

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«РАКУРС-ИНЖИНИРИНГ»



Заказчик: ООО «Ракурс-инжиниринг»

Договор №: B07.2016

Руководство пользователя/администратора

Программа отображения архивных данных «**Trends**»

САНКТ-ПЕТЕРБУРГ
2017

Наименование организации, предприятия	Должность исполнителя	Фамилия, инициалы	Подпись	Дата
ООО «Ракурс-инжиниринг»	Разработал	Лаушкин Д.С.		
	Проверил	Игнатенков С.Б.		
	Нормоконтроль	Игнатенков С.Б.		
	Утвердил	Игнатенков С.Б.		

Настоящее руководство устанавливает основные правила эксплуатации и назначение программного обеспечения отображения архивных данных «Trends».

Содержание

1	Настройка.....	6
1.1	Структура файла Trends.ini	6
1.2	Структура базы данных. Источники и формат данных.....	11
1.2.1	Raw-файлы (данные FinsLogger).....	11
1.2.2	Текстовые файлы (txt).....	12
1.2.3	База данных с поддержкой SQL и ADO	13
1.2.4	SQLite.....	14
1.3	Поддержка многоязычности	14
1.4	Списки (справочники)	14
1.4.1	Общие	14
1.4.2	Аналоговые сигналы	15
1.4.3	История событий.....	17
1.4.4	Отчёты.....	19
1.4.5	Аварийные срезы.....	20
1.5	Шаблоны отчётов	20
1.5.1	Стандартный способ доступа к данным	21
1.5.2	Прямой доступ к данным	21
1.5.3	Форматирование и редактирование	23
1.5.4	Графики в отчётах.....	23
1.6	Параметры командной строки.....	24
2	Эксплуатация	26
2.1	Главное меню	26
2.1.1	Просмотр данных.....	26
2.1.2	Управление временным интервалом	26
2.1.3	Управление окнами.....	26
2.1.4	Управление доступом.....	26
2.1.5	Язык интерфейса и информации	27
2.1.6	Информация о программе	27
2.2	Графики	27
2.2.1	Меню	27
2.2.2	Управление графиком	29
2.3	Отчёты	29
2.3.1	Меню	29
2.4	История событий	30
2.5	Управление интервалом времени.....	31
2.6	Управление пользователями.....	31
2.6.1	Меню	31

Программное обеспечение «Trends» (далее ПО) для отображения истории событий, истории изменения аналоговых сигналов (графиков) и отчётов, для просмотра и анализа накопленной информации об изменении параметров системы и возникших событиях.

Программа состоит из следующих файлов:

- **Trends.exe** – исполняемый файл программы;
- **Trends.ini** – файл основных настроек программы;
- **TrendsUser.ini** – пользовательские настройки (файл создаётся автоматически);
- **SavedTrends.ini** – наборы графиков (при сетевом использовании объединяются файлы на сервере и локальный; при необходимости файл создаётся автоматически);
- **ReportEdit3.exe** – редактор шаблонов отчётов и готовых отчётов.

1 НАСТРОЙКА

1.1 Структура файла Trends.ini

Все основные настройки программы хранятся в файле Trends.ini. Этот файл должен находиться в каталоге программы. В случае работы с сетевого диска, файл Trends.ini располагается в каталоге Windows и может содержать параметры регистрации программы (секция [Register]) и любые другие секции файла Trends.ini со специфическими для данного компьютера настройками. Все остальные настройки берутся из файла Trends.ini в каталоге программы на сетевом диске. При локальном запуске и отсутствии Trends.ini в каталоге программы, будет предложено выбрать файл настроек в любом каталоге. Таким образом, для запуска программы достаточно одного файла Trends.exe.

Настройки интерфейса (положение и размер окон) хранятся в файле TrendsUser.ini, который располагается в папке программы, или в каталоге Windows, при работе с сетевого диска.

Одновременно может быть запущена только одна копия программы. Вторая копия может быть запущена из другого каталога или с другим именем исполняемого файла.

[Register]

Ключи регистрации программы. Ключ указывается либо в параметре Key, либо для каждого компьютера указываются в параметре, соответствующем имени компьютера. При запуске незарегистрированной программы, в окне "О программе" будет предложено выполнить регистрацию и будет указан серийный номер, по которому создаётся ключ регистрации, а в папке программы будет создан файл Key.txt с данными для регистрации.

```
Key=...
Workstation1=...
Workstation2=...
```

[Startup]

Настройки программы, определяющие её состояние при запуске.

```
DataBase=DataBase1
```

База данных по умолчанию.

```
AccessRights=1
```

Уровень доступа по умолчанию (0 – пользователь, остальные уровни настраиваются).

```
DefTime=8h
```

Начальный интервал времени. Здесь и далее время указывается с суффиксами: d – дней, h – часов, m – минут, s – секунд, z – миллисекунд (например: 7d, 15m, 30s, 500z). Конструкции вида 1h30m не поддерживаются.

```
Language=Russian
```

Язык интерфейса программы.

```
LanguageKey=119
```

Скан-код клавиши циклического переключения доступных языков. 119 соответствует клавише <F8>.

```
Locale=1049
```

Принудительная установка локализации программы для корректного отображения кириллицы на системах с английской (или другой) локализацией (коды локализации: 1049 – Русская, 1033 – Английская (США)).

```
SystemFont=Arial
SystemFontSize=10
SystemFontStyle=1
```

Шрифт интерфейса программы. Стиль определяется комбинацией значений 1-жирный и 2-курсив.

```
Debug=1
```

Управление отображением отладочной информации.

[Languages]

Список доступных языков.

```
Russian=Ru
English=En
```

Каждому языку соответствует сокращение (Ru, En). Программой изначально поддерживаются русский и английский языки (Russian, English), строки интерфейса и изображения флагов для них содержатся в Trends.exe. Для поддержки других языков или для изменения стандартных строк и флагов в каталоге программы могут находиться файлы <язык>.lng (содержит строки интерфейса программы) и Flag<сокращение>.bmp (содержит изображение флага). Например: English.lng и FlagEn.bmp.

[DataBases]

Список доступных баз данных.

```
DataBase1=База данных #1
DataBase2=База данных #2
```

Каждой базе данных соответствует текстовое описание, которое отображается в меню.

[DataBase1]

Настройки базы данных, задаются отдельно для каждой из баз данных.

```
SQL="SQL:Provider=SQLOLEDB.1;Database=DataBase1;Server=DBServer;UID=sa;PWD="
Raw="Raw:DBServer\DB@9620"
Txt="Txt:DBServer\DB@9620"
```

Список источников данных. Названия источников данных произвольны. Первые буквы описаний источников определяют тип данных (регистр безразличен):

- **SQL** – база данных с поддержкой SQL и ADO;
- **RAW** – база данных в формате RAW (через RawServer);
- **TXT** – текстовые таблицы с данными, разделёнными запятой (через RawServer);
- **SLT** – файл базы данных SQLite.

Для источников типа SQL указывается стандартная строка подключения. Для источников Raw, Txt и Slт используется доступ через RawServer и строка подключения указывается в следующем виде: <источник>="Raw|Txt|Slт:<сервер>[<база>]@<порт>". Прямой доступ к данным начиная с версии 2.7.0 не поддерживается. В строке подключения может быть указано имя или IP-адрес сервера. Имя базы данных может не указываться, если на сервере доступна только одна база данных.

```
Host=DBServer1,DBServer2
SQL="SQL:Provider=SQLOLEDB.1;Database=DataBase1;Server=%host%;UID=sa;PWD="
Raw="Raw:%host%\Data@9620"
```

В случае использования дублированных серверов, их список указывается в параметре Host, а строки подключения и пути могут содержать строку %host%, которая заменяется на текущий используемый сервер. Каждый сервер из списка может быть задан именем или IP-адресом, а также может содержать часть пути к данным, если пути к данным для разных серверов различны. При старте программы все сервера проверяются на доступность, и недоступные исключаются из дублирования на время работы программы. В случае ошибок доступа к данным, происходит автоматическое циклическое переключение на следующий сервер из списка.

Analog=Raw

Источник данных для аналоговых сигналов (графиков).

History=Raw

Источник данных для истории событий.

Report=SQL

Источник данных для отчётов.

List=Txt

Источник данных для списков (списки сигналов, устройств и другая вспомогательная информация).

Plants=1,2,3

Номера доступных установок (агрегатов) для объектов, состоящих из нескольких установок.

DefPlant=1

Номер установки (агрегата) по умолчанию.

HistoryPrintPreview=1

Просмотр истории событий перед печатью (0 – печать без просмотра, 1 – с просмотром (значение по умолчанию)).

HistoryPrintTemplate=\\DBServer\Trends\History.fr3

Шаблон FastReport для печати и сохранения истории событий. Если шаблон не указан, то он будет сформирован автоматически с учётом текущей ширины колонок.

HistoryHtmlHeader=\\DBServer\Trends\HtmlHead.htm

Заголовок HTML-файла для сохранения истории событий в формате HTML. Если заголовок не указан, история будет сохранена со стандартным оформлением.

HistoryUTCTime=1

AnalogUTCTime=1

ReportUTCTime=0

AlarmsUTCTime=0

Признаки использования времени в UTC в источниках данных. Для отчётов существует возможность указывать этот признак для каждого из отчётов в отдельности в таблице Reports (Reports.UTCTime). По умолчанию, используется локальное время (значение - 0).

HistoryFilterTopLevel=0

Верхний уровень в дереве объектов в фильтре истории событий. 0 – контроллеры группируются по агрегатам, 1 – контроллеры отображаются общим списком.

HistoryColorField=-1

Номер поля в таблице истории событий, который определяет цвет всей строки (тех ячеек, что не имеют своего цвета).

HistorySortOff=0

Если не равно 0, то принудительная сортировка историй событий по времени не осуществляется.

[DataBase1HistoryFields]

Поля, отображаемые в истории событий. Поля отображаются в указанном порядке. Каждому полю соответствует указатель на его название в файле интерфейса для каждого из языков. Названия указываются в разделе HistoryView файла интерфейса.

```
Time=FldTime
Plant=FldPlant
PLC=FldPLC
Group=FldGroup
Device=FldDevice
Code=FldCode
Data1=FldData1
PlcState=FldPlcState
```

[DataBase1HistoryFilters]

Список папок, содержащих фильтры истории событий. Для дублированных подключений поддерживается макроподстановка %host% вместо имени сервера.

```
Серверные=\\Server1\Filters\
Персональные=D:\Trends\Filters\
```

[History]

Параметры окон истории событий.

```
ToolBar=1
```

Отображать панель с кнопками управления.

```
Menu=1
```

Отображать контекстное меню при нажатии правой клавиши мыши.

[HistoryTimeFormats]

Форматы даты и времени для окон истории событий.

```
0=dd.mm.yy hh:nn
1=dd.mm.yy hh:nn:ss
Default=dd.mm hh:nn:ss
3=hh:nn:ss
4=hh:nn:ss.zzz
```

Параметр Default используется по умолчанию. Формат соответствует функции FormatDateTime в Delphi.

[Trend]

Параметры окон графиков.

```
ToolBar=1
```

Отображать панель с кнопками управления.

```
Menu=1
```

Отображать контекстное меню при нажатии правой клавиши мыши.

```
MaxSeries=6
```

Максимальное количество серий данных в одном окне. По умолчанию: 100.

```
Colors=clLime,clBlue,clGray,$80FF,255
```

Цвета серий данных. Допустимы обозначения цветов, принятые в Delphi, шестнадцатеричные (с префиксом "\$") и десятичные числа, соответствующие цветам в системе RGB. По умолчанию: "clRed,clGreen,clBlue,clTeal,clPurple,\$80FF".

```
VertTimeLabels=1
```

Отображать метки оси времени вертикально (0 или 1). По умолчанию: 1.

```
SortPoints=1
```

Принудительная сортировка точек по времени (0 или 1). По умолчанию: 0.

```
StopAutoRefreshOnTouch=0
```

Отключение автообновления данных при щелчке в области графика. По умолчанию: 1.

[TrendTimeFormats]

Форматы даты и времени для окон графиков.

```
0=dd.mm.yy hh:nn  
1=dd.mm hh:nn  
2=dd.mm hh:nn:ss  
3=hh:nn  
Default=hh:nn:ss  
5=nn:ss.zzz
```

Параметр Default используется по умолчанию.

[MainForm]

Параметры главного окна программы.

```
CloseTime=0s
```

Время закрытия неактивной программы при переключении на другое приложение (0 - не закрывать при неактивности).

```
AlwaysOnTop=0
```

Всегда располагать окно программы поверх других окон приложений (1 – поверх других).

```
FullScreen=1
```

Открывать окно программы во весь экран, без строки заголовка и рамки и с блокировкой завершения программы по Alt+F4. Этот режим может быть использован для рабочих станций, предназначенных исключительно для отображения архивной информации.

```
NoFrame=1
```

Отображать окно программы без строки заголовка и рамки.

```
BorderIcons
```

Настройка кнопок управления окном.

[Buttons]

Настройка кнопок управления всех окон программы.

```
MainForm.BtnReport.Enabled=-1  
MainForm.BtnLanguage.Visible=-1  
AccessDlg.BtnAdd.Enabled=3  
AccessDlg.BtnDel.Enabled=3  
AboutDlg.Hit.Enabled=3
```

Настройка видимости и доступности отдельных кнопок и элементов контекстных меню для всех окон. Используется следующий формат: Окно.Элемент.[Visible|Enabled]=N. Список имён окон и элементов приведён в файлах интерфейса для каждого из языков. Значение свойства определяется следующим образом: N = -1 – "нет", N = 0 – "да", N >= 1 – "да", если уровень доступа >= N.

[Limits]

Ограничение максимального количества открытых окон каждого типа. На данный момент не поддерживается.

```
MaxTrends=4
MaxReports=1
MaxHistory=2
```

[AccessLevels]

Перечень уровней доступа. Нумерация с 1, чем больше значение, тем больше прав доступа.

```
Оператор=1
Начальник смены=2
Администратор=3
```

[Users]

Список зарегистрированных пользователей и их зашифрованных паролей.

```
admin=jolsisgpinfigo
```

[RefreshRates], [TimeIntervals]

Данные списки содержат периоды обновления и типовые отрезки времени, и могут быть указаны как в виде секций в Trends.ini, так и в виде таблиц в базе данных. Приоритет у таблиц в базе данных. Состав полей приведен в описаниях соответствующих таблиц базы данных.

1.2 Структура базы данных. Источники и формат данных

Независимо от формата хранения данных, предполагается, что база данных имеет определённую структуру: состав таблиц (таблиц, отображений или хранимых процедур, при доступе через ADO) с определёнными полями. В общем случае база данных состоит из неизменяемой информации – списков, и из накапливаемой информации – данных. Списки и различные виды данных (аналоговые сигналы, история событий, данные для отчётов) могут получаться из различных источников, каждый из которых настраивается отдельно в файле Trends.ini.

1.2.1 Raw-файлы (данные FinsLogger)

Доступ к Raw-файлам начиная с версии 2.7.0 возможен только через RawServer. При доступе через RawServer, каталог базы данных указывается в настройках RawServer, а в настройках Trends указывается имя или адрес сервера, имя базы данных и порт для доступа к RawServer.

Для определения структуры файлов Raw и Txt служит файл Tables.ini, который должен находиться в каталоге базы данных и иметь следующую структуру:

```
[Settings]
CatalogFmt=yyyy\mm\dd

[Tables]
SensorsGroups=Id,Parent,Name
Sensors=Id,Name,Code,Group,Period,Plant,Point,Report,Scale,Max,Min,Step,Units
TimeIntervals=Id,Name,Start,Duration
Analog=Value:SH
Msg*=Msec:USH,Group:USH,Device:USH,Code:UC,Error:UC,Data1:F,Data2:F
SensorsAUG?_avg_3600=S:QF/80
```

В файле должны быть перечислены все используемые таблицы типов Raw и Txt с указанием названий полей. Для Raw-файлов названия полей должны дополняться идентификатором типа данных, которые в этом поле хранятся. Кроме стандартных типов данных (SH, DH, F и т.д.), введён дополнительный тип VOID, который описывает пустое поле и предназначен для обеспечения возможности объединения таблиц с различной

структурой (в частности, в истории событий). Названия полей чувствительны к регистру символов! Поле Time для Raw-файлов не указывается. Названия таблиц могут быть заданы маской с символами "*" и "?".

При хранении аналоговых сигналов в формате Raw с записью каждого параметра в отдельный файл, имя которого соответствует номеру сигнала (Sensors.id), в файле Tables.ini должна быть описана таблица Analog, которая определяет структуру всех таких файлов. Данный формат хранения использовать не рекомендуется, он поддерживается для обеспечения совместимости.

Возможно, и в большинстве случаев оправдано, хранение нескольких аналоговых сигналов в одном файле (например, все сигналы каждого контроллера в отдельном файле). В этом случае в файле Tables.ini должны быть описаны структуры всех этих файлов, а в таблице Sensors для каждого сигнала должен быть указан файл, в котором он хранится (Sensors.Source). Хранение нескольких сигналов в одном файле позволяет увеличить скорость записи данных и экономит дисковое пространство, но несколько снижает скорость доступа к данным.

История событий может храниться как в одном файле для всех контроллеров, так и в отдельных файлах, которые в этом случае должны иметь одинаковый состав полей, хотя поля могут иметь различные типы, которые должны быть описаны.

В секции [Settings] ключ CatalogFmt определяет структуру каталогов данных. Ключ представляет собой формат даты, который используется для формирования суточных каталогов данных при записи (FinsLogger) и при чтении (Trends) данных. CatalogFmt=# соответствует устаревшему формату, при котором суточные каталоги имеют сквозную нумерацию с начала эры.

1.2.2 Текстовые файлы (txt)

Текстовые файлы могут использоваться для хранения списков и, в отдельных случаях, для хранения данных (доступ к ним осуществляется значительно медленнее). Структура текстового файла, как и Raw-файла, определяется списком полей в файле Tables.ini, но может быть указана в первой строке самого текстового файла после символа «#», что является предпочтительным.

Как и к Raw-файлам, доступ к текстовым файлам начиная с версии 2.7.0 возможен только через RawServer.

Данные в файле разделяются запятыми. Вещественные числа должны иметь точку в качестве разделителя целой и дробной части. Текстовые значения, содержащие символы-разделители (в том числе пробелы), должны быть заключены в кавычки. Рекомендуется все текстовые значения заключать в кавычки. Точка с запятой в начале строки может использоваться для комментариев или временного отключения строк, такие строки игнорируются. Допускаются пустые строки.

В случае большого числа повторяющихся строк, может быть использован расширенный синтаксис файла. Изменяющиеся значения полей в повторяющихся строках могут быть записаны в виде n1..n2, при этом автоматически будут созданы строки для этих и всех промежуточных значений (n2-n1+1 строк). При этом значения других полей могут содержать выражения и вычисляться исходя из значений изменяемых полей. Выражение должно заключаться в скобки следующего вида {#<выражение>} и может находиться в любой части поля. Результат выражения автоматически преобразуется к текстовому виду. Примеры исходных и результирующих строк:

```
#Id,Name
1..3,"Агрегат {#Id}" или 1..3,"{'Агрегат '+Id}"
1,"Агрегат 1"
2,"Агрегат 2"
3,"Агрегат 3"
```

```
#Id,Name,HistoryTable,Plant
{#Plant}2,"МНУ {#Plant}",Msg{#Plant},1..3
12,"МНУ 1",Msg1,1
22,"МНУ 2",Msg2,2
32,"МНУ 3",Msg3,3
```

При нескольких изменяемых полях создаются строки для всех комбинаций их значений.

```
#Id,Name,Plant,Index
{#100*Plant+Index},"A{#Plant}.Датчик {#Index+1}",1..2,0..2
100,"A1.Датчик 1",1,0
101,"A1.Датчик 2",1,1
102,"A1.Датчик 3",1,2
200,"A2.Датчик 1",2,0
201,"A2.Датчик 2",2,1
202,"A2.Датчик 3",2,2
```

Для вставки данных из других (произвольных) таблиц, в выражениях можно использовать функцию Join, которая имеет следующий формат:

```
Join(Table, Field, Field1, Param1, Field2, Param2, ..., Default)
```

Table – имя таблицы, из которой вставляются данные;

Field – имя поля в этой таблице, содержимое которого будет вставлено;

FieldX, ParamX – каждая пара параметров задаёт условие для поиска в таблице нужного значения (условие: **Table.FieldX = ParamX**). Пар условий может быть несколько, в этом случае они объединяются по "И", то есть должны выполняться все;

Default – значение по умолчанию (последний параметр, если функция имеет нечётное их число).

Пример:

```
Неисправность: {#Join('Errors','Message','Id',Id,'Group',Group)}
```

В данном случае таблица Errors должна иметь структуру: (Id, Group, Message). В сообщении вставляется строка "Неисправность: " и содержимое поля Message из таблицы Errors, поиск осуществляется по полям Id и Group, значения которых должны совпадать со значениями полей Id и Group таблицы, то есть (Errors.Id = Id)and(Errors.Group = Group).

1.2.3 База данных с поддержкой SQL и ADO

Данные могут быть получены из любого ADO источника, поддерживающего SQL (далее просто SQL). Для источников такого типа указывается строка подключения. Списки хранятся в таблицах, а для доступа к данным используются хранимые процедуры (Analog – аналоговые сигналы, Messages – история событий). При этом в качестве первых двух параметров хранимых процедур передаётся время начала и конца выбранного отрезка времени, третьим параметром передаётся номер сигнала (Sensors.id) для аналоговых сигналов.

```
create procedure Analog @BeginTime datetime, @EndTime datetime, @id int
create procedure Messages @BeginTime datetime, @EndTime datetime
```

Наборы данных, возвращаемые хранимыми процедурами или хранящиеся в других источниках данных, должны иметь следующую структуру:

Analog (Time, Value, [Id])

- **Time** – время регистрации значения;
- **Value** – значение;

- **Id** – номер аналогового сигнала (необходим в случае хранения нескольких аналоговых сигналов в одном Raw-файле).

Messages (Time, PLC, Group, Device, Code, [Data1], [Data2], [PlcState], ...)

- **Time** – время регистрации события;
- **PLC** – номер контроллера (PlcList.Id);
- **Group** – номер группы устройств (DeviceGroups.Id);
- **Device** – номер устройства (Devices.Id);
- **Code** – код события (MsgCodes.Id);
- **Data1, Data2** – необязательные дополнительные данные, которые могут быть использованы при формировании сообщения о событии (например, значения параметров);
- **PlcState** - необязательное поле, содержащее признаки «контроллер активен» (бит 0) и «контроллер исправен» (бит 1), которые позволяют определить состояние контроллера в момент формирования события.

1.2.4 SQLite

База SQLite может использоваться только для хранения списков. Использование аналогично текстовым файлам, при этом расширенный синтаксис для формирования повторяющихся строк не поддерживается.

1.3 Поддержка многоязычности

Для обеспечения многоязычности, все таблицы могут содержать текстовые поля для каждого из языков. К названию поля при этом добавляется сокращённое название языка. Например: NameEn, TemplateRu. При отсутствии в таблице поля для данного языка, используется поле без добавления признака языка.

Пример:

Id	Name	NameEn	Start	Duration
0	1 смена	1st shift	8h	8h
1	2 смена	2nd shift	16h	8h

В данном случае поле NameEn будет использоваться для английского языка, а поле Name – для всех остальных языков.

1.4 Списки (справочники)

Списки могут храниться в форматах Txt, SQLite или получаться из ADO-источника. Отдельные списки (таблицы) могут отсутствовать, если они не нужны для функционирования программы. Так, например, если не используется автоматическое обновление данных (соответствующая кнопка скрыта), то может отсутствовать таблица RefreshRates; если в системе нет отчётов, то могут отсутствовать таблицы Reports и ReportsGroups. Порядок расположения полей произвольный, поля в квадратных скобках могут отсутствовать.

1.4.1 Общие

TimeIntervals (Id, Name, Start, Duration)

Интервалы времени, за которые отображаются данные.

- **Id** – номер интервала;
- **Name** – название интервала;
- **Start** – время начала интервала;
- **Duration** – его продолжительность.

Если начальное время указано, то оно отсчитывается от начала суток, за которые в данный момент отображаются данные. Если начальное время не указано, интервал отсчитывается от текущего момента времени назад.

Пример:

Id	Name	Start	Duration
0	1 смена	8h	8h
1	2 смена	16h	8h
2	Сутки	0h	1d
3	Последний час		1h
4	Последний год		365d

RefreshRates (Id, Name, Time)

Интервалы автоматического обновления данных.

- **Id** – номер интервала обновления;
- **Name** – текст надписи в меню;
- **Time** – период обновления.

Пример:

Id	Name	Time
0	500 миллисекунд	500z
1	1 секунда	1s
2	2 секунды	2s
...
10	30 минут	30m
11	1 час	1h

Эти списки могут быть, вместо таблиц, приведены в секциях в Trends.ini:

```
[TimeIntervals]
0="1 смена",8h,8h
1="2 смена",16h,8h
2="Сутки",0h,1d
3="Последний час",,1h
4="Последний год",,365d

[RefreshRates]
500z=500 миллисекунд
1s=1 секунда
2s=2 секунды
...
30m=30 минут
1h=1 час
```

1.4.2 Аналоговые сигналы

Для аналоговых сигналов создаётся дерево сигналов, узлы которого (группы) описываются в таблице SensorsGroups, а сами сигналы в таблице Sensors.

SensorsGroups (Id, Parent, Name, [Plant])

Группы аналоговых сигналов.

- **Id** – номер группы аналоговых сигналов (нумерация с 1);
- **Parent** – номер родительской группы в древовидной структуре (0 – расположение в корневой группе);
- **Name** – название группы;
- **Plant** – номер установки, к которой относится группа (Plants.Id).

Пример:

Id	Parent	Name	Plant
1	0	Вода	0
2	0	Пар	0
3	1	Техническая вода	0

Sensors (Id, Name, Code, Units, Group, [Plant], [Scale], Min, Max, [NAU, NPU, VPU, VAU], [Source], [Field], [Reference], [UTCTime])

Список аналоговых сигналов.

- **Id** – номер аналогового сигнала (должен быть уникальным для соответствующего источника данных);
- **Name** – наименование;
- **Code** – обозначение;
- **Units** – единицы измерения;
- **Group** – номер группы (SensorsGroups.Id);
- **Plant** – номер установки (Plants.Id);
- **Scale** – масштабный коэффициент для сигналов, также может содержать аддитивную добавку (например: '0.01' ($X \cdot 0.01$), '1-10' ($X \cdot 1 - 10$), '50+100' ($X \cdot 50 + 100$));
- **Min, Max** – пределы измерения сигнала;
- **NAU, NPU, VPU, VAU** – уставки: нижняя аварийная, нижняя предупредительная, верхняя предупредительная, верхняя аварийная, соответственно (используются для отображения на графиках, могут отсутствовать);
- **Source** – источник данных (для сигналов, хранящихся отдельно от остальных аналоговых сигналов, или для нескольких сигналов, хранящихся в одном файле) в формате: [Src@]TableName[id], где Src – имя источника, TableName – имя таблицы, id – идентификатор сигнала в таблице, если в ней используется не общая, а собственная нумерация сигналов (например, Txt@ReferenceData или PLC1Sensors\10);
- **Field** – имя поля, в котором находятся данные. Если указан битовый адрес (поле.бит, например: Data2.08), то данный сигнал воспринимается как дискретный, с соответствующим отображением;
- **Reference** – признак эталонной кривой, если 1, то серия данных читается целиком (без фильтрации по времени) и допускается её сдвиг по времени для наложения на другие серии данных;
- **UTCTime** – признак хранения данных в UTC для отдельного сигнала (0 или 1), по умолчанию используется значение, указанное в Trends.ini в параметре AnalogUTCTime.

Пример:

Id	Name	Code	Units	Group	Plant	Scale	Min	Max	NAU	NPU	VPU	VAU
1	Давление воды	P1	МПа	1	0	0.001	0	12	7	7.5	10	10.5
2	Температура пара	T2	°С	2	0	0.1	0	600	495	520	545	575

1.4.3 История событий

Для отображения истории событий требуются справочники для расшифровки кодов, содержащихся в таблицах событий.

Plants (Id, Name)

Список установок (агрегатов), данные по которым содержатся в базе.

- **Id** – номер установки;
- **Name** – название установки (агрегата), отображается в меню.

Пример:

Id	Name
0	Общестанционное
1	Турбина 1
2	Турбина 2

PlcList (Id, Name, HistoryTable, [Color], [Plant], [UTCTime])

Список контроллеров, по которым регистрируется история событий.

- **Id** – номер контроллера;
- **Name** – название контроллера;
- **HistoryTable** – имя таблицы истории событий для данного контроллера (в одной таблице может храниться история событий нескольких контроллеров, в этом случае таблица должна содержать поле PLC с номером контроллера);
- **Color** – цвет, которым отображается соответствующая ячейка в таблице истории событий (здесь и далее в этом разделе), допускается текстовое или числовое указание цвета (см. описание [Trend]/Colors);
- **Plant** – номер установки, к которой относится контроллер (Plants.Id);
- **UTCTime** – признак хранения данных истории событий в UTC (0 или 1), по умолчанию используется значение, указанное в Trends.ini в параметре HistoryUTCTime.

Пример:

Id	Name	HistoryTable	Color
1	ТГ1	MsgTG1	clGreen
2	ТГ2	MsgTG2	clRed

DeviceGroups (Id, Name, [Color])

Перечень групп устройств, по которым регистрируется история событий.

- **Id** – номер группы устройств;
- **Name** – название группы устройств.

Пример:

Id	Name	Color
2	Регулятор	clWhite
3	Датчик	clWhite

6	Защита	clRed
7	Предупреждение	clYellow

Devices (Id, Group, PLC, Name, [Color], [Plant])

Перечень устройств, по которым накапливается история событий.

- **Id** – номер устройства в контроллере;
- **Group** – группа устройства (DeviceGroups.Id);
- **PLC** – контроллер (PlcList.Id), к которому относится устройство;
- **Name** – название устройства.

Пример:

Id	Group	PLC	Name
4	2	1	[ЗРВ3] Регулятор уровня воды
11	3	1	[L184] Уровень воды
7	6	1	Неисправность в СУ

MsgCodes (Id, Group, Name, [Color])

Список сообщений для каждой группы устройств.

- **Id** – код сообщения;
- **Group** – группа устройства, к которому относится сообщение (DeviceGroups.id);
- **Name** – текст сообщения. В тексте сообщения в фигурных скобках можно указать название поля таблицы истории событий или произвольное выражение: числовое, текстовое или комбинированное (синтаксис и список функций соответствует библиотеке CalcExpress.pas). В этом случае в сообщение будет вставлено значение поля или результат выражения. В качестве параметров в выражении можно использовать поля таблицы истории событий.

Пример:

Id	Group	Name
2	3	Неисправность блока
132	3	Подмена значения на {Data1}
136	3	Изменение верхней уставки с {Data1} на {Data2}
1	2	Неисправность: {Join('Errors','Message','id',Data1,'Group',Group)}
3	1	Подмена значения на {Data1*Join('Sensors','Scale','id',Device)} {Join('Sensors','Units','id',Device)}
20	1	Изменение уставки с {FormatFloat('0.00',Data1)} на {FormatFloat('0.00',Data2)}

PlcState (Id, Name, [Color])

Список состояний контроллера (используется при наличии поля PlcState в таблицах историй событий).

- **Id** – идентификатор состояния, значение слова состояния контроллера;
- **Name** – текстовое описание состояния.

Пример:

Id	Name	Color
0	РН (резерв, неисправ.)	\$0000FF
1	АН (активен, неисправ.)	\$FF00FF
2	РИ (резерв, исправен)	\$00FFFF
3	АИ (активен, исправен)	\$00FF00

1.4.4 Отчёты

ReportsGroups (Id, Parent, Name, [Plant])

Группы отчётов.

- **Id** – номер группы;
- **Parent** – номер родительской группы в древовидной структуре (0 – расположение в корневой группе);
- **Name** – название группы;
- **Plant** – номер установки, к которой относится группа (Plants.id).

Пример:

Id	Parent	Name
1	0	Суточные отчеты
2	0	Месячные отчеты

Reports (Id, Name, Group, Source, Template, [Plant], [Params], [UTCTime], [TimeStep])

Список отчётов (см. раздел "Шаблоны отчётов").

- **Id** – номер отчёта (должен быть уникальным);
- **Name** – название отчёта;
- **Group** – номер группы (ReportsGroups.Id);
- **Source** – список источников данных для отчёта (таблиц для Raw или хранимых процедур для ADO) в формате: "имя1=источник1,имя2=источник2,...". В простейшем случае может содержать один источник без указания его имени. По умолчанию используется тип источника данных, указанный в Trends.ini в параметре Report, но можно использовать другие типы источников данных, отделив их символом "@". При использовании прямого доступа к данным данное поле может быть оставлено пустым;
- **Template** – имя файла шаблона отчёта в формате FastReport 3 или RTF, в котором может быть использована строка %host%, заменяемая на имя текущего сервера. Возможна загрузка шаблона с сервера средствами RawServer, в этом случае шаблон указывается в формате: "папка:шаблон", где папка – опубликованная в RawServer папка, шаблон – имя файла шаблона с расширением;
- **Plant** – номер установки, к которой относится отчёт (Plants.Id);
- **Params** – строка со списком параметров отчёта, разделённых запятыми;
- **UTCTime** – признак хранения данных для отчёта в UTC (0 или 1), по умолчанию используется значение, указанное в Trends.ini в параметре ReportUTCTime;
- **TimeStep** – время, на которое будет сдвигаться выбранный интервал времени при его изменении кнопками "Вперёд", "Назад" (например, '1d' - для суточной ведомости).

Пример:

Id	Name	Group	Source	Template	Params
1	Суточная ведомость	0	DayReportProc	\\DBServer\Reports\DayReport.fr3	Unit=1,Block=3
2	Основные параметры	3	Raw@MainParams	\\DBServer\Reports\MainParams.fr3	
3	Термоконтроль	5	Thermo	Reports:Thermo.fr3	

1.4.5 Аварийные срезы

Механизм аварийных срезов предполагает формирование списка аварийных событий (Alarms) и наличие таблицы AlarmCodes для их расшифровки и привязки к сопутствующим аналоговым сигналам.

Alarms (Start, Duration, Plant, Code[, Comment])

Список аварийных событий, который предназначен для упрощения выбора данных за период возникновения и развития аварийной ситуации. Может формироваться автоматически или вручную.

- **Start** – начальное время (как правило, за несколько минут до регистрации аварии), формат времени фиксированный: "yyyy-mm-dd hh:mm:ss.zzz";
- **Duration** – продолжительность события (например: 20s, 3m и т.д.);
- **Plant** – номер агрегата (Plants.Id);
- **Code** – код аварии (AlarmCodes.Id);
- **Comment** – опциональный текстовый комментарий.

Пример:

Start	Duration	Plant	Code	Comment
2019-11-24 22:37:25.123	1h	2	24	Отказ насоса

AlarmCodes (Id, Plant, Name[, Channels])

Список возможных аварийных событий для каждого из агрегатов, с указанием сопутствующих аналоговых сигналов, которые будут автоматически отображены при просмотре аварийного среза.

- **Id** – идентификатор аварийного события;
- **Plant** – номер агрегата (Plants.Id);
- **Name** – описание аварийного события;
- **Channels** – опциональный перечень аналоговых сигналов (Sensors.id), которые будут отображаться при просмотре аварийного среза. Коды сигналов перечисляются через запятую. Может быть указан диапазон кодов в формате код1-код2. Группы кодов, разделённые «/» вместо запятой, отображаются в отдельных окнах.

Пример:

Id	Plant	Name	Channels
24	2	Отказ насоса	210,215,220-224/250-257

1.5 Шаблоны отчётов

Для создания и редактирования шаблонов отчётов используется программа ReportEdit3 из комплекта ПО. Шаблон отчёта может быть отредактирован из окна просмотра отчёта. При этом редактор шаблонов не обязательно закрывать после внесения изменений, достаточно сохранить шаблон и обновить сам отчёт.

Шаблон отчёта содержит закладки **Code**, **Data** и одну или несколько закладок **Page**. Code содержит скрипты, обеспечивающие формирование отчёта, Data может содержать дополнительные данные, используемые в отчёте, Page содержат страницы макета отчёта.

На закладке Code располагается описание внутренних переменных, которые могут быть использованы как в скрипте, так и на страницах, секция инициализации между begin и end., которая выполняется в начале формирования отчёта, обработчики событий элементов страниц (для подготовки данных перед формированием элемента используется обработчик OnBeforePrint).

В скрипте доступны следующие переменные:

- **@BeginTime**, **@EndTime** – начальное и конечное время выбранного диапазона;
- **@Now** – текущее локальное время;
- **@ReportName** – имя отчёта из таблицы Reports;
- **@Param0..@ParamN** - параметр из поля Params, которое представляет собой список дополнительных параметров отчёта; для корректной работы в случае параметров с разделителями в текстовой таблице, необходимо писать строку Params в виде ""Test,1"", ""Test 2"";
- **@Поля записи** из таблицы Reports (@Id, @Name и т.д.).

В скрипте внешние переменные указываются в угловых скобках (например, <@Now>).

Страницы макета состоят из полос (band) и расположенных на них текстовых полях (memo). Основной полосой, непосредственно содержащей данные, является **MasterData**. Основной элемент отображения – **MemoView**. Параметр MasterData.RowCount определяет число строк в таблице и может задаваться в шаблоне, в скрипте, или определяться автоматически при стандартном способе доступа к данным.

1.5.1 Стандартный способ доступа к данным

В свойстве **Description** полосы MasterData указываются дополнительные свойства таблицы, в частности список источников данных (если в отчёте используется несколько источников). Список источников данных задаётся в виде: "Tables=имя1+имя2+...", где перечислены имена источников данных (из Reports.Source). Двойные кавычки в начале и конце строки в данном случае обязательны. Если данные из разных источников требуется объединять не по времени, а по номерам строк, то список указывается в виде: "Tables=имя1,имя2,...".

В полосе MasterData для стандартного доступа к данным необходимо установить число строк RowCount равным 1 (single row).

Данные в полосе MasterData отображаются в объектах Text object (Memo), в которых указывается имя поля данных или выражение. При отображении отдельного поля данных, его имя указывается в квадратных скобках – [S10]. При использовании нескольких источников данных, к имени поля добавляется имя источника данных – [имя1.S10]. При отображении выражения, все поля указываются в кавычках и угловых скобках, всё выражение заключается в квадратные скобки – [<"S8">+<"S10">].

Для дополнительной обработки данных возможно использование скриптов (закладка Events в Object Inspector). В частности, для обработки данных перед их отображением можно использовать скрипт OnBeforePrint, в котором поля данных указываются так же, как и в случае выражений – <"S10">.

Данный способ рекомендуется использовать только для обеспечения совместимости с имеющимися отчётами.

1.5.2 Прямой доступ к данным

Прямой доступ к данным является предпочтительной альтернативой стандартному и предполагает запрос данных из скрипта шаблона отчёта. При этом есть возможность

управлять диапазоном времени запрашиваемых данных, запрашивать данные многократно из разных источников, а также обрабатывать данные произвольной структуры.

Указание источников данных (Reports.Source) при этом не требуется. В шаблоне в полосе MasterData число строк необходимо при создании установить равным нулю, а в скрипте в процессе его выполнения задать необходимое число строк.

Программный код запроса данных и подготовки отчёта может располагаться в любых обработчиках событий элементов отчёта, но наиболее логично разместить его в конце шаблона отчёта, между ключевыми словами begin и end – этот блок выполняется непосредственно в начале формирования отчёта.

Обработанные данные размещаются во внутренних переменных шаблона и могут быть отображены в элементах отчёта по имени. В элементах отчёта можно ссылаться на массивы, описанные в шаблоне, для нумерации строк при этом служит переменная Line (строки нумеруются с 1). Например: Val[<Line>-1,7].

Для запроса данных служат следующие функции:

```
function SelectData(Table, Filter: String = ''; BeginTime: TDateTime = 0; EndTime: TDateTime = 0): Integer;
```

Запрос данных из таблицы Table за указанный отрезок времени. Также может быть запрошен справочник, при этом указывается только параметр Table. Одновременно может быть открыта только одна выборка. По завершению чтения данных, она должна быть закрыта.

- **Table** – имя таблицы (без указания расширения файла), может содержать имя источника данных в формате Src@Table, например: Txt@Sensors;
- **Filter** – Id запрашиваемого параметра в текстовом формате (если указана пустая строка или в таблице нет поля Id, возвращаются все параметры);
- **BeginTime, EndTime** – начало и конец отрезка времени.

```
procedure CloseData;
```

Завершение работы с выборкой.

```
function NextRec: Boolean;
```

Проверка на конец выборки и переход к следующей строке, возвращает false если достигнут конец данных.

```
function RecNo: Integer;
```

Возвращает номер текущей строки в выборке (нумеруются с нуля).

```
function Field(Name: String): Variant;
```

Возвращает значение указанного поля из текущей строки.

Пример:

```
// чтение архивных данных
SelectData('PLC1', '32', int(<@BeginTime>), int(<@BeginTime>)+1);
while NextRec do begin
    T[RecNo]:=Field('Time'); V[RecNo]:=Field('Value');
end;
CloseData;
// чтение справочника
SelectData('Txt@Sensors');
while NextRec do begin
    Id[RecNo]:=Field('Id'); Name[RecNo]:=Field('Name');
end;
CloseData;
```

1.5.3 Форматирование и редактирование

Время (поле Time) представляет собой вещественное число, целая часть которого является датой, а дробная временем. Таким образом, выражение вида [`<"Time">-1/24`] уменьшает время на 1 час.

Формат отображения данных определяется параметром DisplayFormat, который можно отредактировать в окне, вызываемом в контекстном меню, или в Object Inspector.

Отдельные объекты типа Мемо можно сделать редактируемыми при построении отчёта, для этого нужно объекту указать тип курсора HandPoint (Cursor=crHandPoint). После построения отчёта, на указанных объектах курсор будет менять вид, и при двойном щелчке по объекту можно будет отредактировать отображаемый в нём текст. Внесённые изменения можно использовать для сохранения или печати отчёта, при повторном построении отчёта они не сохраняются.

1.5.4 Графики в отчётах

Существует возможность добавить в отчёт графики. Для этого необходимо добавить в шаблон отчёта объект Picture object (PictureView) для размещения графика и сформировать график вызовом процедуры MakeTrend, которая имеет следующий синтаксис:

```
procedure MakeTrend(Name, Params, BeginDate, EndDate);
```

Name – имя объекта Picture object, который будет отображать график.

Params – список параметров в формате: "key1=value1","key2=value2",...

BeginDate, EndDate – отрезок времени, за который строится график.

Параметры отображения графика указываются в виде списка в формате ключ=значение. Ключ нечувствителен к регистру. Параметры могут повторяться, что имеет смысл для параметров добавления серий данных и линий. Можно использовать следующие параметры:

```
sensor = Id[,Color]
```

Добавляет серию данных с указанным Id и цветом, если указан. Цвет может быть задан десятичным, шестнадцатеричным числом или именем цвета (например: clRed). Id может быть просто идентификатором (Sensors.Id), может быть ключом для поиска серии в таблице Sensors (например: Code=P37), может быть диапазоном Id в формате Id1-Id2.

```
sensors = id1,id2,...
```

Добавляет список серий данных (например: Sensors = 1025,1032,1040-1043,Code=P37)

```
hline = pos[,color]
vline = pos[,color]
```

Добавить вертикальную или горизонтальную линию в указанной позиции в координатах графика. Может быть указан цвет (например: HLine = 70,clBlue)

```
xfmt = format[,horiz_text]
yfmt = format
```

Определяет формат меток на горизонтальной и вертикальной осях. Для горизонтальной оси может быть указано отображать ли метки горизонтально (horiz_text=1) или вертикально (например: XFmt = "dd.mm.yy hh:mm",0)

```
scale = scale
```

Масштаб отображения графика (по умолчанию 1).

```
image = emf|bmp
```

Определяет тип изображения: emf – масштабируемое (по умолчанию), bmp – битмап.

1.6 Параметры командной строки

Параметры командной строки позволяют управлять программой из другого приложения, в частности из SCADA-системы. Набор команд позволяет открыть графики аналоговых сигналов, историю событий, отчёт, управлять интервалом времени, а также позволяет выбрать для работы одну из нескольких баз данных.

Команды могут произвольно комбинироваться, и выполняются в том порядке, в каком они перечислены в командной строке. Каждая команда состоит из одной буквы (в отдельных случаях, с предшествующим символом "/") и параметров (для некоторых команд). Если команда подразумевает параметры, они должны следовать за ней без пробела. Команды, параметры которых содержат пробелы, запятые и другие разделители, должны заключаться в двойные кавычки (").

```
T<[Fld1=]n1,[Fld2=]n2,...>[|заголовок]
```

Открыть график аналоговых сигналов, параметры которых перечислены через запятую (без пробелов). Параметры представляют собой поля из таблицы Sensors. По умолчанию, если имя поля не указано, а указан только сам параметр, используется поле id. В команде может быть указан заголовок окна, отделённый от списка сигналов вертикальной чертой. Если заголовок не указан, он будет сформирован из списка названий сигналов.

Пример:

```
Trends.exe "T20,21,22,23,24"
```

```
Trends.exe "TCode=T1,Code=T2|Параметры T1 и T2"
```

```
R<id>
```

Открыть отчёт с соответствующим номером (Reports.id).

```
H
```

Открыть историю событий.

```
F<файл>
```


Выбор и применение фильтра истории событий. Имя файла указывается с расширением. Путь к файлу указывается в том случае, если он необходим для доступа к файлу. Ключ F применяется совместно с ключом H и должен предшествовать ему.

Пример:

Trends.exe "FD:\Trends\Filters\Мой фильтр.ftt" H

Q или /Q

Завершить программу.

M

Свернуть окно программы.

X

Максимизировать окно программы.

В<имя>

Выбрать для работы базу данных с указанным именем.

O<Id>

Выбрать для работы установку с указанным номером (Plants.Id).

I<Id>

Выбор интервала времени в соответствии с полем Id из таблицы TimeInterval.

D<начало | конец> или D<начало+длительность>

Задание интервала времени, за который отображаются данные. В первом случае указывается время начала и конца интервала. Во втором случае указывается время начала и длительность интервала (см. синтаксис в описании секции [Startup]). Время должно содержать дату, само время опционально.

Пример:

Trends.exe "D20.10.14 08:00|21.10.14 16:30"

Trends.exe "D20.10.14 08:00+12h"

Trends.exe "D20.10.14+8h"

A<period>

Включить автообновление с заданным в миллисекундах периодом.

/P

Напечатать текущий открытый документ (график, историю событий или отчёт).

/S<файл>

Сохранить текущий открытый отчёт или тренд в указанный файл. Формат определяется по расширению имени файла. Для трендов может быть указано расширение ".1s", в этом случае будут записаны секундные срезы данных в формате XML с расширением ".xml". После сохранения текущий документ закрывается.

WC или WT

Расположить окна каскадом или мозаикой, соответственно.

2 ЭКСПЛУАТАЦИЯ

2.1 Главное меню

Главное меню программы условно разбито на группы кнопок. Некоторые кнопки могут быть скрыты или недоступны – это зависит от настроек программы и текущего уровня доступа.

2.1.1 Просмотр данных



Просмотр графиков. Для выбора нужного сигнала нужно нажать на кнопку и выбрать его в выпадающем иерархическом списке



Построение отчета. Отчет можно выбрать аналогичным с графиками образом



Просмотр истории событий

2.1.2 Управление временным интервалом



Выбор временного интервала, за который будут отображаться данные. Интервал устанавливается одинаковым для всех окон программы



Сдвиг текущего временного интервала назад или вперед



Отображение данные за последнее время. Продолжительность интервала времени зависит от настроек программы



Обновление информации во всех окнах. Нажав стрелку, можно выбрать период автоматического обновления информации, выключить автоматическое обновление или включить смещение интервала времени при каждом обновлении (сдвиг времени)

2.1.3 Управление окнами



Выбор одного из открытых окон и управление окнами

2.1.4 Управление доступом



Ввод пароля, открывающего доступ к различным функциям программы. При наличии соответствующих прав доступа, в окне авторизации также возможно управление пользователями



Сброс текущих прав доступа



Выбор объекта позволяет просматривать только ту часть информации, которая относится к выбранному объекту (например, к одной технологической установке из нескольких)



Выбор базы данных позволяет подключаться к одной из нескольких баз данных. При выборе новой базы данных обновляются списки сигналов, информация в открытых ранее окнах становится недоступна для обновления или просмотра за другое время.

2.1.5 Язык интерфейса и информации



Выбор языка, на котором будет отображаться интерфейс программы и вся информация

2.1.6 Информация о программе



Информация о программе

2.2 Графики

В окне графиков может одновременно отображаться несколько сигналов, максимальное количество которых определяется настройками программы. Также могут быть использованы наборы графиков, созданные пользователем.

2.2.1 Меню



Отображение дерева аналоговых сигналов. Можно добавлять просматриваемые сигналы перетаскиванием их из дерева сигналов в рабочую область окна или двойным щелчком на выбранном сигнале



Сохранение текущего набора графиков. В появившемся окне надо указать название набора, выбрать группу, в которой он будет сохранён, и нажать ОК. При сетевом использовании можно выбрать, будет ли набор графиков доступен локально (только на данном компьютере) или доступен всем пользователям. Для перемещения набора в другую группу достаточно сохранить его в этой группе под тем же именем. Имя набора должно быть уникальным



Удаление набора графиков, выбранного в дереве аналоговых сигналов



Сохранение копии окна графиков в одном из следующих форматов: Windows Bitmap (.bmp), CompuServe GIF (.gif), Portable Network Graphics (.png), Enhanced Metafile (.emf). Сохраняется только рабочая область без дерева сигналов и табличных данных. Если данные отображаются в табличном виде, то доступен формат Data (.txt), выбрав который можно сохранить данные в текстовом виде



Печать графика



Отображение легенды. Легенда позволяет управлять видимостью отдельных сигналов, а также настраивать их параметры при помощи меню, вызываемого нажатием правой клавишей мыши на названии сигнала. Удерживая левую клавишу мыши на рамке легенды, её можно перенести в

произвольное место графика. Двойной щелчок мыши возвращает её в исходное положение



Отображение двух горизонтальных и двух вертикальных линий, которые позволяют уточнять значение параметра в определенной точке. Линии могут быть перемещены левой клавишей мыши. При включении, линии располагаются по краям рабочей области



Отображение всех сигналов на отдельных осях. Этот режим позволяет одновременно просматривать на одном графике разномасштабные сигналы



Отображение предупредительных и аварийных уставок жёлтыми и красными линиями соответственно



Включить отображение всех сигналов ступенчатыми линиями



Включить отображение всех сигналов в виде барграфа



Включение режима автоматического масштаба, в котором сигналы занимают всю высоту окна



Отображение панели навигации, которая позволяет управлять перемещением и масштабом графика (при нажатом состоянии кнопки "Перемещение/масштаб" – масштабом)



Выбор формата отображения даты и времени на шкале времени



Просмотр значений сигналов в табличном виде. Для просмотра надо выбрать сигнал из списка, а для выключения режима – нажать саму кнопку

В меню, вызываемом нажатием правой клавиши мыши на названии сигнала в легенде, доступны некоторые из перечисленных пунктов меню применительно к одному отдельному сигналу, а также следующие:

- | | | |
|--|------------------|---|
| | Метки времени | Отображение меток времени у каждой из точек сигнала |
| | Метки узлов | Маркировка точек сигнала |
| | Цвет... | Управление цветом отображения сигнала |
| | Толщина линии | Управление толщиной линии |
| | Табличные данные | Отображение данных в табличном виде |
| | Удалить серию | Удаление сигнала из списка отображаемых |

2.2.2 Управление графиком

Всё управление масштабом и областью отображения графиков реализовано при помощи левой клавиши мыши и колеса прокрутки.

Задача	Действие (возможно несколько вариантов)
Перемещение по вертикали	Используйте колесо прокрутки или перетащите вертикальную ось нажатой клавишей мыши
Изменение вертикального масштаба	Используйте колесо прокрутки или перетащите вертикальную ось нажатой клавишей мыши при нажатой клавише <Ctrl> или используйте колесо прокрутки, удерживая его нажатым. Масштабирование будет производиться относительно текущего положения курсора
Перемещение по горизонтали	Перетащите горизонтальную ось нажатой клавишей мыши
Изменение горизонтального масштаба	Перетащите горизонтальную ось нажатой клавишей мыши при нажатой клавише <Ctrl>. Масштабирование будет производиться относительно текущего положения курсора
Увеличение области	Удерживая клавишу мыши, выделите требуемый прямоугольник из левого верхнего угла в правый нижний угол, и отпустите её
Возврат к исходному масштабу	Удерживая клавишу мыши, выделите произвольный прямоугольник из правого нижнего угла в левый верхний угол, и отпустите её
Произвольное перемещение	Переместите график, удерживая его правой клавишей мыши
Информация об определённой точке	Подведите курсор к интересующей точке, при этом отобразится подсказка с информацией о точке (название сигнала, время, значение). При нажатой клавише <Ctrl> нажатие на точку позволяет просмотреть данные по выбранному сигналу в табличном виде (указанная точка будет выделена в таблице)







2.3 Отчёты

Для построения отчётов используется генератор отчётов FastReport 3. Программа позволяет просмотреть отчёт за выбранный период времени, сохранить его в одном из наиболее распространённых форматов, распечатать на принтере, а также отредактировать шаблон отчёта, если пользователь имеет соответствующий уровень доступа.

2.3.1 Меню









Сохранение отчёта в одном из следующих форматов: FastReport (.fp3), HTML (.htm), XML (.xml), Microsoft Word (.rtf), Microsoft Excel (.xls), Adobe

-  PDF (.pdf), Text (.txt).
-  Печать отчёта
-  Управление отображением отдельных частей отчёта (если отчёт предполагает такую возможность)
-  Управление параметрами отчёта (если отчёт предполагает такую возможность)
-  Редактирование шаблона отчёта
-  Отображение исходных данных отчёта в табличном виде

2.4 История событий

Программа позволяет просматривать отфильтрованные события за выбранный период времени. Фильтр определяет, какие группы устройств, устройства или отдельные события должны быть отображены. Фильтр представляет собой древовидную структуру с уровнями: "агрегат", "контроллер", "группы устройств", "устройства", "события". Переключение видимости каждого из узлов распространяется на все последующие уровни, то есть можно включать и исключать отдельные устройства и группы устройств или контроллеры целиком. Для применения фильтра следует нажать соответствующую кнопку в меню или делать изменения в фильтре, удерживая клавишу <Ctrl>, в этом случае все изменения будут применяться сразу.

-  Сохранение истории событий в одном из следующих форматов: FastReport (.fp3), HTML (.htm), XML (.xml), MS Word (.rtf), MS Excel (.xls), Adobe PDF (.pdf), Text (.csv)
-  Печать истории событий
-  Применение выбранного фильтра
-  Отключение фильтрации
-  Выбор формата отображения даты и времени
-  Установка интервала времени по выбранным событиям

В меню, вызываемом нажатием правой клавиши мыши на определённом событии, можно также скрыть все сообщения, относящиеся к данному агрегату, контроллеру, группе устройств, устройству или событию.

2.5 Управление интервалом времени

Интервал времени задаётся единым для всех отображаемых окон. Изменить его можно или кнопками управления интервалом времени в главном меню или открыв окно "Интервал времени". В этом окне можно выбрать один из predetermined интервалов или указать дату и время начала и конца интервала вручную.

2.6 Управление пользователями

Управление пользователя предназначено для реализации разграничения доступа пользователей к различным функциям программы. Уровень доступа, необходимый для работы с теми или иными функциями программы, в том числе и для управления пользователями, определяется настройками программы.

Для управления пользователями предназначено окно Авторизация. Оно же служит для ввода имени и пароля.

2.6.1 Меню



Ввод имени пользователя и пароля. Нажатие этой кнопки можно заменить нажатием <Enter>



Добавление пользователя. Для этого должны быть указаны имя, пароль и уровень доступа создаваемого пользователя. Если пользователь с таким именем уже существует, его данные будут заменены.



Удаление пользователя с именем, указанным в поле Пользователь