

ТЕКРОРТ

Драйвер взаимодействия с теплоэнергоконтроллерами семейства ТЭКОН

Руководство администратора

1. Общие сведения.

Драйвер **tekport** предназначен для взаимодействия аппаратно-программного комплекса (АПК) «Net Server ТМ» (www.stm.rest.ru) с теплоэнергоконтроллерами семейства ТЭКОН (ТЭКОН-10, ТЭКОН-17). Использование данного драйвера совместно с программным обеспечением сервера телеметрии позволяет наиболее полно реализовать возможности устройств семейства ТЭКОН, а также получить доступ к данным, передаваемым устройствами ТЭКОН рабочим станциям, использующим технологию «клиент-сервер».

При взаимодействии с ТЭКОН драйвер функционирует в составе программного обеспечения (ПО) АПК «Net Server ТМ» и является его неотъемлемой частью.

Драйвер обеспечивает работу с любыми конфигурациями ТЭКОН и обеспечивает связь с ними АПК «Net Server ТМ» через СОМ-порт (интерфейс RS-232) по выделенному либо коммутируемому телефонному каналу с использованием Hayes-модемов.

Драйвер производит опрос параметров датчиков и параметров трубопроводов ТЭКОН [1]. Эти параметры ТЭКОН имеют номера с 0 до 63 по числу подключаемых датчиков, и номера с 0 до 15 по числу вычисляемых в ТЭКОН расходов на трубопроводах. Через команды запросов (систему параметров в транскрипции ТЭКОН) драйверу доступны также системные и архивные параметры ТЭКОН.

2. Системные требования.

Аппаратные средства:

- Требования не установлены

Программные средства:

- Для АПК «Net Server ТМ»:
ПО «Net Server ТМ» на основе ОС семейств UNIX (LINUX).
- Для рабочей станции пользователя:
ПО «Igel» на основе ОС Windows NT 4.0/ Windows 2000/ Windows XP.

3. Структура и описание программы.

Программа разбита на несколько файлов с исходными текстами, но фактически является единой, файлы включаются в основной файл **tekport.c** директивой `include`. Ниже приведен список этих файлов и их краткая характеристика:

tekport.c - main-процедура с основным (бесконечным) циклом, глобальные описания (`define`), переменные и массивы, а также некоторые внутрипрограммные процедуры;

teini.c - подпрограммы разбора командной строки, установки некоторых глобальных переменных и инициализации сокета (термин ОС семейств UNIX (LINUX));

terea.c - подпрограммы чтения сокетных пакетов, их разбора и формирования внутренних форм поступивших запросов;

tewor.c - подпрограммы взаимодействия с ТЭКОН:

- формирование запросов к ТЭКОН из внутренних форм;
- оценка состояния ТЭКОН;
- формирование результатов во внутренних формах;

tedeb.c - подпрограммы формирования и передачи ответов на запросы в сокетный порт (по результатам обработки), также вывод "отслеживающей" (отладочной) информации на стандартный вывод;

tecom.c - подпрограммы обращения к СОМ-порту, файл введён для облегчения переносимости (portability) программы на другую платформу;

temod.c - подпрограммы общения с модемом;

teerr.h - мнемонические коды завершения подпрограмм, используются в основном сообщениями отладки;

sflmin.h - краткая версия sfl.h (при использовании пакета SFL).

4. Процедура настройки драйвера.

Без изменения алгоритма программа может настраиваться тремя способами:

- параметрами командной строки;
- перекомпиляцией программы с заданием define - констант;
- корректировкой глобальных переменных в файле tekport.c с последующей компиляцией.

4.1. Запуск и формат командной строки.

tekport SERIAL=dev,speed,parity,data_b,stop_b PORT=Nport DEVICES=N1,...,Nn
[TKILL=suic_tout] [MINIT=modem_init_string],

где:

dev - устройство, обслуживающее СОМ-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).

1 - предпочтительна при работе с модемом;

2 - при непосредственной связи с ТЭКОН через СОМ-порт;

Nport - сокетный порт, по которому поступают запросы и отправляются ответы;
N1,...,Nn - список разделённых запятыми телефонных номеров. Каждый номер имеет двойкий смысл: с одной стороны - это код ТЭКОН - партнёра, с другой - это конец строки модемной команды, начинающейся с "atdp". Номер может содержать не только цифры, но не может содержать запятые и пробелы. Если в списке задан единственный одно- или двухсимвольный код, то он воспринимается как код единственного ТЭКОН-партнёра, связь с которым осуществляется без модема непосредственно через СОМ-порт. Весь программный блок работы с модемом при этом не задействуется, поэтому, если связь реально идёт через модем и выделенную линию, то модем должен быть настроен так, чтобы быть "абсолютно прозрачным" для программы;

suic_tout - таймаут в секундах "самоубийства" программы при отсутствии сокетных запросов (по умолчанию - 0 (никогда));

modem_init_string - строка инициализации модема при запуске программы, модем должен ответить "ОК". При неуспешном обращении предусмотрен один повтор. После повторного неуспешного обращения, программа завершается. Умолчанием является "ate0". Данный параметр должен применяться скорее при настройке модема, чем в реальной работе, поскольку в процессе работы программа не берёт на себя смелость определять, когда модем был включён и требуется инициализационная строка. Т.е. настроенный на работу модем, после выключения-включения, в ответ на сцепку "+++ " - "ate0h" должен послать "ОК".

Примеры реализации командной строки:

```
tekport SERIAL=/dev/ttyS0,38400,n,8,1 PORT=7720 DEVICES=123456,8W0957890123,34-56-78 MINIT=ATZ1
```

```
tekport SERIAL=/dev/ttyS1,9600,n,8,2 PORT=7720 DEVICES=7 TKILL=3600
```

4.2. Задаваемые при компиляции define-константы.

1) **SFL_YES** (без значения) - использовать укороченную SFL-библиотеку.

Без этого определения компилятор будет использовать только стандартные библиотеки. При задании определения требуется также включить в строку компиляции библиотеку **sflmin.a**, например:

```
gcc tekport.c sflmin.a -otekport -DSFL_YES
```

При неадекватном поведении программы, первый шаг в процессе локализации ошибки - отключение SFL-библиотеки.

2) **MAX_DIALS=** максимальное количество коммутируемых ТЭКОН - устройств. Ввиду того, что список опрашиваемых ТЭКОН жёстко задаётся при запуске, и объёмы хранимой программой

по каждому ТЭКОН информации относительно невелики, программа не использует динамического распределения памяти. Вместо этого задаётся константа MAX_DIALS, которую можно уменьшить или увеличить. *Текущее значение: 30.*

Пример:

```
gcc tekport.c -otekport -DMAX_DIALS=4
```

3) **MAX_REQ_ERRS**= количество ошибок подряд при обращении к ТЭКОН для разрыва соединения и повторного набора номера. На случай, если с точки зрения модема связь есть, но "нет результатов работы" с точки зрения программы. *Текущее значение: 30.*

4) **TOUT_4**= таймаут (в сотых долях секунды) ожидания ответа (0x4) на "hello" (0x5), посланный программой ТЭКОН'у (каждый мини-диалог начинается с обмена 5-4, после чего программа посылает запрос). *Текущее значение: 200.*

5) **TOUT_BEG**= таймаут (в сотых долях секунды) в ожидании первого байта ответа на запрос. *Текущее значение: 400.*

6) **TOUT_SYM**= таймаут (в сотых долях секунды) в ожидании последующих (после первого) байтов ответа на запрос. *Текущее значение: 100.*

7) **TOUT_INTW**= таймаут (в сотых долях секунды) "собственного" опроса ТЭКОН (таблиц описателей архивов и т.п.) при отсутствии запроса по сокетному порту. *Текущее значение: 400.*

8) **TOUT_HANG**= таймаут (в сотых долях секунды) разрыва соединения с ТЭКОН после окончания последней работы (иницированной извне или изнутри). Запускается после исполнения всех внутренних работ с ТЭКОН и возобновляется после каждого "внешнего" запроса. *Текущее значение: 3000.*

9) **TOUT_MCOMM**= пауза (в сотых долях секунды) после команд модему. Например, в эти "паузные скобки" помещается команда "+++" для разрыва соединения. *Текущее значение: 30.*

10) **TOUT_MCALL**= время (в сотых долях секунды) выделенное модему для установления соединения при наборе номера. Если за это время соединение не было установлено, связь разрывается, и следует новая попытка набора номера. Кроме того, эта константа учитывается при оценке того, следует ли вообще пытаться набрать номер - отталкиваясь от таймаута в запросе. Если

оставшееся время меньше этой константы, то попытка соединения отменяется, в посылаемый через сокет ответ проставляется код истечения таймаута. *Текущее значение: 6000.*

11) **TOUT_DIFF**= интервал (в секундах!) корректировки (в сторону уменьшения) таймаута, указанного в запросе - для того, чтобы запрашивающая программа не сделала вывод, что эта программа "умерла". *Текущее значение: 0.*

12) **CONTROL_COUNT=0** - запрет на "неформализованные обращения" к программе через сокет (см. Алгоритм). *Текущее значение: 3.*

13) **DBG**= режим вывода отладочной информации на экран. Обычно: 0 - запретить вывод, 0xffff - разрешить полный вывод. *Текущее значение: 0xffff.*

4.3. Корректировка глобальных переменных.

Не останавливаясь на корректировке как очевидных констант, таких как размеры буферов или модемной строки "+++", так и на константах, требующих знакомства с текстом программы имеет смысл вкратце рассмотреть основную таблицу описания запросов - **par_list** (файл tekport.c). Эту таблицу надо понимать, а при необходимости и редактировать.

Элементы структуры таблицы:

- 1) имя параметра в запросе;
- 2) имя1 параметра в ответе;
- 3) имя2 параметра в ответе;
- 4) тип в запросе;
- 5) первый символ поля "arg" в запросе, если этот символ - не шестнадцатеричная (в дальнейшем, hex-) цифра, или 1 - в противном случае.

б) флаги:

FP_P - в поле "arg" hex-номер (от 0) запрашиваемой трубы,

FP_T - в запросе должно быть поле "time",

FP_T1 - в запросе должно быть поле "time1",

FP_T2 - в запросе должно быть поле "time2",

FP_MV - параметр ответа имеет имя1 или имя2 в зависимости от того, измерение по трубе ведётся соответственно по массе или по объёму;

7) внутренний код работы;

8) первое полуполе (первый байт) основного описателя архива;

9) ТЭКОН – номер параметра;

10) длина ответа (ТЭКОН) на запрос;

11) код интерпретации данных ТЭКОН - ответа;

Запись таблицы будет считаться релевантной (соответствующей) для запроса, если:

- 1) совпало имя параметра &
- 2) совпал тип &
- 3) соответствует поле "arc" &
- 4) флаги соответствуют элементам запроса.

Если табличный номер параметра не нулевой и задан "флаг трубы" (FP_P) - для найденной релевантной записи, то номером искомого параметра программа будет считать сумму табличного номера параметра и номер трубы (из поля "arc").

Когда запрос касается архивного значения (при поднятом флаге FP_P), то релевантным архивом будет считаться архив с основным описателем 0x0040 и описателем положительной составляющей равным искомому параметру. Кроме того, если в табличной записи задано ненулевое первое полуполе основного описателя архива, то релевантным архивом будет также считаться архив с основным описателем, в котором первое полуполе совпадает с табличным, а второе - совпадает с номером трубы.

Например: в табличной записи поля 8) и 9) равны, соответственно, 2 и 0x1c80; поступил соответствующий этой записи запрос с полем arc=3. Тогда будет производиться поиск архива с основным описателем 0x0040 и описателем положительной составляющей 0x831c (перекрутка байтов), а если таковой не найден, то с основным описателем 0x0203.

В запросе может также непосредственно указываться номер архива, предваряемый литерой 'n', например, n1A. Релевантная запись таблицы будет в этом случае содержать поле 5) (признак поля "arc") равным 'n'.

Приведённое выше описание таблицы **par_list** не претендует на полноту, для сложных случаев необходимо ознакомление с текстом программы.

Кроме таблицы **par_list** следует также рассмотреть 48-битовое поле **mask4042**.

Когда программа **tekport** получила запрошенное значение, то прежде, чем выдать в сокет "хороший" ответ, она оценивает, не следует ли отправляемый ответ пометить признаком недостоверности. Для этой оценки программа запрашивает у ТЭКОН 64-битный параметр 0x4042 (текущие отказы блока и труб). Последние 16 бит этого поля - битовая карта "ошибочных" с точки зрения ТЭКОН труб - как правило речь идёт об отказе или выходе за уставку одного из связанных с трубой датчиков. Первые же 48 бит - карта неисправностей ТЭКОН в целом.

Tekport выставляет недостоверность в одном из следующих двух случаев:

1) запрос был по трубе (флаг FP_P) и эта труба расценивается ТЭКОН как содержащая текущую неисправность (по карте труб);

2) карта неисправностей ТЭКОН с наложенной на неё маской **mask4042** (битовое "И") содержит хотя бы один ненулевой бит.

Такая оценка не свидетельствует о подлинной недостоверности (особенно в случае запроса архивов), являясь скорее предупреждающим сигналом (при "недостоверности" **tekport**

переправляет запрашивающей программе эти 64 бита). Ясно также, что заполнение маски "требуется вдумчивости и знания вопроса" и, может быть, даже зависит от конкретики.

Полная карта неисправностей ТЭЖОН приведена в таблицах 6.19 и 5.10 документа [2].

5. Описание подключения объектов ТЭЖОН.

Используя конфигуратор сервера телеметрии (www.stm.rest.ru), администраторы могут производить настройку различных параметров конфигурации драйвера.

При конфигурации драйвера при помощи конфигуратора сервера телеметрии производятся следующие операции:

- настройка параметров подключения;
- определение скриптов обработки событий;
- определение типов объектов и их параметров;
- редактирование объектов и их параметров;
- экспорт конфигурации в формат сервера телеметрии.

Описание подключения датчиков и трубопроводов (в транскрипции сервера телеметрии – объектов) к драйверу ТЭЖОН и серверу телеметрии производится с использованием существующей конфигурации - **uvz**.

5.1. Настройка параметров подключения.

Настройка параметров подключения производится в секции «Источники данных» с заполнением полей выводимого окна (рис.1). В штатных режимах значения заполняемых полей соответствуют показанным на рис.1, изменяется только имя источника и IP-адрес соединения.

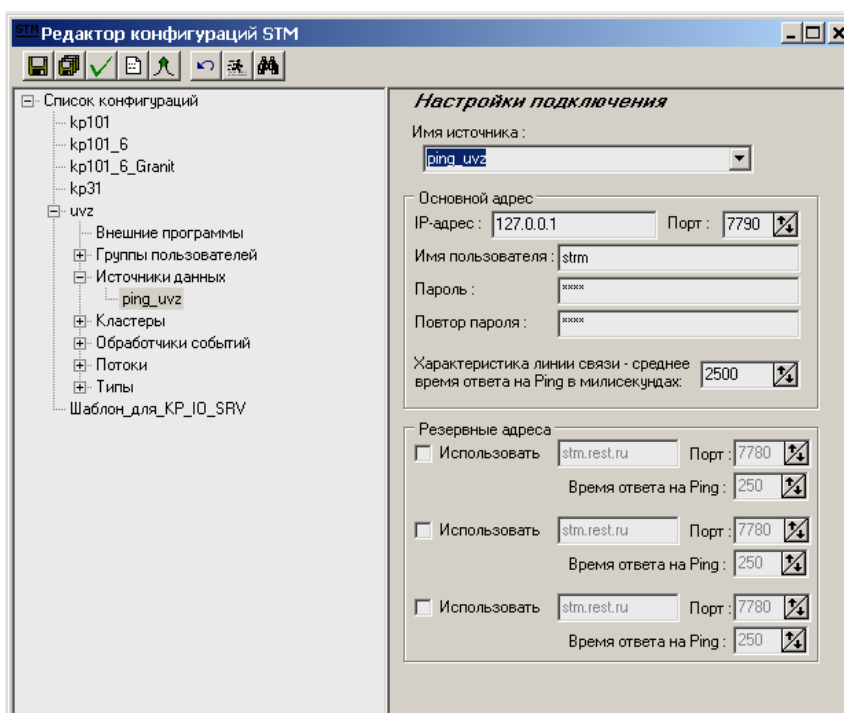


Рис.1.

5.2. Определение скриптов обработки событий.

Скрипты обработки событий для большинства конфигураций ТЭКОН являются стандартными и определяется секцией «Обработчики событий» конфигуратора. При необходимости изменения алгоритмов обработки производится редактирование текстовой части скрипта в соответствующем поле выводимого окна (рис.2).

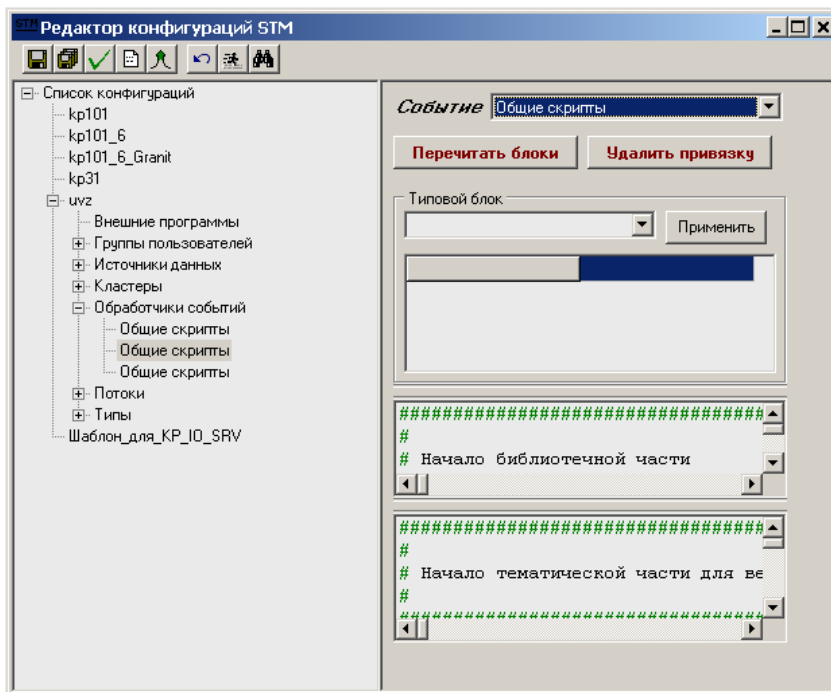


Рис.2.

5.3. Определение типов объектов и их параметров.

Определение типов объектов и их параметров производится в секции «Типы» (рис.3) конфигуратора с заполнением полей выводимого окна для каждого типа объектов.

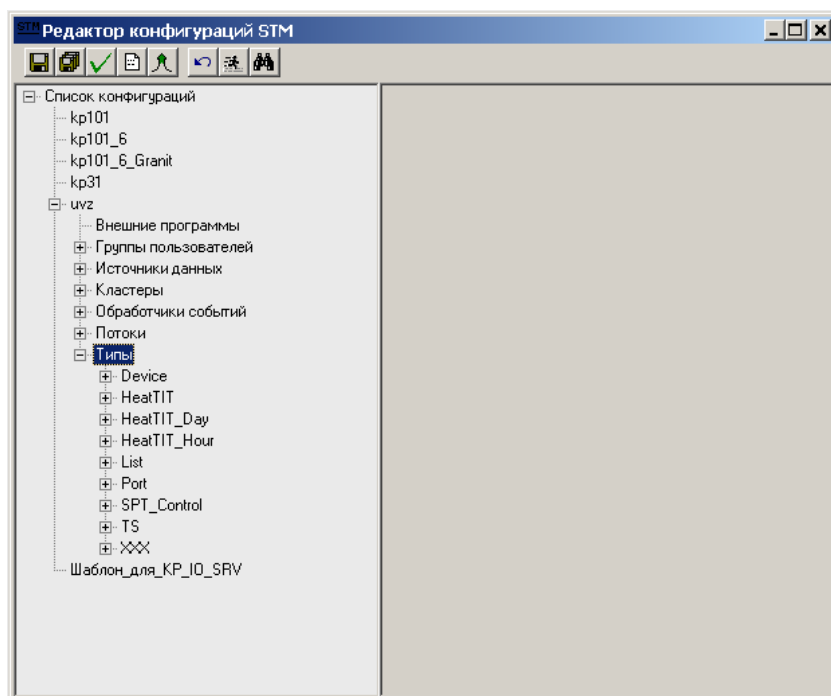


Рис.3.

В стандартной конфигурации ТЭКОН присутствуют следующие типы объектов:

- **Device** – Описание прибора;
- **HeatTIT** – Тепловой TIT - текущий;
- **HeatTIT_Day** – Тепловой TIT – суточный;
- **HeatTIT_Hour** – Тепловой TIT – часовой;
- **List** – Журнал событий;
- **Port** – Com-port;
- **SPT_Control** – Тепловой TIT – текущий;
- **TS** – не используется;
- **XXX** – Служебный тип (Получение имени клиента).

Параметры типов объектов по умолчанию указаны в полях выводимого окна для каждого типа объектов в существующей конфигурации - **uvz**.

5.4. Редактирование объектов и их параметров

Редактирование объектов и их параметров производится конфигуратором в секции «Объекты» каждого имеющегося типа объектов (рис.4) с заполнением полей выводимого окна для каждого объекта, относящегося к данному типу.

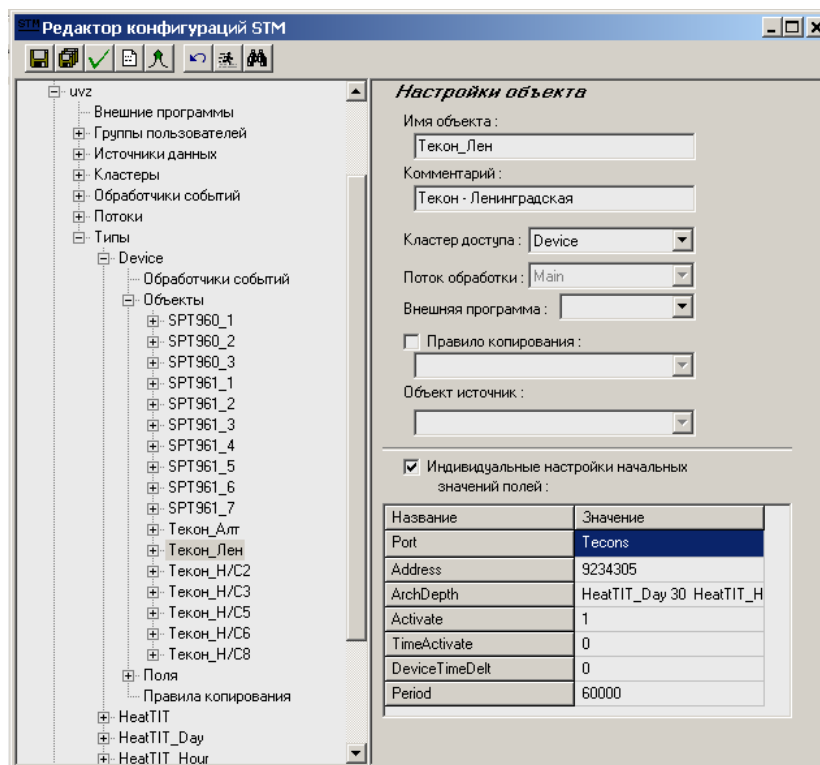


Рис.4.

Далее приводится описание заполняемых полей объектов в стандартной конфигурации ТЭКОН.

Для объектов типа **Device** заполняются следующие поля:

- **Port** – имя объекта типа порт, которому соответствует прибор;
- **Address** – адрес устройства или телефон;
- **ArchDepth** – глубина анализа архивов (пары ИМЯ_ТИПА ЧИСЛО_ТОЧЕК);
- **Activate** – активация (1 – прибор в работе, 0 – прибор выведен из работы);
- **TimeActivate** – время вывода прибора из работы;
- **DeviceTimeDelt** – опережение времени устройства относительно времени сервера (в секундах);
- **Period** – период опроса текущих значений (в миллисекундах).

Для объектов типа **HeatTIT** заполняются следующие поля:

- **Value** – значение телеизмерения;
- **Man** – признак режима ручного ввода (0/1);
- **Dif** – текущее значение «скачка»;
- **Proc** – процент от номинала;
- **_nom** – номинальное значение;
- **_avalue** – значение телеизмерения от оборудования;
- **_mvalue** – значение телеизмерения от ручного ввода;
- **HiAlarm** – значение верхней аварийной уставки;
- **LoAlarm** – значение нижней аварийной уставки;
- **HiTech** – значение верхней предупредительной уставки;
- **LoTech** – значение нижней предупредительной уставки;
- **Limit** – признак выхода за уставку (-2...2);
- **Comment** – комментарий;
- **Calc** – формула пересчета;
- **Device** – имя прибора, к которому привязан ТИТ;
- **Arc** – номер параметра в приборе;
- **_avalid** – значение достоверности от оборудования.

Для объектов типа **HeatTIT_Day** и **HeatTIT_Hour** заполняются следующие поля:

- **Value** – значение телеизмерения;
- **Man** – признак режима ручного ввода (0/1);
- **Dif** – текущее значение «скачка»;
- **Proc** – процент от номинала;
- **_nom** – номинальное значение;
- **_avalue** – значение телеизмерения от оборудования;
- **_mvalue** – значение телеизмерения от ручного ввода;

- **_max2_is** – признак контроля верхней аварийной уставки;
- **_min2_is** – признак контроля нижней аварийной уставки;
- **_max1_is** – признак контроля верхней предупредительной уставки;
- **_min1_is** – признак контроля нижней предупредительной уставки;
- **_maxd_is** – признак контроля скачка вверх (роста);
- **_mind_is** – признак контроля скачка вниз (падения);
- **Limit** – признак выхода за уставку (0 – в норме, 1 – за предупредительной, 2 – за аварийной);
- **Comment** – комментарий;
- **Calc** – формула пересчета;
- **Device** – имя прибора, к которому привязан ТИТ;
- **Arc** – номер архива.

Для объектов типа **List** заполняются следующие поля:

- **Description** – описание события;
- **EventTime** – время события;
- **Event** – код события (0 – не определен, 1 – квитирование);
- **ObjectDescr** – объект, с которым связано событие;
- **SrcName** – источник информации о событии.

Для объектов типа **Port** заполняются следующие поля:

- **PortData** – характеристики порта;
- **TOutRequest** – таймаут, отправляемый в запросе;
- **Parallel** – признак параллельности;
- **TOutTest** – таймаут тестовых запросов;
- **Driver** – командная строка запуска драйверного модуля.

Для объектов типа **SPT_Control** заполняются следующие поля:

- **Value** – значение телеизмерения;
- **Man** – признак режима ручного ввода (0/1);
- **_avalue** – значение телеизмерения от оборудования;
- **_mvalue** – значение телеизмерения от ручного ввода;
- **Comment** – комментарий;
- **Calc** – формула пересчета;
- **Device** – имя прибора, к которому привязан ТИТ;
- **Arc** – номер параметра в приборе;

- **_avalid** – значение достоверности от оборудования.

Добавление нового объекта имеющегося типа производится стандартно с помощью конфигуратора с заполнением соответствующих полей выводимого окна.

Значения по умолчанию и начальные значения полей вновь добавляемых объектов присваиваются аналогично подобным значениям существующих объектов конфигурации.

5. Средства отладки.

Основным средством отладки драйвера ТЭКОН является вывод различных дампов на экран монитора в процессе работы драйвера. Дампы содержат различные диагностические сообщения о трассировке объектов, поставленных на контроль, сообщения о штатных и нештатных ситуациях, обнаруженных драйвером. Для вывода отладочной информации необходимо включение этого режима при помощи задаваемой при компиляции драйвера `define` - константы « **DBG=** ».

Константа принимает два значения:

0 - запретить вывод;

0xffff - разрешить полный вывод.

Запись драйвером отладочной информации производится в кольцевой log-файл (по умолчанию **common.log**), Имя файла и его местонахождение может быть изменено с помощью конфигуратора сервера телеметрии. Выбор записываемой информации в файл **common.log** определяется файлом **common.conf**.

Для просмотра в темпе записи используется стандартная утилита направления потока на экран:

tail -f common.log.

Утилиту необходимо запускать из каталога, в котором пишется файл **common.log**. Прервать вывод информации на экран можно нажатием **Ctrl-C**.

Файл **common.log** - кольцевой и содержит не более 50000 строк.

ЛИТЕРАТУРА

1. Теплоэнергоконтроллеры ТЭКОН-10, ТЭКОН-17. Обмен по последовательному каналу. Руководство программиста. Т10.06.59РД.: ЗАО ИВП КРЕЙТ, Екатеринбург, 2002г.
2. Теплоэнергоконтроллер ТЭКОН 10. Руководство по эксплуатации. Т10.00.13РЭ.: ЗАО ИВП КРЕЙТ, Екатеринбург, 2001г.

Алгоритм и описание работы программы.

Процедура **main** (tekport.c) последовательно вызывает подпрограммы чтения командной строки **fill_arg** (teini.c), инициализации порта **init_com** (tecom.c) (которая, в свою очередь, вызывает **init_modem** (temod.c)) и инициализации сокета **init_sock** (teini.c). Если какой-то из вызываемых модулей возвращает не ноль, **main** выдаёт на стандартный вывод соответствующее сообщение об ошибке и завершает работу (весь вывод осуществляется на **stdout**).

Затем **main** входит в основной рабочий цикл.

Формально результаты работы цикла оцениваются в начале, но де-факто - в конце. Далее будет рассмотрена итерация цикла.

1. Определение таймаута ожидания поступления данных (запроса) по сокету **get_mtout** (tekport.c). Если с ТЭКОН - партнёром установлено соединение, то **get_mtout** оценивает, есть ли внутренние работы по сбору информации ТЭКОН. К ним в настоящий момент относятся:

- сбор информации по описателям труб;
- сбор информации по описателям часовых архивов;
- сбор информации по описателям суточных архивов.

Если какая-то из этих работ не выполнена, то устанавливается таймаут **TOUT_INTW**, с тем чтобы, если в работе возникла пауза, можно было "упреждающе" собрать необходимую информацию. Собранные данные считаются актуальными на всё время работы программы - в т.ч. при повторных соединениях. Данный таймаут предполагается достаточно коротким (секунды).

Если все внутренние работы выполнены, то назначается таймаут разрыва соединения при отсутствии запросов к ТЭКОН - **TOUT_HANG** (предполагаются десятки секунд или минуты).

Если соединение разорвано (также в случае выделенной линии) и определён ненулевой таймаут завершения работы программы при отсутствии сокетных запросов **TKILL**, то назначается этот таймаут.

Если ничего из вышперечисленного, то назначается бесконечное ожидание данных по сокету.

2. **Main** осуществляет чтение информации из сокета с учётом полученного таймаута. Если сокет оказался закрыт, формируется код выхода из программы.

3. Если истёк таймаут, то подпрограмма **mtout_task** (tekport.c) окончательно формирует либо задание на внутреннюю работу, либо результат окончания итерации цикла.

Если поступили данные, то вызывается подпрограмма **ch_spack** (tereа.c), которая вычленяет из сокетного потока пакет, причём предполагается, что более одного пакета получено быть не может, и последние полученные символы есть символы завершения пакета.

Если код возврата **ch_spack** не 0, это может означать следующее:

- законченного пакета нет;
- пакет есть, но он содержит "синтаксическую" ошибку;
- поступило "неформализованное обращение" к программе, которое было обработано подпрограммой **control** (terea.c).

В настоящее время реализовано 3 таких обращения:

- **{quit}** - завершить работу;
- **{debug=hhhh}** - установить режим вывода отладочной информации (hhhh - hex-число);
- **{disc}** - разорвать модемное соединение.

"Неформализованные обращения" могут быть запрещены `define` - константой **CONTROL_COUNT**.

При ненулевом коде возврата **ch_spack** основной цикл уйдёт на конец итерации.

Если **ch_spack** сочла, что получен рабочий, синтаксически верный пакет, она формирует массив ссылок на его значения и возвращает 0.

В последнем случае **main** запускает подпрограмму **get_task** (terea.c), которая формирует, согласно рассматривавшейся таблице **par_list**, задание на работу с ТЭКОН во внутренней форме. Если это удалось сделать (успешно считаны поля данных, найдена релевантная запись таблицы, найден ТЭКОН - партнёр в списке), **get_task**, разорвав соединение в случае нового партнёра, возвращает 0, иначе - код ошибки (с которым **main** перейдёт на конец итерации).

4. При успешном формировании задания на работу с ТЭКОН (**mtout_task** или **get_task**) **main** вызывает "ключевую" подпрограмму **do_task** (tewor.c). **Do_task** делает следующее:

1) запрашивает текущие отказы блока и труб (параметр 0x4042);

2) проверяет, собрана ли с ТЭКОН необходимая информация: по описателям труб - для параметров, по архивам - для архивов. Если нет, то вызывает соответственно подпрограммы **set_pipe** (tewor.c) или **set_a** (tewor.c);

3) для архивов: запрашивает время ТЭКОН - **get_te_time** (tewor.c);

4) проверяет, поддерживает ли текущая конфигурация ТЭКОН нужную информацию:

check_pipe (tewor.c) - для параметра,

find_ha (tewor.c) - для часовых архивов,

find_da (tewor.c) - для суточных архивов.

5) запрашивает у ТЭКОН нужный параметр: **get_par** (tewor.c).

6) для архивов: запрашивает время и проверяет, не изменилось ли поле часов. Если изменилось, возвращается к поз.4.

7) размещает в структуре внутренней работы результат и возвращает код 0.

4А. Большая часть работы с ТЭКОН проходит через **get_par** (tewor.c), которая, в свою очередь вызывает подпрограмму **cust_req** (tewor.c). **Cust_req** в сцепке с подпрограммой **dial_req** (tewor.c) является узловым пунктом коммуникации. Ниже излагается этот коммуникационный алгоритм.

1) Проверяется, есть ли соединение - **is_conn** (tewor.c). Пока нет соединения и установленный лимит времени на начало соединения (берётся как разность таймаута в запросе и таймаута на установление соединения) не исчерпан (**is_ring_tout** (tewor.c)), вызывается подпрограмма набора номера (**conn_attempt** (temod.c)). Когда лимит времени истекает, **cust_req** возвращает соответствующий код ошибки.

2) Когда соединение есть, через подпрограмму **dial_req** запрашивается требуемый параметр. После успешного диалога с ТЭКОН **cust_req** возвращает 0.

3) Если диалог был неуспешен, и истёк таймаут запроса (**is_conn_tout** (tewor.c)), **cust_req** возвращает ошибку.

4) Если количество неуспешных диалогов подряд превысило константу **MAX_REQ_ERRS**, соединение разрывается.

5) Переход к поз.1.

4В. Обращаясь к модему командой "**ate0h**" и ожидая от него ответа "**OK**" (**clr_conn** (temod.c)), программа входит в бесконечный цикл и может, таким образом, "зависнуть". Предполагается, что при долгом молчании программа будет снята извне, и что лучше "молчаливо сигнализировать" о неработоспособности, чем создавать иллюзию работы при неработающем модеме. Снятие программы должно быть реализовано средствами ОС, поскольку в основной цикл программа не возвращается. (Ещё лучше включить выключенный модем!).

5. Конец итерации.

Main вызывает подпрограмму **end_w** (tekport.c), чтобы определить, следует ли завершать цикл. **End_w** вызывает подпрограмму **send_sans** (tedeb.c), которая по результирующему коду итерации, полученным данным и параметру "отказы блока и труб", формирует и отправляет (или не формирует и не отправляет) ответ в сокет.

Затем **end_w** по коду завершения определяет, следует ли продолжать работу, и, если да, возвращает в **main** 0.