

***Драйвер обмена с прибором контроля положения
задвигки «ПКП1Т»***

Руководство пользователя

Содержание

1. Назначение драйвера	3
2. Краткое описание прибора «ПКП1 Т»	5
3. Конфигурация драйвера	6
4. Конфигурация сервера	8
5. Структура пакетов обмена между «Модулем опроса» и драйвером	13
6. Протокол обмена драйвера с прибором «ПКП1 Т»	15

1. Назначение драйвера.

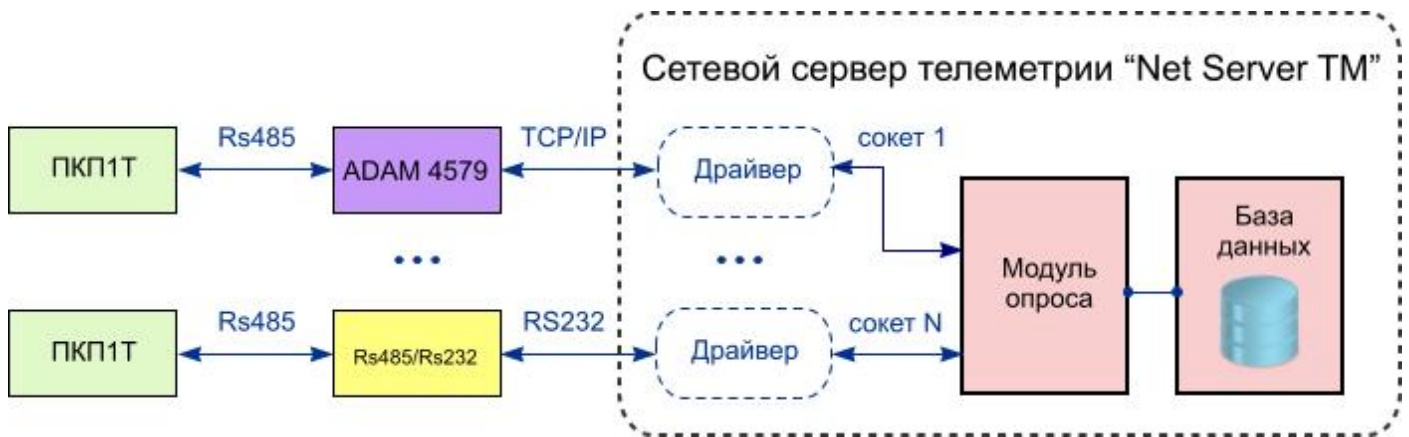


Рис. 1. Структурная схема взаимодействия драйвера с «модулем опроса».

Драйвер служит интерфейсным модулем между прибором «ПКП1Т» (далее по тексту устройством) и «модулем опроса», который входит в состав сервера телеметрии «Net server TM». (См. Рис.1.) «Модуль опроса» общается с драйвером с помощью пакетов-посылок (заказов). Более подробно о структуре пакетов изложено в п.5. Между драйвером и устройством обмен информацией осуществляется с использованием протокола, описанного в п.6. При запуске «модуля опроса» автоматически запускается драйвер с помощью командной строки, сформированной на основании заданных в конфигурации объектов. В командной строке указываются параметры инициализации, среди них: системный IP-адрес и порт для соединения с устройством по протоколу TCP/IP, либо порт и параметры последовательного соединения с устройством, параметры для инициализации диагностики и др. При некорректной инициализации (отсутствие необходимого параметра инициализации, либо присвоение ему некорректного значения) драйвер завершает работу.

После запуска драйвер пытается установить соединение с устройством. В случае неудачного соединения с устройством драйвер повторяет попытки соединения с периодом 20 сек. до нормального завершения. В случае удачного соединения с устройством, драйвер регулярно его опрашивает.

Драйвер на основании полученного от «модуля опроса» заказа формирует и отправляет посылку устройству. Получив ответ от устройства, драйвер преобразует полученную информацию в пакет посылок и отправляет «модулю опроса». В соответствии с заказами «модуля опроса» драйвер может:

- запрашивать текущие данные;
- отдавать команды по управлению задвижкой.

В каждый момент времени драйвер обрабатывает не более одного информационного запроса. При некорректном заказе драйвер выдает «модулю

опроса» признак ошибки обработки заказа. Информационные запросы и команды управления задвижкой выдаются «модулем опроса» по разным сокетам. Драйвер завершает свою работу:

- при закрытии «модуля опроса»;
- при отсутствии заказов от «модуля опроса» в течение времени, заданного в конфигурации.

Для запуска драйвера требуется установленный интерпретатор языка Tcl (Tcl/Tk версии 8.4 и выше), например ActiveTcl 8.4.4.0.

Требования к среде для установки ActiveTcl 8.4.4.0:

ОС:	Аппаратное оборудование:	Версия ОС:
Irix	SGI (Mips)	6.3+
HP-UX	HP (PARISC)	10.20+
Linux	Intel	Red Hat 7.0+ or SuSE 6.0+ *
Solaris	Sun (Sparc)	2.5+
Windows	Intel	NT 3.51/NT 4.0/2000/XP

Свободное место на диске: 2000 Мб

* рекомендуемые требования

2. Краткое описание прибора «ПКП1Т»

1.1 Прибор предназначен для дистанционного управления электроприводом задвижки или затвора без применения конечных выключателей, для чего он контролирует время хода и ток потребляемый электроприводом. Прибор обеспечивает автоматическую остановку электропривода в крайних положениях и формирует сигнал соответствующего конечного положения. Кроме того, прибор отключает электропривод и формирует сигнал «Авария» при заклинивании в процессе движения или холостом ходе механизмов привода. В качестве датчика тока могут применяться либо трансформатор тока (только для приводов переменного тока), либо датчик тока на эффекте Холла

(как для переменного, так и для постоянного тока) с унифицированным выходным сигналом.

1.2 Прибор имеет пять встроенных электромагнитных реле. Два для управления пускателями электропривода или промежуточными реле при управлении цепями постоянного тока, два для выдачи сигналов конечных положений и сигнала «Авария».

1.3 Прибор, по желанию заказчика, может быть оснащен: либо модулем с токовым выходом 4...20 мА для регистрации положения задвижки или затвора, либо модулем связи (RS485) с компьютером для регистрации положения и управления приводом.

1.4 Прибор предназначен для использования в следующих условиях:
Допустимая температура воздуха, окружающего корпус прибора +1...+50 С
Относительная влажность воздуха (при температуре +35 С) не более 80 %
Атмосферное давление 86...107 кПа

Характеристики интерфейса RS485

Скорость обмена 2400...57600 бит/с

Длина линии связи до 1000 м

3. Конфигурация драйвера

3.1. Командная строка вызова.

```
./pkport SERIAL=dev,speed,parity,data_b,stop_b PORT=Nport  
  DEVICES=N1,...,Nn [TKILL=suic_tout] [LOG=log_file]  
  [DEBUG=dbg_val] [TUPORT=Ntuport] [CONF=config_file]
```

или

```
./pkpport IP=ip_addr_or_name:ip_port PORT=Nport  
  DEVICES=N1,...,Nn [TKILL=suic_tout] [LOG=log_file]  
  [DEBUG=dbg_val] [TUPORT=Ntuport] [CONF=config_file]
```

где pkpport – имя запускаемого модуля (имя драйвера);

dev – устройство, обслуживающее COM-порт;

speed – скорость в бодах;

parity – чётность (всегда n, сохранена для «общности» формы);

data_b – количество бит в байте (всегда 8, сохранено для «общности» формы);

stop_b – количество стоповых бит (1 или 2);

ip_addr_or_name:ip_port – ip-адрес и порт (сокет), через который драйвер ведёт диалог с устройствами (драйвер является клиентом);

Nport – сокетный порт верхнего уровня, по которому поступают информационные запросы и отправляются ответы;

Ntuport – сокетный порт верхнего уровня, по которому поступают команды на управление задвижкой;

N1...Nn= имена устройств, с первой встреченной в имени десятичной цифры начинается адрес устройства;

suic_tout – таймаут в секундах «самоубийства» программы при отсутствии сокетных запросов (умолчание: 0 – никогда);

log_file – файл журнала (умолчание: стандартный вывод – экран);

config_file – файл конфигурации (умолчание: tekport.conf);

dbg_val – битовое поле (hex) разрешения вывода в журнал отладочной информации:

FD_OK 1 (OK message)

FD_CPACK 2 (device dialog)

FD_MESS 4 (result message)

FD_INPACK 8 (socket dialog - in packets)

FD_OUTPACK 10 (socket dialog - out packets)

FD_TIME 20 (time output)

Примеры:

```
pkpport SERIAL=/dev/ttyS1,19200,n,8,2 PORT=7720 TUPORT=7721 DEVICES=7  
TKILL=3600
```

```
pkpport IP=10.0.1.27:5201 PORT=7720 TUPORT=7721 DEVICES=1,tc16,22
LOG=dep.log DEBUG=18
```

Строка запуска также поясняется при вызове драйвера без аргументов:
./pkpport

3.2. Список параметров

Имя	Ответ	Флаги	Назначение
«Zdv»	«Zdv=»	0	Опрос текущего положения задвижки
«treg»	«treg=»	FP_U	Команда управления задвижкой
«s-time»	«time=»	0	Опрос времени

Флаги: FP_U – запрос по сокету управления задвижкой.

При запросе параметр "s-time" для устройства возвращается время драйвера.

3.3. Файл конфигурации

Файл `pkpport.conf` (имя может быть специфицировано в строке запуска драйвера) регулярно перепрочитывается (с интервалом 10 секунд). Строка относящаяся к устройству `dev` имеет вид:

```
dev [oktout=5] [tutout=1] [debug=ffff] [log=]
```

Указанные значения – умолчания. Примеры:

```
1 debug=18
```

```
s50 ertout=600 poll=3
```

dev - `dev`, указываемый при обращении к драйверу через сокет;

oktout - таймаут в секундах опроса устройства;

tutout - таймаут в секундах опроса положения задвижки после выдачи команды управления задвижкой;

debug - режим вывода отладочных сообщений (см. строку запуска);

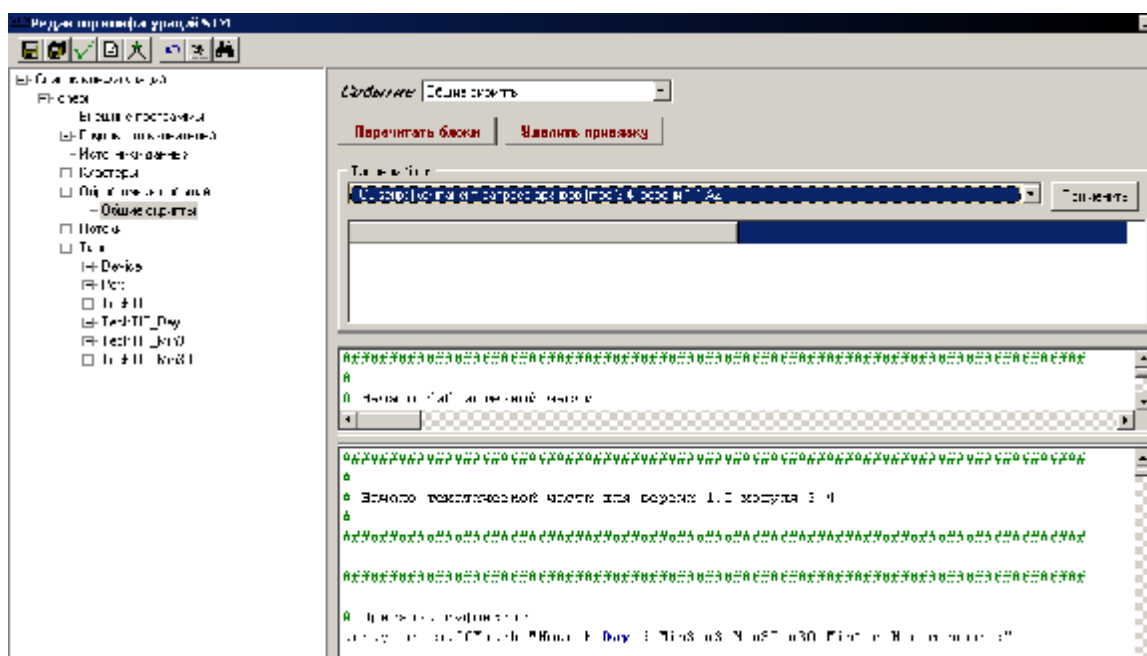
log - файл журнала.

Область действия **debug** & **log** - весь драйвер, а не конкретное устройство (но в строке устройства обслуживаемого драйвером).

4. Конфигурация сервера.

Создание конфигурации осуществляется с помощью программы **STMConf** (C:\Igel\StmConf\StmConf.exe). В данном разделе будут описаны настройки, необходимые для взаимодействия сервера телеметрии и драйвера. Подробное описание работы с конфигуратором **StmConf** и создание конфигураций содержит документ «Конфигуратор сервера телеметрии StmConf. Руководство пользователя». Также дополнительную информацию можно найти в web-справочнике, нажав F1 в IgelView3, раздел Средства конфигурации > Конфигуратор сервера телеметрии StmConf.

- 4.1. Обработчики событий.** В конфигурации сервера телеметрии должны присутствовать следующие обработчики событий
- Модуль3-4 (Компонент запроса архивов (mod 3-4) версии 1.1 AZ)



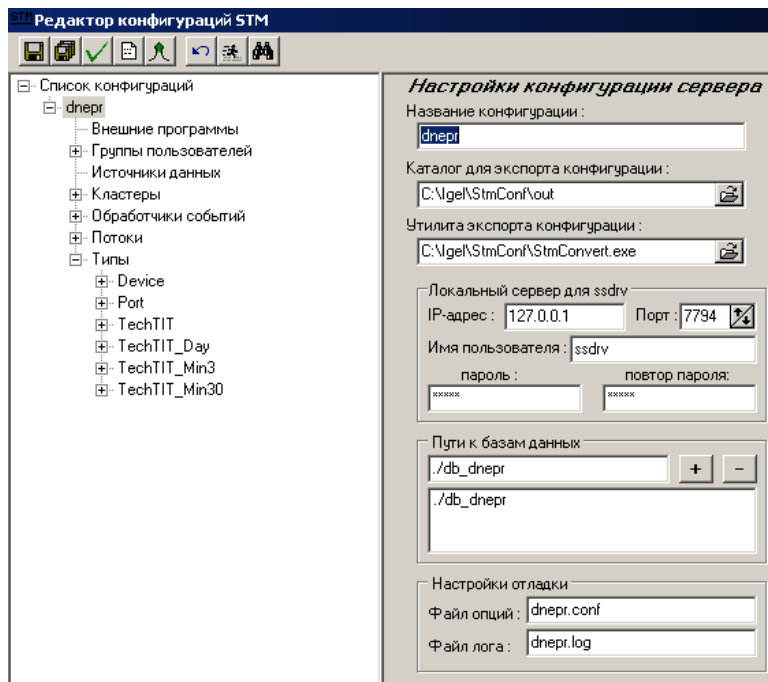
- 4.2. Типы объектов и устройств.** В конфигурации должны присутствовать следующие типы

Port – Приборный порт, описывает режимы работы с конвертерами ADAM 4579 или MOXA;

Device – Описание прибора;

TechTIT – положение задвижки – текущие значения.

На основании введенной информации в объектах типа **Port** и **Device** «модуль опроса» сформирует командные строки и запустит копии драйвера.

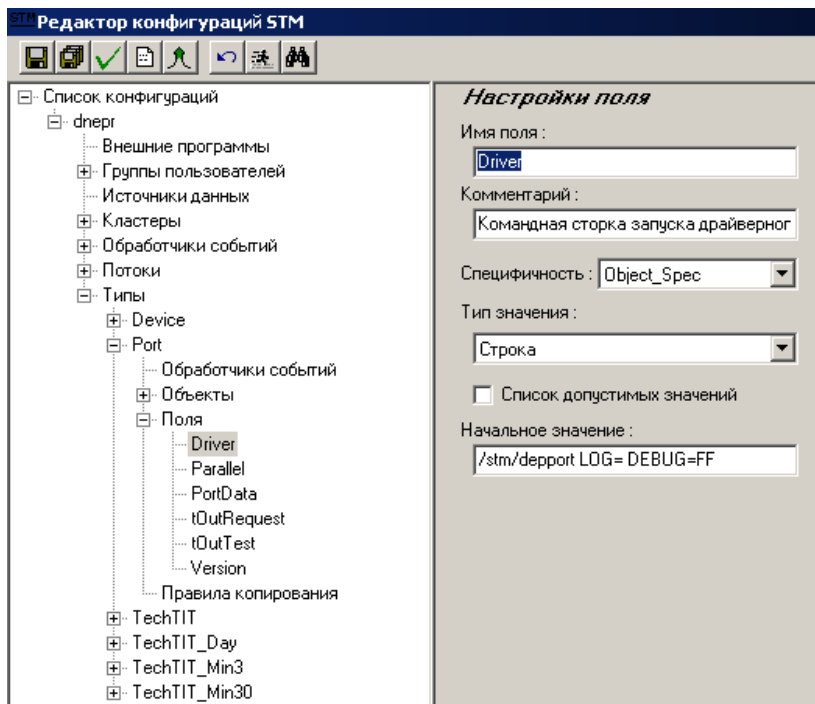


4.3. Структура типов.

Типы должны иметь следующую структуру полей:

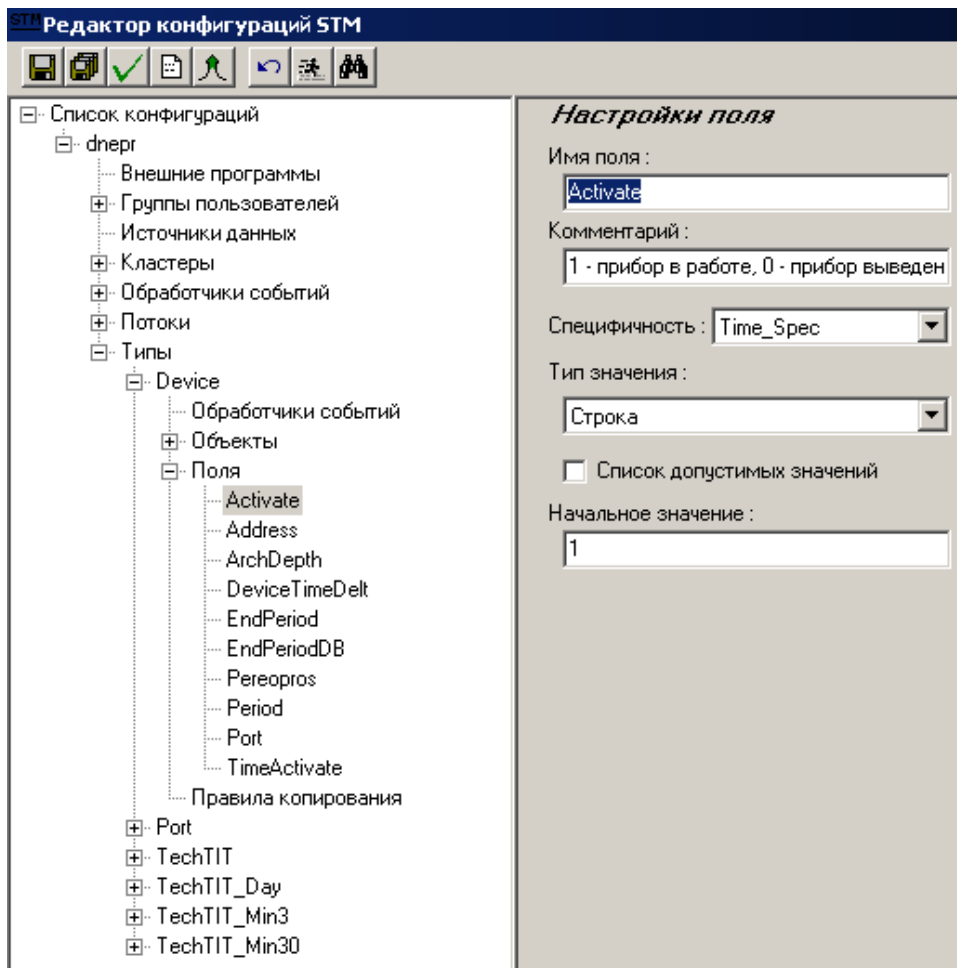
Port

Тип поля	Наим. Поля	Комментарий	Пример значения
String	Driver	Командная строка запуска драйверного модуля	/stm/pkppport LOG=130Ps25st1.log DEBUG=FF
Byte	Parallel	Признак параллельности	1
String	PortData	Характеристики порта	10.12.20.16:4001
Long	tOutRequest	таймаут отправляемый в запросе	15000
Long	tOutTest	таймаут тестовых запросов	80000
Byte	Version	Признак для модуля № 4	0



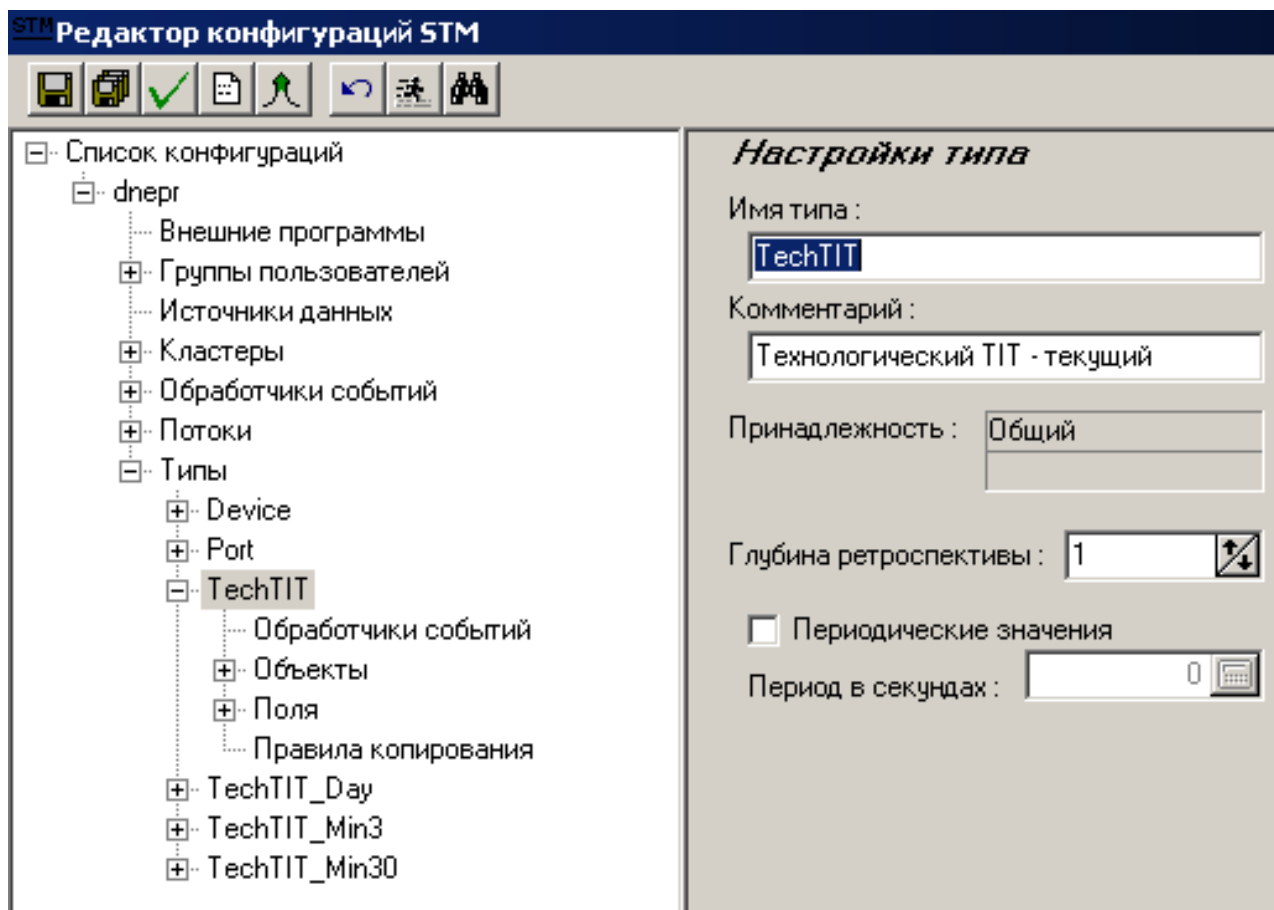
Device

Тип поля	Наим. Поля	Комментарий	Пример значения
Byte	Activate	1 - прибор в работе, 0 – прибор выведен из работы	1
Int	Address	Адрес устройства	15
Long	DeviceTimeDelt	Опережение времени устройства относительно времени сервера (в секундах)	0
Int	EndPeriod	Запрос по концу/началу	
Int	EndPeriodDB	Запись в базу по концу/началу	
Date	Pereopros	Переопрос устройства с дата:время	
uLong	Period	Период опроса текущих значений в мс	0
String	Port	Имя объекта типа порт, которому соответствует прибор	port1
Time	TimeActivate	время вывода прибора из работы	0



TechTIT

Тип поля	Наим. поля	Коммент.	Пример значения
Byte	_avalid	Значение достоверности от оборудования	
Double	_avalue	значение телеизмерения от оборудования	
Double	_mvalue	значение телеизмерения от ручного ввода	
Double	_nom	Номинальное значение	
Byte	Arc	Номер параметра в приборе	
String	Calc	Формула пересчета	
String	Comment	Комментарий	
String	Device	Имя прибора, к которому привязан тит	
Double	HiAlarm	Значение верхней аварийной уставки	
Double	HiTech	Значение верхней предупредительной уставки	
Byte	Limit	Признак выхода за уставку (-2,...,2)	
Double	LoAlarm	Значение нижней аварийной уставки	
Double	LoTech	Значение нижней предупредительной уставки	
Byte	Man,	признак режима ручного ввода (0/1)	
uLong	Proc	Время работы за период	
Double	Value	значение телеизмерения	



4.4. Имена объектов в типах должны иметь следующий вид:
<объект>-W, в соответствии с документом «Посылки между модулями № 2 и № 4 сервера ввода-вывода электрической тематики»
Например: Obj34-W

5. Структура пакетов обмена между «Модулем опроса» и драйвером.

Структура пакетов обмена между «модулем опроса» и драйвером подробно описана в документе ««Модуль опроса». Руководство пользователя». Типы запросов, ответов и значения параметров запросов приводятся в таблице 1, 2 и 3.

Таблица 1. Типы запросов.

Пакет	Описание
{ num=N }\n	Пакет контроля работоспособности соединения
{ num=N type=c par=x dev=y arc=z tout=TO } \n	Запрос текущего значения параметра x устройства у
{ num=N type=c par=x dev=y arc=z trac=1 tout=TO } \n	Запрос текущего значения параметра x устройства у с отслеживанием асинхронных ответов (от устройства будут приходить посылки с номером N на любое изменение параметра x до тех пор, пока не придет запрос текущего значения на параметр x с trac=0)
{ num=N type=c par=s-time dev=y tout=TO }\n	Запрос текущего времени устройства у.

Таблица 2. Типы ответов.

Пакет	Описание
{ num=N }\n	Пакет контроля работоспособности соединения
{ num=N type=c par=x dev=y arc=z sit=s «параметр»=«значение» } \n	Ответ на запрос текущего значения параметра x устройства у

Список кодов ошибок, возвращаемых как значение параметра S.

1. Критическая ошибка. Дальнейшее выполнение невозможно. Необходим рестарт модуля и/или среды выполнения.
2. Тяжелая протокольная ошибка. Нарушение формата пакетов. Разбор пакета невозможен.
Пример : неверная структура пакета, недопустимые символы и т.д.
3. Протокольная ошибка. Нераспознанные ключевые слова в пакете. Пакет разбирается нормально, нераспознанные ключевые слова игнорируются.
4. Ошибка оборудования. Недоступный порт связи
5. Ошибка оборудования. Недоступное устройство.
6. Ошибка оборудования. Отказ устройства.

7. Ошибка оборудования. Отсутствие датчика.
8. Ошибка оборудования. Неисправность датчика.
9. Ошибка оборудования. Отсутствие связи с устройством по таймауту.
10. Ошибка конфигурации. Отсутствие запрошенного параметра в приборе.

Алгоритм выставления значения параметра «sit».

При запросе текущих значений:

Н: значение с устоятва получено.

В: от устоятва пришел ответ с признаком недоступности данного параметра (запрашиваемый «arc» на электросчетчике не существует).

С: устоятво недоступно

Е: запрос от «модуля опроса» некорректен

Примечание: значения уставок можно редактировать, изменяя значения массива «limit».

Таблица 3. Возможные значения параметров запроса.

Пара-метр	Описание	Значения
Num	Номер запроса по порядку. Целое число.	Меняется циклически до 1000000
Type	Тип запроса. Строка без пробелов и спецсимволов.	с - текущий параметр
Par	Имя параметра запроса. Строка без пробелов и спецсимволов.	См п.3.2.
Dev	Имя (адрес) устройства на порту	Целое число от 1 до 2047.
Arc	Код архива в устройстве.	№- номер трубы.
Tout	Таймаут ответа на запрос в миллисекундах	Целое число.
”пара метр ”	Значение параметра	Строка без пробелов и спецсимволов. Параметр равен имени параметра запроса.
sit	Статус ответа	Т - истек таймаут ответа С - связь с устройством отсутствует В - отказ в получении параметра (например - отсутствует в электросчетчике) Е - ошибка обработки запроса (запрос данного формата не поддерживается модулем опроса) Н - получено измеренное значение U - значение получено, но недостоверно Р - пауза (значение не получено, необходимо повторить запрос)

6. Протокол обмена драйвера с прибором «ПКП1Т».

КАНАЛЬНЫЙ уровень протокола ОВЕН

Каждый байт передаваемого или принимаемого сообщения упаковываются по методу "тетрада-в-ASCII символ";

Метод передачи байта "Тетрада-в-ASCII-символ"

Передающее устройство должно преобразовать каждую тетраду исходного сообщения в ASCII символ, в зависимости от требуемого формата передачи байта закодированный 7-ми или 8-ми битовым кодом, добавить или не добавить к нему соответствующий бит четности, и снабдить каждый пересылаемый байт старт-битом и соответствующим количеством стоповых бит.

В задачу данного уровня входит преобразование "чистой" двоичной информации в допустимые для передачи по каналу ASCII-символы.

Выбран вариант кодирования тетрад, позволяющий нам иметь собственное адресное пространство, которое не зависит от адресного пространства чужих приборов. Тетрады кодируются следующим образом:

0000 -> 'G' = 0x47
0001 -> 'H' = 0x48
0010 -> 'I' = 0x49
0011 -> 'J' = 0x4a
0100 -> 'K' = 0x4b
0101 -> 'L' = 0x4c
0110 -> 'M' = 0x4d
0111 -> 'N' = 0x4e
1000 -> 'O' = 0x4f
1001 -> 'P' = 0x50
1010 -> 'Q' = 0x51
1011 -> 'R' = 0x52
1100 -> 'S' = 0x53
1101 -> 'T' = 0x54
1110 -> 'U' = 0x55
1111 -> 'V' = 0x56

На приемной стороне после отбрасывания символов начала и конца кадра производится обратное преобразование ASCII байтов в тетрады с последующим "склеиванием" каждой пары тетрад в полноценный байт двоичной информации. На этом этапе избыточность можно использовать для дополнительного контроля правильности приема, поскольку все символы, не

входящие в набор из 16-ти вышеперечисленных, являются ошибочными, символы с не совпавшей вычисленной и переданной четностью также являются сбойными.

Маркировка начала и конца сообщения (кадра)

Начало и конец кадра обозначаются специальными маркерами. Для этого используются символы, не встречающиеся в других местах кадра, и поэтому позволяющие легко осуществлять локализацию полей принимаемых сообщений и отделять сообщения друг от друга. Хотя для данных целей достаточно использовать лишь один из данных маркеров, для увеличения надежности обмена лучше использовать два. В качестве маркера начала кадра в данном протоколе используется ASCII-символ "#" (код 0x23), а в качестве маркера конца кадра - ASCII символ <CR> (символ возврата каретки, код 0x0d).

Структура кадра

Сообщение (кадр) имеет следующую структуру:

- 8-битовая старшая часть адреса узла сети (прибора)
- 3-битовое расширение адреса узла сети
- 1-битовый признак удаленного запроса
- 4-битовый размер блока данных канального уровня в байтах -2
- Блок данных канального уровня длиной от 2 до 17 байт
- 16 битовая контрольная сумма

Старшая часть адреса сети и расширение.

В протоколе используются разно-размерная адресация узлов сети: 11-и и 8-и битная.

11-битный адрес узла сети

8-битовый адрес узла сети (старшие биты полного адреса) и 3-битовое расширение адреса узла сети (младшие биты полного адреса) представляют собой полный 11-битный адрес узла сети (прибора или "канала").

В одноканальном приборе такой узел единственный, и поэтому 11-битный адрес относится ко всему прибору. Многоканальный прибор с любой сложностью внутренней иерархической структуры благодаря использованию одномерной индексации, с точки зрения сети, рассматривается как несколько одноканальных приборов с расположенными подряд адресами. В качестве

"общего", или Базового адреса прибора берется 11-битный адрес "младшего" канала.

Единственным исключением является посылка широковещательного адреса, который служит для передачи сообщения всем приборам одновременно. В этом случае в базовом адресе прибора посылаются все «1», а значение дополнительного игнорируется.

8-битный адрес узла сети

Для упрощения, как частный случай, применяется и 8-битная адресация. При этом младшие биты полного адреса заполняются нулями. Порядок рассмотрения многоканального прибора с точки зрения одномерной индексации, сохраняется, как и с 11-битной адресацией.

Признак удаленного запроса

Он служит для маркировки команды запроса значения параметра («1») или кадра, содержащего значение параметра («0»).

Запрос на чтение параметра должен содержать признак удаленного запроса в «1», при этом поле данных пустое или содержит индекс запрошенного параметра. При ответе или при команде на запись значения параметра признак удаленного запроса устанавливается в «0», а поле данных содержит данные.

Локальный идентификатор параметра

Локальный идентификатор параметра представляет из себя Hash-свертку имени передаваемого параметра.

Контрольная сумма

Вычисляется согласно алгоритма, указанного в п.0. В контрольную сумму входят все полученные от передающего устройства вышперечисленные биты информации (в т.ч. адрес, длина и поле данных) и не входят маркеры начала и конца пакета.

Блок данных канального уровня

Блок данных содержит данные, передаваемые следующим уровням.

Структура квитанции

Если в принятом сообщении не обнаружены ошибки, то ведомом узлом (прибором) передается квитанция, аналогичная посылке, но с признаком удаленного запроса равным «0».

Уровень ПРЕДСТАВЛЕНИЯ протокола ОВЕН

Структура данных

Блок данных, принятый от канального уровня протокола, состоит из:

- 16-битового локального идентификатора параметра (HASH-кода)
- n-байт (от 0 до 15) поля данных

Локальный идентификатор параметра

Локальный идентификатор параметра в протоколе ОВЕН представляет из себя Hash-свертку имени передаваемых параметров.

Соглашение об именах параметров

Используем 4-х символьные имена пользовательских параметров в виде следующего набора разрешенных символов:

- 1) **26** букв английского алфавита (от "A" до "Z") без различия строчных и прописных букв
- 2) **10** арабских цифр (от "0" до "9")

Примечание: На 7-ми сегментном индикаторе начертание цифры "5" совпадает с начертанием буквы "S"

3) Следующие специальные символы:

пробел

"_"

" "

"_"

"/"

Итого - набор из 40 символов. Каждый символ может иметь или не иметь "при себе" точку, таким образом, количество кодовых комбинаций удваивается, и в именах можно будет использовать и точки, не занимая "отдельного" символьного места.

Вышеозначенные символы кодируются (нумеруются) подряд идущими **ЧЕТНЫМИ** цифрами, начиная с 0000000b, причем сначала нумеруются символы-цифры ("0"..."9"), затем без перерыва в алфавитном (английский) порядке буквы (от "A" до "Z"), а затем специальные знаки "-", "_", "/" и, наконец, пробел.

Символам с точкой присваивается код символа без точки, увеличенный на 1 (нечетные коды). Пример:

```
"0"      - 0000000b
"0."     - 0000001b
"1"      - 0000010b
"1."     - 0000011b
...
"a"="A"  - 0010100b
"a."="A." - 0010101b
```

Таким образом, имена "Comp", "comp", "COMP" будут относиться к одному и тому же параметру, но имена "com.P" или "Co.mp" являются уже именами других параметров и никак не связаны с первым параметром.

Пробелы в имени возможны в рамках 4-х символов, но в конце имени.

Далее с помощью хеширования имя параметра превращается в его локальный идентификатор, который в совокупности с адресом узла образуют глобальный идентификатор параметра (или сообщения).

Преобразование из ASCII-кодов имени параметра в код символа

Преобразование из ASCII кодов (делается на PC перед hash-сверткой или внутри прибора):

```
char ch;                               //преобразуемый ASCII символ
char char_for_hash;                    //символ для обработки с помощью hash-
функции
int toupper(int c);                    //преобразование в символы верхнего
регистра
ch=toupper(ch);                        //преобразуем все в верхний регистр
switch (ch)
{
  case "0".."9": char_for_hash=ch-"0"; break;
  case "A".."Z": char_for_hash=ch-"A"+10; break;
  case "-":char_for_hash=10+26; break;
  case "_":char_for_hash=10+26+1; break;
  case "/":char_for_hash=10+26+2; break;
  case " ":char_for_hash=10+26+3; break;
  default: обработка_ошибки(); break;
```

```

}
char_for_hash*=2;           //удвоить      код
символа
if (за_ним_следует_точка) char_for_hash+=1; //прибавить 1

```

Также можно воспользоваться готовой таблицей (однако следует понимать, что работа с таблицей происходит медленнее):

Код	Символ	Код	Символ	Код	Символ	Код	Символ
0	0	20	A	40	K	60	U
2	1	22	b	42	L	62	v
4	2	24	C	44	m	64	w
6	3	26	d	46	n	66	x
8	4	28	E	48	o	68	Y
10	5	30	F	50	P	70	z
12	6	32	G	52	q	72	-
14	7	34	H	54	r	74	_(подч)
16	8	36	i	56	S	76	/
18	9	38	J	58	t	78	пробел

Г При наличии точки после символа к его коду прибавляется единица.

Хеширование имен параметров и вычисление контрольной суммы сообщения

Для вычисления hash-функции и CRC-суммы используется один и тот же полином, а именно:

$$x^{16}+x^{15}+x^{11}+x^{10}+x^9+x^8+x^6+x^4+x^2+x^1+x^0.$$

Алгоритм вычисления hash-функции и контрольной суммы следующий:

```

unsigned Hash(char Byte, char nbit, unsigned CRC)
{
    for (int i = 0; i < nbit; i++, Byte <<= 1)
    {
        if ( ( Byte ^ (CRC>>8) ) & 0x80 )
        {
            CRC <<= 1; CRC ^= 0x8F57;
        }
        else CRC <<= 1;
    }
}

```

```
    return CRC;  
}
```

Обратите внимание на то, что «добавление» очередного байта к hash-свертке или к контрольной сумме начинается со старшего бита. Для представления символа сетевого имени параметра используется 7 бит, вследствие чего, при расчете hash-свертки имени, старшим битом считается **именно 7-й, а не 8-й**, всегда равный нулю бит. 8-й бит символа в вычислении hash-свертки имени никогда не участвует:

```
crc = Hash(command [3] << 1, 7,  
           Hash(command [2] << 1, 7,  
           Hash(command [1] << 1, 7,  
           Hash(command [0] << 1, 7, 0))));
```

Структура квитанции

Если в сообщении, принятом ведомым узлом, не обнаружены ошибки, передается квитанция, аналогичная посылке, но с признаком удаленного запроса равным «0».