

Система передачи аварийных событий  
в режиме реального времени СПА-РВ

описание программного обеспечения  
и руководство пользователя

AuraHub

## Содержание

Содержание .....	2
1. Общие сведения .....	2
1.1 Назначение .....	2
1.2 Системные требования .....	2
1.3 Состав .....	3
1.4 Установка и удаление .....	3
1.5 Архитектура .....	3
2. Использование .....	5
2.1 Администрирование .....	5
2.1.1 Таблица подключений .....	5
2.1.2 Дерево сигналов .....	6
2.1.3 Параметры определения места повреждения (ОМП) .....	9
2.1.4 Автоматическое получение файлов конфигурации и осциллограмм .....	9
2.1.5 Резервное копирование и экспорт данных .....	10
2.2 Текстовые сообщения о произошедших событиях .....	10
2.2.1 Сводка событий, архив событий, уведомления о событиях .....	11
2.3 Просмотр графиков аварийных процессов .....	11
2.3.1 Модификация графиков для проверки скриптов .....	13
2.3.2 Векторные диаграммы .....	13
2.3.3 Осциллограммы .....	14
3. Установка и первоначальная настройка .....	14
3.1 Первый запуск .....	14
3.2 Окно настройки .....	15
3.2.1 Настройка подключения к базе данных .....	15
3.2.2 Настройка службы (управление службой) .....	17
4. Дополнительная информация .....	17
4.1 Ручная настройка .....	17
4.1.1 Файл программной конфигурации .....	17
4.1.2 Файл команд ежедневного обслуживания .....	19
4.2 Составление скриптов генерации отчётов .....	19
4.2.1 Объект Record (запись) .....	20
4.2.2 Объект Report (отчёт) .....	21
4.2.3 Примеры скриптов .....	21
5. Изменения .....	22
5.1 Новое в версии 0.9.0 .....	22
5.2 Новое в версии 0.10.0 .....	23
5.3 Новое в версии 0.11.0 .....	23
5.4 Новое в версии 0.12.0 .....	23
5.5 Новое в версии 1.0.0 .....	23

## 1. Общие сведения

### 1.1 Назначение

Программа AuraHub предназначена для получения с регистраторов аварийных событий в режиме реального времени аварийных записей по протоколу МЭК 60870-5-104, помещения их в базу данных Microsoft SQL Server, автоматического анализа с составлением текстового отчета и последующего просмотра. С регистраторов, поддерживающих спорадический режим протокола МЭК 60870-5-104, в базу данных могут поступать данные нормальных режимов.

### 1.2 Системные требования

Программе для работы требуется компьютер с операционной системой Windows и установленным Microsoft .NET Framework 2.0 (входит в новые версии Windows). Аппаратная часть определяется требованиями операционной системы, дополнительных требований к ней не предъявляется.

Также требуется доступ к Microsoft SQL Server, версии 2005 или новее, который может быть установлен на том же или на другом компьютере. Для ограниченных объёмов данных возможно применение Microsoft SQL Server Express Edition, в остальных случаях рекомендуется Standard Edition.

### 1.3 Состав

Устанавливаемый комплект программного обеспечения состоит из нескольких исполняемых файлов (AuraHub.exe, AuraTolec104.dll, unzip.exe, unrar.exe; из них на рабочем месте пользователя необходим только AuraHub.exe), необязательного файла программной конфигурации (AuraHub.config), необязательного файла команд ежедневного обслуживания (daily.cmd) и настоящего руководства пользователя.

В процессе работы программы создаётся зашифрованный файл пользовательской конфигурации (AuraHub.cfg) с параметрами подключения к базе данных и др. Также могут создаваться файлы трассировки (\*.log; подробнее см. 4.1.1.1).

### 1.4 Установка и удаление

Установка программы осуществляется копированием либо разархивированием из дистрибутива. Затем производится настройка подключения к базе данных (см. 3.2.1) и внесение программы в список служб Windows (см. 3.2.2). Удаление осуществляется в обратном порядке.

### 1.5 Архитектура

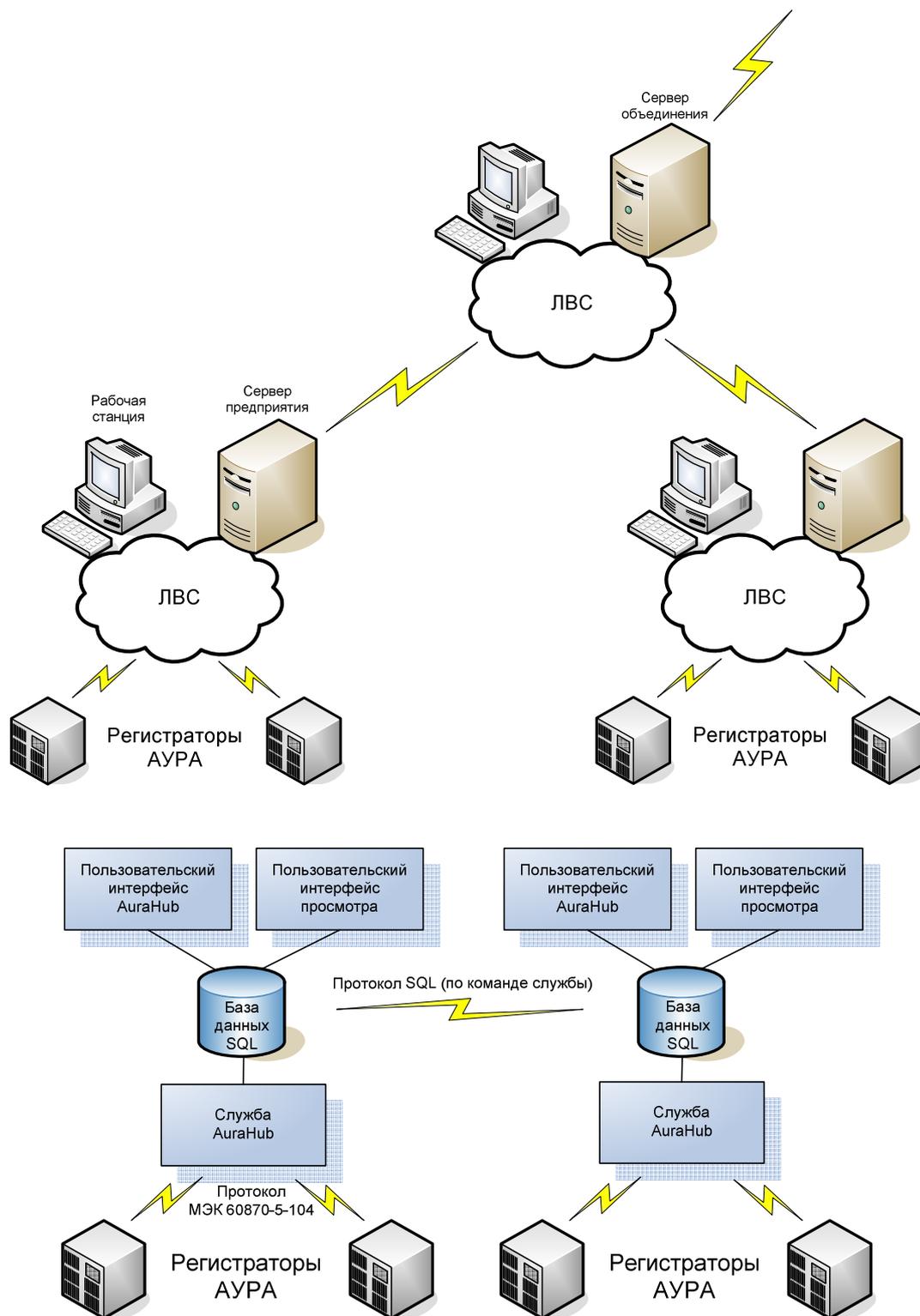
Ниже приведены два рисунка, соответствующих физической и логической архитектуре системы сбора аварийных данных в реальном времени. На первом рисунке изображена двухуровневая архитектура и обозначена связь к возможным вышестоящим уровням. На втором рисунке показана одноуровневая архитектура, использующая обмен данными между серверами одного уровня.

Служба AuraHub, работает в автоматическом режиме, собирая данные с регистраторов и серверов согласно конфигурации и помещая их в базу данных.

База данных состоит из ряда таблиц, каждая из которых содержит информацию какого-то одного определённого типа в пригодном для машинной обработки формате. В таблицах базы данных хранятся:

- перечень подключенных к системе регистраторов, с параметрами связи
- список сигналов, получаемых с каждого регистратора
- сводный структурированный пользователем список (дерево) сигналов
- измеренные значения сигналов, полученные от регистраторов (подробнее см. ниже)
- метки начала и окончания аварийных записей
- текстовые отчёты, создаваемые в результате обработки аварийных записей
- другая необходимая информация

Служба устанавливает и постоянно поддерживает соединение в протоколе МЭК 60870-5-104 с каждым из входящих в систему регистраторов. Периодически, с заданным интервалом, например один раз в 30 минут, служба выполняет запрос измеренных значений нормальных режимов и помещает эти значения в базу данных. В случае возникновения аварийной ситуации регистратор записывает аварийный файл и сразу же спорадически передаёт значения предаварийного, аварийного и послеаварийного режимов. При этом передаются только изменения действующего значения первой гармонической, фазы, частоты или постоянного напряжения, а также изменения состояния дискретных каналов. Эти значения также помещаются службой AuraHub в базу данных.



Передача и сохранение только изменений действующих значений позволяет существенно сократить объём обрабатываемой информации, в то же время сохраняя достаточно как для анализа аварии, так и для анализа статистики работы оборудования (расчёта ресурса выключателей и т.п.). Кроме самих измеренных значений и отметок о начале и окончании аварийной записи в базу данных помещается текстовый отчёт об аварии, включая результаты анализа аварии пользовательским скриптом (программой на языке JScript или VBScript), если таковой был задан, и результаты ОМП (если были заданы параметры линии).

Программа, запущенная на рабочих местах операторов, отслеживает появление в базе данных новых аварийных записей и информирует операторов. В типовом случае,

пользователь получит всплывающее сообщение с отчётом об аварии не позднее, чем через 10 секунд с момента её окончания.

Для обеспечения возможности более детального анализа аварий в системе также реализована передача файлов осциллограмм. Осциллограммы передаются по протоколам FTP или SMB, по инициативе регистратора или по запросу.

## 2. Использование

Этот раздел предполагает, что программное обеспечение на Вашем рабочем месте уже установлено и прошло первоначальную настройку (например, администратором или интегратором). Если это не так, переходите к разделу 3.

### 2.1 Администрирование

К администрированию комплекса программного и аппаратного обеспечения в процессе использования относятся отслеживание работоспособности уже настроенного программного и аппаратного обеспечения и внесение необходимых изменений в соответствии с подключением, отключением и переконфигурированием регистраторов (установка и первоначальная настройка программного обеспечения описаны в разделе 3).

Основными задачами администрирования в процессе эксплуатации являются подключение к системе новых регистраторов и пополнение/изменение общего списка регистрируемых сигналов и линий электропередач при изменении конфигурации регистраторов. Для выполнения этих задач применяются две формы представления режима администрирования: таблица подключений и дерево сигналов, переход между которыми осуществляется расположенными в панели инструментов кнопками  "К таблице подключений" и  "К дереву сигналов".

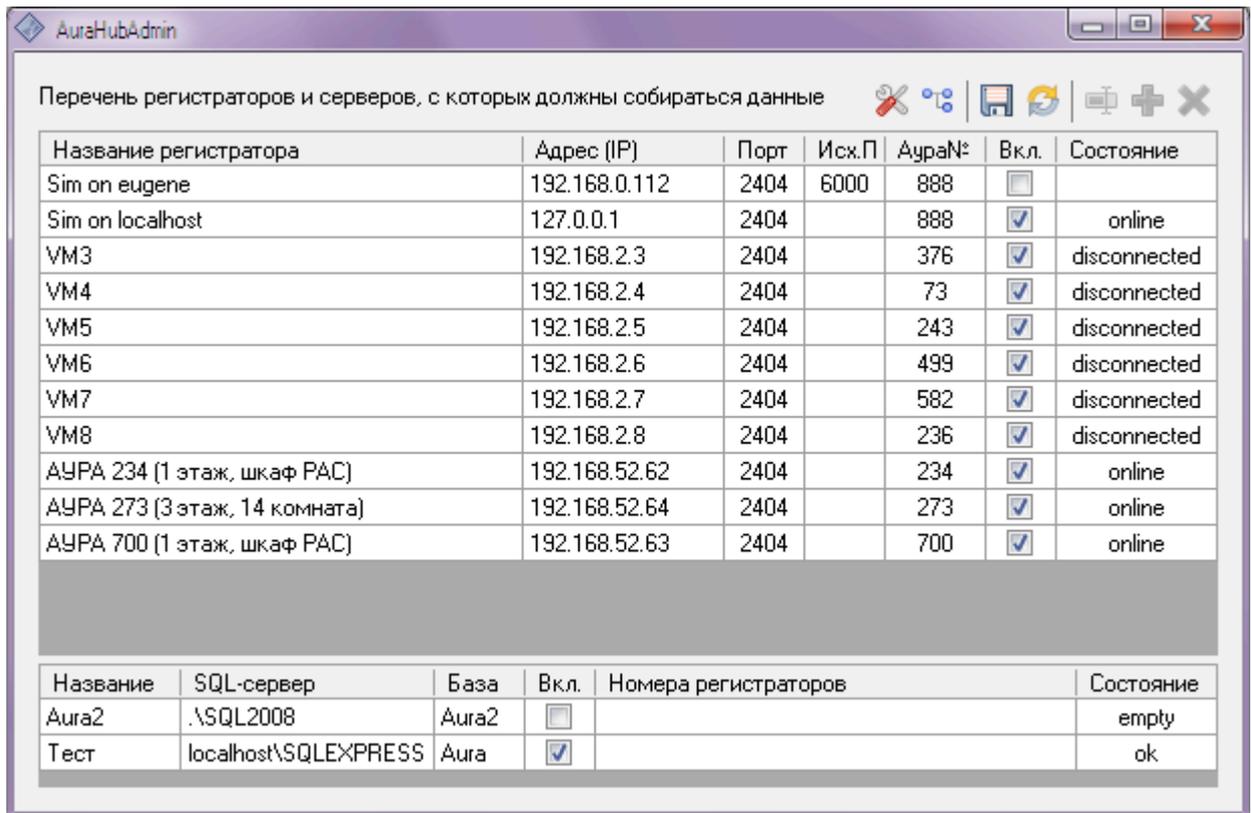
Пользователь вносит изменения в конфигурацию сигналов и подключений как при помощи кнопок панели инструментов -  "Переименовать",  "Создать",  "Удалить", так и при помощи правки данных непосредственно по месту их нахождения в таблице либо путём перетаскивания узла дерева.

Все изменения, вносимые пользователем в главном окне и в диалоговом окне параметров линий, сохраняются в базе данных только по нажатию кнопки  "Сохранить" в панели инструментов, причём в режиме таблицы подключений сохраняется только таблица подключений, а в режиме дерева - только дерево.

#### 2.1.1 Таблица подключений

В режиме таблицы подключений в главном окне программы показываются список подключений к регистраторам (в средней части) и список подключений к нижестоящим SQL-серверам иерархии (в нижней части).

При подключении к регистраторам, по протоколу МЭК 60870-5-104, служба AuraHub является клиентом TCP, а регистраторы - серверами TCP. Каждая строка соответствует одному регистратору. При этом указывается произвольное текстовое название регистратора, его IP адрес и TCP порт (стандартный порт 2404 для МЭК 60870-5-104), исходящий TCP порт (не обязательно; если не указан, он будет назначен операционной системой и может быть разным при каждом подключении), номер регистратора. Галочкой отмечается следует ли использовать указанное подключение или оно временно отключено. Последний столбец не доступен для изменения и, при работающей службе, показывает текущий статус подключения.

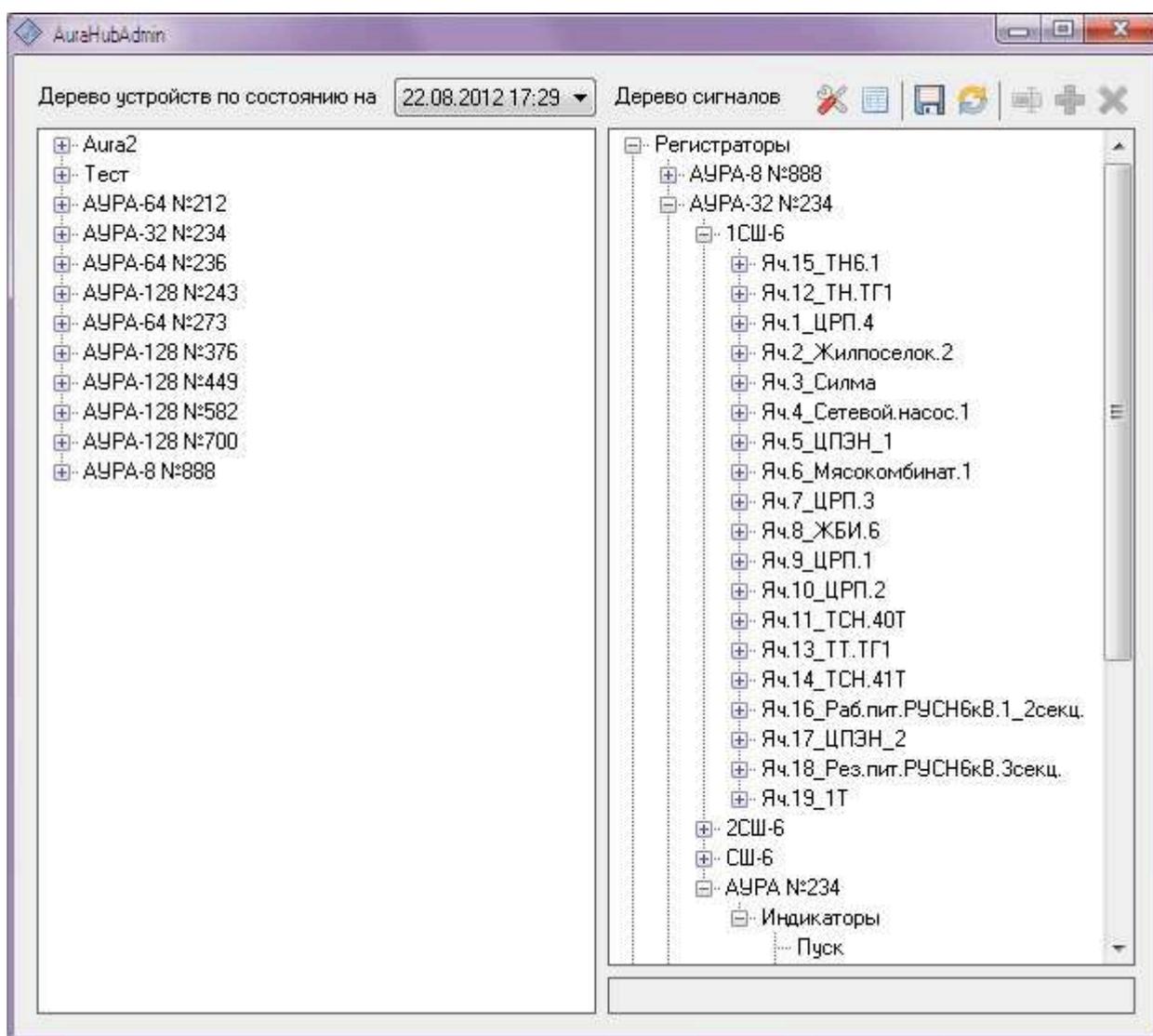


При подключении к SQL-серверам служба AuraHub использует любой из подходящих протоколов (TCP, Named Pipes или Shared Memory) в зависимости от указанного адреса сервера и от его доступности. Общий подход к указанию имени сервера аналогичен описанному в 3.2.1. Имя пользователя и пароль указывать не требуется, так как служба использует специально для этой цели создаваемое имя пользователя AuraHubRemote с автоматически сгенерированным сложным паролем. При необходимости (например, при двустороннем обмене) можно ограничить объём получаемой информации, указав через запятую номера регистраторов, данные с которых следует получать (в случае если список пуст, будут получены все данные, имеющиеся на соответствующем сервере). Последний столбец не доступен для изменения и показывает текущий статус получения данных с соответствующего сервера.

### 2.1.2 Дерево сигналов

Список каналов базы данных составляется из списков каналов регистраторов и организуется в иерархическом виде с произвольным количеством уровней. В базу данных могут помещаться все каналы регистраторов целиком, либо группами, либо по-отдельности. Группировка каналов по присоединениям может отличаться от группировки их в регистраторах.

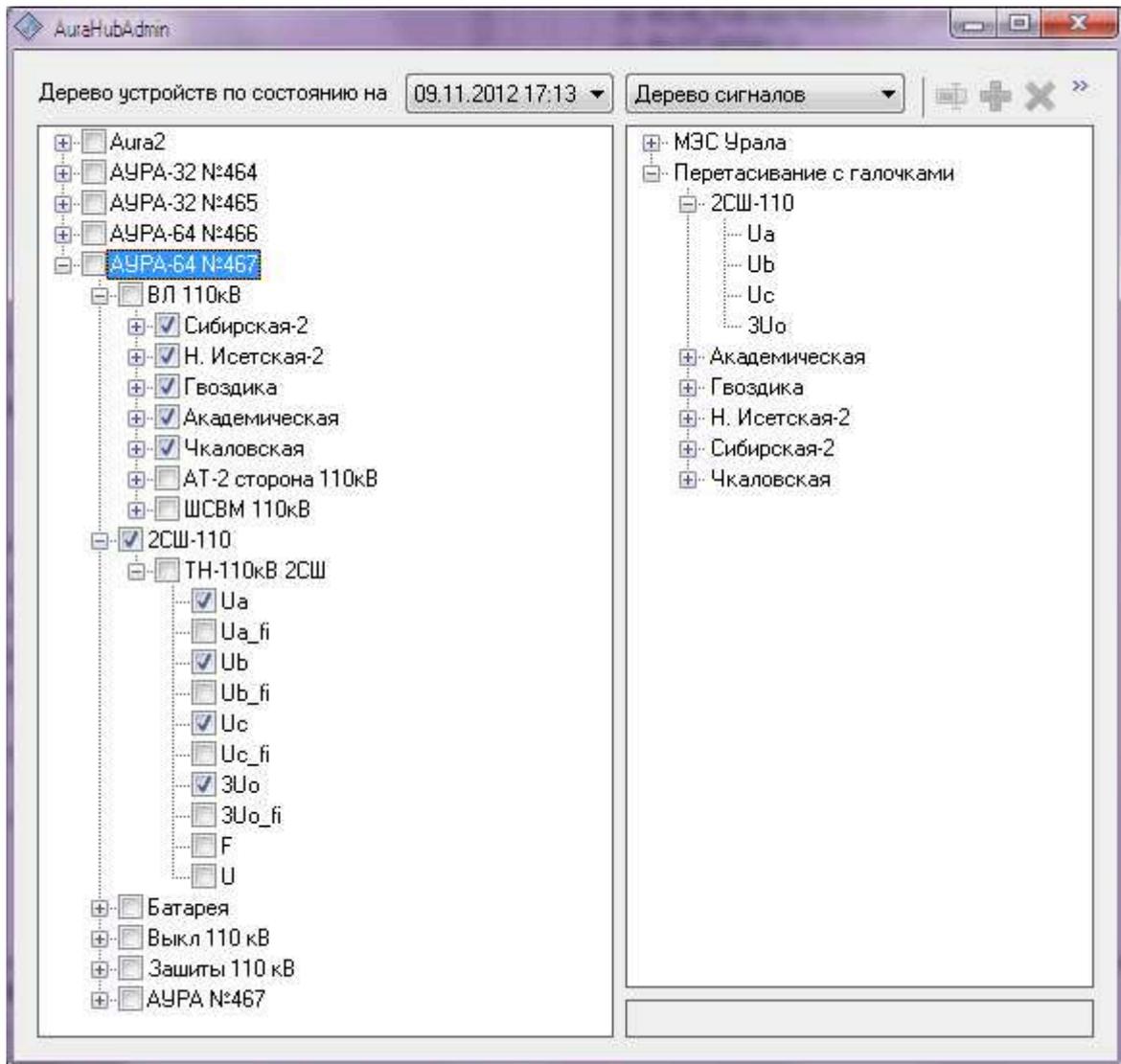
В режиме дерева сигналов в главном окне программы показываются как само дерево сигналов (в правой части), так и дерево устройств (в левой). Дерево устройств может быть отображено в ретроспективе по состоянию на любой момент его изменения (это касается только сигналов регистраторов; SQL-серверы показываются по состоянию на момент обращения), оно используется для составления (путём перетаскивания) компоуемого пользователем дерева в правой части, описывающего логическую иерархию сигналов в удобной для дальнейшего использования форме.



### 2.1.2.1 Режим с галочками

Включение и отключение этого режима производится через контекстное меню дерева устройств. Если режим включен, то при перетаскивании части дерева устройств части, не отмеченные галочками, будут проигнорированы. Пример использования этого режима приведен на рисунке ниже.

Обратите внимание на то, что при перетаскивании "АУРА-64 №467" не отмеченные галочками "АУРА-64 №467" и "ТН-110кВ 2СШ" не были помещены в дерево сигналов, а отмеченные пять из семи воздушных линий и напряжения внутри "2СШ-110" - были.

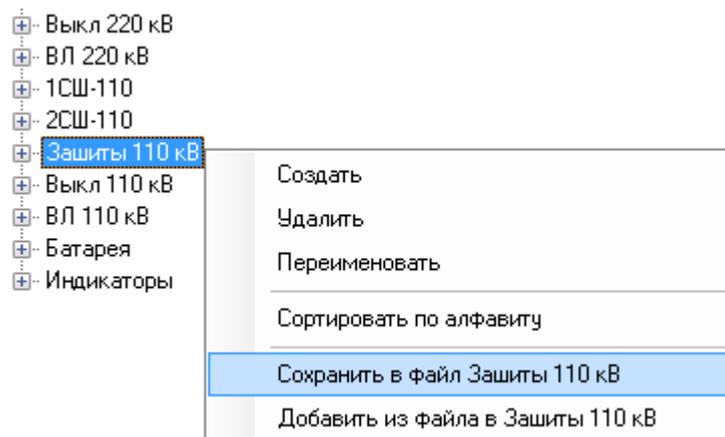


### 2.1.2.2 Сортировка

Команда сортировки по алфавиту находится в контекстном меню дерева сигналов. Ручная сортировка осуществляется с клавиатуры стрелками вверх/вниз при нажатой клавише Ctrl.

### 2.1.2.3 Сохранение части дерева в файл, добавление файла в дерево

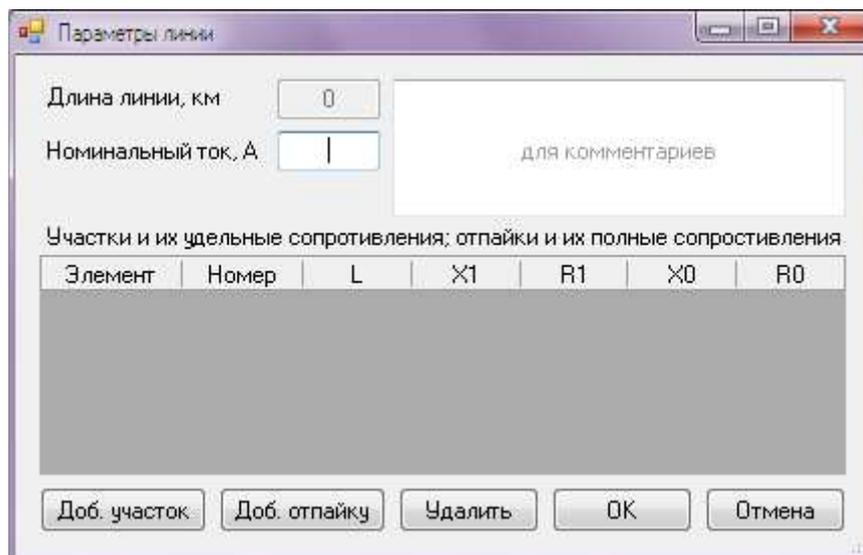
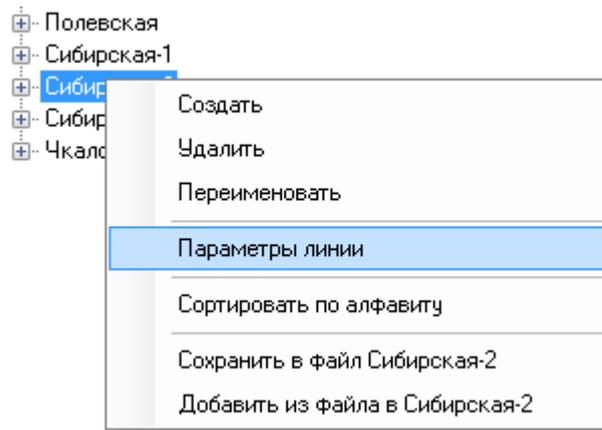
В контекстном меню дерева сигналов находятся команды сохранения в файл выбранной части дерева и добавления файла в выбранную часть дерева. Если вызвать контекстное меню ниже последнего из элементов дерева, так чтобы ни один из элементов не был выбран, то можно сохранить всё дерево либо добавить файл непосредственно в корень дерева.



Файл сохраняется в кодировке UTF-8 и по умолчанию имеет расширение .tsv.

### 2.1.3 Параметры определения места повреждения (ОМП)

В режиме дерева регистрируемые сигналы сопоставляются с линиями электропередач. Для каждой линии должны быть определены (перетаскиванием из дерева устройств) как минимум три канала тока ( $I_a$ ,  $I_b$ ,  $I_c$ ) и три канала напряжения ( $U_a$ ,  $U_b$ ,  $U_c$ ). После этого в контекстном меню для линии становится доступным пункт "Параметры линии", позволяющий указать участки и отпайки на этой линии для целей определения места повреждения (для каждого участка указывается его длина, а также реактивное и активное сопротивления прямой и нулевой последовательностей; общая длина линии считается автоматически как сумма длин участков и может использоваться для контроля правильности ввода длин участков).



### 2.1.4 Автоматическое получение файлов конфигурации и осциллограмм

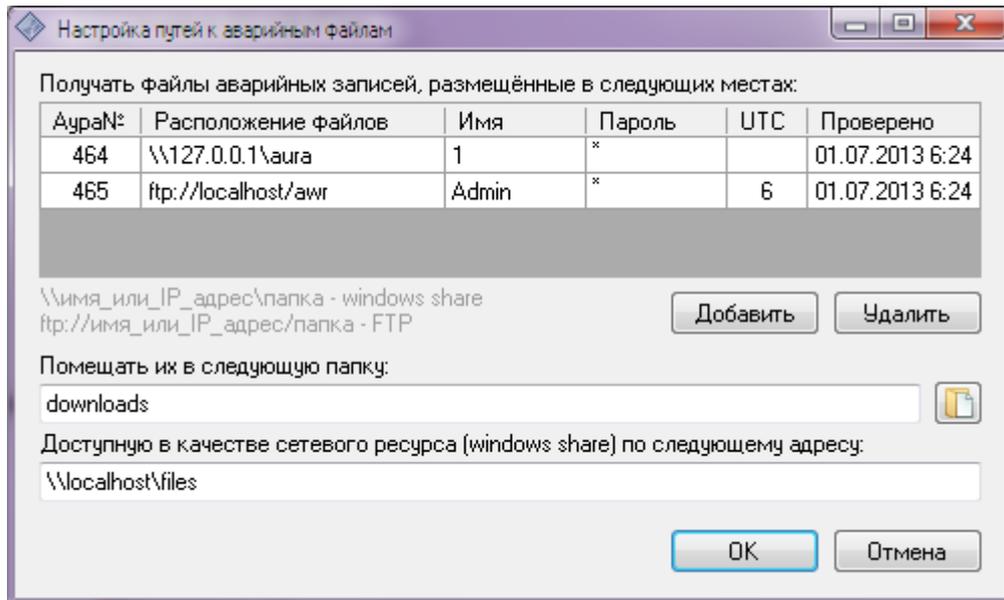
После получения аварийной записи по протоколу МЭК 60870-5-104 и её обработки, служба может обратиться к регистратору за получением более детальной записи аварии, а именно осциллограммы. Полученные осциллограммы сохраняются в виде файлов на сервере и просматриваются программой Aura2000.exe, входящей в состав комплекса.

Расположенная в панели инструментов кнопка "Настройка получения файлов"  открывает окно, представленное на рисунке ниже. В верхней части окна расположена таблица с параметрами подключения к регистраторам. В ней указываются:

- Аура№ - номер регистратора
- Расположение файлов - FTP или UNC адрес/путь к папке, в которую регистратор помещает конфигурационные и аварийные файлы; в случае, если это два разных места (как правило, это так), то для регистратора создаётся две отдельных записи -

путь к конфигурационным файлам на одной строке и путь к аварийным файлам на другой

- Имя - имя пользователя для авторизации при обращении к регистратору
- Пароль - для авторизации при обращении к регистратору
- UTC - смещение времени регистратора относительно UTC, в часах; при работе всех регистраторов, серверов, и АРМ пользователей в одной временной зоне допускается не указывать
- Проверено - время последнего успешного обращения за файлами к указанному расположению, независимо от того были там на этот момент новые файлы или нет



В нижней части окна указываются путь к папке, в которую служба будет помещать полученные от регистраторов файлы. Файлы размещаются по подпапкам: аварийные файлы в папки с номерами регистраторов, конфигурационные файлы в общую папку config.

В самом низу указывается путь UNC к папке выбранной выше. К этому сетевому ресурсу, который рекомендуется делать доступным только для чтения, будут обращаться пользователи со своих рабочих мест для получения файлов осциллограмм.

### 2.1.5 Резервное копирование и экспорт данных

Расположенная в панели инструментов кнопка резервного копирования  позволяет загрузить с сервера и сохранить в указываемое место полную резервную копию базы данных в стандартном для Microsoft SQL-Server формате .bak. Программа не допускает автоматическое восстановление базы данных из резервной копии, оставляя эту работу администратору.

Альтернативно, кнопка резервного копирования может быть нажата с клавишей Shift или Shift+Ctrl для получения данных в формате нескольких текстовых файлов (за две недели или за весь период эксплуатации соответственно) с сохранением в подкаталог tsv в каталоге программы. Это служебная диагностическая функциональность, которая в дальнейшем может быть изменена или расширена.

### 2.2 Текстовые сообщения о произошедших событиях

После получения аварийной записи от регистратора, служба AuraHub немедленно проводит её обработку, как встроенными в программу методами, так и настраиваемыми методами, с применением пользовательских скриптов на языках JavaScript и VBScript.

По результатам обработки аварийной записи формируются текстовые сообщения о том, что именно произошло - какие дискретные сигналы изменили состояние, были ли короткие замыкания (если да, и известны параметры линии, указывается расстояние до места

короткого замыкания), а также любой другой анализ, описанный в пользовательском скрипте, например оценка правильности срабатывания защиты.

На рабочем месте оператора программа AuraHub запускается в режиме отслеживания новых событий, позволяя просматривать сводку событий, их архив и оперативно уведомляя оператора о произошедших событиях.

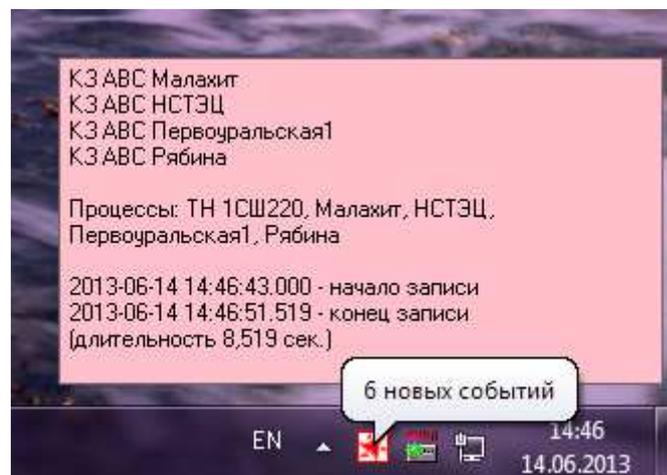
### 2.2.1 Сводка событий, архив событий, уведомления о событиях

В сводке отображаются все недавние несквитированные события, а также может отражаться часть более давних и сквитированных событий, общим числом до 100, для создания исторического контекста для новой информации. При необходимости просмотра событий, произошедших раньше и не попавших в сводку, в контекстном меню сводки следует выбрать пункт "Архив событий", который позволит подгрузить для просмотра события за любой указанный период.

Квит	Время	Место	Событие
<input checked="" type="checkbox"/>	2002-10-02 19:34:56.980	АУРА №25	===== Конец аварийной записи =====
<input checked="" type="checkbox"/>	2002-10-02 19:34:54.280	ВЛ500_СГРЭС2	Двухфазное КЗ (АС) с 2,280 по 2,420 сек. на 110,3 км
<input checked="" type="checkbox"/>	2002-10-02 19:34:54.280	ВЛ500_СГРЭС1	Двухфазное КЗ (АС) с 2,280 по 2,420 сек. на -91,3 км
<input checked="" type="checkbox"/>	2002-10-02 19:34:52.140	ВЛ500_СГРЭС2	Двухфазное КЗ (АС) с 0,140 по 0,240 сек. на 119,4 км
<input checked="" type="checkbox"/>	2002-10-02 19:34:52.140	ВЛ500_СГРЭС1	Двухфазное КЗ (АС) с 0,140 по 0,240 сек. на -89,5 км
<input checked="" type="checkbox"/>	2002-10-02 19:34:52.000	АУРА №25	===== Начало аварийной записи =====

Наличие в сводке несквитированных событий вызывает мигание и красный цвет иконки, расположенной в системном трее; после квитирования оператором всех поступивших событий иконка перестает мигать и принимает зелёный цвет (серый цвет иконки говорит об отсутствии связи с сервером или со службой).

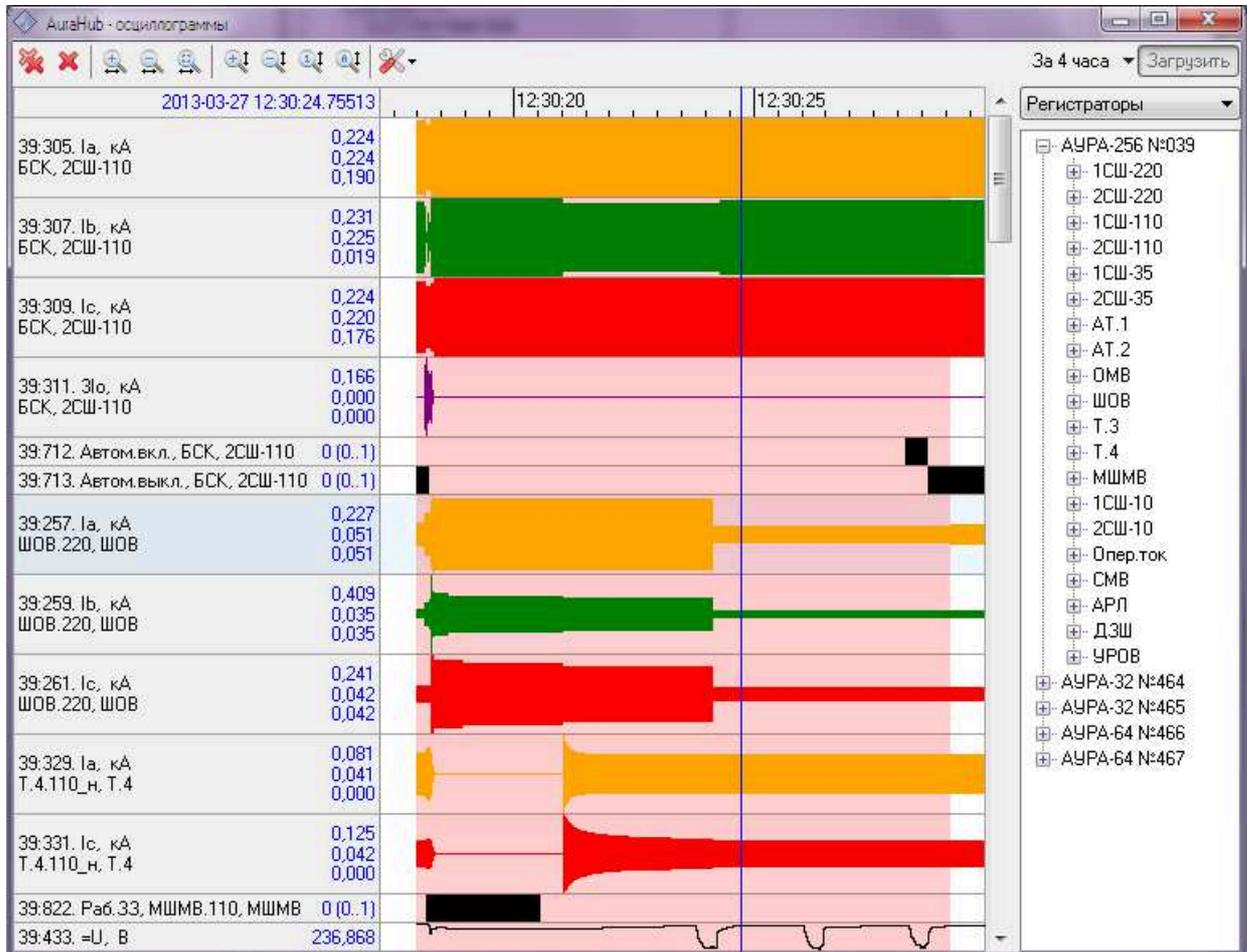
Также при поступлении новых событий появляется всплывающая подсказка с указанием их общего количества, а некоторые события (результаты работы скриптов и сообщения о пропадании каналов связи с регистраторами) сразу выводятся на рабочий стол.



### 2.3 Просмотр графиков аварийных процессов

Графики, полученные по протоколу МЭК 60870-5-104, представлены в виде действующих значений сигналов. Графики фазных напряжений и токов приближены по виду к огибающим синусоидальным графикам (изображаются закрашиванием симметрично относительно центральной линии, соответствующей нулевому уровню). Графики прочих аналоговых

значений изображаются тонкой линией, изменяясь от минимального значения внизу до максимального вверх. В обоих случаях масштаб по вертикали выбирается автоматически, исходя из высоты строки и фактического диапазона изменения сигнала на выбранном интервале времени. Дискретные сигналы изображаются наличием закрашивания (логический уровень 1, включено) или его отсутствием (логический уровень 0, выключено).



Добавление каналов на график осуществляется путём перетаскивания из представленного в виде дерева общего списка, расположенного в правой части окна.

При открытии окна выбранный интервал времени равен либо одному дню, либо, при переходе из сводки событий, длительности аварийной записи, соответствующей событию. Выбрать другой интервал можно из выпадающего списка в правой верхней части окна. При выборе длительного (более трёх месяцев) интервала графики сигналов показаны не будут, а будет показано только наличие аварийных записей на то или иное время. После масштабирования и уточнения интервала следует нажать расположенную в правом верхнем углу кнопку "Загрузить" для загрузки и отображения графиков.

Масштабирование по горизонтали осуществляется либо кнопками в панели инструментов либо при помощи мыши. Для приближения следует нажать левую клавишу мыши в начале интересующего интервала и, удерживая её, перевести курсор на момент окончания интервала (вправо), где и отпустить левую клавишу. Аналогичная операция, с движением курсора влево возвращает к полному выбранному масштабу времени. Графики можно смещать вправо и влево перетаскиванием используя правую клавишу мыши.

Масштабирование по вертикали осуществляется увеличением и уменьшением высоты строк при помощи кнопок панели инструментов либо перетаскиванием границы между заголовками строк в левой части окна. Внутри каждой строки масштабирование автоматическое.

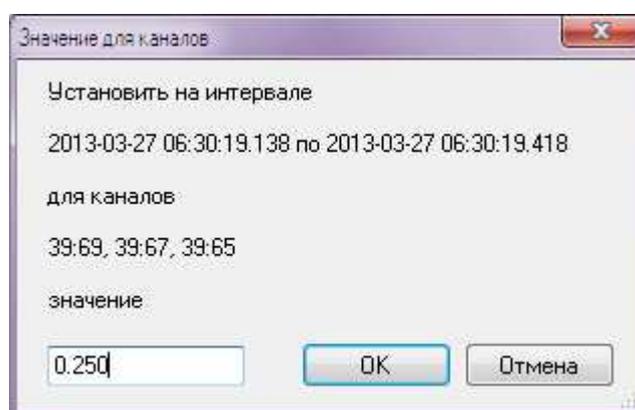
Кнопки панели инструментов: "Удалить все каналы", "Удалить выбранные каналы", "Увеличить масштаб по горизонтали", "Уменьшить масштаб по горизонтали", "Вместить"

по горизонтали", "Увеличить масштаб по вертикали", "Уменьшить масштаб по вертикали", "Автоматический масштаб по вертикали", "Единый масштаб по вертикали", "Инструменты".

### 2.3.1 Модификация графиков для проверки скриптов

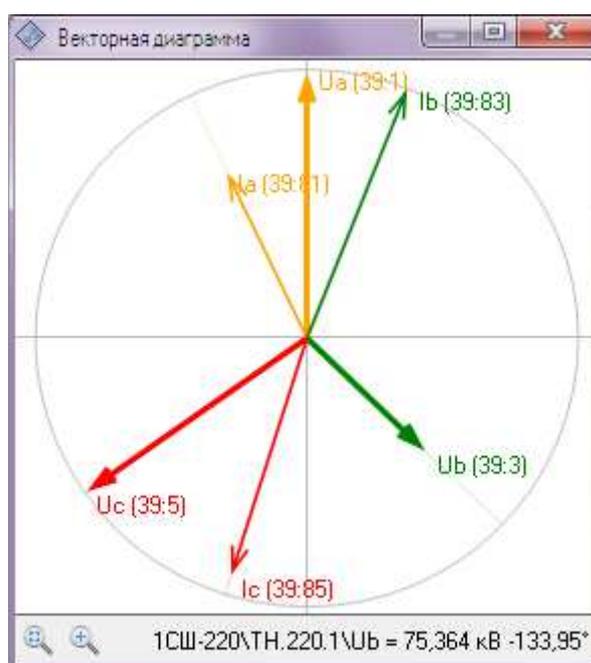
В выпадающем меню "Инструменты" расположены команды для сохранения видимого фрагмента в файл особого формата, загрузки из файла для просмотра, формирования и просмотра (без помещения в базу) сводки событий, соответствующей видимому фрагменту. Этими командами, можно пользоваться для проверки функционирования скриптов.

Также для проверки функционирования скриптов предусмотрена операция редактирования графика. Для этого следует выполнить действия, аналогичные приближению при помощи мыши (нажать - потянуть вправо - отпустить), но с нажатой кнопкой Alt. При этом появится окно, в котором перечислены выделенные каналы и предлагается указать для них новое значение. Все внесённые изменения сохраняются только до закрытия окна и не отражаются в базе, но могут быть сохранены в файл для последующей загрузки.



### 2.3.2 Векторные диаграммы

При нажатии на расположенную на панели инструментов кнопку "Векторная диаграмма" открывается окно, пример которого приведён ниже. Вектора для отображения выбираются двойным кликом мыши на левом столбце окна графиков, содержащем названия каналов.



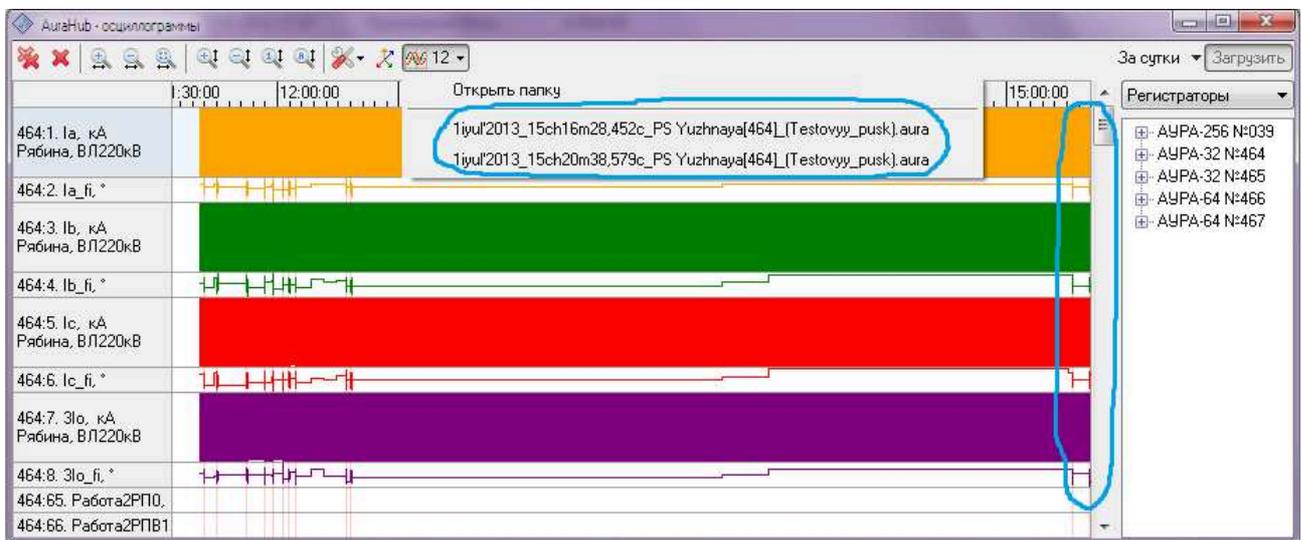
В данном случае отображаются три тока и три напряжения, причём вектора напряжений выделяются более толстой линией и сплошной стрелкой. Каждый вектор подписан и при

наведении курсора мыши на подпись в нижней части окна появляется более подробная информация о названии и значении вектора.

Также в нижней части окна расположены кнопки масштаба: "Неизменный масштаб"  (при нажатии которой масштаб выбирается по наибольшему значению за весь обозреваемый период и не меняется при сдвиге визира) и "Увеличить"  (при нажатии которой в размеры окна вписывается не весь круг с векторной диаграммой, а только часть реально занятая векторами).

### 2.3.3 Осциллограммы

На рисунке ниже показан вид графиков тока по одной из линий регистратора №464. Видно, что за период с 11:30 по 15:20 было сделано 12 записей, последние две из которых доступны в выпадающем меню "Осциллограммы" . Видно, что это тестовые пуски. Остальные 10 файлов в выпадающем меню отсутствуют, скорее всего потому, что это также были тестовые пуски и они были удалены оператором. При выборе в выпадающем меню одного из файлов он будет открыт в программе, ассоциированной с этим типом файла (например, для файлов \*.aura это скорее всего Aura2000.exe).

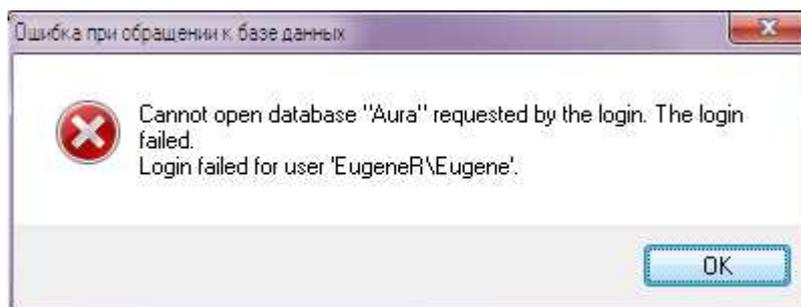
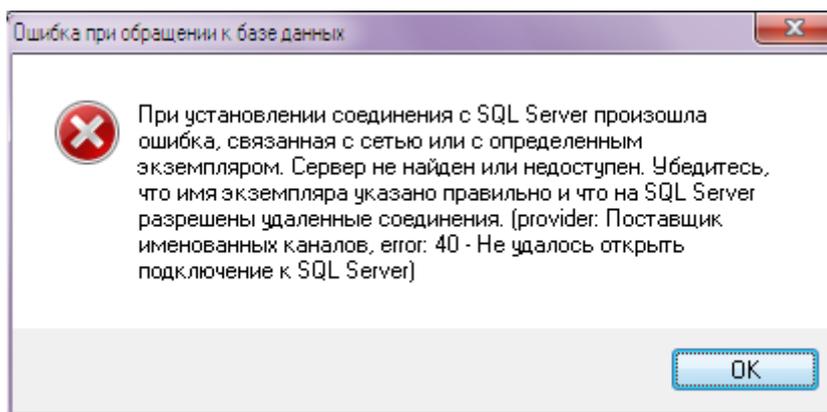


## 3. Установка и первоначальная настройка

### 3.1 Первый запуск

При запуске AuraHub пытается установить связь с базой данных и отобразить конфигурацию и текущее состояние подключений. По умолчанию (например, при отсутствии пользовательской конфигурации - AuraHub.cfg) производится попытка подключения к базе данных "Aura" на локальном компьютере.

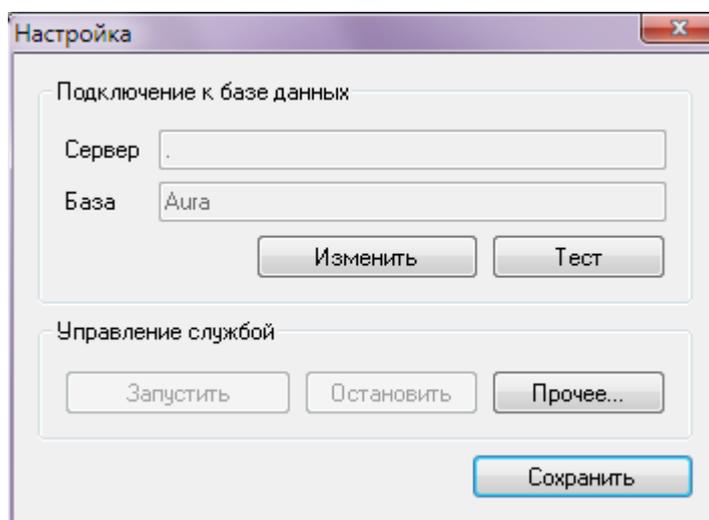
В случае если связь с базой данных установить не удалось, будет показано сообщение об ошибке. Ниже приведены два примера такого сообщения об ошибке - отсутствие соединения с сервером SQL и отсутствие на сервере SQL базы данных "Aura".



Убедившись, что сервер SQL запущен, нажатием кнопки  "Настройка" в панели инструментов главного окна программы открываем окно настройки.

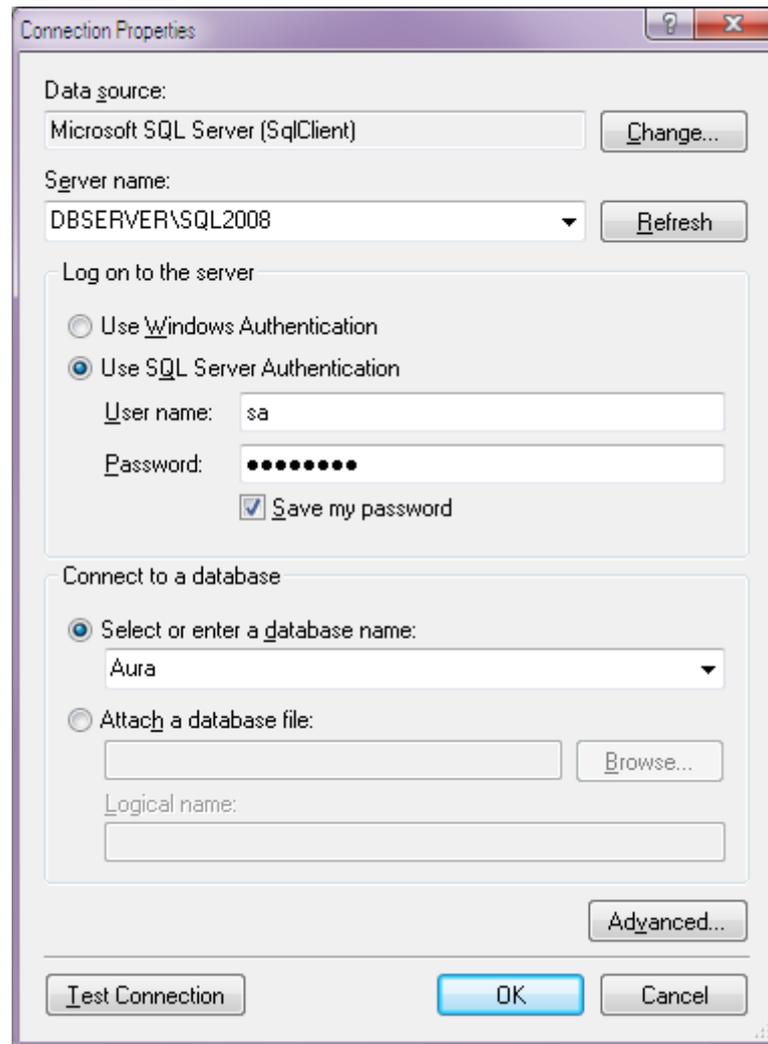
### 3.2 Окно настройки

Ниже показано окно настройки со значениями по умолчанию: в качестве сервера базы данных указан локальный компьютер ".", база данных "Aura", служба не установлена.



#### 3.2.1 Настройка подключения к базе данных

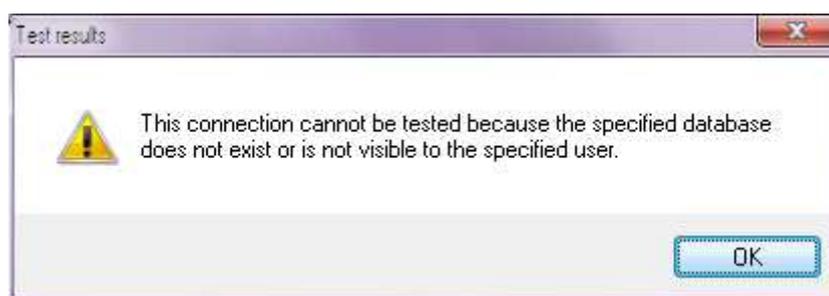
По кнопке "Изменить" открывается стандартный диалог настройки подключения к базе данных, который для примера приведён ниже.



В приведённой конфигурации используется сервер "DBSERVER", инстанция "SQL2008", авторизация SQL Server, пользователь "sa" (запись "sa" это администратор базы данных), база данных "Аура". В каждом конкретном случае параметры могут быть другими.

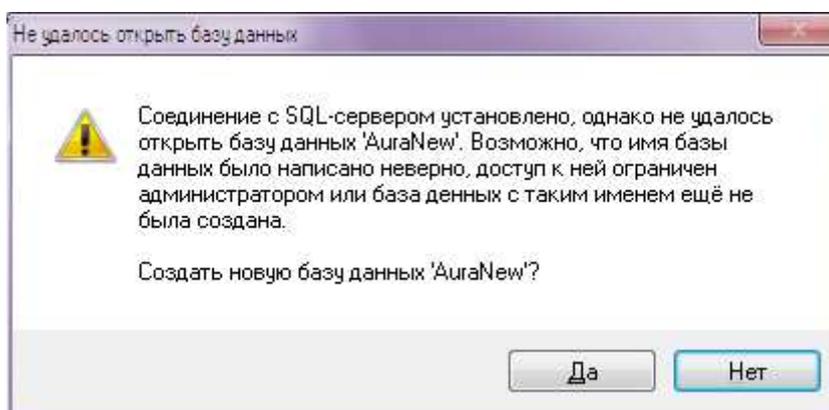
- Data source - источник данных; это значение менять не следует.
- Server name - сервер баз данных; может быть указан по имени (например, "SomeComputer") по имени с указанием инстанции SQL-сервера ("SomeComputer\SQLEXPRESS"), по IP-адресу (например, "192.168.0.123"), по IP-адресу с указанием через запятую номера порта ("127.0.0.1,1433"). Вместо имени локального компьютера может использоваться точка (".", "\SQLEXPRESS").
- Log on to the server - параметры авторизации на сервере SQL; возможны два варианта: с использованием учётных записей Windows (например, в домене или на локальном компьютере) либо с авторизацией непосредственно SQL-сервером (с явным указанием имени пользователя и пароля).
- Connect to a database - выбор базы данных; следует указать имя базы данных "Aura".

Расположенная в левом нижнем углу кнопка "Test Connection" проверяет доступность сервера и правильность введённых настроек. Если сервер доступен, но база данных "Aura" на нём ещё не создана, будет показано сообщение, приведённое ниже.



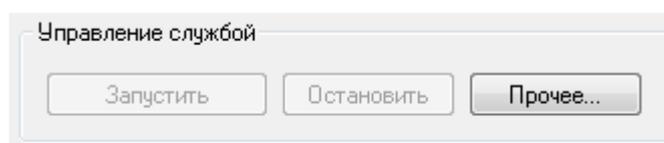
Такое сообщение означает, что все параметры указаны верно, однако необходимо создать базу данных.

По кнопке "Тест", расположенной справа от кнопки "Изменить" в окне настроек, производится проверка подключения, проверка структуры базы данных и, при необходимости, создание новой базы данных или обновление структуры существующей базы данных до уровня, требуемого программой.



### 3.2.2 Настройка службы (управление службой)

Для выполнения своих функций по непрерывному фоновому сбору данных программа AuraHub должна быть установлена как служба Windows. Команды установки и удаления службы находятся в меню, доступном при нажатии кнопки "Прочее...".



После того, как служба установлена, становятся доступны кнопки "Запустить" и "Остановить". Этими же кнопками следует пользоваться для перезапуска службы после изменения конфигурации.

## 4. Дополнительная информация

### 4.1 Ручная настройка

Как правило, ручная настройка не требуется. Этот раздел включён в документацию для полноты информации.

#### 4.1.1 Файл программной конфигурации

Файл программной конфигурации (AuraHub.config), если он присутствует в одном каталоге с программой, позволяет менять принятые программой настройки по умолчанию, в частности, параметры трассировки (протокола, лога) и ряд других. Это обычный XML-файл, который можно править в текстовом редакторе, однако ошибки в нём могут привести к

неработоспособности программы либо ухудшить её работу. Свяжитесь с разработчиками прежде чем вносить изменения в этот файл.

#### 4.1.1.1 Параметры трассировки (протокола, лога)

Для трассировки используются стандартные средства, входящие в состав Microsoft .NET Framework, и библиотека компонентов Essential.Diagnostics (присоединена внутрь AuraHub.exe). Используемые средства вводят следующие понятия:

- Источник трассировки - TraceSource; какая-либо функциональная часть программы
- Прослушиватель трассировки - TraceListener; программный модуль, осуществляющий какие-либо операции с трассировочными сообщениями (например, запись их в файл или системный журнал)
- Уровень трассировки - TraceLevel; уровень детализации, указывающий насколько подробно должна вестись трассировка

В программе определены следующие источники трассировки:

- Service - запуск/останов службы, а также всё, не включенное в какой-либо другой отдельный источник
- Iec104 - протокол МЭК 60870-5-104 (получение данных от регистраторов)
- DataWriter - запись полученных данных в базу SQL Server
- EventNarrator - формирование текстовых записей о событиях
- Downloader - получение аварийных и конфигурационных файлов

Определены следующие уровни трассировки, в порядке увеличения детализации:

- Critical
- Error
- Warning
- Information
- Verbose
- All

Прослушиватели трассировки указываются по мере необходимости, подробности см:

<http://essentialdiagnostics.codeplex.com/documentation>

<http://msdn.microsoft.com/ru-ru/library/1txedc80.aspx>

<http://msdn.microsoft.com/ru-ru/library/zs6s4h68.aspx>

Ниже приведён фрагмент файла конфигурации с трассировкой службы и подробной трассировкой МЭК 60870-5-104. Изменяемые части, включая шаблон имени файла трассировки, выделены курсивом и прерывистым подчёркиванием.

```
<configuration>
  <system.diagnostics>
    <sources>
      <source name="Service">
        <listeners>
          <clear />
          <add name="RollingFile"/>
        </listeners>
      </source>
      <source name="Iec104">
        <listeners>
          <clear />
          <add name="RollingFile"/>
        </listeners>
      </source>
    </sources>
    <switches>
      <add name="Service" value="Information" />
      <add name="Iec104" value="All" />
    </switches>
  </system.diagnostics>
</configuration>
```

```
</switches>
<sharedListeners>
  <add name="RollingFile"
    type="Essential.Diagnostics.RollingFileTraceListener, Essential.Diagnostics"
    initializeData="{ApplicationName}-{DateTime:yyyy-MM-dd}.log"
    template="{LocalDateTime:yyyy-MM-dd HH:mm:ss.fff} {Message}{Data}"
  />
</sharedListeners>
<trace autoflush="true"/>
</system.diagnostics>
...
</configuration>
```

#### 4.1.1.2 Период общего опроса

Период общего опроса, в секундах.

```
<configuration>
...
<appSettings>
...
  <add key="GI.Period" value="1800"/>
</appSettings>
</configuration>
```

#### 4.1.1.3 Web-интерфейс

Web-интерфейс службы весьма примитивен и служит в основном для передачи информации о состоянии службы пользовательскому интерфейсу программы. Впрочем, возможен доступ и напрямую из браузера. По умолчанию используется порт 4790 на локальной машине, без доступа по локальной сети (<http://localhost:4790/AuraHub/>). Ниже приведён фрагмент файла конфигурации, делающего интерфейс доступным на всех IP адресах ("+") на стандартном для HTTP порту 80 (значение параметра "WebUI.Prefix"). В соответствии с этим также изменён адрес, по которому к службе будет обращаться пользовательский интерфейс (значение параметра "WebUI.Address").

```
<configuration>
...
<appSettings>
  <add key="WebUI.Prefix" value="http://+:80/AuraHub/" />
  <add key="WebUI.Address" value="http://localhost/AuraHub/" />
...
</appSettings>
</configuration>
```

#### 4.1.2 Файл команд ежедневного обслуживания

Файл команд ежедневного обслуживания (daily.cmd), запускается, на выполнение службой AuraHub каждую ночь около 3:00, если он находится в одном каталоге с AuraHub.exe. Может содержать команды для сжатия/удаления/пересылки журналов, резервного копирования базы данных и т.п.

#### 4.2 Составление скриптов генерации отчётов

Скрипты составляются на языке JScript или VBScript во встроенном текстовом редакторе и сохраняются в базе данных с указанием названия скрипта и номера регистратора, записи которого этим скриптом должны обрабатываться.

Кроме обычных объектов и функций в соответствии с документацией (см. интернет и/или литературу) скриптам в AuraHub доступны ещё два объекта: аварийная запись и отчёт. Через первый из этих объектов скрипт получает входные данные об аварии, через второй выводит результаты её обработки.

При поступлении новых данных (новой аварийной записи) от регистратора, скрипты, сопоставленные с этим регистратором будут вызваны автоматически.

```

26
27 Report.AppendLine("")
28 if (Record.FindFault(97, 99, 101, 1.2, 0)) {
29     Report.AppendLine(
30         "КЗ СРЭС_1: " + Record.T1.toFixed(3) +
31         " по " + Record.T2.toFixed(3) + " мс"
32     )
33     while (Record.FindFault(97, 99, 101, 1.2, Record.T2)) {
34         Report.AppendLine(
35             "Повтор КЗ СРЭС_1: " + Record.T1.toFixed(3) +
36             " по " + Record.T2.toFixed(3) + " мс"
37         )
38     }
39 }
40

```

#### 4.2.1 Объект Record (запись)

`double` T1

Значение устанавливается функциями `FindAnalogOver`, `FindAnalogUnder`, `FindDiscrete`, `FindFault` и соответствует (первому) найденному моменту времени.

`double` T2

Значение устанавливается функцией `FindFault` и соответствует второму найденному моменту времени.

`double` Duration

Длительность записи, в секундах.

`double` Analog(`int` N, `double` Time)  
`double` Discrete(`int` N, `double` Time)

Аналоговое или дискретное значение по каналу N на момент времени Time. Нумерация каналов в соответствии с тем как они передаются регистратором по протоколу МЭК-870-5-104, что отличается от аппаратной нумерации.

`bool` FindAnalogOver(`int` N, `double` StartTime, `double` EndTime, `double` Value)  
`bool` FindAnalogOverEx(`int` N, `double` StartTime, `double` EndTime, `double` Value, `out double` FoundTime)  
`bool` FindAnalogUnder(`int` N, `double` StartTime, `double` EndTime, `double` Value)  
`bool` FindAnalogUnderEx(`int` N, `double` StartTime, `double` EndTime, `double` Value, `out double` FoundTime)

Найти момент времени, в который аналоговое значение по каналу N станет больше или меньше, чем Value. Поиск ограничен интервалом от StartTime до EndTime. Функции `FindAnalogOver` и `FindAnalogUnder` помещают найденный момент времени в T1. Функции `FindAnalogOverEx` и `FindAnalogUnderEx` возвращают его в выходной параметр FoundTime (способ работает только в VBScript).

```
bool FindDiscrete(int N, double StartTime, double EndTime, double Value)
bool FindDiscreteEx(int N, double StartTime, double EndTime, double Value,
out double FoundTime)
```

Найти момент времени, в который аналоговое значение по каналу N станет равно Value, где Value это 1 или 0. Поиск ограничен интервалом от StartTime до EndTime. Функция FindDiscrete помещает найденный момент времени в T1. Функция FindDiscreteEx возвращает его в выходной параметр FoundTime (способ работает только в VBScript).

```
bool FindFault(int NIa, int NIb, int NIc, double Inom, double StartTime)
bool FindFaultEx(int NIa, int NIb, int NIc, double Inom, double StartTime,
out double FoundTime1, out double FoundTime2)
```

Найти моменты времени начала и окончания режима на основании передаваемых номеров каналов NIa, NIb, NIc и номинального значения тока Inom. Поиск ведётся с момента времени StartTime до конца записи. Функция FindFault помещает найденные моменты времени в T1 (начало аварийного режима) и T2 (окончание аварийного режима). Функция FindFaultEx возвращает их в выходные параметры FoundTime1 и FoundTime2 (способ работает только в VBScript).

```
double U0(int Na, int Nb, int Nc, double Time)
double U1(int Na, int Nb, int Nc, double Time)
double U2(int Na, int Nb, int Nc, double Time)
```

Возвращает значение нулевой, прямой или обратной последовательности на основании передаваемых номеров каналов Na, Nb, Nc на момент времени Time.

```
string ATime(double Time)
```

Преобразует время из числа секунд с момента начала записи в строковый формат даты и времени "гггг-мм-дд чч:мм:сс.xxx".

#### 4.2.2 Объект Report (отчёт)

При обращении к объекту "отчёт" название объекта можно не указывать. Так, оба следующих обращения: Report.Clear и Clear одинаково верны.

```
Clear()
```

Полностью очищает формируемый отчёт.

```
bool HasText
```

Указывает на наличие какого-либо текста в отчёте (используется, например, если требуется дописать что-либо в отчёт только если отчёт не пуст).

```
Append(string text)
AppendLine(string text)
```

Добавляет к отчёту указанный текст либо текст и перевод строки.

#### 4.2.3 Примеры скриптов

Пример скрипта на языке JScript

```
Append("Авария на ПС Холмогорская от " + Record.ATime(0))
AppendLine(" длительностью " + Record.Duration.toFixed(3))
```

```

AppendLine("")

if (Record.FindFault(97, 99, 101, 1.2, 0)) {
    Append("КЗ СГРЭС_1: " + Record.T1.toFixed(3))
    AppendLine(" по " + Record.T2.toFixed(3) + " мс")
    while (Record.FindFault(97, 99, 101, 1.2, Record.T2)) {
        Append("Повтор КЗ СГРЭС_1: " + Record.T1.toFixed(3))
        AppendLine(" по " + Record.T2.toFixed(3) + " мс")
    }
}
}

```

Пример результата выполнения:

Авария на ПС Холмогорская от 2002-10-02 13:34:52.000 длительностью 4.980

КЗ СГРЭС\_1: 0.140 по 0.260 мс

Повтор КЗ СГРЭС\_1: 2.280 по 2.440 мс

Пример скрипта на языке VBScript

```

if Record.FindDiscreteEx(530, 0, Record.Duration, 1, T) then
    AppendLine "Пуск ВЧ Тарко-Сале:"
    AppendLine Round(T, 3) & " - срабатывание"
    while Record.FindDiscreteEx(530, T, Record.Duration, NextState, T)
        if NextState = 0 then
            AppendLine Round(T, 3) & " - возврат"
        else
            AppendLine Round(T, 3) & " - срабатывание"
        end if
        if NextState = 0 then NextState = 1 else NextState = 0
    wend
end if
AppendLine ""

```

Пример результата выполнения:

Пуск ВЧ Тарко-Сале:

0,158 - срабатывание

0,740 - возврат

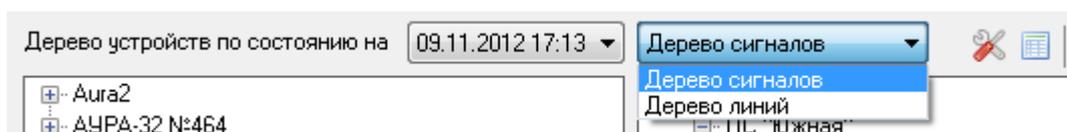
2,304 - срабатывание

2,878 - возврат

## 5. Изменения

### 5.1 Новое в версии 0.9.0

Дерево сигналов теперь не единственное - добавлено дерево линий.

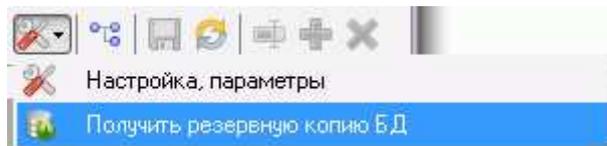


Для упрощения редактирования деревьев добавлены: сортировка (0), выбор интересующих элементов галочками (2.1.2.1).

Добавлена возможность сохранения дерева сигналов в файл (2.1.2.3).

## 5.2 Новое в версии 0.10.0

Добавлена в пользовательский интерфейс возможность резервного копирования базы данных.



Добавлено формирование текстовых сообщений о событиях, их просмотр и уведомление о появлении новых сообщений (п. 2.2.1).

Для формирования произвольных текстовых отчётов реализована интеграция с Windows Scripting Host и поддержка двух скриптовых языков программирования: JScript и VBScript (п. 4.2).

## 5.3 Новое в версии 0.11.0

Добавлена возможность просмотра графиков (п. 2.3), сохранения видимого фрагмента в файл и загрузка его для просмотра.

Добавлены всплывающие уведомления о событиях (п. 2.2.1).

## 5.4 Новое в версии 0.12.0

Добавлен просмотр векторных диаграмм (п. 2.3.2). Автоматизирована загрузка файлов конфигурации и аварийных записей (п. 2.1.4). Добавлена возможность быстрого перехода к просмотру осциллограмм (п. 2.3.3).

## 5.5 Новое в версии 1.0.0

Добавлена возможность фильтрации в сводке событий. Добавлена возможность проставлять замеры на графиках. Добавлено окно сводки состояния регистраторов. Изменён интерфейс графиков в части операций с мышью (назначение клавиш, операции перетаскивания и т.п.).