Пример значений (hex)

В случае неуспешного выполнения ответ прибора будет иметь следующую структуру:

<u> </u>	
№ байта в	Наименование поля
запросе	
1	Сетевой адрес
2	Функция (установлен старицій бит)

0x1B 0xC8 3 Code Код ошибки чтения 4 Код ошибки записи Code 5 Порядковый номер запроса (старший байт) 0x00 0x01 6 Порядковый номер запроса (младший байт) LRC (CRC16) 7 (8) Контрольная сумма

Пример запроса и ответа в случае ошибки записи в режиме RTU (в шестнадцатеричном представлении):

Запрос: 1B 48 00 1C 00 02 21 66 00 02 00 04 00 01 00 00 00 57 CF

Ответ : 1B C8 00 0E 00 01 42 23

Пример запроса и ответа в случае ошибки записи в режиме ASCII (в шестнадцатеричном представлении):

OTBET: 3A 31 42 43 38 30 30 30 45 30 30 30 31 30 45 0D 0A

Пример запроса и ответа в случае ошибки записи в режиме РРР (в шестнадцатеричном представлении):

 $\tt 3anpoc\colon$ 7E 7D 3B 48 7D 20 7D 3C 7D 20 7D 22 21 66 7D 20 7D 22 7D 20 7D 24 7D 20 7D 21 7D 20 7D 20 7D 20 7D 20 57 CF 7F

OTBET: 7E 7D 3B C8 7D 20 7D 2E 7D 20 7D 21 42 23 7F

5 Описание параметров, доступных по сети

Значения настроечных, текущих, архивных параметров всегда передаются в системе единиц СИ независимо от системы единиц отображения, установленной в приборе.

5.1 Используемые типы данных и условные обозначения

- unsigned char восьмибитное беззнаковое целое число;
- signed char восьмибитное знаковое целое число;
- unsigned short шестнадцатибитное беззнаковое целое число;
- signed short шестнадцатибитное знаковое целое число;
- unsigned long тридцатидвухразрядное беззнаковое целое число;
- signed long тридцатидвухразрядное знаковое целое число;
- float вещественное число одинарной точности с плавающей запятой, соответствующее стандарту «IEEE 754»;
- double вещественное число двойной точности 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;
- AC архив суточный;
- AM архив месячных;
- АИ архив итоговый;
- ТВ тепловой ввод;
- ДП дополнительный параметр;

5.2 Существующие типы значений и терминология

Измеряемые и вычисляемые параметры прибора можно разделить на мгновенные и архивные.

Мгновенные данные дополнительно терминологически подразделяются на Текущие, и Текущие итоговые. Под Текущими данными понимаются мгновенные измеренные или вычисленные параметры такие как температуры, давления, расходы теплоносителя, тепловой поток, разность температур и т.д. Под Текущими итоговыми данными понимаются интегральные параметры теплопотребления от момента сброса архива прибора до текущего момента. Т.е. это данные, представляющие «нарастающий итог» параметров теплопотребления (объемы, массы, тепло, разность масс, времена нормальной работы и отсутствия счета и т.д.). Дискретность обновления мгновенных 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 теплопотреблении 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 теплопотребления 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». Часовая архивная запись с хронологической меткой «10 часов» хранит параметры теплопотребления, зафиксированные на интервале времени с 10:00:00 до 11:00:00. 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 234.:

• 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.).

5.3 Считывание мгновенных данных

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

5.4 Организация архивов и считывание архивных данных

Синхронные и асинхронные архивы организованы как кольцевые индексируемые 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.

5.4.1 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) и типа архива, который надо прочитать;
- Операция чтения регистров. Адрес стартового регистра для чтения зависит от типа читаемого архива. Количество регистров соответствует длине архивной записи, или длине фрагмента архивной записи. За один прием можно прочитать содержимое только одной архивной записи.

5.4.2 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 сделан 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. «<u>Тип читаемых данных</u>»). В ответ прибор присылает подтверждение записи;
- Операция чтения регистров. Адрес стартового регистра для чтения зависит от типа архива (см. «<u>Архивные записи асинхронных архивов</u>»). 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. «<u>Тип читаемых данных</u>»). 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.

5.5 Рекомендуемая последовательность считывания данных

В большинстве случаев задача считывания данных сводится к чтению текущих показаний вычислителя 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 встроенного программного обеспечения, серийного номера и т.д.;
- Чтение настроек прибора;
- Чтение информации о синхронных и асинхронных архивах, интервал дат архивов;
- Чтение текущих, итоговых текущих значений;
- Чтение синхронных архивов;
- Чтение асинхронных архивов.

6 Карта переменных

В данном разделе описываются структуры данных, доступные для чтения или записи. Для каждого параметра указывается адрес (или смещение), размер параметра, его тип и режим доступа.

6.1 Информация об устройстве

Данная структура предназначена для получения информации о типе и модели подключенного устройства, его серийном номере, версии ПО, порядке передачи байт и регистров и т.д. Для получения информации об устройстве необходимо выполнить операцию чтения содержимого регистров.

Описание структуры данных

Название	Адрес	Размер (байт)	Тип	Доступ	Примечание			
Тип устройства	0	2	unsigned short	R/O				
Программная версия	1	2	unsigned short	R/O				
Аппаратная версия	2	2	unsigned short	R/O				
Контрольная сумма ПО	3	2	unsigned short	R/O				
Модель	4	1	unsigned char	R/O	Биты 0-7			
Резерв	4	1	unsigned char	R/O	Биты 8-15			
Серийный номер	5	4	unsigned long	R/O				

Возможные значения:

- Тип устройства: допустимое значение типа ТВ7-5890 (0х1702).
- Программная и аппаратная версии: старший байт содержит номер версии, младший номер редакции.
- Контрольная сумма программного обеспечения.
- Модель: модель вычислителя (0-3).
- Серийный номер прибора.

6.2 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, буферирует их, и передает системе верхнего уровня. Т.о. чтение архивных данных выглядит как несколько последовательных серий операций «запись регистров, чтение регистров».

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

Описание структуры данных								
Название	Адрес	Размер (байт)	Тип	Доступ	Примечание			
Дата: день	99	1	unsigned char	R/W	Биты 0-7			
Дата: месяц	99	1	unsigned char	R/W	Биты 8-15			
Дата: год-2000	100	1	unsigned char	R/W	Биты 0-7			
Время: час	100	1	unsigned char	R/W	Биты 8-15			
Время: минута	101	1	unsigned char	R/W	Биты 0-7			
Время: секунда	101	1	unsigned char	R/W	Биты 8-15			
Тип архива	102	2	unsigned short	R/W				
Номер записи	103	2	unsigned short	R/W				
Резерв 1	104	2	unsigned short	R/W				

Дата: день: 1-31;Дата: месяц: 1-12;

• Дата: год: Значение года минус 2000;

Время: час: 0-23;Время: минута: 0-59;Время: секунда: 0-59;

- Тип архива. Принимает значения: 0-часовой архив, 1-суточный архив, 2-месячный архив, 3-итоговый архив;
- Номер записи. 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.

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)»).

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»).

6.3 Настройки вычислителя

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

6.3.1 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. «<u>Сервисная команда</u>»);

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, выполняет автоматический сброс архивов параметров и начинает работать в соответствии с «новыми» настройками.

6.3.1.1 Системные настройки

Описание структуры данных

Название	Адрес	Размер (байт)	Тип	Доступ	Примечание
Час отчета	105	1	unsigned char	R/W*	Биты 0-7
Дата отчета	105	1	unsigned char	R/W*	Биты 8-15
БД1 с: день	106	1	unsigned char	R/W*	Биты 0-7
БД1 с: месяц	106	1	unsigned char	R/W*	Биты 8-15
БД1 с: год	107	1	unsigned char	R/W*	Биты 0-7
БД1 с: час	107	1	unsigned char	R/W*	Биты 8-15
БД2 с: день	108	1	unsigned char	R/W*	Биты 0-7
БД2 с: месяц	108	1	unsigned char	R/W*	Биты 8-15
БД2 с: год (не исп.)	109	1	unsigned char	R/W*	Биты 0-7
БД2 с: час	109	1	unsigned char	R/W*	Биты 8-15
Использование БД2	110	1(!)	bit field	R/W*	Бит 0
Способ переключ. БД	110	2(!)	bit field	R/W*	Бит 1-2
Доступ к перекл. БД с					
клавиат,	110	3(!)	bit field	R/W*	Бит 3-5
Доступ к перекл. БД по					
сети	110	3(!)	bit field	R/W*	Бит 6-8
Резерв 1	110	7(!)	bit field	R/W*	Бит 9-15
Ед. измерения	111	1(!)	bit field	R/W*	Бит 0
Тип термодатчиков	111	4(!)	bit field	R/W*	Биты 1-4
Резерв 2	111	1(!)	bit field	R/W*	Бит 5
Резерв 3	111	25(!)	bit field	R/W*	Биты 5-31

Список полей структуры:

- Час отчета: 0-23;
- Дата отчета: 1-31;
- БД1/2 с: день/месяц/год/час: дата и время автоматического переключения активной базы данных. Поле «год» игнорируется;
- Использование БД2: 0-БД2 не используется, 1-БД2 используется;
- Способ переключения БД: 0-вручную, 1-автоматически по дате;
- Доступ к изменению БД с клавиатуры: 0-запрет, 1-с паролем, 2-при нажатой кнопке «Доступ»;
- Доступ к изменению БД по сети: 0-запрет, 1-с паролем;
- Единицы измерения: 0-СИ, 1-МКС;
- Тип термодатчиков: 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);

6.3.1.2 Настройки труб

Вычислитель хранит настройки двух баз данных. В каждой базе данных может быть задействовано до 6-ти труб одновременно. Структуры настроек всех труб идентичны. В данном разделе будут представлены базовые адреса структур настроек всех труб и смещения (в количестве регистров) до параметров труб относительно базового адреса.

Список базовых адресов

Наименование	Базовые адреса
БД1 ТВ1 Труба 1	113
БД1 ТВ1 Труба 2	130
БД1 ТВ1 Труба 3	147
БД1 ТВ2 Труба 1	164
БД1 ТВ2 Труба 2	181
БД1 ТВ2 Труба 3	198
БД2 ТВ1 Труба 1	233
БД2 ТВ1 Труба 2	250
БД2 ТВ1 Труба 3	267
БД2 ТВ2 Труба 1	284
БД2 ТВ2 Труба 2	301
БД2 ТВ2 Труба 3	318

Описание структуры данных

Chineanine crpykryp		1			
Название	Смещение	Размер (байт)	Тип	Доступ	Примечание
НС по V	0	3(!)	bit field	R/W*	Биты 0-2
Контроль раб.	0	2(!)	bit field	R/W*	Биты 3-4
Датчик Р	0	1(!)	bit field	R/W*	Бит 5
Тип ВС	0	2(!)	bit field	R/W*	Биты 6-7
Резерв 1	0	8(!)	bit field	R/W*	Биты 8-15
tдог.	1	4	float	R/W*	
Рдог.	3	4	float	R/W*	
Рп	5	4	float	R/W*	
Рв	7	4	float	R/W*	
Вес имп.	9	4	float	R/W*	
Vчн	11	4	float	R/W*	
Vчв	13	4	float	R/W*	
Vдог	15	4	float	R/W*	

Список полей структуры:

- НС по V: способ обработки НС по объему. 0-нет, 1-Без подст., 2-С подст., 3-С подс. и контр. U, 4-Счет отменен;
- Контр. раб.: контроль работоспособности расходомера. 0-нет, 1-Сеть (общий);2-Индивид.;
- Датчик Р: наличие датчика измерения давления. 0-нет, 1-есть;
- Тип ВС: тип водосчетчика. 0-механический, 1-электронный;
- tдог: договорное значение температуры (0-175 °C);
- Рдог: договорное значение давления (0–1,667 МПа (17 кгс/см2));
- Рп: поправка на высоту водяного столба (±0-9.99 м);
- Рв: верхний предел измерения датчика (0-1,667 МПа (17 кгс/см2));
- Вес имп.: вес импульса (0–9999.9999 литр);
- Vчн: нижняя уставка на часовой объем (0-99999,9 м3);
- Vчв: верхняя уставка на часовой объем (0-99999,9 м3);
- Удог: договорной часовой объем (0-99999,9 м3).

6.3.1.3 Настройки тепловых вводов

Вычислитель хранит настройки двух баз данных. В каждой базе данных может быть задействовано до 2-х тепловых вводов одновременно. Структуры настроек тепловых вводов идентичны. В данном разделе будут представлены базовые адреса структур настроек тепловых вводов и смещения (в количестве регистров) до параметров тепловых вводов относительно базового адреса.

Список базовых адресов

Наименование	Базовые адреса
БД1 TB1	215
БД1 TB2	224
БД2 TB1	335
БД2 ТВ2	344

Описание структуры данных

Название	Смещение	Размер (байт)	Тип	Доступ	Примечание
	Смещение	i asivich (davii)			Примечание
СИ	0	1	unsigned char	R/W*	
Резерв 1	0	1	unsigned char	R/W*	
dMmax	1	4	float	R/W*	
tхдог	3	4	float	R/W*	
Рхдог	5	4	float	R/W*	
KT3	7	3(!)	bit field	R/W*	Биты 0-2
ФРТ	7	3(!)	bit field	R/W*	Биты 3-6
HC по t	7	2(!)	bit field	R/W*	Биты 7-8
HC по dM	7	3(!)	bit field	R/W*	Биты 9-11
НС по Q	7	3(!)	bit field	R/W*	Биты 12-14
HC по dt	7	3(!)	bit field	R/W*	Биты 15-17
dtmin	7	3(!)	bit field	R/W*	Биты 18-20
Исп.tx	7	4(!)	bit field	R/W*	Биты 21-24
Исп. tнв	7	1(!)	bit field	R/W*	Бит 25
Резерв 2	7	6(!)	bit field	R/W*	Биты 26-31

Список полей структуры:

- СИ: схема измерения (0–14, 0–ТВ не используется);
- dMmax: уставка на небаланс часовой массы (0-4%);
- txдог: договорное значение температуры холодной воды (0-99 °C);
- Рхдог: договорное значение давления холодной воды (0-1,667 МПа (17 кгс/см2));
- КТЗ: 0-труба 3 не используется, 1-измерение объема в трубе 3, 2-измерение температуры в трубе 3; 3-измерение объема и температурыв трубе 3;
- ФРТ: формула расчета тепла:
 - 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)
- HC по t: способ обработки HC по t (0-с подст., 1-счет отмен.);
- HC по dM: способ обработки HC по dM (0-нет, 1-без подст. 1, 2- без подст. 2, 3-с подст. 1, 4- подст. 2);
- НС по Q: способ обработки НС по Q (0-нет, 1-без подст., 2-с подст., 3-счет отменен):
- HC по dt: способ обработки HC по dt (0-без подст., 1-с подст., 2-счет отмен.);
- dtmin: минимально допустимая разность t (0-2°C, 1-3°C);
- Исп. tx: способ использования температуры холодной воды (0-hx не учитывается, 1-договорная, 2-измеряется в данном ТВ, 3-измеряется в другом ТВ);

• Исп. tнв: способ использования температуры наружного воздуха (0-не измеряется, 1-измеряется);

6.3.1.4 Настройки дополнительного импульсного входа

Описание структуры данных

Название	Адрес	Размер (байт)	Тип	Доступ	Примечание
Способ исп.	353	3(!)	bit field	R/W*	Биты 0-2
Уровень	353	1(!)	bit field	R/W*	Бит 3
Ед.изм.	353	4(!)	bit field	R/W*	Биты 4-7
Резерв 1	353	8(!)	bit field	R/W*	Биты 8-15
Вес имп./Тподтв.	354	4	float	R/W*	
Знач.ЧН	356	4	float	R/W*	
Знач.ЧВ	354	4	float	R/W*	

Список полей структуры:

- Способ исп.: способ использования дополнительного импульсного входа (0-нет, 1-контр.сети, 2-счет импульсов, 3-сигнализация);
- Уровень: уровень тревоги сигнализации (0-нормально разомкнутый, 1-нормально замкнутый);
- Ед.измерения: 0-м3, 1-кВт*ч;
- Вес имп./Тподтв.: вес импульса при использовании дополнительного входа для счета импульсов (0-9999.9999) или время подтверждения тревоги (0.1-9.9 сек.) при использовании входа как входа сигнализации;
- Знач.ЧВ/ЧН: максимальное/минимальное часовое значение параметра (0-99999.9) в режиме счета импульсов.

6.3.2 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;

Описание структуры данных

 Simodiffic Cipykiypsi Hamisix							
Название	Адрес	Размер (байт)	Тип	Доступ	Примечание		
Сет.адрес	2153	1	unsigned char	R/W	Биты 0-7		
Резерв 1	2153	1	unsigned char	R/W	Биты 8-15		
Код орган.	2154	10	char[10]	R/W			
Договор	2159	10	char[10]	R/W			

Список полей структуры:

- Сет. адрес: значение сетевого адреса вычислителя (0-255);
- <u>Код орган.</u>: символы кода организации. Допустимы только цифровые символы ('0'- '9'). Последний символ последовательности, задающий длину строки, должен иметь значение двоичного нуля;
- <u>Договор</u>: символы номера договора. Допустимы только цифровые символы ('0'-'9'). Последний символ последовательности, задающий длину строки, должен иметь значение двоичного нуля;

6.4 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».

Описание структуры данных

Название	Адрес	Размер (байт)	Тип	Доступ	Примечание
Команда	2665	2	unsigned short	R/W*	

Список полей структуры:

- Команда: в зависимости от установленного значения вычислителем выполняются следующие действия:
 - 1 Старт записи настроек;
 - 2 Окончание записи настроек;
 - 7 Установить активной БД1;
 - 8 Установить активной БД2.

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 0х20). Например, пароль «123» должен быть представлен в виде массива байт: 0х31, 0х32, 0х33, 0х20, 0х20, 0х00.

6.5 Информация о настройках

Структура предназначена для операций с такой информацией, как контрольные суммы настроек, номер выбранной активной базы данных и т.д.. Поля «Контрольная сумма...» должны устанавливаться системой верхнего уровня при изменении настроек прибора (см. «Настройки вычислителя»).

Описание структурь	і данных				
Название	Адрес	Размер (байт)	Тип	Доступ	Примечание
Контрольная сумма					
защищенных настроек	2669	2	unsigned short	R/W*	
Активная БД	2670	1	unsigned char	R/O	Биты 0-7
Резерв 1	2670	1	unsigned char	R/O	Биты 8-15
Резерв 2	2671	2	unsigned short	R/O	
Резерв 3	2672	2	unsigned short	R/O	
Контрольная сумма					
незащищенных					
настроек	2673	2	unsigned short	R/W	

- Контрольная сумма защищенных настроек;
- Активная БД: 0-активна база данных 1, 1-активна база данных 2; Поле доступно только на чтение.
- Контрольная сумма незащищенных настроек.

6.6 Информация о датах начала/конца архивов

Структура предназначена для чтения дат начала/конца архивов параметров.

Название	ī	Размер (байт)	Тип	Доступ	Примечание
Дата начала АЧ, день:	2676	1	unsigned char	R/O	Биты 0-7
Дата начала АЧ, месяц:	2676	1	unsigned char	R/O	Биты 8-15
Дата начала АЧ, год-2000:	2677	1	unsigned char	R/O	Биты 0-7
Время начала АЧ, час:	2677	1	unsigned char	R/O	Биты 8-15
Время начала АЧ, минута:	2678	1	unsigned char	R/O	Биты 0-7
Время начала АЧ, секунда:	2678	1	unsigned char	R/O	Биты 8-15
Дата начала АС, день:	2679	1	unsigned char	R/O	Биты 0-7
Дата начала АС, месяц:	2679	1	unsigned char	R/O	Биты 8-15
Дата начала АС, год-2000:	2680	1	unsigned char	R/O	Биты 0-7
Время начала АС, час:	2680	1	unsigned char	R/O	Биты 8-15
Время начала АС, минута:	2681	1	unsigned char	R/O	Биты 0-7
Время начала АС, секунда:	2681	1	unsigned char	R/O	Биты 8-15
Дата начала АМ, день:	2682	1	unsigned char	R/O	Биты 0-7
Дата начала АМ, месяц:	2682	1	unsigned char	R/O	Биты 8-15
Дата начала АМ, год-2000:	2683	1	unsigned char	R/O	Биты 0-7
Время начала АМ, час:	2683	1	unsigned char	R/O	Биты 8-15
Время начала АМ, минута:	2684	1	unsigned char	R/O	Биты 0-7
Время начала АМ, секунда:	2684	1	unsigned char	R/O	Биты 8-15
Дата начала АИ, день:	2685	1	unsigned char	R/O	Биты 0-7
Дата начала АИ, месяц:	2685	1	unsigned char	R/O	Биты 8-15
Дата начала АИ, год-2000:	2686	1	unsigned char	R/O	Биты 0-7
Время начала АИ, час:	2686	1	unsigned char	R/O	Биты 8-15
Время начала АИ, минута:	2687	1	unsigned char	R/O	Биты 0-7
Время начала АИ, секунда:	2687	1	unsigned char	R/O	Биты 8-15
Дата конца АЧ, день:	2688	1	unsigned char	R/O	Биты 0-7
Дата конца АЧ, месяц:	2688	1	unsigned char	R/O	Биты 8-15
Дата конца АЧ, год-2000:	2689	1	unsigned char	R/O	Биты 0-7
Время конца АЧ, час:	2689	1	unsigned char	R/O	Биты 8-15
Время конца АЧ, минута:	2690	1	unsigned char	R/O	Биты 0-7
Время конца АЧ, секунда:	2690	1	unsigned char	R/O	Биты 8-15
Дата конца АС, день:	2691	1	unsigned char	R/O	Биты 0-7
Дата конца АС, месяц:	2691	1	unsigned char	R/O	Биты 8-15
Дата конца АС, год-2000:	2692	1	unsigned char	R/O	Биты 0-7
Время конца АС, час:	2692	1	unsigned char	R/O	Биты 8-15
Время конца АС, минута:	2693	1	unsigned char	R/O	Биты 0-7
Время конца АС, секунда:	2693	1	unsigned char	R/O	Биты 8-15
Дата конца АМ, день:	2694	1	unsigned char	R/O	Биты 0-7
Дата конца АМ, месяц:	2694	1	unsigned char	R/O	Биты 8-15
Дата конца АМ, год-2000:	2695	1	unsigned char	R/O	Биты 0-7
Время конца АМ, час:	2695	1	unsigned char	R/O	Биты 8-15
Время конца АМ, минута:	2696	1	unsigned char	R/O	Биты 0-7
Время конца АМ, секунда:	2696	1	unsigned char	R/O	Биты 8-15
Дата конца АИ, день:	2697	1	unsigned char	R/O	Биты 0-7
Дата конца АИ, месяц:	2697	1	unsigned char	R/O	Биты 8-15
Дата конца АИ, год-2000:	2698	1	unsigned char	R/O	Биты 0-7

Название	Адрес	Размер (байт)	Тип	Доступ	Примечание
Время конца АИ, час:	2698	1	unsigned char	R/O	Биты 8-15
Время конца АИ, минута:	2699	1	unsigned char	R/O	Биты 0-7
Время конца АИ, секунда:	2699	1	unsigned char	R/O	Биты 8-15
Дата сброса архива, день:	2700	1	unsigned char	R/O	Биты 0-7
Дата сброса архива, месяц:	2700	1	unsigned char	R/O	Биты 8-15
Дата сброса архива, год-2000:	2701	1	unsigned char	R/O	Биты 0-7
Время сброса архива, час:	2701	1	unsigned char	R/O	Биты 8-15
Время сброса архива, минута:	2702	1	unsigned char	R/O	Биты 0-7
Время сброса архива, секунда:	2702	1	unsigned char	R/O	Биты 8-15

- Даты начала/конца архивов, дата сброса:
 - День 1-31;
 - Месяц 1-12;
 - Год 0-255;
 - 4ac 0-23;
 - Минута 0-59;
 - Секунда 0-59.

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

6.7 Информация об индексах архивов параметров

Структура предназначена для чтения служебной информации о ведении архивов.

Описание структуры данных						
Название	Адрес	Размер (байт)	Тип	Доступ	Примечание	
Дата сброса архива, день:	2708	1	unsigned char	R/O	Биты 0-7	
Дата сброса архива, месяц:	2708	1	unsigned char	R/O	Биты 8-15	
Дата сброса архива, год-2000:	2709	1	unsigned char	R/O	Биты 0-7	
Время сброса архива, час:	2709	1	unsigned char	R/O	Биты 8-15	
Время сброса архива, минута:	2710	1	unsigned char	R/O	Биты 0-7	
Время сброса архива, секунда:	2710	1	unsigned char	R/O	Биты 8-15	
Макс.кол-во записей (АЧ)	2711	2	unsigned short	R/O		
Индекс записи (АЧ)	2712	2	unsigned short	R/O		
Размер записи, байт (АЧ)	2713	2	unsigned short	R/O		
Признак закольц. (АЧ)	2714	1(!)	bit field	R/O	Бит 0	
Резерв 1 (АЧ)	2714	15(!)	bit field	R/O	Биты 1-15	
Макс.кол-во записей (АС)	2715	2	unsigned short	R/O		
Индекс записи (АС)	2716	2	unsigned short	R/O		
Размер записи, байт (АС)	2717	2	unsigned short	R/O		
Признак закольц. (АС)	2718	1(!)	bit field	R/O	Бит 0	
Резерв 1 (АС)	2718	15(!)	bit field	R/O	Биты 1-15	
Макс.кол-во записей (АМ)	2719	2	unsigned short	R/O		
Индекс записи (АМ)	2720	2	unsigned short	R/O		
Размер записи, байт (АМ)	2721	2	unsigned short	R/O		
Признак закольц. (АМ)	2722	1(!)	bit field	R/O	Бит 0	
Резерв 1 (АМ)	2722	15(!)	bit field	R/O	Биты 1-15	
Макс.кол-во записей (АИ)	2723	2	unsigned short	R/O		
Индекс записи (АИ)	2724	2	unsigned short	R/O		
Размер записи, байт (АИ)	2725	2	unsigned short	R/O		
Признак закольц. (АИ)	2726	1(!)	bit field	R/O	Бит 0	
Резерв 1 (АИ)	2726	15(!)	bit field	R/O	Биты 1-15	

- Дата сброса архива: допустимые значения полей:
 - День 1-31;
 - Месяц 1-12;
 - Год 0-255;
 - 4ac 0-23;
 - Минута 0-59;
 - Секунда 0-59.
- Макс. количество записей: представляет значение максимального количества записей для архива данного типа;
- Индекс записи: представляет значение в диапазоне 0-«Макс. количество записей», и означающее номер записи, в который будет произведено сохранение архивной информации в следующий раз;
- Размер записи: означает размер архивной записи в байтах;
- Признак закольц.: значение 0 означает, что архив еще не заполнен, значение 1 означает, что архив заполнен полностью и сохранение новой записи вызывает стирание самой старой записи архива данного типа.

6.8 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»).

Название	Адрес	Размер (байт)	Тип	Доступ	Примечание
Дата: день	2740	1	unsigned char	R/O	Биты 0-7
Дата: месяц	2740	1	unsigned char	R/O	Биты 8-15
Дата: год-2000	2741	1	unsigned char	R/O	Биты 0-7
Время: час	2741	1	unsigned char	R/O	Биты 8-15
Значения по трубам					См. Значения по трубам
Значения по тепловым вводам					См. <u>Значения по ТВ</u>
Значение по ДП	2826	4	float	R/O	
НС по ТВ1 Труба 1	2828	1	unsigned char	R/O	Биты 0-7
НС по ТВ1 Труба 2	2828	1	unsigned char	R/O	Биты 8-15
НС по ТВ1 Труба 3	2829	1	unsigned char	R/O	Биты 0-7
НС по ТВ2 Труба 1	2829	1	unsigned char	R/O	Биты 8-15
НС по ТВ2 Труба 2	2830	1	unsigned char	R/O	Биты 0-7
НС по ТВ2 Труба 3	2830	1	unsigned char	R/O	Биты 8-15
НС по ТВ1	2831	2	unsigned short	R/O	
НС по ТВ2	2832	2	unsigned short	R/O	
НС по ДП	2833	1	unsigned char	R/O	Биты 0-7
Резерв 1	2833	1	unsigned char	R/O	Биты 8-15
Признаки событий	2834	2	unsigned short	R/O	
Резерв 2	2835	2	unsigned short	R/O	
Длит.работы по сети	2836	2	unsigned short	R/O	
Длит.работы дисплея	2837	2	unsigned short	R/O	
Длит.отсут.сет.питания	2838	2	unsigned short	R/O	
Активная БД	2839	1	unsigned char	R/O	Биты 0-7
СИ ТВ1	2839	1	unsigned char	R/O	Биты 8-15
KT3 TB1	2840	1	unsigned char	R/O	Биты 0-7
ΦPT TB1	2840	1	unsigned char	R/O	Биты 8-15
Активная БД	2841	1	unsigned char	R/O	Биты 0-7
СИ ТВ2	2841	1	unsigned char	R/O	Биты 8-15
KT3 TB2	2842	1	unsigned char	R/O	Биты 0-7
ΦPT TB2	2842	1	unsigned char	R/O	Биты 8-15

- Дата/время: представляет дату/время интервала архивирования. Значение «21.04.11 02» в часовом архиве означает, что архивная запись содержит данные, которые были зарегистрированы по окончании «21.04.11 02». Допустимые значения:
 - День 1-31;
 - Месяц 1-12;
 - Год 0-255;
 - Yac 0-23.
- Значения по трубам: см далее <u>Значения по трубам;</u>
- Значения по тепловым вводам: см. далее Значения по тепловым вводам;
- Значение по ДП: архивное значение по дополнительному параметру, измеренному на дополнительном импульсном входе, в режиме счета импульсов;
- НС по трубам: биты кодируют наличие нештатных ситуаций в следующем порядке (0-НС отсутствует, 1-НС присутствует):
 - Бит 0 HC t<min;
 - Бит 1 HC t>max;
 - Бит 2 НС неиспр. датчика t;
 - Бит 3 HC P<min;
 - Бит 4 HC P>max;
 - Бит 5 HC V<min;
 - Бит 6 HC V>max;
 - Бит 7 НС неиспр. или отсут. питания ВС;
- НС по ТВ: биты кодируют наличие нештатных ситуаций в следующем порядке (0-НС отсутствует, 1-НС присутствует):
 - Бит 0 HC по dt;
 - Бит 1 HC по dM;
 - Бит 2 НС по Qтв;
 - Бит 3 HC tx<min;
 - Бит 4 HC tx>max;
 - Бит 5 НС неиспр. датчика tx;
 - Бит 6 HC tнв<min;
 - Бит 7 HC tнв>max;
 - Бит 8 НС неиспр. датчика tнв
 - Бит 9 НС по Q12;
 - Бит 10 HC по Qг;
 - Бит 11 HC по Px<min:
 - Бит 12 НС по Рх>тах;
- НС по ДП: биты кодируют наличие нештатных ситуаций в следующем порядке (0-НС отсутствует, 1-НС присутствует):
 - Бит 0 HC по ДП<min, 1 HC по ДП>max;
- Признаки событий: биты кодируют наличие событий в следующем порядке
 - 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;

- Длит. работы по сети: значение длительности работы по сети в минутах. Фиксируется только при отсутствии внешнего сетевого питания;
- Длит. работы дисплея: значение длительности работы дисплея в минутах. Фиксируется только при отсутствии внешнего сетевого питания;
- Длит. отсут. сет. питания: значение длительности отсутствия внешнего сетевого питания в минутах;
- Активная БД: 0-активна БД1, 1-активна БД2;
- СИ ТВ: значение конфигурационного параметра ТВ;
- КТЗ ТВ: значение конфигурационного параметра КТЗ;
- ФРТ ТВ: значение конфигурационного параметра ФРТ;

6.8.1 Значения по трубам

Архивная запись хранит данные 6-ти труб. Структуры данных по трубам идентичны. В данном разделе будут представлены базовые адреса структур данных по трубам и смещения (в количестве регистров) до параметров по трубам относительно базового адреса.

Список базовых адресов

Наименование	Базовые адреса
ТВ1 Труба 1	2742
ТВ1 Труба 2	2750
ТВ1 Труба 3	2758
ТВ2 Труба 1	2766
ТВ2 Труба 2	2774
ТВ2 Труба 3	2782

Описание структуры данных

Название	Смещение	Размер (байт)	Тип	Доступ	Примечани е
t	0	4	float	R/O	
Р	2	4	float	R/O	
V	4	4	float	R/O	
M	6	4	float	R/O	

Список полей структуры:

t, P, V, M – параметры температура, давление, объем и масса;

6.8.2 Значения по тепловым вводам

Архивная запись хранит данные 2-х тепловых вводов. Структуры данных по ТВ идентичны. В данном разделе будут представлены базовые адреса структур данных по ТВ и смещения (в количестве регистров) до параметров по ТВ относительно базового адреса.

Список базовых адресов

Наименование	Базовые адреса
TB1	2790
TB2	2808

Описание структуры данных

Название	Смещение	Размер (байт)	Тип	Доступ	Примечание
tнв	0	4	float	R/O	
tx	2	4	float	R/O	
Px	4	4	float	R/O	
dt	6	4	float	R/O	
dM	8	4	float	R/O	
Qтв	10	4	float	R/O	
Q12	12	4	float	R/O	
Qr	14	4	float	R/O	
BHP	16	2	unsigned short	R/O	
BOC	17	2	unsigned short	R/O	

Список полей структуры:

• tнв, tx, Px, dt, dM, Qтв, Q12, Qг, BHP, BOC –температура наружного воздуха, температура холодной воды, давление холодной воды, разность температур, разность масс, тепло по тепловому вводу, тепло контура труб 1,2, тепло горячего водоснабжения, время нормальной работы (ч), время отсутствия счета (ч);

6.9 Архивная запись (итоговый архив)

Структура предназначена для чтения значений архивных параметров из итогового архива. Для получения архивной записи необходимо сообщить прибору за какую дату необходимо извлекать данные (см. «Тип читаемых данных»).

Описание структуры данных

Название	Адрес	Размер (байт)	Тип	Доступ	Примечание
Дата: день	2868	1	unsigned char	R/O	Биты 0-7
Дата: месяц	2868	1	unsigned char	R/O	Биты 8-15
Дата: год-2000	2869	1	unsigned char	R/O	Биты 0-7
Время: час	2869	1	unsigned char	R/O	Биты 8-15
Значения по трубам					См. <u>Значения по</u> <u>трубам (итоги)</u>
Значения по тепловым					См. Значения по ТВ
вводам					<u>(итоги)</u>
Значение по ДП	2964	8	double	R/O	
Длит.работы по сети	2968	4	unsigned long	R/O	
Длит.работы дисплея	2970	4	unsigned long	R/O	
Длит.отсут.сет.питания	2972	4	unsigned long	R/O	
Активная БД	2974	1	unsigned char	R/O	Биты 0-7
СИ ТВ1	2974	1	unsigned char	R/O	Биты 8-15
KT3 TB1	2975	1	unsigned char	R/O	Биты 0-7
ΦPT TB1	2975	1	unsigned char	R/O	Биты 8-15
Активная БД	2976	1	unsigned char	R/O	Биты 0-7
СИ ТВ2	2976	1	unsigned char	R/O	Биты 8-15
KT3 TB2	2977	1	unsigned char	R/O	Биты 0-7
ΦPT TB2	2977	1	unsigned char	R/O	Биты 8-15

Список полей структуры:

- Дата/время: представляет дату/время интервала архивирования. Значение «21.04.11 02» в часовом архиве означает, что архивная запись содержит данные, которые были зарегистрированы по окончании «21.04.11 02». Допустимые значения:
 - День 1-31;
 - Месяц 1-12;
 - Год 0-255;
 - 4ac 0-23.

- Значения по трубам: см далее Значения по трубам (итоги);
- Значения по тепловым вводам: см.далее Значения по тепловым вводам (итоги);
- Значение по ДП: архивное значение по дополнительному параметру, измеренному на дополнительном импульсном входе, в режиме счета импульсов;
- Длит. работы по сети: значение длительности работы по сети в минутах. Фиксируется только при отсутствии внешнего сетевого питания;
- Длит. работы дисплея: значение длительности работы дисплея в минутах. Фиксируется только при отсутствии внешнего сетевого питания;
- Длит. отсут. сет. питания: значение длительности отсутствия внешнего сетевого питания в минутах;
- Активная БД: 0-активна БД1, 1-активна БД2;
- СИ ТВ: значение конфигурационного параметра ТВ;
- КТЗ ТВ: значение конфигурационного параметра КТЗ;
- ФРТ ТВ: значение конфигурационного параметра ФРТ;

6.9.1 Значения по трубам (итоги)

Архивная запись хранит данные 6-ти труб. Структуры данных по трубам идентичны. В данном разделе будут представлены базовые адреса структур данных по трубам и смещения (в количестве регистров) до параметров по трубам относительно базового адреса.

Список базовых адресов

Наименование	Базовые адреса
ТВ1 Труба 1	2870
ТВ1 Труба 2	2878
ТВ1 Труба 3	2886
ТВ2 Труба 1	2894
ТВ2 Труба 2	2902
ТВ2 Труба 3	2910

Описание структуры данных

Название	Смещение	Размер (байт)	Тип	Доступ	Примечание
Пазвание	Смещение	rasmep (dam)	Ιип	доступ	Примечание
V	0	8	double	R/O	
M	4	8	double	R/O	

Список полей структуры:

V, М – параметры объем и масса;

6.9.2 Значения по тепловым вводам (итоги)

Архивная запись хранит данные 2-х тепловых вводов. Структуры данных по ТВ идентичны. В данном разделе будут представлены базовые адреса структур данных по ТВ и смещения (в количестве регистров) до параметров по ТВ относительно базового адреса.

Список базовых адресов

Наименование	Базовые адреса
TB1	2918
TB2	2941

Описание структуры данных

Название	Смещение	Размер (байт)	Тип	Доступ	Примечание
dM	0	8	double	R/O	
Qтв	4	8	double	R/O	
Q12	8	8	double	R/O	
Qг	12	8	double	R/O	
BHP	16	2	unsigned short	R/O	
BOC	17	2	unsigned short	R/O	
TVmin	18	2	unsigned short	R/O	
TVmax	19	2	unsigned short	R/O	
Tdt	20	2	unsigned short	R/O	
Тбез.пит.	21	2	unsigned short	R/O	
Ттнеиспр.	22	2	unsigned short	R/O	

Список полей структуры:

• dM, Qтв, Q12, Qг, BHP, BOC – разность масс, тепло по тепловому вводу, тепло контура труб 1,2, тепло горячего водоснабжения, время нормальной работы (ч), время отсутствия счета (ч), время при HC V<min (ч), время при HC V>max (ч), время при HC по dt (ч), время отключения внешнего сетевого питания (ч), время неисправности t1 или t2;

6.10 Информация об асинхронных архивах

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

Описание структуры данных

Название	Адрес	Размер (байт)	Тип	Доступ	Примечание
Макс.кол-во записей (АИБД)	2996	2	unsigned short	R/O	
Индекс записи (АИБД)	2997	2	unsigned short	R/O	
Размер записи, байт (АИБД)	2998	2	unsigned short	R/O	
Признак закольц. (АИБД)	2999	1(!)	bit field	R/O	Бит 0
Резерв 1 (АИБД)	2999	15(!)	bit field	R/O	Биты 1-15
Макс.кол-во записей (ААС)	3000	2	unsigned short	R/O	
Индекс записи (ААС)	3001	2	unsigned short	R/O	
Размер записи, байт (ААС)	3002	2	unsigned short	R/O	
Признак закольц. (ААС)	3003	1(!)	bit field	R/O	Бит 0
Резерв 1 (ААС)	3003	15(!)	bit field	R/O	Биты 1-15
Макс.кол-во записей (АД)	3004	2	unsigned short	R/O	
Индекс записи (АД)	3005	2	unsigned short	R/O	
Размер записи, байт (АД)	3006	2	unsigned short	R/O	
Признак закольц. (АД)	3007	1(!)	bit field	R/O	Бит 0
Резерв 1 (АД)	3007	15(!)	bit field	R/O	Биты 1-15

Список полей структуры:

- Макс. количество записей: представляет значение максимального количества записей для архива данного типа;
- Индекс записи: представляет значение в диапазоне 0-«Макс. количество записей», и означающее номер записи, в который будет произведено сохранение архивной информации в следующий раз;
- Размер записи: означает размер архивной записи в байтах;
- Признак закольц.: значение 0 означает, что архив еще не заполнен, значение 1 означает, что архив заполнен полностью и сохранение новой записи вызывает стирание самой старой записи архива данного типа.

6.11 Архивные записи асинхронных архивов

Структура предназначена для чтения записей архива изменений базы данных, архива событий и диагностического архива. В данном разделе будут представлены базовые адреса архивных записей и смещения (в количестве регистров) до полей архивных записей относительно базового адреса. Возможно чтение сразу нескольких архивных записей. В этом случае система верхнего уровня указывает индекс первой желаемой записи и запрашивает количество регистров, соответствующее 1, 2, 3...N записям.

Список базовых адресов

Наименование	Базовые адреса первой записи	Количество записей
АИБД	3028	16
AAC	3156	16
АД	3284	16

Описание структуры данных							
Название	Смещение	Размер (байт)	Тип	Доступ	Примечание		
АИБД Дата: день	0	1	unsigned char	R/O	Биты 0-7		
АИБД Дата: месяц	0	1	unsigned char	R/O	Биты 8-15		
АИБД Дата: год - 2000	1	1	unsigned char	R/O	Биты 0-7		
АИБД Время: час	1	1	unsigned char	R/O	Биты 8-15		
АИБД Время: минута	2	1	unsigned char	R/O	Биты 0-7		
АИБД Время: секунда	2	1	unsigned char	R/O	Биты 8-15		
АИБД Событие/Параметр	3	9(!)	bit field	R/O	Биты 0-8		
АИБД № ТВ/Трубы	3	3(!)	bit field	R/O	Биты 9-11		
АИБД Резерв 1	3	3(!)	bit field	R/O	Биты 12-14		
АИБД БД	3	1(!)	bit field	R/O	Бит 15		
АИБД Старое значение	4	4	unsigned char[4]	R/O			
АИБД Новое значение	6	4	unsigned char[4]	R/O			
ААС Дата: день	0	1	unsigned char	R/O	Биты 0-7		
ААС Дата: месяц	0	1	unsigned char	R/O	Биты 8-15		
ААС Дата: год - 2000	1	1	unsigned char	R/O	Биты 0-7		
ААС Время: час	1	1	unsigned char	R/O	Биты 8-15		
ААС Время: минута	2	1	unsigned char	R/O	Биты 0-7		
ААС Время: секунда	2	1	unsigned char	R/O	Биты 8-15		
ААС Событие/Параметр	3	9(!)	bit field	R/O	Биты 0-8		
ААС № ТВ/Трубы	3	3(!)	bit field	R/O	Биты 9-11		
ААС Резерв 1	3	3(!)	bit field	R/O	Биты 12-14		
ААС БД	3	1(!)	bit field	R/O	Бит 15		
ААС Старое значение	4	4	unsigned char[4]	R/O			
ААС Новое значение	6	4	unsigned char[4]	R/O			
АД Дата: день	0	1	unsigned char	R/O	Биты 0-7		
АД Дата: месяц	0	1	unsigned char	R/O	Биты 8-15		
АД Дата: год - 2000	1	1	unsigned char	R/O	Биты 0-7		
АД Время: час	1	1	unsigned char	R/O	Биты 8-15		
АД Время: минута	2	1	unsigned char	R/O	Биты 0-7		
АД Время: секунда	2	1	unsigned char	R/O	Биты 8-15		
АД Событие/Параметр	3	9(!)	bit field	R/O	Биты 0-8		
АД № ТВ/Трубы	3	3(!)	bit field	R/O	Биты 9-11		
АД Резерв 1	3	3(!)	bit field	R/O	Биты 12-14		
АД БД	3	1(!)	bit field	R/O	Бит 15		
АД Старое значение	4	4	unsigned char[4]				
АД Новое значение	6	4	unsigned char[4]	R/O			

- Дата/время: представляет дату/время создания архивной записи. Допустимые значения:
 - День 1-31;
 - Месяц 1-12;
 - Год 0-255;
 - 4ac 0-23
- Событие/Параметр: условный код события или наименование параметра, вызвавшего создание архивной записи;
- №ТВ/Трубы: контекстно означает номер теплового ввода или трубы, в которой произошло изменение параметра (только для АИБД);
- БД: номер базы данных, в которой произошло изменение параметра (только для АИБД);
- Старое/Новое значение: может быть представлено различными типами данных в зависимости от измененного параметра.

6.11.1 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 кодировке от 0 до 5-ти. При этом значения 0-2 кодируют трубы 1-3 ТВ1, значения 3-5 — трубы 1-з ТВ2. Если в столбце «№ ТВ/трубы» указано «ТВ», значит, параметр принадлежит тепловому вводу. Значение 0 в поле «№ТВ/Трубы» кодирует ТВ1, значение 1 — ТВ2. Значение 0 поля «БД» кодирует БД1, значение 1 — БД2.

Nº	Событие	№ TB/Tp.	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	
					0-запрет, 1-с паролем, 2-с	
2	С клав.	-	-	unsigned char	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	ı	-			
10	Резерв 5	-	-			
11	Резерв 6	ı	-			
12	Резерв 7	ı	-			
13	Резерв 8	ı	-			
14	Резерв 9	ı	-			
15	Резерв 10	ı	-			
16	Час отчета	ı	-	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	√ 0	Событие	№ TB/Tp.	БД	Тип значений	Список значений	Примечание
Сист. термопреоб 22 портоворовороворовороворовороворовороворо	20	Логовор		_			
21 единиц - unsigned char 0-CIV, 1-MKC Термопреоб р. р. р - unsigned char 0-100П, 1-500П, 2-Pt100, 3-Pt500 Перев. 23 часов - - unsigned char 0-Bыкл, 1-Bкл 24 Время - - unsigned char[3] Время 25 Пароль - - - - 26 Дата - - - - 27 Peзерв 11 - - - - 28 Peзерв 12 - - - - 29 Peзерв 13 - - - - 30 Peзерв 16 - - - - 31 Peзерв 16 - - - - 32 Peзерв 18 - - - - 31 Peзерв 19 - - - - 32 Peзерв 19 - - - - 34 Peзерв 20 - - - - 37 Kohtp. V typ6a +	-0			_		формируются	
Термопреоб р. перев. 23 часов - чиnsigned char перев. 34 часов - чиnsigned char перев. 35 пароль - чиnsigned char перев. 36 пароль - чиnsigned char перев. 37 пароль - чиnsigned char перев. 38 пароль - чиnsigned char перев. 38 пароль - чиnsigned char перев. 39 пароль - чиnsigned char перев. 39 пароль - чиnsigned char перев. 39 пароль - чиnsigned char перев. 30 пароль перев. 30	1		_	_	unsigned char	0-СИ 1-МКС	
Percent					anoignou onai	· · · · · · · · · · · · · · · · · · ·	
Перев. часов			-	_	unsigned char		
23 часов - unsigned char 0-Bыкл, 1-Вкл 24 Время - unsigned char[3] Время 25 Пароль - -						12.2.2	
25 Пароль - - - - старое и новое зн. не формируются 26 Дата - <td>23</td> <td>•</td> <td>-</td> <td>-</td> <td>unsigned char</td> <td>0-Выкл, 1-Вкл</td> <td></td>	23	•	-	-	unsigned char	0-Выкл, 1-Вкл	
25 Пароль - -	24	Время	-	-	unsigned char[3]	Время	ЧЧ:ММ:СС
26 Дата - - unsigned char[3] Дата 27 PesepB 11 - - - - 28 PesepB 12 - - - - 29 PesepB 13 - - - - 30 PesepB 14 - - - - 31 PesepB 15 - - - - - 32 PesepB 16 -						Старое и новое зн. не	
27 Резерв 11 - -			-	-			
28 Peзерв 12 - <td< td=""><td></td><td></td><td>-</td><td>-</td><td>unsigned char[3]</td><td>Дата</td><td>ДД.ММ.ГГ</td></td<>			-	-	unsigned char[3]	Дата	ДД.ММ.ГГ
29 Резерв 13 - -			-	-			
30			-	-			
31 Резерв 15 - - - - - -			-	-			
32 Резерв 16				-			
33							
34 Резерв 18		•		-			
35 Резерв 19 - - </td <td></td> <td></td> <td>-</td> <td>-</td> <td></td> <td></td> <td></td>			-	-			
36 Резерв 20 - - <t< td=""><td></td><td></td><td>-</td><td>-</td><td></td><td></td><td></td></t<>			-	-			
37 Контр. V труба + unsigned char 0-нет, 1-без подст., 2-Подст и контр U, 4-счет отменен 38 Контр. ВС труба + unsigned char 0-нет, 1-сеть(общ.), 2-Резерв 39 Датчик Р труба + unsigned char 0-нет, 1-да 40 Тип ВС труба + unsigned char 0-механич., 1-Электрон. 41 Рдог труба + float 42 tдог труба + float 43 Рп труба + float 44 Рв труба + float 44 Рв труба + float 46 Vmin труба + float 47 Vmax труба + float 48 Vдог труба + float 49 Резерв 21 - 50 Резерв 23 - - 51 Резерв 2			-	-			
Rohtp.V труба + Unsigned char Подст.,3-Подст и контр U,4-Счет отменен О-нет, 1-сеть(общ.),2-Верв Заратчик Р труба + Unsigned char О-нет, 1-да О-нет, 1-да О-нет, 1-да О-нет, 1-да О-нет, 1-да О-нет, 1-да О-механич.,1-Электрон. О-Механич.,1-	30	Резерв 20	-	-			
37 Контр. V труба + unsigned char Счет отменен							
38 Контр. ВС труба + unsigned char 0-нет, 1-сеть(общ.),2-Резерв 39 Датчик Р труба + unsigned char 0-нет, 1-да 40 Тип ВС труба + unsigned char 0-Механич.,1-Электрон. 41 Рдог труба + float 42 труба + float 43 Рп труба + float 44 Рв труба + float 44 Рв труба + float 46 Vmin труба + float 47 Vmax труба + float 48 Удог труба + float 49 Резерв 21 - 50 Резерв 23 - 51 Резерв 24 - - 52 Резерв 25 - - 53 </td <td>27</td> <td>Kouto V</td> <td>тпуба</td> <td>_</td> <td>unsigned char</td> <td>•</td> <td></td>	27	Kouto V	тпуба	_	unsigned char	•	
38 Контр. BC труба + unsigned char Резерв 39 Датчик Р 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 44 Рв труба + float 46 Vmin труба + float 47 Vmax труба + float 48 Удог труба + float 49 Резерв 21 - - 50 Резерв 23 - - 51 Резерв 24 - - 52 Резерв 25 - - 55) /	контр. у	труба	Т	unsigned chai		
39 Датчик Р труба + unsigned char 0-нет, 1-да 40 Тип ВС труба + unsigned char 0-Механич.,1-Электрон. 41 Рдог труба + float 42 тдог труба + float 43 Рп труба + float 44 Рв труба + float 45 импульса труба + float 46 Vmin труба + float 47 Vmax труба + float 48 Vдог труба + float 49 Резерв 21 - - 50 Резерв 23 - - 51 Резерв 23 - - 52 Резерв 24 - - 53 Резерв 26 - - 55 Резерв 27 <td< td=""><td>88</td><td>Конто ВС</td><td>труба</td><td>+</td><td colspan="2"></td><td></td></td<>	88	Конто ВС	труба	+			
40 Тип BC труба + unsigned char 0-Механич.,1-Электрон. 41 Рдог труба + float 42 tдог труба + float 43 Рп труба + float 44 Рв труба + float 45 импульса труба + float 46 Vmin труба + float 47 Vmax труба + float 48 Vдог труба + float 49 Резерв 21 - 50 Резерв 22 - 51 Резерв 23 - 52 Резерв 24 - 53 Резерв 26 - 54 Резерв 27 - 55 Резерв 28 - <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>							
41 Рдог труба + float 42 tдог труба + float 43 Рп труба + float 44 Рв труба + float 45 импульса труба + float 46 Vmin труба + float 47 Vmax труба + float 48 Vдог труба + float 49 Резерв 21 - 50 Резерв 22 - - 51 Резерв 23 - - 52 Резерв 24 - - 53 Резерв 25 - - 54 Резерв 26 - - 55 Резерв 27 - 56 Резерв 28 - - 57 Резерв 29 - -							
42 tдог труба + float 43 Pп труба + float 44 Pв труба + float 45 импульса труба + float 46 Vmin труба + float 47 Vmax труба + float 48 Vдог труба + float 49 Peзерв 21 - 50 Peзерв 22 - 51 Peзерв 23 - - 52 Peзерв 24 - 52 Peзерв 25 - - 53 Peзерв 26 - - 54 Peзерв 27 - - 55 Peзерв 28 - - 57 Peзерв 29 - -							
43 Pп труба + float 44 Pв труба + float 45 импульса труба + float 46 Vmin труба + float 47 Vmax труба + float 48 Vдог труба + float 49 Peзерв 21 - 50 Peзерв 22 - - 51 Peзерв 23 - - 52 Peзерв 23 - - -				+			
44 Pв труба + float 45 импульса труба + float 46 Vmin труба + float 47 Vmax труба + float 48 Vдог труба + float 49 Peзерв 21 - 50 Peзерв 22 - 51 Peзерв 23 - 52 Peзерв 24 - - <	13			+	float		
45 импульса труба + float 46 Vmin труба + float 47 Vmax труба + float 48 Vдог труба + float 49 Peзерв 21 - 50 Peзерв 22 - 51 Peзерв 23 - 52 Peзерв 23 - 52 Peзерв 24 - - 53 Peзерв 25 - 54 Peзерв 26 - - 55 Peзерв 27 - 56 Peзерв 28 - - 57 Peзерв 29 - -		Рв		+			
46 Vmin труба + float 47 Vmax труба + float 48 Vдог труба + float 49 Peзерв 21 - 50 Peзерв 22 - 51 Peзерв 23 - 52 Peзерв 24 - 53 Peзерв 25 - 54 Peзерв 26 - 55 Peзерв 27 - 56 Peзерв 28 - - 57 Peзерв 29 - -		Bec					
47 Vmax труба + float 48 Vдог труба + float 49 Peзерв 21 - 50 Peзерв 22 - 51 Peзерв 23 - 52 Peзерв 24 - 53 Peзерв 25 - 54 Peзерв 26 - 55 Peзерв 27 - 56 Peзерв 28 - 57 Peзерв 29 -			труба	+			
48 Vдог труба + float 49 Резерв 21				+			
49 Pesepb 21 - <t< td=""><td></td><td></td><td></td><td>+</td><td></td><td></td><td></td></t<>				+			
50 Pesepb 22 - <t< td=""><td></td><td></td><td>труба</td><td>+</td><td>float</td><td></td><td></td></t<>			труба	+	float		
51 Pesepb 23 - <t< td=""><td></td><td></td><td>-</td><td>-</td><td></td><td></td><td></td></t<>			-	-			
52 Pesepb 24 - <t< td=""><td></td><td></td><td>-</td><td>-</td><td></td><td></td><td></td></t<>			-	-			
53 Резерв 25 - 54 Резерв 26 -			-	-			
54 Резерв 26 - 55 Резерв 27 - 56 Резерв 28 - 57 Резерв 29 -				-			
55 Резерв 27 - 56 Резерв 28 - 57 Резерв 29 -							
56 Резерв 28 - 57 Резерв 29 -							
57 Резерв 29							
			- TD	-	uncianad char		
)		
62 Рхд ТВ + float 63 KT3 ТВ + unsigned char							
64 ΦPT TB + unsigned char	34						

		Nº			_	
Nº	Событие	TB/Tp.	БД	Тип значений	Список значений	Примечание
65	Контр.t	TB	+	unsigned char	0-с подст.,1-счет отмен.	
					0-Нет,1-Без подст.1,2-Без	
					подст.2,3-С подст.1,4-С	
66	Контр.dM	TB	+	unsigned char	подст.2	
					0-Нет, 1-Без подст., 2-С	
67	Контр.Q	TB	+	unsigned char	подст.	
				0-без подст., 1-с подст., 2-		
68	Контр.dt	TB	+	unsigned char	счет отмен.	
69	dtmin	TB	+	unsigned char 0-2°C, 1-3°C		
				0-Не исп., 1-догов., 2-изм.в		
					данном ТВ, 3-изм. в другом	
70	Исп.tx	TB	+	unsigned char	ТВ	
71	Исп.Рх	TB	+	unsigned char	0-Не изм., 1-Изм.	
72	Резерв 31	-	-			
73	Резерв 32	-	-			
74	Резерв 33	-	-			
75	Резерв 34	-	-			
76	Резерв 35	-	-			
77	Резерв 36	-	-			
78	Резерв 37	-	-			
79	Резерв 38	-	-			
80	Резерв 39	-	-			
81	Резерв 30	-	-			
					0-нет, 1-контр.сети, 2-счет	
82	Назнач. ДП	-	-	unsigned char	имп., 3-сигн-ия	
83	Уровень	-	-	unsigned char	0-HP, 1-H3	
84	Ед.изм.	-	-	unsigned char	0-м3, 1кВт*ч	
85	Вес имп.	-	-	float	,	
86	ДПmin	-	-	float		
87	ДПтах	_	-	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	-	_			
JU	i coche 40					

6.11.2 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.

авто	ризацию. 10 имеет тип инзідней юпу.		
Nº	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.t(ПК)	+	
6	Калибр.Р(ПК)	+	
7	Калибр.t		
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. сервцентра	+	
28	Авто перевод часов		
29	Очистка архивов	+	
30	Замена батареи		

6.11.3 Интерпретация диагностического архива

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

Nº	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	

6.12 Текущие итоговые значения

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

Описание структуры данных

Название	Адрес	Размер (байт)	Тип	Доступ	Примечание
Дата: день	3412	1	unsigned char	R/O	Биты 0-7
Дата: месяц	3412	1	unsigned char	R/O	Биты 8-15
Дата: год-2000	3413	1	unsigned char	R/O	Биты 0-7
Время: час	3413	1	unsigned char	R/O	Биты 8-15
Время: минута	3414	1	unsigned char	R/O	Биты 0-7
Время: секунда	3414	1	unsigned char	R/O	Биты 8-15
Значения по трубам					См. <u>Знач. по трубам</u> (тек.итоги)
Значения по тепловым вводам					См. <u>Знач. по ТВ</u> (тек.итоги)
Значение по ДП	3509	8	double	R/O	
Длит.работы по сети	3513	4	unsigned long	R/O	
Длит.работы дисплея	3515	4	unsigned long	R/O	
Длит.отсут.сет.питания	3517	4	unsigned long	R/O	
Активная БД	3519	1	unsigned char	R/O	Биты 0-7
СИ ТВ1	3519	1	unsigned char	R/O	Биты 8-15
KT3 TB1	3520	1	unsigned char	R/O	Биты 0-7
ΦPT TB1	3520	1	unsigned char	R/O	Биты 8-15
Активная БД	3521	1	unsigned char	R/O	Биты 0-7
СИ ТВ2	3521	1	unsigned char	R/O	Биты 8-15
KT3 TB2	3522	1	unsigned char	R/O	Биты 0-7
ΦPT TB2	3522	1	unsigned char	R/O	Биты 8-15

Список полей структуры:

- Дата/время: представляет дату/время в приборе на момент съема информации.
 Допустимые значения:
 - День 1-31;
 - Месяц 1-12;
 - Год 0-255;
 - 4ac 0-23;
 - Минута;
 - Секунда.
- Значения по трубам: см Значения по трубам (текущие итоги);
- Значения по тепловым вводам: см. <u>Значения по тепловым вводам (текущие итоги)</u>;
- Значение по ДП: текущее итоговое значение по дополнительному параметру, измеренному на дополнительном импульсном входе, в режиме счета импульсов;
- Длит. работы по сети: значение длительности работы по сети в минутах. Фиксируется только при отсутствии внешнего сетевого питания;
- Длит. работы дисплея: значение длительности работы дисплея в минутах. Фиксируется только при отсутствии внешнего сетевого питания;
- Длит. отсут. сет. питания: значение длительности отсутствия внешнего сетевого питания в минутах;
- Активная БД: 0-активна БД1, 1-активна БД2;
- СИ ТВ: значение конфигурационного параметра ТВ;
- КТЗ ТВ: значение конфигурационного параметра КТЗ;
- ФРТ ТВ: значение конфигурационного параметра ФРТ;

6.12.1 Значения по трубам (текущие итоги)

Структура хранит данные 6-ти труб. Структуры данных по трубам идентичны. В данном разделе будут представлены базовые адреса структур данных по трубам и смещения (в количестве регистров) до параметров по трубам относительно базового адреса.

Список базовых адресов

Наименование	Базовые адреса
ТВ1 Труба 1	3415
ТВ1 Труба 2	3423
ТВ1 Труба 3	3431
ТВ2 Труба 1	3439
ТВ2 Труба 2	3447
ТВ2 Труба 3	3455

Описание структуры данных

Название	Смещение	Размер (байт)	Тип	Доступ	Примечание
V	0	8	double	R/O	
M	4	8	double	R/O	

Список полей структуры:

• V, M – параметры объем и масса;

6.12.2 Значения по тепловым вводам (текущие итоги)

Структура хранит данные 2-х тепловых вводов. Структуры данных по ТВ идентичны. В данном разделе будут представлены базовые адреса структур данных по ТВ и смещения (в количестве регистров) до параметров по ТВ относительно базового адреса.

Список базовых адресов

Наименование	Базовые адреса
TB1	3463
TB2	3486

Описание структуры данных

описание структуры данных							
Название	Смещение	Размер (байт)	Тип	Доступ	Примечание		
dM	0	8	double	R/O			
Qтв	4	8	double	R/O			
Q12	8	8	double	R/O			
Qг	12	8	double	R/O			
BHP	16	2	unsigned short	R/O			
BOC	17	2	unsigned short	R/O			
TVmin	18	2	unsigned short	R/O			
TVmax	19	2	unsigned short	R/O			
Tdt	20	2	unsigned short	R/O			
Тбез.пит.	21	2	unsigned short	R/O			
Ttнеиспр.	22	2	unsigned short	R/O			

Список полей структуры:

dM, Qтв, Q12, Qг, BHP, BOC – разность масс, тепло по тепловому вводу, тепло контура труб 1,2, тепло горячего водоснабжения, время нормальной работы (ч), время отсутствия счета (ч), время при HC V<min (ч), время при HC V>max (ч), время при HC по dt (ч), время отключения внешнего сетевого питания (ч)), время неисправности t1 или t2;

6.13 Текущие значения

Структура предназначена для чтения текущих измеренных (вычисленных) параметров.

Описание структуры данных

Название	Адрес	Размер (байт)	Тип	Доступ	Примечание
Дата: день	3540	1	unsigned char	R/O	Биты 0-7
Дата: месяц	3540	1	unsigned char	R/O	Биты 8-15
Дата: год-2000	3541	1	unsigned char	R/O	Биты 0-7
Время: час	3541	1	unsigned char	R/O	Биты 8-15
Время: минута	3542	1	unsigned char	R/O	Биты 0-7
Время: секунда	3542	1	unsigned char	R/O	Биты 8-15
t1-t6	3543	24	float[6]	R/O	TB1: t1-t3, TB2: t4-t6
P1-P6	3555	24	float[6]	R/O	TB1: P1-P3, TB2: P4-P6
Go1-Go6	3567	24	float[6]	R/O	TB1: Go1-Go3, TB2: Go4-Go6
					ТВ1: Gм1-Gм3, ТВ2: Gм4-
Gм1-Gм6	3579	24	float[6]	R/O	Gм6
Ф1-Ф6	3591	24	float[6]	R/O	ТВ1: Ф1-Ф3, ТВ2: Ф4-Ф6
h1-h6	3603	24	float[6]	R/O	TB1: h1-h3, TB2: h4-h6
Фтв1-Фтв2	3615	8	float[2]	R/O	
hx1-hx2	3619	8	float[2]	R/O	
ДП	3623	4	float	R/O	
НС по трубам 1-6	3625	6	unsigned char[6]	R/O	TB1: HC1-HC3, TB2: HC4-HC6
НС по ТВ1,2	3628	4	unsigned short[2]	R/O	
НС по ДП	3630	1	unsigned char	R/O	Биты 0-7
Резерв 1	3630	1	unsigned char	R/O	Биты 8-15
Признаки событий	3631	2	unsigned short	R/O	
Резерв 2	3632	2	unsigned short	R/O	
tx1-tx2	3633	8	float[2]	R/O	
Px1-Px2	3637	8	float[2]	R/O	
dt1-dt2	3641	8	float[2]	R/O	
tнв1-tнв2	3645	8	float[2]	R/O	
Активная БД	3649	1(!)	bit field	R/O	Бит 0
Резерв 1	3649	15(!)	bit field	R/O	Биты 1-15

Список полей структуры:

Дата/время: представляет дату/время в приборе на момент съема информации. Допустимые значения:

- День 1-31;
- Месяц 1-12;
- Год 0-255;
- 4ac 0-23;
- Минута;
- Секунда.
- t1-t6: значения температур. Значения t1, t2, t3 представляют t1, t2, t3 по ТВ1. Значения t4, t5, t6 представляют t1, t2, t3 по ТВ2;
- Р1- Р6: значения давлений. Значения Р1, Р2, Р3 представляют Р1, Р2, Р3 по ТВ1. Значения Р4, Р5, Р6 представляют Р1, Р2, Р3 по ТВ2;
- Go1-Go6: значения объемного расходов. Значения Go1, Go2, Go3 представляют Go1, Go2, Go3 по ТВ1. Значения Go4, Go5, Go6 представляют Go1, Go2, Go3 по ТВ2;
- Gм1-Gм6: значения массового расходов. Значения Gм1, Gм2, Gм3 представляют Gм1, Gм2, Gм3 по ТВ1. Значения Gм4, Gм5, Gм6 представляют Gм1, Gм2, Gм3 по ТВ2;
- Ф1-Ф6: значения теплового потока по отдельным трубам. Значения Ф1, Ф2, Ф3 представляют Ф1, Ф2, Ф3 по ТВ1; Значения Ф4, Ф5, Ф6 представляют Ф1, Ф2, Ф3 по ТВ2.
- h1-h6: значения энтальпии. Значения h1, h2, h3 представляют h1, h2, h3 по ТВ1; Значения h4, h5, h6 представляют h1, h2, h3 по ТВ2;

- Фтв1-Фтв2: значения теплового потока по тепловым вводам;
- hx1- hx2: значения энтальпии холодной воды по тепловым вводам;
- ДП: значение «расхода» по параметру, измеренному на дополнительном импульсном входе в режиме счета импульсов;
- НС по трубам 1-5. НС1-НС3 представляют нештатные ситуации по трубам ТВ1, НС4-НС6 — ТВ2. Кодирование НС битами полностью аналогично кодированию в архиве (см. Архивная запись (часовой, суточный, месячный архивы));
- НС по ТВ1,2: Кодирование НС битами полностью аналогично кодированию в архиве (см. Архивная запись (часовой, суточный, месячный архивы));
- НС по ДП: Кодирование НС битами полностью аналогично кодированию в архиве (см. <u>Архивная запись (часовой, суточный, месячный архивы)</u>);
- Признаки событий: биты кодируют наличие различных событий. Кодирование битами полностью аналогично кодированию в архиве (см. <u>Архивная запись (часовой, суточный, месячный архивы)</u>);
- tx1-tx2: Значения температуры холодной воды в тепловых вводах;
- Px1-Px2: Значения давления холодной воды в тепловых вводах;
- dt1- dt2: Значения разности температур в тепловых вводах;
- thв1- thв2: Значения температур наружного воздуха в тепловых вводах;
- Активная БД: 0-активна БД1, 1-активна БД2;

6.14 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.

Описание структуры данных

Название	Адрес	Размер (байт)	Тип	Доступ	Примечание
Номер параметра	3796	2	unsigned short	R/W*	
Значение параметра	3797	4	float	R/W*	
Резерв 1	3799	4	Unsigned char[4]	R/W*	

Поле «Номера параметра» может принимать следующие значения:

- 0 коррекция времени. Поле «Значение параметра» содержит величину коррекции времени. Допустимый интервал ±900 секунд. Если текущее время прибора находится в пределах ЧЧ:20:00 ÷ 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-значение txд TB1 или TB2 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.

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

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

```
WORD Crc16(BYTE *Data, ULONG size)
{
   {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++)</pre>
        {/*обработка байта*/
        if((Sum.w&0x1) == 1)
        Sum.w=(unsigned short)((Sum.w>>1)^0xa001U);
   else
        Sum.w>>=1;
   }
   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 Longitudinal Redundancy Checking (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;
}</pre>
```

где:

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

Приложение 3. Функции преобразования из ASCII формата в двоичный и обратного преобразования из двоичного в **ASCII**

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

```
unsigned char CharToBin[23]=\{0,1,2,3,4,5,6,7,8,9,
0,0,0,0,0,0,0,
10,11,12,13,14,15};
        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:

описатель даппых имеет вледующую структуру.							
Поле	Тип	Размер	Примечание				
		(байт)					
Тип	целое	4	Определяет тип хранимых данных				
Длина	целое	4	Определяет длину одного элемента данных (байт)				
Количество	целое	4	Определяет количество элементов данных (*)				
Доп.	целое	4	Описывается контекстно в зависимости от поля				
информация			«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\r\rangle\r
- 1 элемент данных представлен структурой «Информация об устройстве»;
- 2 элемент данных представлен структурой «Информация о настройках»;
- 3 элемент данных представлен структурой «Настройки вычислителя (настройки защищенные)»;

- 4 Элемент данных содержит значение контрольной суммы незащищенных настроек. Остальные байты анализировать не нужно;
- 5 элемент данных представлен структурой «Настройки вычислителя (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»;
- 11 элемент данных представлен структурой «Текущие итоговые значения»;
- 12, 13, 14 элементы данных представлены структурой «Архивная запись (часовой, суточный, месячный архивы)». Тип 12 записи представляют часовой архив, 13 суточный архив, 14 месячный архив;
- 15 элемент данных представлен структурой «Архивная запись (итоговый архив)»:
- 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;
- 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];
                     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);
```

```
HID WriteFile(THidDevice * pHidDevice, BYTE* buffer, DWORD bufferSize);
void HID Init
              (THidDevice * pHidDevice);
void
     HID UnInit (THidDevice * pHidDevice);
     HID Open (THidDevice * pHidDevice, WORD vid, WORD pid, TTrackSerialNumbers * pTrackSerialNumbers);
     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++) {</pre>
      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++) {</pre>
        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) {</pre>
                                             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) {</pre>
            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) {</pre>
              if(count == 0) {
                      hHidDeviceHandle = CreateFile(devicePath, GENERIC READ | GENERIC WRITE, 0, 0, 0PEN 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(preparsedData, &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 FreePreparsedData(preparsedData);
              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// DEBUG
       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;
                             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;
             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)) {</pre>
                                  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;</pre>
             nRetCode = 1;
      else
             THidDevice hidDevice;
             TTrackSerialNumbersAr serialNumListAr;
             BYTE outbuffer[] = \{0x00, 0x48, 0x00, 0x00, 0x00, 0x13, 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);
       default:
              MyAddLogString("Read error!", true);
              break;
       HID Close(&hidDevice);
return nRetCode;
```