



ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ «ЭКРА»

Waves

Руководство пользователя
ЭКРА.00090-01 90 01

Версия 4.0 (от 09.04.2024)

Приложение *Waves* предназначено для анализа осциллограмм аварийных процессов, записанных цифровыми осциллографами и устройствами релейной защиты.

Поддерживаются следующие форматы осциллограмм:

Формат	Маска	Примечание
ЭКРА БЭ2502, БЭ2704	<i>*.dfr</i>	Один пуск может состоять из нескольких файлов (до 16)
ЭКРА БЭ2702(М)	<i>DR???F?.???</i>	Один пуск может состоять из нескольких файлов (до 4)
ЭКРА БЭ2701	<i>DATA??*.???</i>	Один пуск может состоять из нескольких файлов (до 4)
ЭКРА	<i>*.zfr</i>	Может содержать пуски БЭ2702(М), БЭ2704 и БЭ2502
<i>Comtrade</i>	<i>*.cfg + *.dat *.cff</i>	Поддерживаются все существующие версии стандарта.
<i>Waves</i>	<i>*.waves</i>	Собственный формат <i>Waves</i> , в котором можно сохранить любую осциллограмму после ее анализа со всеми внесенными изменениями

Waves распространяется либо как установочный пакет, либо как портативный архив.

Для работы *Waves* необходим компьютер под управлением ОС *Microsoft Windows XP* и новее, или под управлением ОС *Linux* с использованием пакета *Wine*..

Подготовка компьютера с ОС *Linux* для работы *Waves* описана в главе 17

СОДЕРЖАНИЕ

1 Термины и определения.....	10
1.1 Сигналы	10
1.2 Комплексные сигналы.....	10
1.3 Трехфазные цепи	10
1.4 Текущий отсчет.....	10
1.5 Элементы панели осциллограммы.....	11
1.6 Элементы ленты.....	11
2 Настройка программы.....	13
2.1 Настройка интерфейса пользователя.....	13
2.1.1 Страница: Вид по умолчанию.....	13
2.1.2 Страница: Масштаб единиц измерения	13
2.2 Привязка файлов.....	13
3 Открытие осциллограммы	14
3.1 Особенности форматов ЭКРА.....	14
3.1.1 Пуски.....	14
3.1.2 Частота дискретизации.....	14
3.2 Особенности формата Comtrade.....	14
3.2.1 Кодировка символов.....	14
3.2.2 Первичные и вторичные величины аналоговых сигналов	15
3.2.3 Трехфазные цепи.....	15
4 Работа с осциллограммой	16
4.1 Ярлык страницы осциллограммы.....	16
4.2 Перемещение курсора.....	16
4.3 Прокрутка по времени.....	16
4.4 Масштабирование по времени.....	17
4.5 Прокрутка списка сигналов	17
4.6 Масштабирование сигналов по амплитуде.....	17
4.7 Масштабный коэффициент ленты	18
4.8 Изменение высоты ленты.....	19
4.9 Перемещение сигналов.....	19
4.10 Инверсия сигналов.....	19
4.11 Видимость сигналов	19
4.12 Удаление сигналов.....	20

4.13 Свойства сигнала.....	20
4.13.1 Основные свойства сигналов.....	21
4.13.2 Дополнительные свойства аналоговых сигналов.....	21
4.14 Вычисление сигналов.....	21
4.14.1 Вычисление сигнала путем задания выражения.....	22
4.14.2 Вычисление сигнала с использованием меню.....	22
4.14.3 Изменение расчетного выражения.....	22
4.14.4 Зависимости.....	22
4.15 Отображение значений аналоговых сигналов.....	22
4.15.1 Первичные и вторичные величины.....	22
4.15.2 Значения аналоговых сигналов.....	23
4.15.3 Единицы измерения сигналов.....	23
4.16 Ограничение видимого диапазона.....	24
5 Инструменты на осциллограмме.....	25
5.1 Маркеры.....	25
5.1.1 Добавление.....	25
5.1.2 Перемещение.....	25
5.1.3 Поиск.....	25
5.1.4 Настройка.....	25
5.1.5 Удаление.....	25
5.1.6 Интервалы времени.....	26
5.2 Замеры.....	26
5.2.1 Добавление.....	26
5.2.2 Перемещение.....	26
5.2.3 Поиск.....	26
5.2.4 Настройка.....	26
5.2.5 Удаление.....	27
5.3 Уровни.....	27
5.3.1 Добавление.....	27
5.3.2 Перемещение.....	27
5.3.3 Поиск.....	27
5.3.4 Настройка.....	27
5.3.5 Удаление.....	27
6 Инструменты анализа.....	28
6.1 Векторная диаграмма.....	28

6.2 Гармоники.....	29
6.3 Диаграмма Потье.....	30
6.4 Определение места повреждения.....	31
6.5 Спектр.....	31
6.6 Статистика.....	32
6.7 Таблица значений.....	33
6.8 Модели ИО БЭ2704/БЭ2502.....	33
6.8.1 Состав.....	33
6.8.2 Настройка.....	34
6.8.3 Работа с комплексной плоскостью.....	34
6.9 Модели ИО ЭКРА 200.....	35
7 Печать осциллограммы.....	36
8 Сохранение осциллограммы.....	37
9 Экспорт осциллограммы.....	38
9.1 Экспорт в Comtrade.....	38
9.2 Экспорт в CSV.....	38
10 Совмещение осциллограмм.....	39
10.1 Объединение.....	39
10.1.1 Автоматическое объединение по абсолютному времени.....	39
10.1.2 Ручное объединение.....	40
10.1.3 Замечания.....	41
10.1.4 Источники.....	41
10.2 Стыковка.....	41
10.2.1 Автоматическая стыковка по абсолютному времени.....	42
10.2.2 Ручная стыковка.....	42
10.2.3 Замечания.....	43
11 Прочие команды для осциллограмм ЭКРА.....	44
11.1 Внедрение файла конфигурации устройства.....	44
11.2 Причина пуска.....	44
11.3 Состояние в момент пуска.....	44
11.4 Добавление сигналов из базы данных.....	44
11.4.1 Проблема пропуска событий.....	45
11.4.2 Проблема округления времени.....	45
11.5 Уставки.....	45

12 Шаблоны обработки осциллограмм	46
13 Выражения	47
13.1 Сигналы.....	47
13.2 Операторы.....	47
13.2.1 Логические операторы	47
13.2.2 Арифметические операторы.....	48
13.3 Функции.....	48
13.3.1 Вызов функций	48
13.3.2 Описание функций в документе	48
13.3.3 Типы параметров функций	49
13.4 Константы	49
13.4.1 Логические константы.....	49
13.4.2 Целочисленные константы.....	50
13.4.3 Вещественные константы	50
13.4.4 Константа nil	50
13.5 Преобразование типа сигнала	50
13.5.1 Binary.....	50
13.5.2 Float.....	51
13.5.3 Complex	51
13.6 Сдвиг сигнала: Shift.....	52
13.7 Сборка и разборка комплексного сигнала	52
13.7.1 ComplexA, ComplexP.....	52
13.7.2 Re, Im, Mag, Ang.....	52
13.8 Симметричные составляющие	52
13.8.1 Pos(2), Neg(2), Zero(2).....	53
13.8.2 PosF(2), NegF(2), ZeroF(2).....	53
13.9 Мощность трех фаз: Power(2).....	53
13.10 Частота сигнала: Freq.....	54
13.11 Компенсация емкостного тока линии: RmCC	54
13.12 Компенсация нагрузочного небаланса: RmLd.....	54
13.13 Сравнение аналоговых сигналов.....	55
13.13.1 Cmp, CmpE, CmpG, CmpGE, CmpL, CmpLE, CmpNE.....	55
13.13.2 Min, Max.....	55
13.14 Преобразования дискретных сигналов.....	56
13.14.1 Выдержка времени: Delay.....	56

13.15 Преобразования аналоговых сигналов	56
13.15.1 Абсолютная величина: Abs.....	56
13.15.2 Арккосинус: ACos	56
13.15.3 Арккотангенс: ACtg.....	56
13.15.4 Арксинус: ASin	57
13.15.5 Арктангенс: ATg.....	57
13.15.6 Постоянная составляющая: Const	57
13.15.7 Косинус: Cos	57
13.15.8 Котангенс: Ctg.....	57
13.15.9 Производная: Der.....	57
13.15.10 Среднее значение: Mean	57
13.15.11 Пиковый детектор: Peak.....	58
13.15.12 Среднеквадратическое значение: RMS	58
13.15.13 Синус: Sin.....	58
13.15.14 Квадратный корень: Sqrt.....	58
13.15.15 Тангенс: Tg	58
13.15.16 Нормализация угла: To180	58
13.16 Преобразования комплексных сигналов	59
13.16.1 Сопряженное значение: Conj.....	59
13.16.2 Поворот: Rot.....	59
13.17 Счетчик: Count.....	59
14 Скрипт	60
14.1 Редактор кода.....	60
14.2 Комментарии.....	61
14.3 Литералы	61
14.4 Переменные.....	61
14.5 Динамическое преобразование типа.....	62
14.6 Числовая арифметика.....	63
14.7 Конкатенация строк.....	63
14.8 Блоки	64
14.9 Циклы.....	64
14.10 Макросы	65
14.11 Команды.....	65
14.12 Команды категории Rtm.....	66
14.12.1 Запись сообщения в лог: Rtm.Log	66

14.12.2 Прерывание выполнения: Rtm.Stop	67
14.13 Команды категории Sig.....	67
14.13.1 Создание сигнала: Sig.Create	67
14.13.2 Изменение свойств сигнала: Sig.Edit	70
14.13.3 Дублирование сигнала: Sig.Dup	71
14.13.4 Скрытие сигналов: Sig.Hide(...).....	71
14.13.5 Инверсия сигналов: Sig.Invert	71
14.13.6 Удаление сигналов: Sig.Remove(...).....	71
14.13.7 Показ сигналов: Sig.Show(...).....	72
14.14 Команды категории Tri	72
14.14.1 Создание трехфазной цепи: Tri.Create	72
14.14.2 Изменение трехфазной цепи: Tri.Edit.....	73
14.14.3 Скрытие сигналов трехфазных цепей: Tri.Hide	73
14.14.4 Удаление трехфазных цепей: Tri.Remove(...).....	73
14.14.5 Показ сигналов трехфазных цепей: Tri.Show	73
14.15 Команды категории Osc	74
14.15.1 Настройка панели инструментов: Osc.Setup.....	74
14.15.2 Перемещение сигналов: Osc.Move(...)	75
15 Нюансы преобразования старых скриптов.....	76
16 Конвертер etc.....	77
17 Работа под управлением Linux.....	78
17.1 Установка Wine.....	78
17.1.1 Окружение Windows.....	78
17.1.2 Отключение декорирования окон.....	78
17.2 Установка/распаковка Waves.....	79
17.2.1 Ярлык запуска Waves	79
17.3 Отличия в поведении	79

1 ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

1.1 СИГНАЛЫ

Осциллограмма состоит из набора сигналов, записанных или рассчитанных на общем временном интервале. Поддерживаются *аналоговые* и *дискретные* сигналы.

Для всех типов сигналов поддерживается специальное значение отсчета - *Нет данных*, которое не отображается на графиках. Такое значение отсчетов могут принимать, если в процессе их вычисления происходит ошибка. Также этим значением заполняются пробелы между фрагментами осциллограмм при их совмещении.

1.2 КОМПЛЕКСНЫЕ СИГНАЛЫ

В дополнение к обычным аналоговым сигналам, *Waves* поддерживает так называемые аналоговые сигналы, представленные в комплексной форме. Такие сигналы представляют собой массив (по количеству отсчетов в осциллограмме), каждый элемент которого является комплексным числом, рассчитанным на предшествующем периоде определенной частоты. По умолчанию используется период основной частоты файла (как правило, 50 Гц), но также имеется возможность задавать частоту вручную.

В дальнейшем, если требуется что-то особо отметить для таких сигналов по отношению к обычным аналоговым сигналам, они, для простоты, будут именоваться *комплексными* сигналами.

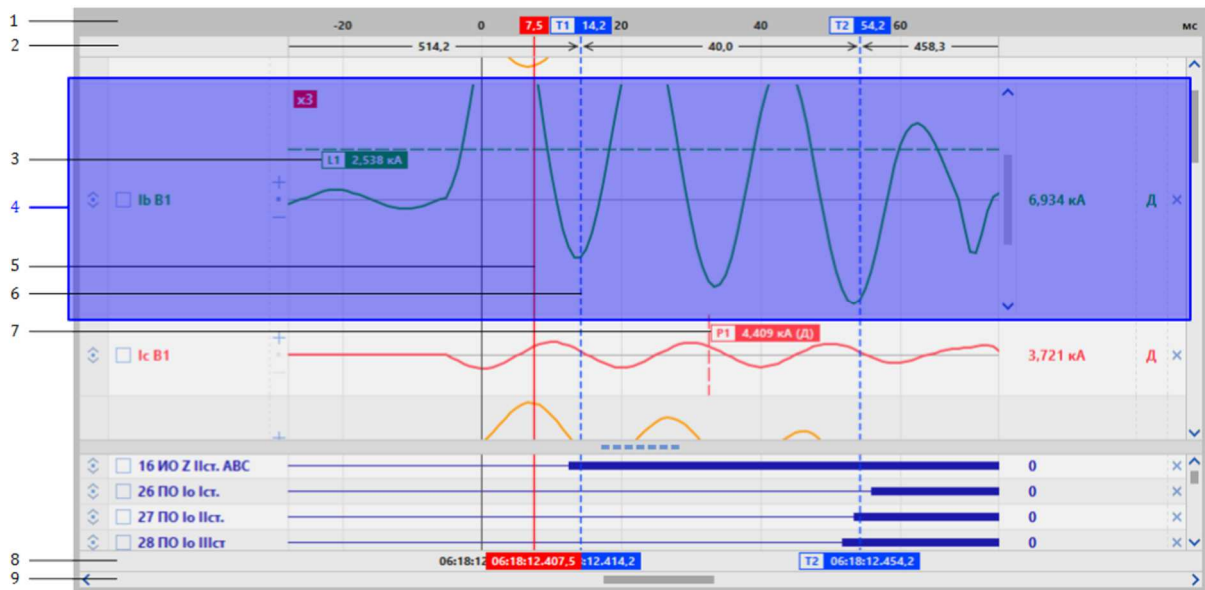
1.3 ТРЕХФАЗНЫЕ ЦЕПИ

Аналоговые и комплексные сигналы могут объединяться в трехфазные цепи, которые могут быть использованы для совместного (группового) масштабирования сигналов по амплитуде, а также в ряде расчетов для упрощения задания параметров. Таблица трехфазных цепей представлена в окне *Трехфазные цепи*, которое открывается при выполнении команды *Сигналы > Трехфазные цепи*. В ней можно посмотреть состав трехфазных цепей открытой осциллограммы, а также создать новые цепи вручную. Если имя цепи выводится на розовом фоне, значит, она определена некорректно и следует проанализировать сигналы, входящие в нее на предмет идентичности их единиц измерения и коэффициентов трансформации. Некорректно заданные цепи не отображаются в списках выбора цепей различных инструментов.

1.4 ТЕКУЩИЙ ОТСЧЕТ

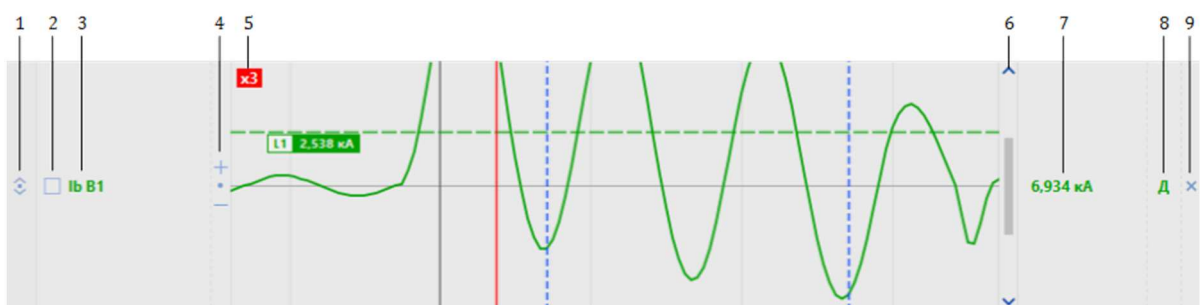
В *Waves* существует понятие *текущий отсчет*. Он задается *курсором* (вертикальная линия красного цвета), который всегда присутствует в области сигналов и который можно перемещать по оси времени. В заголовке курсора отображается метка времени текущего отсчета. Текущий отсчет используется для отображения значений сигналов в области их значений, а также в различных инструментах анализа.

1.5 ЭЛЕМЕНТЫ ПАНЕЛИ ОСЦИЛЛОГРАММЫ



№	Элемент
1	Шкала относительного времени (относительно момента пуска)
2	Шкала интервалов между соседними маркерами
3	Инструмент <i>Уровень</i>
4	Лента
5	Курсор
6	Инструмент <i>Маркер</i>
7	Инструмент <i>Замер</i>
8	Шкала абсолютного времени
9	Полоса прокрутки по времени

1.6 ЭЛЕМЕНТЫ ЛЕНТЫ



№	Элемент
1	Зона захвата для перемещения ленты на новое место
2	Флажок выделения сигнала (присутствует только в специальных режимах)
3	Имя сигнала
4	Кнопки изменения масштабного коэффициента ленты
5	Масштабный коэффициент ленты
6	Полоса прокрутки ленты (присутствует, только если задан масштабный коэффициент ленты)
7	Значение сигнала
8	Метка типа величины значения сигнала
9	Кнопка скрытия ленты (всех сигналов расположенных на ней)

2 НАСТРОЙКА ПРОГРАММЫ

2.1 НАСТРОЙКА ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

Для доступа к настройкам необходимо выполнить команду *Настройки*, главного меню программы

2.1.1 СТРАНИЦА: ВИД ПО УМОЛЧАНИЮ

Страница позволяет настроить состояния элементов панели инструментов осциллограммы, в которых они будут находиться при открытии новой осциллограммы.

Кроме файлов в формате *Waves*, которые все текущее состояние интерфейса сохраняют в себе

2.1.2 СТРАНИЦА: МАСШТАБ ЕДИНИЦ ИЗМЕРЕНИЯ

Страница содержит таблицу, в которой для каждой поддерживаемой физической величины, единицы измерения которой могут масштабироваться (например ток можно отображать в А, кА, мА и т.д.), можно задать масштаб единицы измерения для отображения в режиме *ручного* масштабирования единиц измерения. Этот масштаб будет использоваться по умолчанию для вычисляемых сигналов, а в некоторых случаях и для загружаемых из файлов.

2.2 ПРИВЯЗКА ФАЙЛОВ

Для доступа к окну привязки расширений файлов осциллограмм к *Waves*, необходимо выполнить команду *Привязка файлов* главного меню программы.

Привязка расширений позволяет открывать осциллограммы соответствующих форматов двойным щелчком по имени файла в *Проводнике Windows*.

Установочная версия сама регистрирует все привязки в процессе установки, но для портативной версии возможность привязки прямо в программе единственно возможная.

Также функция будет полезна, если по каким-то причинам информация о привязках в *Windows* была испорчена или перепривязана другим программным обеспечением к себе, а вам этого не нужно. Пользуясь данной функцией, вы всегда можете легко восстановить все утерянные привязки.

3 ОТКРЫТИЕ ОСЦИЛЛОГРАММЫ

Waves позволяет открывать одновременно несколько осциллограмм. Каждая осциллограмма открывается на собственной вкладке. Для того, чтобы открыть осциллограмму, необходимо выполнить команду *Открыть* из главного меню, которая открывает окно *Обзор файлов и папок*. В этом окне для каждой осциллограммы, помимо имени файла, указывается время пуска осциллографа, информация об устройстве, записавшем осциллограмму и о его месте установки.

Для получения этой информации производится быстрый анализ каждого файла в папке. Если анализ выявил некорректность файла, его имя отображается красным без дополнительной информации.

Если папка является сетевой или содержит много осциллограмм, может возникать небольшая задержка при выводе списка файлов на экран.

3.1 ОСОБЕННОСТИ ФОРМАТОВ ЭКРА

3.1.1 Пуски

Осциллограммы в форматах ЭКРА могут состоять из нескольких файлов (*фрагментов*), совокупность которых называется *пуском*. В окне *Обзор файлов и папок* файлы, составляющие пуск, визуально объединены в группу.

Объединение в пуски работает только, если файлы имеют оригинальный формат имени. В связи с чем, мы крайне не рекомендуем переименовывать файлы в форматах ЭКРА.

3.1.2 ЧАСТОТА ДИСКРЕТИЗАЦИИ

Устройства ЭКРА оцифровывают входные сигналы с частотой *минимум* 2400 Гц и все расчетные алгоритмы в них работают именно с такими данными. При записи осциллограмм происходит прореживание данных до частоты 1200 Гц. В результате такого огрубления, анализ поведения устройств ЭКРА по осциллограммам не всегда точен. Особенно сильно расхождения проявляются в переходных режимах на алгоритмах, в которых участвуют производные.

3.2 ОСОБЕННОСТИ ФОРМАТА COMTRADE

3.2.1 КОДИРОВКА СИМВОЛОВ

Текстовые конфигурационные файлы (*.cfg) или соответствующие им секции составных файлов (*.cff) формата *Comtrade*, сформированные российскими устройствами записи осциллограмм, как правило, используют однобайтовые символы ANSI. При этом, символы кириллицы, могут быть записаны в кодировках *DOS* (кодировка 866) или *Windows* (кодировка 1251). Кроме того стандарт 2013 года вводит кодировку интернациональную UTF8.

Waves автоматически определяет кодировку файла. Алгоритм определения основан на вероятностных методах и, в редких случаях, может быть неточен. Если

кодировка была определена неверно, то после открытия файла следует вручную изменить ее с помощью команды *Файл > Изменить кодировку*. Это действие не приводит к повторному открытию осциллограммы и не влияет на имена сигналов, добавленных уже после открытия файла.

3.2.2 ПЕРВИЧНЫЕ И ВТОРИЧНЫЕ ВЕЛИЧИНЫ АНАЛОГОВЫХ СИГНАЛОВ

В некоторых файлах *Comtrade* может отсутствовать информация о том, в каких величинах (первичных или вторичных) записаны отсчеты аналоговых сигналов. На этапе загрузки такого файла *Waves* выдаст окно с запросом указать эту информацию вручную.

3.2.3 ТРЕХФАЗНЫЕ ЦЕПИ

В стандарте *Comtrade* отсутствует явная возможность объединения сигналов в трехфазные цепи, однако *Waves* при загрузке таких файлов, производит анализ имен сигналов и их параметров и, в некоторых случаях, может выполнять такое объединение.

4 РАБОТА С ОСЦИЛЛОГРАММОЙ

4.1 ЯРЛЫК СТРАНИЦЫ ОСЦИЛЛОГРАММЫ

Ярлык страницы отображает имя файла открытой осциллограммы. При наведении курсора мыши на ярлык показывается подсказка с полной информацией об осциллограмме. Для закрытия осциллограммы необходимо щелкнуть на крестик в правой части ярлыка.

При помощи захвата ярлыка мышью можно перемещать вкладку из одного окна *Waves* в другое, что позволяет видеть несколько осциллограмм одновременно.

4.2 ПЕРЕМЕЩЕНИЕ КУРСОРА

– *Перетаскивание мышью.* При этом за курсором тянется шлейф желтого цвета, в котором указано смещение по времени относительно предыдущего положения;

– *Щелчок левой кнопкой мыши* в области отображения графиков сигналов переводит курсор на отсчет, соответствующий месту щелчка;

– Клавиши *Left* и *Right* смещают курсор влево и вправо на один отсчет;

– Кнопка *Переместить курсор к моменту пуска* на панели инструментов переводит курсор на момент пуска;

– Кнопка *Переместить курсор к заданному моменту времени* на панели инструментов позволяет перевести курсор на момент времени, указав его значение вручную.

4.3 ПРОКРУТКА ПО ВРЕМЕНИ

– Клавиатура:

Клавиши	Действие
<i>Shift + Left</i>	Плавная прокрутка влево
<i>Shift + Right</i>	Плавная прокрутка вправо
<i>Shift + Page Up</i>	Постраничная прокрутка влево
<i>Shift + Page Down</i>	Постраничная прокрутка вправо
<i>Shift + Home</i>	Прокрутка к началу осциллограммы
<i>Shift + End</i>	Прокрутка к концу осциллограммы

– Использование полосы прокрутки по времени;

– *Shift + Вращением колеса мыши;*

4.4 МАСШТАБИРОВАНИЕ ПО ВРЕМЕНИ

– Клавиатура:

Клавиши	Действие
<i>Ctrl + Left</i>	Увеличение масштаба
<i>Ctrl + Right</i>	Уменьшение масштаба

– Кнопки *Растянуть по времени* и *Сжать по времени* панели инструментов;

– *Ctrl + Shift + Вращением колеса мыши*;

4.5 ПРОКРУТКА СПИСКА СИГНАЛОВ

– Клавиатура:

Клавиши	Действие
<i>Up</i>	Плавная прокрутка вверх
<i>Down</i>	Плавная прокрутка вниз
<i>Page Up</i>	Постраничная прокрутка вверх
<i>Page Down</i>	Постраничная прокрутка вниз
<i>Home</i>	Прокрутка к началу списка
<i>End</i>	Прокрутка к концу списка

– Использование полосы вертикальной прокрутки списка;

– *Вращением колеса мыши*;

4.6 МАСШТАБИРОВАНИЕ СИГНАЛОВ ПО АМПЛИТУДЕ

По умолчанию, если открытая осциллограмма не подразумевает объединения сигналов в трехфазные цепи, все сигналы масштабируются независимо.

Трехфазные цепи, позволяют объединять до трех сигналов (фазных или линейных), что, в частности, дает возможность организовать совместное масштабирование объединенных в цепь сигналов даже если используется индивидуальное масштабирование. Тем не менее, основное назначение трехфазных цепей это упрощение выбора сигналов в различных инструментах и расчетах, а их масштабирующая функция – лишь удобный побочный эффект.

Во многих случаях необходимо совместно масштабировать более трех сигналов. Для решения данной задачи *Waves* содержит инструмент *Группы масштабирования*. Каждая такая группа может включать в себя любые сигналы и трехфазные цепи. Все, что объединено в группу масштабирования, масштабируется по совместно и независимо от всего, что в нее не входит.

Редактирование групп масштабирования доступно при выполнении команды *Сигналы > Группы масштабирования*

Таким образом, график аналогового сигнала внутри ленты может быть отображен в одном из четырех режимов масштабирования по амплитуде.

Режимы переключаются при помощи выпадающего списка *Масштаб* на панели инструментов.

Режим	Описание
Индивидуально	Каждый график масштабируется индивидуально, независимо от остальных графиков, и занимает всю высоту ленты
По единицам	Графики сигналов с одной и той же единицей измерения масштабируются совместно (независимо от ленты, в которой они сейчас находятся)
Групповой (по единицам)	Графики сигналов, входящих в одну трехфазную цепь или/и в одну группу масштабирования, масштабируются совместно, а все остальные, не входящие ни в одну трехфазную цепь или/и группу масштабирования – по единицам измерения
Групповой (индивидуально)	Графики сигналов, входящих в одну трехфазную цепь или/и в одну группу масштабирования, масштабируются совместно, а все остальные, не входящие ни в одну трехфазную цепь или/и группу масштабирования – индивидуально

В совместном масштабировании скрытые сигналы не участвуют. Поэтому при скрытии или показе сигналов в режимах совместного масштабирования, размах по амплитуде других видимых сигналов может меняться.

4.7 МАСШТАБНЫЙ КОЭФФИЦИЕНТ ЛЕНТЫ

Если некоторый аналоговый сигнал имеет большой динамический диапазон, области с малыми значениями будут теряться на фоне областей с большими значениями. Для того чтобы рассмотреть области малых значений имеется возможность изменить масштабный коэффициент ленты в диапазоне 1 ... 100.

Если на ленте присутствует хотя бы один аналоговый сигнал, становится доступной область изменения масштабного коэффициента этой ленты. По умолчанию масштабный коэффициент равен 1. При этом графики данной ленты не выходят за ее границы по высоте и видны целиком.

Кнопка **+** увеличивает масштабный коэффициент на 1. Кнопка **-** уменьшает на 1. Кнопка **•** устанавливает коэффициент в значение 1. Быстрое изменение масштабного коэффициента с шагом 1 возможно при помощи вращения колеса мыши, когда курсор мыши находится в области изменения масштабного коэффициента.

Также можно задать коэффициент вручную, выполнив команду *Изменить масштабный коэффициент* из контекстного меню ленты. При этом доступен ввод и дробных значений коэффициента.

Если масштабный коэффициент становится больше 1, то графики сигналов выходят за пределы вертикальных границ ленты. При этом справа от графиков появляется полоса вертикальной прокрутки, которая позволяет просматривать в увеличенном масштабе любое место на графике.

4.8 ИЗМЕНЕНИЕ ВЫСОТЫ ЛЕНТЫ

Высоту ленты можно изменять, если на ней имеется хотя бы один аналоговый сигнал. Возможно как индивидуальное изменение высоты ленты, так и одновременное изменение высоты всех лент.

Для индивидуального изменения необходимо ухватиться мышью за нижнюю границу ленты и переместить ее вверх или вниз.

Для одновременного изменения высоты всех лент, можно воспользоваться следующими способами:

– Клавиатура:


Клавиши	Действие
<i>Ctrl + Down</i>	Увеличение высоты лент
<i>Ctrl + Up</i>	Уменьшение высоты лент

– Кнопки *Растянуть по амплитуде* и *Сжать по амплитуде* панели инструментов

– *Ctrl + Вращением колеса мыши*;

4.9 ПЕРЕМЕЩЕНИЕ СИГНАЛОВ

Для перемещения сигнала необходимо ухватиться за его имя мышью и перенести на новое место. В процессе перемещения красная горизонтальная линия отображает место вставки. Можно располагать сигналы, как на отдельных лентах, так и совмещать их в одной ленте.

Для перемещения ленты целиком, необходимо ухватиться за зону захвата (символ  в левой части) ленты мышью и перенести на новое место.

Кроме перемещения сигналов и лент внутри панели осциллограммы, можно перетаскивать их на некоторые инструменты анализа (векторная диаграмма, спектр, таблица значений), что эквивалентно выбору соответствующих сигналов в настройках инструмента.

Для быстрого разнесения сигналов, собранных на одной ленте по индивидуальным лентам необходимо выполнить команду *Разгруппировать ленту*, ее контекстного меню.

4.10 ИНВЕРСИЯ СИГНАЛОВ

Waves позволяет выполнить инвертирование сигналов. Перед именем инвертированного сигнала ставится символ “~” (тильда). Повторное инвертирование возвращает отсчеты сигнала в исходное состояние и убирает тильду. Инвертирование выполняется командой *Инвертировать* контекстного меню сигнала или в окне *Свойства сигнала*.

4.11 ВИДИМОСТЬ СИГНАЛОВ

Сигналы, которые в текущий момент не нужны на экране, можно скрыть.

– Команда *Скрыть ленту* контекстного меню ленты позволяет скрыть сразу все сигналы, находящиеся на ней;

- Команда *Скрыть* контекстного меню сигнала скрывает соответствующий сигнал;
- Команда *Сигналы > Видимость > Изменить* позволяет отметить флажками сигналы, которые должны быть видимы;
- Команда *Сигналы > Видимость > Скрыть неизменяющиеся* скрывает все неизменяющиеся в рамках осциллограммы сигналы;
- Команда *Сигналы > Видимость > Скрыть нулевые* скрывает все неизменяющиеся в рамках осциллограммы сигналы со значением отсчетов равным нулю;
- Команда *Сигналы > Видимость > Скрыть шум* скрывает все сигналы, помеченные как шум;
- Команда *Сигналы > Видимость > Показать скрытые* делает все скрытые сигналы видимыми.

4.12 УДАЛЕНИЕ СИГНАЛОВ

Возможно полное удаление информации о сигнале из осциллограммы. Сигнал можно удалить, только если от него не зависят другие сигналы. Если какие-то сигналы нельзя удалить, но они вам мешают, скройте их.

Для удаления сигнала необходимо выполнить команду *Удалить* из его контекстного меню.

4.13 СВОЙСТВА СИГНАЛА

Значение основных свойств сигнала можно изменять на странице *Свойства* в окне *Свойства сигнала*, которое открывается по команде *Свойства* контекстного меню сигнала или по двойному щелчку левой кнопкой мыши на имени сигнала.

Для расчетных сигналов часть свойств может вычисляться автоматически. Поэтому перед их значениями в окне появляются флажки, которые по умолчанию сброшены, что и говорит о том, что свойство заполняется автоматически. Тем не менее, вы можете и сами определять значения этих свойств, для этого необходимо установить соответствующий флажок и задать требуемое значение вручную.

4.13.1 ОСНОВНЫЕ СВОЙСТВА СИГНАЛОВ

Свойство	Описание
Имя	Имя сигнала
Тип ¹⁾	Тип сигнала. _____
	Аналоговый Дискретный Комплексный
Перв ²⁾	Первичная составляющая коэффициента трансформации, и ее единица измерения
Втор ²⁾	Вторичная составляющая коэффициента трансформации, и ее единица измерения
Инверсия	Флажок <i>Инвертировать отсчеты</i> производит инверсию отчетов сигнала
Цепь ^{1,2)}	Имя трехфазной цепи, в которую входит сигнал
Цвет	Цвет сигнала _____
	Синий Желтый Зеленый Красный Лиловый Бирюзовый Коричневый Серый Темный
Линия ²⁾	Стиль линии, которой рисуется сигнал _____
	Сплошная Пунктирная

¹⁾ только чтение

²⁾ отсутствует у дискретных сигналов

4.13.2 ДОПОЛНИТЕЛЬНЫЕ СВОЙСТВА АНАЛОГОВЫХ СИГНАЛОВ

Для аналоговых сигналов возможно отображение не только мгновенного значения, но и любых других вариантов, которые можно выбрать в списке *Отображаемое значение аналоговых сигналов*, на панели инструментов осциллограммы. Тем не менее, при помощи флажка *Дополнительные параметры > Отображать только мгновенное значение*, можно включить режим, при котором всегда отображается мгновенное значение.

Если формат осциллограммы поддерживает возможность указания режима отображения только мгновенного значения для аналоговых сигналов, то данный флажок может быть установлен для того или иного сигнала сразу после загрузки осциллограммы автоматически.

Для сигналов частоты доступна возможность смещения оси на указанный уровень. По умолчанию смещение устанавливается на значение основной частоты осциллограммы (50 или 60 Гц).

4.14 ВЫЧИСЛЕНИЕ СИГНАЛОВ

Waves предоставляет широкие возможности по созданию расчетных сигналов.

С каждым вычисленным сигналом связывается полная информация о том, как и на основе каких сигналов, он был рассчитан. Поэтому в любой момент можно изменить параметры расчета и пересчитать сигнал. Более того, если в осциллограмме существуют сигналы, вычисленные на базе изменяемого сигнала, они автоматически будут рассчитаны повторно.

Расчеты могут производиться как в первичных, так и во вторичных величинах.

Мы **НЕ** рекомендуем выбирать расчет во вторичных величинах, если у вас нет твердой уверенности, что именно этого требуют ваши вычисления. Как правило, расчет во вторичных величинах требуется при вычислении специфических формул, которые специально записаны для вторичных величин и учитывают все различия в коэффициентах трансформации сигналов, используемых в расчете. Например, при расчете БНН, уставка которого задается только для вторичных величин.

4.14.1 ВЫЧИСЛЕНИЕ СИГНАЛА ПУТЕМ ЗАДАНИЯ ВЫРАЖЕНИЯ

Все расчеты в *Waves* выполняются на основе выражений. При помощи команды *Расчеты > Произвольное выражение* можно создать сигнал непосредственно вводя требуемые выражения с клавиатуры. Это наиболее гибкий и функциональный способ расчета новых сигналов, однако, он требует знания правил составления выражений (см. г. 13), и рассчитан на опытного пользователя.

4.14.2 ВЫЧИСЛЕНИЕ СИГНАЛА С ИСПОЛЬЗОВАНИЕМ МЕНЮ

Waves предоставляет набор из наиболее часто используемых расчетов в виде команд меню. Каждая такая команда может создать один или несколько сигналов, выражения, для вычисления которых, генерируются автоматически, на основании выполненной вами настройки.

Все доступные команды собраны в дереве, которое расположено в разделе меню *Расчеты*. Команды разбиты на логические группы и снабжены всплывающими подсказками.

Три самые нижние группы с именами вида *Преобразования <тип сигнала> сигналов*, содержат команды, работающие с единственным входным сигналом соответствующего группе типа. Состав этих групп эквивалентен составу группы *Вычислить* контекстных меню соответствующих типов сигналов.

4.14.3 ИЗМЕНЕНИЕ РАСЧЕТНОГО ВЫРАЖЕНИЯ

В окне *Свойства сигнала* для вычисленного сигнала присутствует кнопка *Изменить расчетное выражение*, щелкнув на которой можно вызвать окно *Редактор выражений* для внесения изменений в выражение, по которому рассчитывается сигнал.

4.14.4 ЗАВИСИМОСТИ

В окне *Свойства сигнала* на странице *Зависимости* можно видеть список сигналов, от которых зависит данный сигнал (если он расчетный), а также список сигналов, которые используют данный сигнал для расчета себя.

4.15 ОТОБРАЖЕНИЕ ЗНАЧЕНИЙ АНАЛОГОВЫХ СИГНАЛОВ

4.15.1 ПЕРВИЧНЫЕ И ВТОРИЧНЫЕ ВЕЛИЧИНЫ

Значения аналоговых сигналов могут отображаться в первичных и вторичных величинах. Переключение режима отображения выполняется с помощью переключателя *Показывать первичные/вторичные значения (П/В)* панели инструментов.

4.15.2 ЗНАЧЕНИЯ АНАЛОГОВЫХ СИГНАЛОВ

Выпадающий список *Отображаемое значение аналоговых сигналов* на панели инструментов позволяет выбрать величину, в которой будут отображаться значения аналоговых сигналов. При этом для удобства рядом со значением будет указываться поясняющая метка выбранной величины: Д - действующее значение; С – среднее значение; nГ – действующее значение n-й гармоники.

Для аналоговых сигналов, помеченных флажком *Отображать только мгновенное значение*, всегда отображается фактическое (мгновенное) значение текущего отсчета.

Для комплексных сигналов, чей частотный базис отличен от базиса по умолчанию (первая гармоника основной частоты осциллограммы), он отображается в поясняющей метке в виде 2Г/25Гц (в данном примере базис: вторая гармоника основной частоты 25 Гц)

4.15.3 ЕДИНИЦЫ ИЗМЕРЕНИЯ СИГНАЛОВ

Аналоговые сигналы имеют первичную и вторичную единицы измерения. Отсчеты таких сигналов всегда хранятся в базовой единице измерения. Т.е. если сигнал является напряжением, то отсчеты хранятся в вольтах.

Waves имеет два режима отображения значений, которые переключаются путем выбора одного из подпунктов группы *Сигналы > Масштаб единиц измерения*.

РЕЖИМ: АВТОМАТИЧЕСКИЙ

Этот режим выставлен по умолчанию. При отображении значений сигналов *Waves* масштабирует единицу измерения автоматически, чтобы на экране не было очень больших или очень маленьких чисел. Например:

110000 В отображается как **110 кВ**
0.0123 А отображается как **12.3 мА**

РЕЖИМ: РУЧНОЙ

Масштаб единицы измерения для отображения задается вручную. Т.е. если сигнал является напряжением и хранит свои отсчеты в вольтах, можно заставить программу отображать его всегда, например, в киловольтах.

При переходе в данный режим, в поле значения сигнала на осциллограмме появляется треугольник позволяющий выбрать необходимый масштаб для отображения.

Для сигналов, загруженных из осциллограмм, масштаб единиц измерения задается в файле осциллограммы. Для вычисленных сигналов - назначается при их создании согласно таблице *Единицы измерения окна Общие настройки*.

Формат Comtrade позволяет задать масштаб единицы измерения только для тех величин (первичных или вторичных), в которых записан файл. Масштаб единицы измерения альтернативной величины назначается согласно таблице *Единицы измерения окна Общие настройки*. Например, если файл записан в первичных величинах, то масштаб вторичной единицы измерения будет взят из таблицы.

4.16 ОГРАНИЧЕНИЕ ВИДИМОГО ДИАПАЗОНА

Можно ограничить видимый диапазон осциллограммы по времени. Для этого необходимо нажать кнопку *Диапазон*, на панели инструментов и двумя щелчками мыши по полю графиков выбрать участок осциллограммы, который должен остаться видимым. Все слева и справа от него будет скрыто.

Для показа полного временного диапазона вновь, необходимо нажать на кнопку *Сбросить диапазон*.

5 ИНСТРУМЕНТЫ НА ОСЦИЛЛОГРАММЕ

5.1 МАРКЕРЫ

Маркер – это вертикальная пунктирная линия синего цвета, которая служит для отметки важного момента времени. Маркеру можно задать поясняющий текст, который будет отображаться вдоль линии в вертикальном направлении.

Каждый маркер имеет личный идентификатор (T1, T2...), который можно использовать при подготовке документов для ссылки на него.

5.1.1 ДОБАВЛЕНИЕ

– Команда *Маркеры > Создать (Shift + ~)* позволяет добавить новый маркер в текущую позицию (красного) курсора;

– *Shift + Щелчок левой кнопкой мыши* добавляет новый маркер в позицию с моментом времени, соответствующим месту щелчка.

5.1.2 ПЕРЕМЕЩЕНИЕ

Маркер можно перемещать вдоль оси времени ухватившись за него мышью. При перемещении маркера за ним тянется шлейф, показывающий смещение по времени относительно предыдущего положения.

Если произвести захват маркера с удержанием клавиши *Shift* и начать перемещение, то сам маркер остается на месте, а перемещается вновь созданный маркер и шлейф, который при этом тянется за новым маркером, показывает смещение относительно исходного маркера. Эта функция удобна, если требуется добавить новый маркер, отстоящий от уже имеющегося на определенный промежуток времени.

5.1.3 ПОИСК

Для поиска маркера необходимо открыть меню *Маркеры* и выполнить двойной щелчок на интересующем маркере, ориентируясь на его идентификатор и метку времени.

5.1.4 НАСТРОЙКА

– Команда *Маркеры > Изменить*. При этом требуемый маркер необходимо выбрать из списка, ориентируясь на его идентификатор и метку времени;

– Команда *Изменить* контекстного меню маркера

– *Двойной щелчок мышью* на требуемом маркере.

5.1.5 УДАЛЕНИЕ

– Команда *Маркеры > Удалить*. При этом требуемый маркер необходимо выбрать из списка, ориентируясь на его идентификатор и метку времени;

– Команда *Маркеры > Удалить все*. Удаляет сразу все маркеры, добавленные на осциллограмму;

- Команда *Удалить* контекстного меню маркера;

5.1.6 ИНТЕРВАЛЫ ВРЕМЕНИ

- При добавлении на осциллограмму более одного маркера становится видимой шкала, показывающая интервалы времени между соседними маркерами;
- Команда *Инструменты > Таблица интервалов* управляет видимостью на экране панели, которая в табличной форме отображает интервалы времени между всеми добавленными на осциллограмму маркерами.

5.2 ЗАМЕРЫ

Замер – это инструмент, позволяющий отметить конкретный момент времени на конкретном сигнале для отображения его значения в этот момент времени. Цвет Замера совпадает с цветом сигнала, на котором он установлен. Для аналоговых сигналов в параметрах замера можно выбрать величину, которую требуется отображать в качестве значения замера. Вариант *Авто*, который выбран по умолчанию, задает отображение величины, выбранной в списке *Отображаемое значение аналоговых сигналов* на панели инструментов.

Каждый замер имеет личный идентификатор (P1, P2...), который можно использовать при подготовке документов для ссылки на него.

Когда осциллограмма сильно сжата по времени, отображение замеров становится бессмысленным, и они автоматически перестают выводиться, чтобы не загромождать изображение.

5.2.1 ДОБАВЛЕНИЕ

- Команда *Замеры > Создать* позволяет добавить замеры на указанные флажками сигналы в позицию курсора или одного из имеющихся на осциллограмме маркеров;
- *Ctrl + щелчок левой кнопкой мыши* добавляет новый замер на ближайший к месту щелчка сигнал, в позицию с моментом времени соответствующим месту щелчка.

5.2.2 ПЕРЕМЕЩЕНИЕ

Замер можно перемещать вдоль оси времени ухватившись за него мышью.

5.2.3 ПОИСК

Для поиска замера необходимо открыть меню *Замеры* и выполнить двойной щелчок на интересующем замере, ориентируясь на его идентификатор и метку времени.

5.2.4 НАСТРОЙКА

- Команда *Замеры > Изменить*. При этом требуемый замер необходимо выбрать из появляющегося списка. В списке замер представлен своим идентификатором;
- Команда *Изменить* контекстного меню замера;
- *Двойной щелчок мышью* на требуемом замере.

5.2.5 УДАЛЕНИЕ

- Команда *Замеры > Удалить*. При этом требуемый замер необходимо выбрать из появляющегося списка. В списке замер представлен своим идентификатором;
- Команда *Замеры > Удалить все*. Удаляет сразу все замеры, добавленные на осциллограмму;
- Команда *Удалить* контекстного меню замера;

5.3 УРОВНИ

Уровень – это горизонтальная пунктирная линия, которую можно перемещать по вертикали и которая показывает амплитуду сигнала в месте своего расположения. Уровень привязан к конкретному сигналу и имеет его цвет.

Каждый уровень имеет личный идентификатор (L1, L2...), который можно использовать при подготовке документов для ссылки на него.

5.3.1 ДОБАВЛЕНИЕ

- Команда *Уровни > Создать* позволяет добавить новые уровни на отмеченные флажками сигналы;
- *Alt + щелчок левой кнопкой мыши* добавляет новый уровень на ближайший к месту щелчка сигнал.

5.3.2 ПЕРЕМЕЩЕНИЕ

Уровень можно перемещать по вертикали в рамках размаха амплитуды сигнала, тем самым изменяя значение, которое он показывает.

5.3.3 ПОИСК

Для поиска уровня необходимо открыть меню *Уровни* и выполнить двойной щелчок на интересующем уровне, ориентируясь на его идентификатор и метку времени.

5.3.4 НАСТРОЙКА

- Команда *Уровни > Изменить*. При этом требуемый уровень необходимо выбрать из появляющегося списка. В списке уровень представлен своим идентификатором;
- Команда *Изменить* контекстного меню требуемого уровня;
- *Двойной щелчок мышью* на требуемом уровне.

5.3.5 УДАЛЕНИЕ

- Команда *Уровни > Удалить*. При этом требуемый уровень необходимо выбрать из появляющегося списка. В списке уровень представлен своим идентификатором;
- Команда *Уровни > Удалить все*. Удаляет сразу все уровни, добавленные на осциллограмму;
- Команда *Удалить* контекстного меню требуемого уровня;

6 ИНСТРУМЕНТЫ АНАЛИЗА

Инструменты анализа - это панели различного назначения, которые можно выводить на экран в виде плавающих окон или стыковать слева/справа/снизу относительно поля осциллограммы в основное окно. Один и тот же инструмент анализа можно вывести на экран неограниченное число раз. При этом, каждый экземпляр инструмента можно настроить независимо от других. В частности возможен одновременный анализ различных временных участков осциллограммы.

Для добавления экземпляра инструмента анализа необходимо выбрать его название в меню *Инструменты*.

Каждый инструмент анализа содержит заголовок окна, в котором присутствует пиктограмма его собственного меню в левой части, а также стандартный крестик закрытия экземпляра инструмента в правой. Меню также можно вызвать как контекстное, щелкнув правой кнопкой мыши в любом месте инструмента.

Меню каждого инструмента анализа содержит пункт *Сохранить изображение*, который позволяет сохранить текущий вид инструмента в графическом файле.

6.1 ВЕКТОРНАЯ ДИАГРАММА

Отображение векторов на комплексной плоскости.

ОСОБЕННОСТИ

– Позволяет выбрать любой аналоговый сигнал в качестве базового вектора диаграммы.

– Позволяет отображать как текущий момент времени (определяемый положением курсора), так и фиксированный маркером. Это делает возможным, в частности анализ одних и тех же сигналов в разные моменты времени одновременно, путем вывода требуемого количества одинаково настроенных, но привязанных к разным моментам времени экземпляров инструмента.

– Позволяет выводить на диаграмму угловые зоны заданного пользователем цвета, что в ряде случаев улучшает визуальное восприятие информации.

– Поддерживает несколько режимов масштабирования длин векторов на диаграмме, а также режим отображения *Только направления*, при котором все векторы вне зависимости от их модуля, отображаются равными радиусу внешней окружности масштабной сетки диаграммы.

– Поддерживает добавление отображаемых сигналов путем перетаскивания их (а также целых лент сигналов) при помощи мыши.

– Поддерживает показ как имен сигналов на концах векторов (удобно, если сигналы имеют короткие имена), так и их порядковых номеров в таблице значений (удобно, если сигналы имеют длинные имена).

– Позволяет настроить частотный базис векторов (по умолчанию это первая гармоника основной частоты осциллограммы)

МЕНЮ

Команда	Краткое описание
Настроить	Отображение диалога настройки инструмента
Момент времени	Выбор момента времени, отображаемого на диаграмме
Показывать только направление	Включение/отключение отображения только направления векторов
Показывать имена сигналов	Включение/отключение отображения имен сигналов на диаграмме

ОКНО НАСТРОЙКИ - СИГНАЛЫ

Страница позволяет выбрать сигналы, вектора которых следует отображать, указать базовый вектор и выбрать частотный базис.

ОКНО НАСТРОЙКИ - МАСШТАБ

Страница, позволяет задать параметры масштабирования векторов.

Векторы с одинаковой единицей измерения масштабируются совместно и независимо от векторов с иными единицами измерения. Т.е. токи с токами, напряжения с напряжениями и т. д.

Поддерживается два режима, определяющие как масштабировать сигналы:

– *Вычислять [масштаб] для момента времени, отображаемого на диаграмме* – учитываются только модули векторов в момент времени, отображаемый на диаграмме. Таким образом, самый большой по модулю вектор определенной единицы измерения всегда будет иметь длину равную радиусу внешней окружности масштабной сетки. Это позволяет удобно просматривать участки осциллограмм с малыми значениями сигналов, потому, что их отображение не зависит от участков с большими значениями. Визуальное сравнение длины одного и того же вектора в разные моменты времени в данном режиме некорректно.

– *Использовать единый [масштаб] за все время осциллограммы* – учитываются значения сигналов за все время длительности осциллограммы. Режим удобен, когда необходимо анализировать одни и те же вектора в разные моменты времени, потому что позволяет задать им общий масштаб.

Также поддерживаются два варианта учета сигналов влияющих на масштабирование:

– *Учитывать только сигналы, отображаемые на диаграмме* – при расчете масштабов учитываются только сигналы, которые были выведены на диаграмму.

– *Учитывать все сигналы осциллограммы* – при расчете масштаба учитываются все сигналы осциллограммы, даже те, которых на диаграмме нет.

ОКНО НАСТРОЙКИ - ЗОНЫ

Страница позволяет управлять списком отображаемых угловых зон

6.2 ГАРМОНИКИ

Построение диаграмм гармонического состава аналоговых сигналов.

ОСОБЕННОСТИ

– Позволяет отображать как текущий момент времени (определяемый положением курсора), так и фиксированный маркером. Это делает возможным, в частности анализ одних и тех же сигналов в разные моменты времени одновременно, путем вывода требуемого количества одинаково настроенных, но привязанных к разным моментам времени экземпляров инструмента.

– Поддерживает добавление отображаемых сигналов путем перетаскивания их (а также целых лент сигналов) при помощи мыши.

– Поддерживает выбор числа отображаемых гармоник (в пределах допустимого по теореме Котельникова).

МЕНЮ

Команда	Краткое описание
Настроить	Отображение диалога настройки инструмента
Момент времени	Выбор момента времени, отображаемого на диаграмме

ОКНО НАСТРОЙКИ - СИГНАЛЫ

Страница позволяет выбрать сигналы, гармонический состав которых следует отображать.

ОКНО НАСТРОЙКИ - ДИАГРАММА

Позволяет выбрать число отображаемых гармоник и указать, значение какой из них принять за 100%.

6.3 ДИАГРАММА ПОТЬЕ

Анализ тока ротора генератора при помощи диаграммы Потье

МЕНЮ

Команда	Краткое описание
Настроить	Отображение диалога настройки инструмента

ОКНО НАСТРОЙКИ - ВХОДНЫЕ ВОЗДЕЙСТВИЯ

Страница позволяет выбрать трехфазные группы напряжения и тока генератора.

ОКНО НАСТРОЙКИ - УСТАВКИ

Страница, позволяет задать необходимые уставки и базисные величины, а также параметры характеристики КЗ.

Характеристика КЗ выводится на диаграмме жирной линией синего цвета.

ОКНО НАСТРОЙКИ - ХАРАКТЕРИСТИКА ХХ

Страница позволяет задать характеристику холостого хода. Поддерживается от 5 до 8 участков. Характеристика ХХ выводится на диаграмме жирной линией красного цвета.

6.4 ОПРЕДЕЛЕНИЕ МЕСТА ПОВРЕЖДЕНИЯ

Определение места повреждения на линии электропередачи. Поддерживает сложные линии с множеством участков (в том числе и с параллельными линиями) и отпаяк.

ОСОБЕННОСТИ

- Если вы работаете с осциллограммой ЭКРА БЭ2704 или БЭ2502, доступна возможность загрузки описания линии из уставок в теле осциллограммы;
- Поддерживает загрузку описания линии из файлов **.tlc* и **.pline*, используемых для настройки функции ОМП в устройствах ЭКРА БЭ2704 и БЭ2502;
- Поддерживает сохранение описания линии в файлы **.wavespowerline*, а также загрузку из них;
- Отображает все возможные для заданной линии места повреждения.

МЕНЮ

Команда	Краткое описание
Настроить	Отображение диалога настройки инструмента
Предавварийный режим	Выбор момента времени забора данных предаварийного режима

ОКНО НАСТРОЙКИ

Позволяет создать или изменить описание линии.

6.5 СПЕКТР

Построение диаграммы амплитудного спектра (далее спектра) аналоговых сигналов.

ОСОБЕННОСТИ

- Позволяет отображать, как спектр аналогового сигнала по всей длительности осциллограммы, так и спектр отдельных участков однородности сигнала, выделенных маркерами.
- Поддерживает добавление отображаемых сигналов путем перетаскивания их при помощи мыши.
- Отображение значения (в процентах) в месте щелчка мышью на поле спектра.

– Отображаемый диапазон частот спектра ограничивается пределами допустимыми по теореме Котельникова.

МЕНЮ

Команда	Краткое описание
Настроить	Отображение диалога настройки инструмента
Начало спектра	Выбор момента времени, начала участка однородности осциллограммы
Конец спектра	Выбор момента времени, конца участка однородности осциллограммы
Сжать по частоте	Сжатие спектра по горизонтали
Растянуть по частоте	Растяжение спектра по горизонтали
Сжать по амплитуде	Сжатие спектра по вертикали
Растянуть по амплитуде	Растяжение спектра по вертикали

ОКНО НАСТРОЙКИ

Позволяет выбрать сигналы, амплитудные спектры которых следует отображать.

6.6 СТАТИСТИКА

Просмотр статистической информации об аналоговых сигналах за выбранный промежуток времени. Отображается минимальное, максимальное, среднее и медианное значения.

ОСОБЕННОСТИ

– Поддерживает добавление отображаемых сигналов путем перетаскивания их (а также целых лент сигналов) при помощи мыши.

– Позволяет отображать данные как за всю длительность осциллограммы, так для участка, выделенного маркерами.

МЕНЮ

Команда	Краткое описание
Настроить	Отображение диалога настройки инструмента
Начало диапазона	Выбор момента времени, начала участка обработки
Конец диапазона	Выбор момента времени, конца участка обработки

ОКНО НАСТРОЙКИ

Позволяет выбрать сигналы, которые следует отображать в таблице

6.7 ТАБЛИЦА ЗНАЧЕНИЙ

Просмотр значений аналоговых сигналов в моменты времени, отмеченные маркерами в виде таблицы.

ОСОБЕННОСТИ

Поддерживает добавление отображаемых сигналов путем перетаскивания их (а также целых лент сигналов) при помощи мыши.

МЕНЮ

Команда	Краткое описание
Настроить	Отображение диалога настройки инструмента

ОКНО НАСТРОЙКИ

Позволяет выбрать сигналы, которые следует отображать в таблице

6.8 Модели ИО БЭ2704/БЭ2502

Инструменты данной группы являются моделями различных измерительных органов терминалов БЭ2704 и БЭ2502. Принципы их работы поясняются в описаниях на конкретные типы терминалов или шкафов, в которых они реализованы, и в настоящем документе не приводятся.

6.8.1 Состав

Состав моделей органов терминалов БЭ2704 и БЭ2502:

- ИО Z фазный;
- ИО Z междуфазный;
- ИО ИПФ;
- ИО ИПФК.

Модели отображают годограф соответствующего замера сопротивления на комплексной плоскости, а также показывают факты срабатывания компонентов соответствующего ИО.

Модели позволяют задавать точки определяющие начало и конец интервала отображения годографов сопротивления. В качестве начала может быть выбрано начало осциллограммы, текущий момент (определяемый курсором) или любой момент фиксированный маркером. В качестве конца – текущий момент, конец осциллограммы или любой момент фиксированный маркером.

6.8.2 НАСТРОЙКА

МЕНЮ

Команда	Краткое описание
Настроить	Отображение диалога настройки инструмента
Начало годографа	Группа позволяющая выбрать момент времени, используемый как начало интервала отображения годографов сопротивления.
Конец годографа	Группа позволяющая выбрать момент времени, используемый как конец интервала отображения годографов сопротивления.
Показывать годограф/текущую точку	Переключает режим отображения замера сопротивления между годографом за выбранный период времени и единственным значением в текущей точке осциллограммы.

ОКНО НАСТРОЙКИ - ВХОДНЫЕ ВОЗДЕЙСТВИЯ

Выбор входных сигналов для анализируемого ИО.

ОКНО НАСТРОЙКИ - ПАРАМЕТРЫ ЛИНИИ

Параметры линии задаются для всех типов ИО. Если вы работаете с осциллограммой БЭ2704 или БЭ2502, то становится доступна кнопка *Получить из осциллограммы*, которая позволяет загрузить соответствующие уставки, присутствующие в теле осциллограммы.

ОКНО НАСТРОЙКИ - ПАРАМЕТРЫ ОВП

Параметры органа определения вида повреждения (ОВП) задаются для всех типов ИО, кроме *Z междуфазный*. Если вы работаете с осциллограммой БЭ2704 или БЭ2502, то становится доступна кнопка *Получить из осциллограммы*, которая позволяет загрузить соответствующие уставки, присутствующие в теле осциллограммы.

ОКНО НАСТРОЙКИ - ХАРАКТЕРИСТИКА

На странице настройки параметров характеристики дана поясняющая иллюстрация, на которой показаны все параметры, которые необходимо задать.

Имеющийся набор параметров позволяет описать характеристику ИО любого, когда-либо выпускавшегося терминала БЭ2704 или БЭ2502. Если вы работаете с осциллограммой БЭ2704 или БЭ2502, то становится доступна кнопка *Получить из осциллограммы*, которая позволяет загрузить соответствующие уставки, присутствующие в теле осциллограммы.

6.8.3 РАБОТА С КОМПЛЕКСНОЙ ПЛОСКОСТЬЮ

В рамках комплексной плоскости можно производить ряд манипуляций с использованием мыши.

- вращение колеса в то время, как курсор мыши находится над плоскостью, меняет ее масштаб;
- захватив мышью плоскость, можно перемещать ее в рамках окна отображения;

– при остановке курсора мыши над любой точкой плоскости, показывается подсказка с координатами этой точки.

6.9 Модели ИО ЭКРА 200

Инструменты данной группы являются моделями различных органов терминалов ЭКРА 200. Принципы их работы поясняются в описаниях на конкретные типы терминалов или шкафов, в которых они реализованы, и в настоящем документе не приводятся.

Набор состоит из одного *ИО Z междуфазный*. Инструмент полностью аналогичен одноименному инструменту терминалов БЭ2704/БЭ2502, но моделирует ИО междуфазного сопротивления терминалов ЭКРА 200, который отличается принципом замера сопротивления и используемой характеристикой.

7 ПЕЧАТЬ ОСЦИЛЛОГРАММЫ

Команда *Файл > Печать* позволяет распечатать осциллограмму. При выполнении команды открывается окно предварительного просмотра.

Кнопка *Настроить принтер* позволяет выбрать принтер, настроить его параметры, задать параметры страницы и выбрать режим печати.

Кнопка *Параметры печати* позволяет указать ряд дополнительных параметров оформления. Результаты изменений параметров можно сразу увидеть в области предварительного просмотра.

На печать выдается видимый диапазон времени осциллограммы.

Осциллограмма печатается в том виде, в котором она в настоящий момент присутствует на экране со всеми установленными на нее инструментами.

Скрытые сигналы не печатаются.

8 СОХРАНЕНИЕ ОСЦИЛЛОГРАММЫ

Сохранение осциллограмм выполняется в собственный формат *Waves* (*.waves). Сохраняются все исходные данные, которые существовали в осциллограмме записанной устройством, а также все действия, выполненные в процессе анализа. Поэтому, открыв впоследствии такой файл, вы продолжите работу с того места, на котором прервали ее в момент сохранения.

Для сохранения открытой осциллограммы предназначены команды *Сохранить* и *Сохранить как* из меню *Файл*.

Команда *Файл > Сохранить как* всегда предлагает задать путь и имя файла в котором будет произведено сохранение.

Команда *Файл > Сохранить* работает либо также, если осциллограмма еще ни разу не сохранялась, либо сохраняет файл под текущим именем, если осциллограмма открыта из файла *.waves или ранее хоть раз была сохранена.

9 ЭКСПОРТ ОСЦИЛЛОГРАММЫ

Экспорт позволяет сохранить осциллограмму в одном из поддерживаемых открытых форматов.

Экспортируется только видимый диапазон времени осциллограммы.

Экспортируются только видимые (не скрытые) сигналы в порядке их текущего расположения на панели осциллограммы (сверху вниз).

9.1 ЭКСПОРТ В COMTRADE

Экспорт открытой осциллограммы в формат *Comtrade* осуществляется при помощи команд меню *Файл > Экспорт > Comtrade XXXX*, где XXXX это версия стандарта Comtrade.

Формируются только обязательные файлы (или части CFF-файла): CFG и DAT.

При экспорте можно:

- выбрать кодировку CFG: **DOS / Windows / UTF8**;
- выбрать формат DAT: **Text / Binary**;
- выбрать формат отсчетов аналоговых сигналов: **Int16 / Int32 / Float32**;
- выбрать тип величин аналоговых сигналов: **Первичные / Вторичные**;
- указать часовой пояс;
- выполнить передискретизацию отсчетов к требуемой частоте.

9.2 ЭКСПОРТ В CSV

Экспорт открытой осциллограммы в формат CSV осуществляется при помощи команды меню *Файл > Экспорт > CSV*. Файл представляет собой текстовую таблицу значений отсчетов по всем экспортируемым сигналам за экспортируемый отрезок времени.

Файл в формате CSV можно использовать для проведения расчетов в *Microsoft Excel* или с использованием программ собственной разработки. Чтение осциллограммы в формате CSV, выдаваемом *Waves* на порядок проще, чем чтение той же осциллограммы в формате *Comtrade*.

При экспорте можно:

- выбрать кодировку: **DOS / Windows / UTF8**;
- выбрать тип значений аналоговых сигналов: **Мгновенные / Действующие ...**;
- выбрать тип величин аналоговых сигналов: **Первичные / Вторичные**;
- указать, требуется ли включать в файл дискретные сигналы.

10 СОВМЕЩЕНИЕ ОСЦИЛЛОГРАММ

Waves содержит функции совмещения двух и более осциллограмм в единой новой осциллограмме. Доступны два вида совмещения:

Объединение – для совместного анализа записей различных устройств одного и того же процесса.

Стыковка – для стыковки осциллограмм записанных последовательно одним устройством.

10.1 ОБЪЕДИНЕНИЕ

Часто необходим совместный анализ данных одной и той же аварии, записанных различными устройствами РЗА или РАС. Для проведения такого анализа *Waves* предлагает функцию объединения осциллограмм.

Программа поддерживает два режима объединения:

- Автоматический, по абсолютному времени;
- Ручной.

10.1.1 АВТОМАТИЧЕСКОЕ ОБЪЕДИНЕНИЕ ПО АБСОЛЮТНОМУ ВРЕМЕНИ

Позволяет автоматически объединить несколько осциллограмм в одну, ориентируясь на абсолютные времена, указанные в них.

Для использования данного режима необходимо быть уверенным, что устройства, осциллограммы которых объединяются, были точно синхронизированы в момент записи аварии.

Для начала работы необходимо выполнить команду главного меню *Объединить > Автоматически*.

ШАГ 1 – ВЫБОР ОСЦИЛЛОГРАММ

Необходимо выбрать осциллограммы, которые требуется объединить.

ШАГ 2 – ВЫБОР СИГНАЛОВ

Необходимо отметить сигналы, которые следует включить в объединенную осциллограмму.

Флажок *Группировать сигналы с одинаковыми именами* позволяет разместить сигналы с одинаковыми именами рядом друг с другом.

ШАГ 3 – ВВОД ИМЕНИ НОВОЙ ОСЦИЛЛОГРАММЫ

Необходимо задать имя новой осциллограммы.

10.1.2 РУЧНОЕ ОБЪЕДИНЕНИЕ

Позволяет объединить две осциллограммы глядя на формы кривых и векторные диаграммы. Последовательным выполнением этой операции можно объединить любое количество осциллограмм.

Для начала работы необходимо выполнить команду главного меню *Объединить > Вручную*.

ШАГ 1 – ВЫБОР ОСЦИЛЛОГРАММ

Необходимо выбрать две осциллограммы: основную (осциллограмма 1 – красная), которая будет определять временную шкалу объединенной осциллограммы и дополнительную (осциллограмма 2 – синяя), которые требуется объединить.

ШАГ 2 – СОВМЕЩЕНИЕ

Необходимо выполнить точное совмещение этих осциллограмм.

Для этого следует выбрать по одному сигналу из каждой из них, по которым в данном случае удобно выполнять совмещение. В некоторых случаях удобные сигналы может потребоваться предварительно вычислить.

Выбранные сигналы совместно отображаются в поле графиков окна мастера (вертикальные линии цвета сигнала отображают момент пуска соответствующей осциллограммы).

Сигнал осциллограммы 1 (красный) неподвижен, а сигнал осциллограммы 2 (синий) можно свободно перемещать вдоль оси времени, ухватившись за него мышью.

В поле графиков работают такие же правила масштабирования и перемещения по оси времени, как и в основном окне.

Сигналы выводятся в индивидуальном режиме масштабирования. Размер окна мастера можно менять, при этом меняется размер области отображения графиков.

На панели инструментов под графиками сигналов доступны кнопки:

Кнопка	Описание
<i>Совместить > Моменты пуска</i>	Выполняет точное совмещение моментов пуска. Позволяет совместить осциллограммы, которые были записаны от одного возмущающего воздействия, даже если часы в устройствах не были синхронизированы
<i>Совместить > По астрономическому времени</i>	Выполняет совмещение по астрономическому времени. Позволяет совместить осциллограммы, записанные устройствами, которые были точно синхронизированы
<i>Не привязывать к отсчетам</i>	Включает и отключает режим, при котором перемещение сигнала 2 возможно без привязки к периоду дискретизации

Для упрощения совмещения в окне также присутствует векторная диаграмма, на которую можно вывести по одному сигналу из совмещаемых осциллограмм, причем, не обязательно те же самые, что выбраны для отображения графиков.

ШАГ 3 – ВЫБОР СИГНАЛОВ

Необходимо отметить сигналы из обеих осциллограмм, которые следует включить в объединенную осциллограмму.

Флажок *Группировать сигналы с одинаковыми именами* позволяет разместить сигналы с одинаковыми именами рядом друг с другом.

ШАГ 4 – ВВОД ИМЕНИ НОВОЙ ОСЦИЛЛОГРАММЫ

Необходимо задать имя новой осциллограммы.

10.1.3 ЗАМЕЧАНИЯ

В процессе объединения, сигналы исходных осциллограмм могут претерпеть следующие изменения:

- если исходные осциллограммы имеют разную частоту дискретизации, то объединенная осциллограмма будет сформирована с частотой дискретизации, наибольшей из исходных;

- если отсчеты второй и последующих осциллограмм при синхронизации сдвигаются относительно отсчетов осциллограммы 1 на величину, не кратную периоду дискретизации, то аналоговые сигналы объединенной осциллограммы, взятые из них, могут немного изменить свою форму вследствие интерполяции.

10.1.4 Источники

При объединении сигналы каждой из объединяемых осциллограмм собираются в *источник*. Источник хранит полную информацию о своей осциллограмме, которая включает: имя осциллограммы, информацию о записавшем ее устройстве, все дополнительные данные (уставки, причину пуска и т.д.), которые содержали исходные осциллограммы.

Каждый источник имеет собственное подменю в меню *Файл*, с командами, относящимися только к нему.

Источники именованные. По умолчанию имя источника формируется как буква **D** дополненная его порядковым номером в объединенной осциллограмме. Имя можно изменить вручную, при помощи команды *Свойства* меню источника. Имя источника отображается как префикс перед именами сигналов, относящихся к нему.

10.2 СТЫКОВКА

Некоторые устройства РЗА и РАС в процессе одной аварии записывают несколько последовательных осциллограмм. Иногда между ними могут быть паузы в несколько секунд.

В частности устройства ЭКРА в режиме длительного пуска записывают две осциллограммы: первая содержит момент аварии, а вторая момент устранения сигнала срабатывания.

Также встречаются устройства, которые по техническим причинам не могут записывать длинные процессы единым файлом и пишут несколько небольших фрагментов. Некоторые устройства пишут их встык, а некоторые внахлест.

Такие осциллограммы удобно анализировать так, как будто они составляют одну непрерывную запись. При помощи стыковки фрагментов можно выполнить такое совмещение.

В результирующую осциллограмму могут попасть только полностью идентичные сигналы, которые найдены во всех стыкуемых фрагментах. Полностью идентичными считаются сигналы, у которых одинаковы имена, коэффициенты трансформации и единицы измерения. Разумеется, если выбранные осциллограммы на самом деле записаны одним и тем же устройством, то кандидатами на попадание в новую осциллограмму будут абсолютно все сигналы.

Программа поддерживает два режима стыковки: Автоматический, по абсолютному времени и Ручной.

10.2.1 АВТОМАТИЧЕСКАЯ СТЫКОВКА ПО АБСОЛЮТНОМУ ВРЕМЕНИ

Позволяет автоматически стыковать несколько фрагментов, записанных одним устройством в одну осциллограмму ориентируясь на абсолютные времена, указанные в них. В том числе позволяет стыковать фрагменты, имеющие между собой паузы. Паузы между фрагментами заполняются значением *Нет данных*

Для начала работы необходимо выполнить команду главного меню *Состыковать > Автоматически*.

ШАГ 1 – ВЫБОР ФРАГМЕНТОВ

Необходимо отметить фрагменты, которые будут автоматически состыкованы.

ШАГ 2 – ВЫБОР СИГНАЛОВ

Необходимо отметить сигналы, которые следует включить в новую осциллограмму.

ШАГ 3 – ВВОД ИМЕНИ НОВОЙ ОСЦИЛЛОГРАММЫ

Необходимо задать имя новой осциллограммы.

10.2.2 РУЧНАЯ СТЫКОВКА

Позволяет стыковать два фрагмента вручную глядя на формы кривых, а также умеет находить интервалы перекрытия. Последовательным выполнением этой операции можно состыковать любое количество осциллограмм.

Ручную стыковку следует применять только для тех случаев, когда устройство заносит в записываемые фрагменты некорректные показания времени. Важно помнить, что **ни одно устройство, произведенное ООО НПП «ЭКРА», так осциллограммы не пишет**. Данный инструмент разработан с целью упростить стыковку фрагментов осциллограмм ряда **устройств сторонних производителей**, которые имеют подобные недостатки.

Для начала работы необходимо выполнить команду главного меню *Состыковать > Вручную*.

ШАГ 1 – ВЫБОР ФРАГМЕНТОВ

Необходимо выбрать две осциллограммы: основную (начало – красная), которая будет определять временную шкалу состыкованной осциллограммы и дополнительную (конец – синяя), которые требуется состыковать.

ШАГ 2 – СОВМЕЩЕНИЕ

Необходимо выполнить точное совмещение этих осциллограмм. Для этого следует выбрать сигнал, по которому в данном случае удобно выполнять совмещение. Этот сигнал из обоих фрагментов будет показан совместно в поле графиков окна совмещения

(вертикальные линии цвета сигнала отображают момент пуска соответствующей осциллограммы).

Сигнал фрагмента начала (красный) неподвижен, а сигнал фрагмента конца (синий) можно свободно перемещать вдоль оси времени, ухватившись за него мышью.

В поле графиков работают такие же правила масштабирования и перемещения по оси времени, как и в основном окне. Размер окна мастера можно менять, при этом меняется размер области отображения графиков.

На панели инструментов под графиками сигналов доступны кнопки:

Кнопка	Описание
<i>Совместить > По астрономическому времени</i>	Выполняет совмещение отсчетов по астрономическому времени.
<i>Совместить > По общей части</i>	Выполняет поиск общей части фрагментов.
<i>Не привязывать к отсчетам</i>	Включает и отключает режим, при котором перемещение сигнала 2 возможно без привязки к периоду дискретизации

Может показаться, что визуальная стыковка фрагментов, а тем более функция *совмещения по общей части* не требуются. В большинстве случаев так и есть, поэтому при переходе к шагу 2 фрагменты автоматически совмещаются по абсолютному времени и можно сразу нажимать кнопку *Далее*. Однако, как оказалось, в эксплуатации находится довольно большое количество **устройств сторонних производителей**, которые некорректно проставляют метки времени в смежных фрагментах и при этом пишут их внахлест. Такие фрагменты можно состыковать, лишь найдя их общую часть или визуально или при помощи команды поиска, которая также может не всегда находить перекрытие корректно в силу ряда других неточностей допускаемых в таких осциллограммах.

ШАГ 3 – ВЫБОР СИГНАЛОВ

Необходимо отметить сигналы, которые следует включить в новую осциллограмму.

ШАГ 4 – ВВОД ИМЕНИ НОВОЙ ОСЦИЛЛОГРАММЫ

Необходимо задать имя новой осциллограммы.

10.2.3 ЗАМЕЧАНИЯ

Если в качестве исходных фрагментов используются осциллограммы в формате *Comtrade*, возможны микросекундные отклонения в метках времени во втором и далее фрагментах относительно их исходных меток времени. Это связано с низкой точностью задания меток времени в большинстве существующих файлов *Comtrade*.

11 ПРОЧИЕ КОМАНДЫ ДЛЯ ОСЦИЛЛОГРАММ ЭКРА

11.1 ВНЕДРЕНИЕ ФАЙЛА КОНФИГУРАЦИИ УСТРОЙСТВА

Осциллограммы, записанные устройствами БЭ2704 и БЭ2502, содержат уставки, актуальные на момент пуска осциллографа. *Waves* предоставляет ряд функций, использующих их. Для того, чтобы уставки можно было использовать, требуется файл конфигурации соответствующего устройства.

В настоящее время комплекс программ *EKRASMS* при чтении осциллограмм из терминалов автоматически снабжает их нужным файлом конфигурации, но так было не всегда. Вы можете столкнуться с осциллограммой, в которой файла конфигурации нет.

При загрузке осциллограммы, не содержащей файл конфигурации, *Waves* автоматически ищет его на компьютере. Сначала в папке осциллограммы. Затем, если файл не был найден, в папке хранения файлов конфигурации *EKRASMS*. Если и там нет ничего подходящего, в меню появляется команда *Файл > Внедрить файл конфигурации устройства*.

Команда выводит на экран краткую инструкцию, с помощью которой вы сможете вручную внедрить требуемый файл конфигурации в осциллограмму, предварительно запросив его там, откуда вы осциллограмму получили. После успешного внедрения, команда из меню исчезает и заменяется набором команд, которые описаны далее в этом разделе.

11.2 ПРИЧИНА ПУСКА

Команда *Файл > Причина пуска* показывает список дискретных сигналов, послуживших причиной пуска осциллографа в устройствах ЭКРА.

11.3 СОСТОЯНИЕ В МОМЕНТ ПУСКА

Для осциллограмм, записанных устройствами БЭ2704 и БЭ2502 доступна команда *Файл > Состояние в момент пуска*. Она показывает состояние всех дискретных сигналов в момент пуска.

Не все выпущенные устройства регистрируют состояние дискретных сигналов в момент пуска, поэтому выводимый список может быть пустым.

Следует учитывать, что сигналы регистрируются с задержкой 10 мс после реального момента пуска и в общем случае могут не отражать ситуацию, имевшую место в истинный момент пуска.

11.4 ДОБАВЛЕНИЕ СИГНАЛОВ ИЗ БАЗЫ ДАННЫХ

Для осциллограмм, записанных устройствами БЭ2704 и БЭ2502 доступна команда *Файл > Добавить сигналы из базы данных*. Она позволяет добавить в открытую осциллограмму недостающие дискретные сигналы из файлов с выборками из базы данных *EKRASMS*. Поддерживаются все типы файлов с событиями, используемые в пакете *EKRASMS*, а именно: **.timeline*, **.archive*, **.datapak*, **.dbbackup*, **.db*.

11.4.1 ПРОБЛЕМА ПРОПУСКА СОБЫТИЙ

В процессе добавления сигналов в осциллограмму возможно образование отсчетов с признаком недостоверности. Это связано с тем, что при сборе событий в базе данных могут возникать пропуски. Как правило, это говорит о некорректной настройке маски регистрации устройства. В результате события, отражающие изменение сигнала в одну сторону, могут идти последовательно. При этом нельзя сказать, в какой момент между ними сигнал менял состояние в обратную сторону, и поэтому участки недостоверной информации заполняются значением *Нет данных*.

11.4.2 ПРОБЛЕМА ОКРУГЛЕНИЯ ВРЕМЕНИ

События в устройстве формируются с частотой дискретизации осциллографа (обычно 1200 Гц), однако метки времени в них имеют точность 1 мс (т.е. события раскладываются по сетке с частотой 1000 Гц). Округления времени неизбежны и, если сигнал менял свое значение дважды в соседних отсчетах, может получиться так, что соответствующие события в базе получают одну и ту же метку времени. При обнаружении такого конфликта отсчет получает значение *Нет данных*. Также следует понимать, что разница в частоте дискретизации осциллограммы и временной сетки базы данных, может приводить к ошибкам позиционирования изменений состояния добавляемого сигнала на плюс-минус один отсчет.

11.5 УСТАВКИ

Команда *Файл > Уставки*, позволяет показать уставки устройства ЭКРА в момент пуска осциллографа.

Для осциллограмм терминалов БЭ2704 и БЭ2502, в случае, если на ПК установлен пакет *EKRASMS*, уставки будут открыты в приложении *Atlas*.

12 ШАБЛОНЫ ОБРАБОТКИ ОСЦИЛЛОГРАММ

Waves позволяет сохранить практически все выполненные в осциллограмме изменения в виде скрипта-шаблона. Причем, совершенно не важно как вы эти изменения производили через команды дерева расчетов, создавая произвольные выражения (см. г. 13) или даже разрабатывая и выполняя какие-то скрипты (см. г. 14).

Впоследствии скрипт-шаблон можно выполнить в рамках другой осциллограммы полученной с того же (или подобного) устройства и повторить всю выполненную над исходным файлом обработку.

В скрипт-шаблон попадает следующая информация:

- Изменение имен, коэффициентов трансформации и единиц измерения записанных сигналов;
- Инвертирование сигналов;
- Добавление расчетных сигналов с информацией о том, по каким формулам и на основе каких сигналов они были рассчитаны;
- Видимость сигналов;
- Объединение сигналов в трехфазные цепи;
- Визуальные параметры сигналов (цвет, стиль линии)
- Взаимное расположение сигналов на экране;

В скрипт-шаблон **НЕ** сохраняется информация о вынесенных на осциллограмму маркерах, замерах и уровнях, потому что все, что привязано к конкретной ситуации или моментам времени **становится бессмысленным** при переносе в другую осциллограмму.

Для сохранения скрипта-шаблона необходимо выполнить команду *Файл > Сохранить шаблон* из меню осциллограммы.

Для последующего применения скрипта-шаблона необходимо выполнить команду *Файл > Применить шаблон*, которая выполнит указанный шаблон.

Так как скрипт-шаблон это самый обычный скрипт, вы можете вручную открывать редактор скрипта, загружать в него нужный шаблон и выполнять его. Это дает возможность визуально оценить действия, которые будут выполнены, перед тем как их инициировать, что полезно, если вам недостаточно имени файла шаблона для идентификации того, что он делает.

13 ВЫРАЖЕНИЯ

Waves позволяет создавать сигналы, рассчитываемые на основе текстовых выражений. В выражении можно использовать арифметические и логические операторы, числовые константы, сигналы, а также встроенные функции.

Для создания нового сигнала необходимо воспользоваться командой *Расчеты > Произвольное выражение*. Для удобства восприятия доступен ввод выражения в несколько строк.

```
Freq(
  Float(
    Complex("Ia") * 2 + Complex("Ib")
  )
)
```

Выражения не чувствительны к регистру символов. Корректность выражения проверяется только при закрытии окна кнопкой *OK*.

Для упрощения ввода **сигналов** и **функций** можно воспользоваться сочетанием клавиш **Ctrl+Space**, открывающим выпадающий список, из которого можно выбрать требуемое для вставки. Вставка осуществляется в текущую позицию курсора в редакторе выражения. Кроме того, если курсор находится в пределах некоторой функции, при нажатии комбинации клавиш **Ctrl + Shift + Space** на экран выводится подсказка с ее прототипом

13.1 СИГНАЛЫ

Сигналы вводятся в выражения своими именами, взятыми в кавычки. Если осциллограмма получена путем объединения нескольких осциллограмм и, как следствие, содержит несколько источников данных, имена сигналов дополняются в начале именем источника данных и разделителем :: (два двоеточия).

"Ia" - для обычной осциллограммы
 "D1::Ia" - для объединенной осциллограммы

13.2 ОПЕРАТОРЫ

13.2.1 ЛОГИЧЕСКИЕ ОПЕРАТОРЫ

Оператор	Описание	Пример
not	НЕ	not S1
or	ИЛИ	S1 or S2
and	И	S1 and S2
xor	Исключающее ИЛИ	S1 xor S2

Операндами логических операторов могут быть существующие дискретные сигналы и вложенные выражения, вычисляющие дискретный сигнал. Результатом выполнения логического оператора является дискретный сигнал.

13.2.2 АРИФМЕТИЧЕСКИЕ ОПЕРАТОРЫ

Оператор	Описание	Пример
+	Сложение	$S1 + S2$
-	Вычитание	$S1 - S2$
*	Умножение	$3 * S1$
/	Деление	$S1 / 3$
-	Унарный минус	$-S1$

Операндами арифметических операторов могут быть существующие аналоговые сигналы, числовые константы, а также вложенные выражения, вычисляющие аналоговые сигналы.

Оба операнда бинарных (имеющих правый и левый операнды) операторов обязаны быть одного типа, оба вещественные или оба комплексные. При использовании комплексных операндов, необходимо, чтобы их частотный базис был одинаков.

Результатом выполнения арифметического оператора является сигнал того же типа, что и операнды.

Кoeffициент трансформации и единицы измерения результата вычисляются в соответствии с правилами для конкретной операции. Однако, это не всегда возможно. Например, невозможно сказать, какова будет единица измерения результата, если вы по каким-то причинам складываете ток и напряжение. В случае невозможности вычисления значения свойства, ему присваивается значение по умолчанию $K_t = 1$, Единица измерения = <не определено>.

13.3 ФУНКЦИИ

13.3.1 ВЫЗОВ ФУНКЦИЙ

Для вызова встроенной функции необходимо записать ее имя, а затем в скобках через запятую указать значения ее параметров.

```
Pos("Ia", "Ib", "Ic")
```

Результатом выполнения функции является сигнал, тип которого определяется самой функцией.

Если функция вычисляет аналоговый сигнал, она определяет его коэффициент трансформации и единицы измерения.

13.3.2 ОПИСАНИЕ ФУНКЦИЙ В ДОКУМЕНТЕ

При описании функций, даются их прототипы.

Формат объявления прототипов, принятый в данном документе полностью соответствует языку программирования **Pascal**, за исключением того, что ключевое слово **function** не используется.

```
MyFunction(S: AnySignal; P1: Int = 0; P2: Int = 5; P3: Int = 0): FloatSignal
```

Здесь **MyFunction** – имя функции, **S**, **P1**, **P2**, **P3** – имена параметров, **AnySignal** и **Int** типы параметров, значения после знака равно – значения по умолчанию для *опциональных* параметров, а **FloatSignal** после закрывающей скобки это тип результата.

Три последних параметра функции (P1, P2, P3) объявлены как *опциональные*, т.к. для них указаны значения по умолчанию. Опциональными могут быть только один или несколько последних параметров функции.

При использовании функции в выражении, опциональные параметры можно опускать, если вас устраивают их значения по умолчанию. Однако, следует помнить, что опускать опциональные параметры можно только начиная с некоторого из них и до конца. Например, функцию **MyFunction** нельзя вызвать, опустив параметр P1, но указав P2 и опустив P3. В то же время, вызов с указанным параметром P1 и опущенными P2 и P3 допустим.

Следующие друг за другом *обязательные* параметры одного типа могут быть перечислены через запятую с однократным указанием типа для всей группы параметров.

```
MyFunction2(S: AnySignal; P1, P2, P3: Int): FloatSignal
```

13.3.3 ТИПЫ ПАРАМЕТРОВ ФУНКЦИЙ

Тип	Описание
Bool	Логический (возможные значения False и True)
Int ¹⁾	Целое число
Float ¹⁾	Вещественное число
AnySignal	Сигнал любого типа
BinarySignal	Дискретный сигнал
FloatSignal	Аналоговый сигнал
ComplexSignal	Комплексный сигнал

¹⁾ Допускает наложение ограничений на диапазон возможных значений

Результат функции может быть только одним из сигнальных типов (**xxxSignal**).

13.4 КОНСТАНТЫ

13.4.1 ЛОГИЧЕСКИЕ КОНСТАНТЫ

Waves поддерживает две логические константы: **False** (0) и **True** (1).

Логические константы можно использовать в качестве значений параметров функций типа **Bool**, а также в качестве дискретных сигналов постоянного значения, в том числе и используемых как параметры функций типа **BinarySignal** или **AnySignal**.

```
Shift("Ia", -10, True)
```

Выражение, состоящее только из логической константы, создает дискретный сигнал, заполненный значением, соответствующим указанной константе.

13.4.2 ЦЕЛОЧИСЛЕННЫЕ КОНСТАНТЫ

Целочисленные константы можно использовать в качестве значений параметров функций типа **Int** и **Float**, а также в качестве аналоговых сигналов постоянного значения, в том числе и используемых как параметры функций типа **FloatSignal** или **AnySignal**.

```
Abs("Ia" + 10)
```

Выражение, состоящее только из целочисленной константы, создает аналоговый сигнал, заполненный значением, соответствующим указанной константе.

13.4.3 ВЕЩЕСТВЕННЫЕ КОНСТАНТЫ

Вещественные константы можно использовать в качестве значений параметров функций типа **Float**, а также в качестве аналоговых сигналов постоянного значения, в том числе и используемых как параметры функций типа **FloatSignal** или **AnySignal**.

Десятичным разделителем для вещественных констант является символ “.” (точка).

```
12.4 * "Ia"
```

Выражение, состоящее только из вещественной константы, создает аналоговый сигнал, заполненный значением, соответствующим указанной константе.

13.4.4 КОНСТАНТА NIL

Ряд функций (см. далее), которые доступны для вызова из выражения, могут оперировать тремя междуфазными сигналами. Т.к. не всегда доступны все три междуфазных сигнала, допустимо вызывать такие функции, указав вместо одного отсутствующего сигнала специальную константу **Nil**.

13.5 ПРЕОБРАЗОВАНИЕ ТИПА СИГНАЛА

13.5.1 BINARY

Преобразование входного сигнала (S) любого типа в **дискретный** сигнал

```
Binary(S: AnySignal): BinarySignal
```

Тип S	Что будет создано
BinarySignal	Копия входного сигнала
ComplexSignal	Если <i>модуль</i> отсчета входного сигнала =0.0, то отсчет нового сигнала =0. Иначе =1
FloatSignal	Если отсчет входного сигнала =0.0, то отсчет нового сигнала =0. Иначе =1

13.5.2 FLOAT

Преобразование входного сигнал (S) любого типа в **аналоговый** сигнал.

Float(S: AnySignal): FloatSignal

Тип S	Что будет создано
BinarySignal	Если отсчет входного сигнала =0, то отсчет нового сигнала =0.0. Иначе =1.0
ComplexSignal	Обратное дискретное преобразование Фурье из частотного базиса входного сигнала
FloatSignal	Копия входного сигнала

13.5.3 COMPLEX

Преобразование входного сигнала (S) любого типа в **комплексный** сигнал в частотном базисе основной частоты (F) и гармоники (H).

Complex(S: AnySignal; F: Int = 0; H: Int = 1): ComplexSignal

Параметр	Тип	Мин	Макс	Описание
F	Int	0	SR ¹ /2	Основная частота =0 - используется основная частота осциллограммы
H	Int	1	²)	Номер гармоники

²) – SR (Sampling Rate) – частота дискретизации осциллограммы

³) – Определяется тем, что частота гармоники не может превышать SR/2

Тип S	Что будет создано
BinarySignal	Если отсчет входного сигнала =0, то отсчет нового сигнала =0.0/0°. Иначе =1.0/0°
ComplexSignal	Копия входного сигнала с возможностью смены частотного базиса
FloatSignal	Дискретное преобразование Фурье в заданном частотном базисе

13.6 СДВИГ СИГНАЛА: SHIFT

Создание сдвинутой на заданное значение (V) влево ($V < 0$) или вправо ($V > 0$) копии входного сигнала (S).

```
Shift(S: AnySignal; V: Int; InSamples: Bool = False): AnySignal
```

Опциональный параметр **InSamples** задает единицы измерения параметра V . Если **InSamples = True** – сдвиг задается в отсчетах, если **=False** – в миллисекундах, которые при выполнении функции преобразуются к ближайшему целому числу отсчетов

13.7 СБОРКА И РАЗБОРКА КОМПЛЕКСНОГО СИГНАЛА

13.7.1 COMPLEXA, COMPLEXP

Функции **ComplexA** и **ComplexP** предназначены для сборки комплексного сигнала из отдельных составляющих.

ComplexA собирает результат из действительной (Re) и мнимой (Im) частей, а **ComplexP** из модуля (Mag) и угла (Ang) в градусах.

```
ComplexA(Re, Im: FloatSignal; F: Int = 0; H: Int = 1): ComplexSignal
ComplexP(Mag, Ang: FloatSignal; F: Int = 0; H: Int = 1): ComplexSignal
```

Обе функции назначают результату частотный базис основной частоты (F) и гармоники (H). О частотном базисе подробнее см. функцию **Complex**.

13.7.2 RE, IM, MAG, ANG

Функции данного раздела позволяют разобрать **комплексный** сигнал (S) на отдельные составляющие. Каждая из функций создает **аналоговый** сигнал соответствующей составляющей.

```
Re(S: ComplexSignal): FloatSignal - вещественная часть
Im(S: ComplexSignal): FloatSignal - мнимая часть
Mag(S: ComplexSignal): FloatSignal - модуль
Ang(S: ComplexSignal): FloatSignal - угол в градусах
```

13.8 СИММЕТРИЧНЫЕ СОСТАВЛЯЮЩИЕ

Симметричные составляющие можно рассчитывать как по фазным, так и по междуфазным входным величинам (имеют в названиях окончание **2**).

Функции расчета нулевой последовательности содержат последний опциональный параметр **E**, в котором указывается степень коэффициента $(\sqrt{3})^E$, задающего соотношение коэффициентов трансформации по цепям звезды и треугольника, и может быть выбран в диапазоне $[-2..2]$.

13.8.1 Pos(2), Neg(2), Zero(2)

Данный набор функций позволяет рассчитывать симметричные составляющие в комплексной форме (по формулам из учебника ТОЭ).

ПРЯМАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ

```
Pos(A, B, C: ComplexSignal): ComplexSignal
Pos2(AB, BC, CA: ComplexSignal): ComplexSignal
```

ОБРАТНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ

```
Neg(A, B, C: ComplexSignal): ComplexSignal
Neg2(AB, BC, CA: ComplexSignal): ComplexSignal
```

НУЛЕВАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ

```
Zero(A, B, C: ComplexSignal; E: Int = 0): ComplexSignal
Zero2(AB, BC, CA: ComplexSignal; E: Int = 0): ComplexSignal
```

13.8.2 PosF(2), NegF(2), ZeroF(2)

Данный набор функций позволяет рассчитывать симметричные составляющие в аналоговой форме с применением сдвигов входных сигналов и без использования дискретного преобразования Фурье.

Все функции содержат опциональный параметр **F** - основная частота входных сигналов. Значение **0** этого параметра выбирает в качестве нее основную частоту осциллограммы.

ПРЯМАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ

```
PosF(A, B, C: FloatSignal; F: Int = 0): FloatSignal
PosF2(AB, BC, CA: FloatSignal; F: Int = 0): FloatSignal
```

ОБРАТНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ

```
NegF(A, B, C: FloatSignal; F: Int = 0): FloatSignal
NegF2(AB, BC, CA: FloatSignal; F: Int = 0): FloatSignal
```

НУЛЕВАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ

```
ZeroF(A, B, C: FloatSignal; F: Int = 0; E: Int = 0): FloatSignal
ZeroF2(AB, BC, CA: FloatSignal; F: Int = 0; E: Int = 0): FloatSignal
```

13.9 МОЩНОСТЬ ТРЕХ ФАЗ: POWER(2)

Создание **комплексного** сигнала мощности трехфазного присоединения с использованием фазных (Power) или междуфазных (Power2) напряжений.

```
Power(UA, UB, UC, IA, IB, IC: ComplexSignal): ComplexSignal
Power2(UAB, UBC, UCA, IA, IB, IC: ComplexSignal): ComplexSignal
```

13.10 ЧАСТОТА СИГНАЛА: FREQ

Создание **аналогового** сигнала, отсчеты которого представляют собой частоту входного **аналогового** сигнала (S) в соответствующие моменты времени.

```
Freq(S: FloatSignal; AvgPds: Int = 4): FloatSignal
```

Оptionальный параметр **AvgPds** задает *число периодов усреднения* при расчете частоты, которое можно выбрать в диапазоне [1..16]

13.11 КОМПЕНСАЦИЯ ЕМКОСТНОГО ТОКА ЛИНИИ: RmCC

Создание **аналогового** сигнала тока указанной фазы (Ph) линии, в котором скомпенсирован ее емкостной ток.

```
RmCC(UA, UB, UC, IA, IB, IC: FloatSignal; B1, B0, Len: Float; Ph: Int;
F: Int = 0): FloatSignal
```

Параметр	Тип	Мин	Макс	Описание
B1	Float	0	-	Емкостная поперечная проводимость прямой последовательности линии (мкСм/км)
B0	Float	0	-	Емкостная поперечная проводимость нулевой последовательности линии (мкСм/км)
Len	Float	0	-	Длина линии (км)
Ph	Int	0	2	Фаза, которую необходимо рассчитать 0 - A, 1 - B, 2 - C
F	Int	0	SR ¹⁾ /2	Основная частота =0 - используется основная частота осциллограммы

¹⁾ – SR (Sampling Rate) – частота дискретизации осциллограммы

13.12 КОМПЕНСАЦИЯ НАГРУЗОЧНОГО НЕБАЛАНСА: RmLD

Создание **комплексного** сигнала, в отсчетах которого скомпенсирован нагрузочный небаланс входного **комплексного** сигнала.

```
RmLd(S: ComplexSignal): ComplexSignal
```

13.13 СРАВНЕНИЕ АНАЛОГОВЫХ СИГНАЛОВ

13.13.1 CMP, CMPЕ, CMPG, CMPGE, CMPЛ, CMPLE, CMPNE

Функции семейства **Cmp** предназначены для создания **дискретного** сигнала, в отсчет которого заносится **1**, если для соответствующих отсчетов входных **аналоговых** сигналов (S1 и S2) выполняется условие сравнения. В противном случае в отсчет заносится **0**.

Функция **Cmp** позволяет задать операцию сравнения путем выставления значений параметров E, L и G.

Cmp(S1, S2: *FloatSignal*; E, L, G: *Bool*): *BinarySignal*

Параметр	Тип	Описание
E	Bool	Выполнить сравнение S1 = S2
L	Bool	Выполнить сравнение S1 < S2
G	Bool	Выполнить сравнение S1 > S2

Зависимость условия сравнения от параметров E, L, G:

E	L	G	Условие	Пояснение
FALSE	FALSE	FALSE	0	всегда 0
TRUE	FALSE	FALSE	S1 = S2	равно
FALSE	TRUE	FALSE	S1 < S2	меньше
FALSE	FALSE	TRUE	S1 > S2	больше
TRUE	TRUE	FALSE	S1 <= S2	меньше или равно
TRUE	FALSE	TRUE	S1 >= S2	больше или равно
FALSE	TRUE	TRUE	S1 <> S2	не равно
TRUE	TRUE	TRUE	1	всегда 1

Остальные функции семейства, выполняют сравнение по одному из условий.

CmpE(S1, S2: *FloatSignal*): *BinarySignal* – условие S1 = S2
CmpG(S1, S2: *FloatSignal*): *BinarySignal* – условие S1 > S2
CmpGE(S1, S2: *FloatSignal*): *BinarySignal* – условие S1 >= S2
CmpL(S1, S2: *FloatSignal*): *BinarySignal* – условие S1 < S2
CmpLE(S1, S2: *FloatSignal*): *BinarySignal* – условие S1 <= S2
CmpNE(S1, S2: *FloatSignal*): *BinarySignal* – условие S1 <> S2

13.13.2 MIN, MAX

Функции **Min** и **Max** создают **аналоговый** сигнал, отсчеты которого являются соответственно минимумом или максимумом соответствующих отсчетов двух входных **аналоговых** сигналов S1 и S2.

```
Min(S1, S2: FloatSignal): FloatSignal
Max(S1, S2: FloatSignal): FloatSignal
```

13.14 ПРЕОБРАЗОВАНИЯ ДИСКРЕТНЫХ СИГНАЛОВ

Функции данного раздела выполняют преобразование входного **дискретного** сигнала (S) в выходной сигнал, согласно определенному алгоритму.

13.14.1 ВЫДЕРЖКА ВРЕМЕНИ: DELAY

Выдержка времени (T) входного сигнала на срабатывание (Return = False) или возврат (Return = True).

```
Delay(S: BinarySignal; T: Int; Return: Bool = False;
      InSamples: Bool = False): BinarySignal
```

Оptionальный параметр **InSamples** задает единицы измерения параметра T. Если **InSamples = True** – выдержка задается в отсчетах, если **=False** – в миллисекундах, которые при выполнении функции преобразуются к ближайшему целому числу отсчетов

13.15 ПРЕОБРАЗОВАНИЯ АНАЛОГОВЫХ СИГНАЛОВ

Функции данного раздела выполняют преобразование входного **аналогового** сигнала (S) в выходной сигнал, согласно определенному алгоритму.

Часть функций содержат опциональный параметр **F** - основная частота входного сигнала. Значение **0** этого параметра выбирает в качестве основной частоты – основную частоту осциллограммы.

13.15.1 АБСОЛЮТНАЯ ВЕЛИЧИНА: ABS

Абсолютная величина отсчетов входного сигнала.

```
Abs(S: FloatSignal): FloatSignal
```

13.15.2 АРККОСИНОС: ACOS

Арккосинус отсчетов входного сигнала.

```
ACos(S: FloatSignal): FloatSignal
```

13.15.3 АРККОТАНГЕНС: ACTG

Арккотангенс отсчетов входного сигнала.

```
ACTg(S: FloatSignal): FloatSignal
```


13.15.4 АРКСИНОС: ASIN

Арксинус отсчетов входного сигнала.

ASin(S: FloatSignal): FloatSignal

13.15.5 АРКТАНГЕНС: ATG

Арктангенс отсчетов входного сигнала.

ATg(S: FloatSignal): FloatSignal

13.15.6 ПОСТОЯННАЯ СОСТАВЛЯЮЩАЯ: CONST

Выделение постоянной составляющей на периоде входного сигнала.

Const(S: FloatSignal; F: Int = 0): FloatSignal

13.15.7 КОСИНУС: COS

Косинус отсчетов входного сигнала. Выходной сигнал представляет собой угол в градусах.

Cos(S: FloatSignal): FloatSignal

13.15.8 КОТАНГЕНС: CTG

Котангенс отсчетов входного сигнала. Выходной сигнал представляет собой угол в градусах.

Ctg(S: FloatSignal): FloatSignal

13.15.9 ПРОИЗВОДНАЯ: DER

Производная на периоде входного сигнала.

Der(S: FloatSignal; F: Int = 0): FloatSignal

Результирующий сигнал является приведенным к единице измерения входного сигнала, поэтому его масштаб и единица измерения получаются таким же, как и у входного.

13.15.10 СРЕДНЕЕ ЗНАЧЕНИЕ: MEAN

Среднее значение на периоде входного сигнала.

Mean(S: FloatSignal; F: Int = 0): FloatSignal

13.15.11 ПИКОВЫЙ ДЕТЕКТОР: PEAK

Фильтр "Пиковый детектор" входного сигнала.

Peak(S: *FloatSignal*; Tau, Limit: *Int*): *FloatSignal*

Параметр	Тип	Мин	Макс	Описание
Tau	Int	0	10000	Постоянная времени затухания (мс)
Limit	Int	0	10000	Ограничение длительности действия (мс)

Обрабатываются только пики в положительную сторону. Для обработки отрицательных пиков необходимо предварительно инвертировать входной сигнал ($-1*S$). Для обработки и отрицательных и положительных пиков одновременно необходимо предварительно получить абсолютное значение (**Abs**(S)) сигнала.

13.15.12 СРЕДНЕКВАДРАТИЧЕСКОЕ ЗНАЧЕНИЕ: RMS

Среднеквадратическое (действующее) значение на периоде входного сигнала.

RMS(S: *FloatSignal*; F: *Int* = 0): *FloatSignal*

13.15.13 СИНУС: SIN

Синус отсчетов входного сигнала. Выходной сигнал представляет собой угол в градусах.

Sin(S: *FloatSignal*): *FloatSignal*

13.15.14 КВАДРАТНЫЙ КОРЕНЬ: SQRT

Квадратный корень из отсчетов входного сигнала.

Sqrt(S: *FloatSignal*): *FloatSignal*

13.15.15 ТАНГЕНС: TG

Тангенс отсчетов входного сигнала. Выходной сигнал представляет собой угол в градусах.

Tg(S: *FloatSignal*): *FloatSignal*

13.15.16 НОРМАЛИЗАЦИЯ УГЛА: TO180

Приведение входного сигнала угла в градусах в диапазон $[-180^{\circ}..+180^{\circ}]$.

To180(S: *FloatSignal*): *FloatSignal*

13.16 ПРЕОБРАЗОВАНИЯ КОМПЛЕКСНЫХ СИГНАЛОВ

Функции данного раздела выполняют преобразование входного **комплексного** сигнала (S) в выходной сигнал, согласно определенному алгоритму.

13.16.1 СОПРЯЖЕННОЕ ЗНАЧЕНИЕ: CONJ

Сопряженное значение отсчетов входного сигнала.

Conj(S: *ComplexSignal*): *ComplexSignal*

13.16.2 ПОВОРОТ: ROT

Поворот отсчетов входного сигнала на угол (Ang) в градусах.

Rot(S: *ComplexSignal*; Ang: *FloatSignal*): *ComplexSignal*

В качестве типа параметра Ang используется *FloatSignal*, а не *Float* для того, чтобы угол мог быть не постоянным. Тем не менее, т.к. параметр типа *FloatSignal*, как частный случай, принимает и вещественную константу, значение параметра можно задавать и постоянным числом.

13.17 СЧЕТЧИК: COUNT

Счетчик, прибавляющий (или вычитающий) 1 на каждом следующем отсчете заданного интервала осциллограммы.

Count(Start: *Int* = 0; Stop: *Int* = 0; Initial: *Int* = 0; Countdown: *Bool* = *false*): *FloatSignal*

Параметр	Тип	Мин	Описание
Start	Int	0	Индекс отсчета начала интервала счета
Stop	Int	0	Индекс отсчета конца интервала счета (0 – до конца осциллограммы)
Initial	Int		Начальное значение счетчика
Countdown	Bool		Считать в обратную сторону (вычитанием 1)

Индексы отсчетов можно посмотреть прямо на графиках осциллограммы, увеличив масштаб по времени до того момента, когда появится нумерация

14 СКРИПТ

Скрипт – это программа, написанная пользователем в соответствии со специальными правилами. С помощью скрипта можно автоматизировать рутинные действия, которые приходится выполнять постоянно.

Например, вы постоянно получаете осциллограммы с некоторого устройства и всегда перед началом анализа выполняете набор одинаковых действий. Что-то скрываете, организуете определенным образом сигналы, производите определенные расчеты. Вы можете описать все эти действия в скрипте (или проделать их однократно и сохранить шаблонный скрипт, выполнив команду *Файл > Сохранить шаблон*) и выполнять их впоследствии существенно быстрее, буквально в пару кликов.

14.1 РЕДАКТОР КОДА

Для работы со скриптом в *Waves* встроен редактор кода, видимостью которого можно управлять при помощи команды *Инструменты > Скрипт*.

Редактор кода помимо поля ввода программы содержит собственное меню:

Команда	Краткое описание
Файл	
Открыть	Загружает в редактор скрипт, сохраненный в файле
Открыть из старой библиотеки ¹⁾	Загружает текст скрипта, сохраненный в библиотеке <i>Waves 3</i> . После открытия, скрипт следует преобразовать командой <i>Действия > Преобразовать старый скрипт</i> .
Сохранить	Сохраняет текущий скрипт в файле
Правка	
Отменить	Отменяет последнее действие в редакторе кода
Вырезать	Вырезает выделенный текст в буфер обмена
Копировать	Копирует выделенный текст в буфер обмена
Вставить	Вставляет текст из буфера обмена в позицию курсора редактора кода
Удалить	Удаляет выделенный текст
Выделить все	Выделяет весь текст скрипта
Действия	
Преобразовать старый скрипт	Преобразует скрипт, написанный для <i>Waves 3</i> в актуальный
Выполнить	Выполняет скрипт

¹⁾ - Команда отсутствует, если на ПК нет библиотеки *Waves 3*

В редакторе кода работает контекстная вставка. Она позволяет заменить ручной ввод (в процессе которого можно ошибиться) имен сигналов, команд, констант и функций выражений расчета сигналов выбором из выпадающего списка. Список контекстной вставки появляется при нажатии комбинации клавиш **Ctrl + Space**, и содержит все элементы указанных сущностей, которые можно вставить в текущей позиции редактора кода. Если при нажатии указанной комбинации список не появляется, значит, в текущей позиции ничего из перечисленного ввести нельзя.

Кроме того, если курсор находится в пределах некоторой команды, при нажатии комбинации клавиш **Ctrl + Shift + Space** на экран выводится подсказка с ее прототипом

В процессе выполнения скрипта под редактором кода появляется лог – список сообщений выводимых исполняющей системой в процессе работы.

14.2 КОММЕНТАРИИ

Комментарий – это текст, который предназначен для чтения пользователем и пропускается исполняющей системой. Обычно комментарии используются для создания пояснений к написанному коду. Скрипт поддерживает два типа комментариев.

СТРОЧНЫЙ КОММЕНТАРИЙ

Строчный комментарий - это текст, начинающийся с пары символов “//” (слеш, слеш) и продолжающийся до перевода строки:

```
// это строчный комментарий
```

БЛОЧНЫЙ КОММЕНТАРИЙ

Блочный комментарий - это текст между парами символов “/*” (слеш, звездочка) и “*/” (звездочка, слеш), включая переводы строк:

```
/* это блочный
комментарий */
```

14.3 ЛИТЕРАЛЫ

Литерал – это константа, включаемая непосредственно в текст программы. Литералы могут быть числовыми (*целыми и вещественными*) и *строковыми*.

Строковые литералы задаются в кавычках.

```
B = A + 10; // литерал - 10
Rtm.Log(Msg = "My message") // литерал - "My message"
```

14.4 ПЕРЕМЕННЫЕ

Переменная – это именованный объект, который хранит значение или массив значений одного из 3-х типов **string** (строка), **int** (целое число), **float** (вещественное число). Тип переменной определяется в момент объявления неявно по присвоенному

значению. Переменная доступна с точки объявления и до конца блока, в котором она объявлена.

Переменная объявляется при помощи ключевого слова **let**, за которым следует ее имя и, после знака "=" (равно) – начальное значение. Объявление заканчивается символом ";" (точка с запятой).

```
let I = 10;           // целая переменная
let F = 10.5;        // вещественная переменная
let S = "10";        // строковая переменная
let IL = [1, 2, 3];  // массив целых чисел
let FL = [1.0, 2.5, 3.6]; // массив вещественных чисел
let SL = ["1", "2", "3"]; // массив строк
```

Тип объявленной переменной и размер объявленного массива впоследствии изменить нельзя.

Значение, хранимое переменной можно изменять, находясь в области ее видимости, присвоив ей новое значение.

```
I = 11;
F = 34.7;
SL[0] = "hello"; // изменяется 0-й элемент массива
```

14.5 ДИНАМИЧЕСКОЕ ПРЕОБРАЗОВАНИЕ ТИПА

Исполняющая система поддерживает динамическое преобразование значений переменных и литералов к типу переменной (или параметра команды), которой производится присваивание.

В таблице представлены правила динамического преобразования значений к целевому типу. В строках даны результаты присвоения значений разных типов на участке скрипта, где требуется определенный целевой тип значения.

Целевой тип	Присваиваемое значение			
	2 (int)	2.8 (float)	"3.2" (string)	"не число" (string)
int	2	2	3	Ошибка
float	2.0	2.8	3.2	Ошибка
string	"2"	"2.8"	"3.2"	"не число"

В частности видно, что попытка применения вещественного значения там, где требуется значение целого типа, приводит к отсечению его дробной части, а применение строкового значения там, где требуется любой из числовых типов, приводит к попытке преобразовать строку в число соответствующего типа, и если это не удастся, генерируется ошибка выполнения.

14.6 ЧИСЛОВАЯ АРИФМЕТИКА

Над целыми и вещественными переменными и литералами возможно выполнение арифметических операций: сложение (+), вычитание (-), умножение (*) и деление (/).

Все операторы, кроме *деления* дают целый результат, если оба операнда целые и вещественный результат, если хотя бы один из операндов вещественный. *Деление* всегда дает вещественный результат.

```
let A = 10;
let B = 21;
let C = A * B + 10.8; // переменная получит вещественное значение 220.8
```

14.7 КОНКАТЕНАЦИЯ СТРОК

Строковые переменные и литералы допустимо конкатенировать (стыковать). В результате получается строка, составленная из всех подстрок, участвовавших в стыковке в порядке их следования в выражении.

```
let A = "Hello";
let B = A + " world!"; // результат: "Hello world!"
```

В качестве операндов конкатенации могут выступать переменные и литералы числовых типов. При этом они автоматически преобразуются в строковое представление.

```
let x = ", ";
let A = "раз";
let B = 2;
let C = A + x + B + x + "3" + x + 4.5; // результат: "раз, 2, 3, 4.5"
let D = 10 + B + " " + A; // результат "12 раз"
```

Для конкатенации строк и для сложения чисел используется один и тот же оператор (+), поэтому, чтобы правильно писать выражения необходимо запомнить правило:

Если хотя-бы один из операндов является строкой, то оператор выполняет конкатенацию и результатом будет строка (все операции в расчете переменной **C** являются конкатенацией), в противном случае – он выполняет сложение с результатом одного из числовых типов (10 + B в расчете переменной **D** это сложение, а все остальное - конкатенации). При смешивании в одном выражении сложений и конкатенаций рекомендуется использовать скобки для явного обозначения приоритетов операций.

Еще несколько примеров:

```
let A = 10 + 1; // результат int: 11
let B = 10 + "1"; // результат string: "101"

let C = 0; // конкатенация с преобразованием к типу int.
C = 10 + "1"; // результат int: 101
```

14.8 Блоки

Блок - это участок скрипта, ограниченный символами "{" и "}". Переменные, объявленные внутри блока, являются локальными для него и недоступны за его пределами.

```
{
  let I = 10;
}
Rtm.Log(Msg = I) // ОШИБКА: Переменная I недоступна
```

Допустимо объявлять внутри блока переменную с тем же именем, что уже существует вне его. При этом для исполняющей системы эту будут две разные переменные. Обращение к такой переменной внутри блока, вернет значение переменной, объявленной внутри блока, а обращение вне блока – переменной, объявленной вне его.

```
let I = 5;
{
  let I = 10;
  Rtm.Log(Msg = I) // вывод значения 10
}
Rtm.Log(Msg = I) // вывод значения 5
```

14.9 Циклы

Цикл - это блок, который выполняется заданное число раз.

```
Loop(I, 3)
{
  Rtm.Log(Msg = I)
}
```

Цикл объявляется при помощи ключевого слова **Loop**, за которым в скобках через запятую следуют переменная цикла типа **int** (**I**, в примере выше), и число итераций (**3**, в примере выше). Тело цикла (блок, который будет повторен заданное число раз) заключается в фигурные скобки.

Переменная цикла не требует специального объявления и доступна внутри блока цикла. Перед началом работы цикла, ее значение равно 0 и после завершения каждой итерации увеличивается на 1. Соответственно, на последней итерации цикла, ее значение на 1 меньше, чем заданное число итераций.

Явное присвоение значения переменной цикла внутри тела цикла игнорируется.

Число итераций не обязательно задавать литералом, оно может быть и целочисленным выражением.

```
let A = 5;
Loop(I, 2*(A + 4))
{
  Rtm.Log(Msg = I)
}
```


Циклы могут вкладываться друг в друга

```

Loop(i, 3)
{
    .. // здесь доступна только i
    Loop(j, 2)
    {
        .. // здесь доступны и i и j
    }
    .. // здесь доступна только i
}

```

14.10 МАКРОСЫ

Макрос – это средство, позволяющее заменить громоздкое, используемое неоднократно выражение более компактным определением.

Макрос объявляется при помощи ключевого слова **define**, за которым в скобках через запятую следуют его формальные параметры. Далее ставится знак "=" (равно) и записывается выражение, которое макрос будет в дальнейшем олицетворять в скрипте. Завершается объявление символом ";" (точка с запятой). В выражении можно использовать формальные параметры макроса. При вызове они будут заменены переданными значениями.

```

define Desc(a, b, c) = b*b - 4*a*c;

```

После того, как макрос определен, его можно вызывать для выполнения соответствующего расчета.

```

Rtm.Log(Msg = Desc(2, 3, 1)); // в лог попадет результат расчета "1"

```

По принципу действия макрос похож на директиву препроцессора **#define** языка **Си**. В момент выполнения в месте вызова макрос заменяется на выражение, которое он определяет, с расстановкой конкретных значений переданных параметров.

Таким образом, исполняющая система, встретив в скрипте вызов макроса:

```

Desc(2, 3, 1)

```

... заменяет его на соответствующее выражение, с расстановкой в нем переданных значений параметров:

```

(3*3 - 4*2*1)

```

...и только затем вычисляет результат.

14.11 КОМАНДЫ

Команды – это сущности скрипта, вызов которых производит воздействие на открытую осциллограмму, в рамках которой выполняется скрипт.

Вызов команды осуществляется записью ее имени, после которого в скобках через запятую следуют именованные значения ее параметров. Порядок следования именованных параметров не важен.

```
Sig.Create(Name = "MySig", Expr = Abs(2 * "Ia"), Color = Color.Red)
```

Параметры могут быть опциональными. Поведение команд при опускании того или иного опционального параметра определяется самой командой и будет подробно описано в разделе ее описания.

Параметры команд могут быть следующих типов:

Тип	Описание
Bool	Логический (возможные значения False и True)
Int	Целое число
Float	Вещественное число
String	Строка
String[]	Массив (список) строк
Expression	Специальный тип, используемый исключительно для параметра Expr команды Sig.Create , который позволяет передать команде выражение для расчета сигнала, записанное по правилам, изложенным в главе 13.

Команды не возвращают никаких значений и делятся на несколько смысловых категорий:

Категория	Объект воздействия
Rtm	Система исполнения скрипта
Sig	Сигналы
Tri	Трехфазные цепи
Osc	Осциллограмма в целом

14.12 КОМАНДЫ КАТЕГОРИИ RTM

14.12.1 ЗАПИСЬ СООБЩЕНИЯ В ЛОГ: RTM.LOG

Записывает сообщение в лог выполнения скрипта. Команду можно использовать в отладочных целях.

Параметр	Тип	Описание
Msg	String	Сообщение, помещаемое в лог

14.12.2 ПРЕРЫВАНИЕ ВЫПОЛНЕНИЯ: RTM.STOP

Команда прекращает выполнение скрипта. Команду можно использовать в отладочных целях.

Параметр	Тип	Описание	Если отсутствует
Msg	String	Сообщение, помещаемое в лог	Нет записи в лог

14.13 КОМАНДЫ КАТЕГОРИИ SIG

14.13.1 СОЗДАНИЕ СИГНАЛА: SIG.CREATE

Создание нового сигнала.

Параметр	Тип	Описание	Если отсутствует
Name	String	Имя создаваемого сигнала	
Expr	Expression	Выражение для расчета сигнала	
Sec ¹⁾	Bool	Расчет во вторичных величинах	False
Q1 ¹⁾	String	Первичная величина коэффициента трансформации	Результат расчета
Q2 ¹⁾	String	Вторичная величина коэффициента трансформации	Результат расчета
NonPeriodic ²⁾	Bool	Сигнал отображает только мгновенное значение	Результат расчета
Color	Int	Цвет	Color.Black
Dashed ¹⁾	Bool	Пунктир	False
Invert	Bool	Значение флажка: Инвертировать отсчеты	False
Noise ¹⁾	Bool	Значение флажка: Сигнал содержит шум	False
Fail	Bool	³⁾	False

¹⁾ Не используются при создании дискретного сигнала

²⁾ Используется только при создании аналоговых сигналов

³⁾ В случае ошибки в значении Expr, позволяет прервать выполнение скрипта (True) или пропустить выполнение команды (False)

Значение параметра Color - индекс цвета, который можно видеть в списке выбора цвета сигнала. Как следствие, его возможные значения ограничены длиной этого списка. Для удобства задания значений цвета в скрипте доступны именованные константы для каждого индекса в списке цветов:

Константа	Знач.	Описание
Color.Black	0	Черный
Color.Yellow	1	Желтый
Color.Green	2	Зеленый
Color.Red	3	Красный
Color.Blue	4	Синий
Color.Violet	5	Фиолетовый
Color.Turquoise	6	Бирюзовый
Color.Brown	7	Коричневый
Color.Gray	8	Серый
Color.Dark	9	Темный

Параметры Q1 и Q2 имеют тип **String**, при этом строка должна быть составлена по определенным правилам. Строка должна начинаться с числового значения, после которого (с пробелом или без) можно указать одну из единиц измерения, представленных в таблице ниже.

Значение	Физическая Величина	Описание
	Любая	Безразмерная величина
pu	Любая	Относительные единицы
%	Любая	Относительные единицы в процентах
deg	Угол	Градус
C	Температура	Градус Цельсия
Hz	Частота	Герц
V	Напряжение	Вольт
A	Ток	Ампер
ohm	Сопротивление	Ом
S	Проводимость	Сименс
VA	Мощность	Вольт-ампер
W	Активная мощность	Ватт
var	Реактивная мощность	Вольт-ампер реактивный
m	Длина	Метр

Значения Q1 и Q2 вводятся строго в базовой единице измерения системы СИ, масштабирующие префиксы (кило, мега и т.д.) недопустимы. Регистр символов единицы измерения не имеет значения, но рекомендуется строго придерживаться написания из таблицы выше.

```
"110000 V" // 110 кВ
"12 Ohm"   // 12 Ом
```

Параметр `Expr`, специального типа `Expression` задает выражение, согласно которому будет рассчитан новый сигнал. Правила составления выражений расчета сигналов описаны в главе 13.

В выражении `Expression` можно использовать переменные, макросы, а также числовую арифметику скрипта. Все эти сущности будут разобраны и вычислены, с простановкой результата в место вставки.

Единственная возможность скрипта, которая не работает в рамках типа `Expression` это конкатенация строк. Конкатенация вступает в противоречие с возможностью сложения сигналов в выражениях `Expression`, которому в данном случае отдается приоритет.

ПРИМЕР

Пусть в осциллограмме существуют токи 3-х фаз 2-х выключателей "Iв1_a", "Iв1_b", "Iв1_c" и "Iв2_a", "Iв2_b", "Iв2_c". Необходимо посчитать токи 3-х фаз линии, как суммы соответствующих токов фаз выключателей с постоянной составляющей, зависящей от фазы, и создать соответствующие сигналы "Iл_a", "Iл_b", "Iл_c". Переменные A и B участвующие в определении постоянной составляющей считаем рассчитанными по некоторому алгоритму до достижения обсуждаемого фрагмента кода.

```
// Объявляем имена фаз
let ph = ["a", "b", "c"];

// Объявляем макрос составления имени тока выключателя
define mkQName(QNum, phase) = "Iв" + QNum + "_" + ph[phase];

// Объявляем макрос составления имени тока линии
define mkLName(phase) = "Iл_" + ph[phase];

// Объявляем макрос расчета постоянной составляющей
define mkOffset(phase) = (A[phase] + B[phase] * 2);

// Цикл для создания токов 3-х фаз
loop(i, 3)
{
    // Создаем локальные для цикла переменные с собранными именами токов
    // выключателей текущей фазы, т.к. собрать их прямо в выражении
    // Expression не получится из-за запрещенной конкатенации строк
    let I1 = mkQName(1, i);
    let I2 = mkQName(2, i);

    // Создаем сигнал тока линии текущей фазы
    Sig.Create(
        Name = mkLName(i),
        Expr = I1 + I2 + mkOffset(i)
    )
}
```

Обратите внимание, как макросы (даже однократно используемые) могут сделать код чище и понятнее. Представьте, что бы было, если бы вместо них в местах вызова были вставлены соответствующие выражения.

Рассмотрим, какие преобразования претерпит код тела цикла после раскрытия макросов (для $i = 0$):

```

{
  let I1 = "IВ" + 1 + "_" + ph[0];
  let I2 = "IВ" + 2 + "_" + ph[0];

  Sig.Create(
    Name = "Iл_" + ph[0],
    Expr = I1 + I2 + (A[0] + B[0] * 2)
  )
}

```

Предположим, что переменные A и B, к моменту выполнения нашего цикла имеют следующие значения (это не корректный код, а только демонстрация значений):

```

A = [10, 20, 30]
B = [40, 50, 60]

```

Тогда в процессе вычисления переменные I1 и I2, а также используемые параметры команды Sig.Create получают следующие значения (это не корректный код, а только демонстрация значений):

```

I1 = "IВ1_a"
I2 = "IВ2_a"
Sig.Create.Name = "Iл_a"
Sig.Create.Expr = "IВ1_a" + "IВ2_a" + 90

```

Как видим, для параметра Expr вычисление постоянной составляющей (90) выполнено исполняющей системой скрипта и передано системе вычисления сигналов в готовом виде. В то же время конкатенация имен сигналов не выполнялась.

14.13.2 ИЗМЕНЕНИЕ СВОЙСТВ СИГНАЛА: SIG.EDIT

Изменение свойств существующего сигнала.

Параметр	Тип	Описание	Если отсутствует
Signal	String	Имя сигнала	
Name	String	Новое имя сигнала	Не изменяется
Q1	String	см. Sig.Create	Не изменяется
Q2	String	см. Sig.Create	Не изменяется
NonPeriodic	Bool	см. Sig.Create	Не изменяется
Color	Int	см. Sig.Create	Не изменяется
Dashed	Bool	см. Sig.Create	Не изменяется
Invert	Bool	см. Sig.Create	Не изменяется
Noise	Bool	см. Sig.Create	Не изменяется

14.13.3 ДУБЛИРОВАНИЕ СИГНАЛА: SIG.DUP

Дублирование сигнала.

Параметр	Тип	Описание	Если отсутствует
Signal	String	Имя сигнала, который необходимо продублировать	
Name	String	Имя нового сигнала	Генерируется автоматически

14.13.4 СКРЫТИЕ СИГНАЛОВ: SIG.HIDE(...)

Семейство команд предназначенных для скрытия сигналов.

Команда **Sig.Hide** предназначена для скрытия заданного списка сигналов и имеет единственный параметр:

Параметр	Тип	Описание
List	String[]	Список имен скрывааемых сигналов

Прочие команды семейства не имеют параметров и скрывают сигналы, соответствующие различным условиям:

- **Sig.Hide.All** – все сигналы;
- **Sig.Hide.Analog** – все аналоговые (и комплексные) сигналы;
- **Sig.Hide.Binary** – все дискретные сигналы;
- **Sig.Hide.Calc.All** – все вычисленные сигналы;
- **Sig.Hide.Calc.Analog** – все вычисленные аналоговые (и комплексные) сигналы;
- **Sig.Hide.Calc.Binary** – все вычисленные дискретные сигналы;
- **Sig.Hide.Const** – все сигналы, с постоянным значением на протяжении всей осциллограммы;
- **Sig.Hide.Noise** – все сигналы помеченные флажком "Сигнал содержит шум";
- **Sig.Hide.Zero** – все сигналы, со значением 0 на протяжении всей осциллограммы.

14.13.5 ИНВЕРСИЯ СИГНАЛОВ: SIG.INVERT

Инверсия сигналов.

Параметр	Тип	Описание
List	String[]	Список имен инвертируемых сигналов

14.13.6 УДАЛЕНИЕ СИГНАЛОВ: SIG.REMOVE(...)

Семейство команд предназначенных для удаления сигналов.

Команда **Sig.Remove** предназначена для удаления заданного списка сигналов и имеет единственный параметр:

Параметр	Тип	Описание
List	String[]	Список имен удаляемых сигналов

Важно помнить, что удаление сигнала может быть невозможным, если от него зависят какие-то вычисленные сигналы, которые при этом не удаляются вместе с ним. Если сигнал невозможно удалить, команда просто пропускает его.

Прочие команды семейства не имеют параметров и удаляют сигналы, соответствующие различным условиям:

– **Sig.Remove.Calc** – все вычисленные сигналы.

14.13.7 ПОКАЗ СИГНАЛОВ: SIG.SHOW(...)

Семейство команд предназначенных для показа скрытых сигналов.

Команда **Sig.Show** предназначена для показа заданного списка сигналов и имеет единственный параметр:

Параметр	Тип	Описание
List	String[]	Список имен сигналов, которые надо показать

Прочие команды семейства не имеют параметров и показывают сигналы, соответствующие различным условиям:

– **Sig.Show.All** – все сигналы.

14.14 КОМАНДЫ КАТЕГОРИИ TRI

14.14.1 СОЗДАНИЕ ТРЕХФАЗНОЙ ЦЕПИ: TRI.CREATE

Создание трехфазной цепи.

Параметр	Тип	Описание	Если отсутствует
Name	String	Имя цепи	
Phase	Bool	(True) фазная или (False) междуфазная цепь	True
S0	String	Имя сигнала А или АВ	1)
S1	String	Имя сигнала В или ВС	1)
S2	String	Имя сигнала С или СА	1)

1) – Если Phase = TRUE, должны быть указаны все 3 сигнала фаз, если Phase = FALSE, должны быть указаны хотя бы 2 междуфазных сигнала.

14.14.2 ИЗМЕНЕНИЕ ТРЕХФАЗНОЙ ЦЕПИ: **Tri.Edit**

Изменение трехфазной цепи.

Параметр	Тип	Описание	Если отсутствует
Triad	String	Имя изменяемой цепи	
Name	String	Новое имя цепи	Не изменяется
S0	String	см. Tri.Create	Не изменяется
S1	String	см. Tri.Create	Не изменяется
S2	String	см. Tri.Create	Не изменяется

14.14.3 СКРЫТИЕ СИГНАЛОВ ТРЕХФАЗНЫХ ЦЕПЕЙ: **Tri.Hide**

Скрытие всех сигналов указанных трехфазных цепей

Параметр	Тип	Описание
List	String[]	Список имен трехфазных цепей, сигналы которых следует скрыть

14.14.4 УДАЛЕНИЕ ТРЕХФАЗНЫХ ЦЕПЕЙ: **Tri.Remove(...)**

Семейство команд предназначенных для удаления трехфазных цепей.

Команда **Tri.Remove** предназначена для удаления заданного списка трехфазных цепей и имеет единственный параметр:

Параметр	Тип	Описание
List	String[]	Список имен удаляемых трехфазных цепей

Прочие команды семейства не имеют параметров и удаляют цепи, соответствующие различным условиям:

– **Sig.Remove.All** – все трехфазные цепи.

14.14.5 ПОКАЗ СИГНАЛОВ ТРЕХФАЗНЫХ ЦЕПЕЙ: **Tri.Show**

Показ всех сигналов указанных трехфазных цепей

Параметр	Тип	Описание
List	String[]	Список имен трехфазных цепей, сигналы которых следует показать

14.15 КОМАНДЫ КАТЕГОРИИ OSC

14.15.1 НАСТРОЙКА ПАНЕЛИ ИНСТРУМЕНТОВ: OSC.SETUP

Настраивает положения переключающихся элементов управления на панели инструментов осциллограммы.

Параметр	Тип	Изменяет	Если отсутствует
Val	Int	Список: Отображаемое значение аналоговых сигналов	Не изменяется
Zoom	Int	Список: Режим масштабирования аналоговых сигналов	Не изменяется
Sec	Bool	Переключатель: П/В. Если значение False – выбирается П, если True – В.	Не изменяется

Параметры Val и Zoom имеют тип **int** и изменяют индекс выбранного элемента в соответствующих списках. Как следствие, возможные значения каждого из них ограничены длиной списка. Для удобства задания значений в скрипте доступны именованные константы для каждого индекса в обоих списках.

Возможные значения Val:

Константа	Знач.	Описание
Osc.Val.RMS	0	Действующее значение
Osc.Val.Inst	1	Мгновенное значение
Osc.Val.Mean	2	Среднее значение
Osc.Val.H1	3	1-я гармоника
Osc.Val.H2	4	2-я гармоника
Osc.Val.H3	5	3-я гармоника
Osc.Val.H5	6	5-я гармоника

Возможные значения Zoom:

Константа	Знач.	Описание
Osc.Zoom.Ind	0	Индивидуально
Osc.Zoom.Msr	1	По единицам
Osc.Zoom.Tri1	2	Групповой (по единицам)
Osc.Zoom.Tri2	3	Групповой (индивидуально)

14.15.2 ПЕРЕМЕЩЕНИЕ СИГНАЛОВ: **Osc.Move(...)**

Семейство команд предназначенных для перемещения сигналов в панели осциллограммы. Все команды семейства имеют одинаковый список параметров.

Параметр	Тип	Описание
Dst	String	Имя сигнала, относительно которого будут позиционироваться перемещаемые сигналы
List	String[]	Список имен перемещаемых сигналов

Окончание в названии команды определяет принцип размещения сигналов List относительно сигнала Dst:

- **Osc.Move.Before** - в отдельных лентах **над** лентой сигнала Dst;
- **Osc.Move.After** - в отдельных лентах **под** лентой сигнала Dst;
- **Osc.Move.Into.Before** - на ленту сигнала Dst **над** ним;
- **Osc.Move.Into.After** - на ленту сигнала Dst **под** ним.

15 НЮАНСЫ ПРЕОБРАЗОВАНИЯ СТАРЫХ СКРИПТОВ

В главе 14 было сказано о команде *Действия* > *Преобразовать старый скрипт*, в меню редактора скрипта, которая позволяет преобразовать скрипт, написанный для *Waves 3* в скрипт для *Waves 4*.

Данное преобразование всегда генерирует синтаксически корректный скрипт, однако, не всегда его можно выполнить без доработок.

Дело в том, что подсистема расчета сигналов в *Waves 3* делала ряд автоматических преобразований типов сигналов используемых в выражениях (параметр `Expr` команды `Sig.Create`).

В *Waves 4* мы прекратили данную практику, потому что в конечном итоге от нее получилось больше вреда, чем пользы.

Ниже даны все автоматические преобразования, применявшиеся в *Waves 3* с указанием преобразований-обертков, которые позволят проимитировать их в *Waves 4*.

Waves 3 позволял *умножать* и *делить* сигнал типа `ComplexSignal` на сигнал типа `FloatSignal`, при этом последний автоматически приводился к типу `ComplexSignal` с использованием следующего преобразования:

■ `ComplexP(FloatSignal, 0)`

Waves 3 позволял передавать в параметры функций типа `ComplexSignal` сигнал типа `FloatSignal`, при этом последний автоматически приводился к типу `ComplexSignal` с использованием следующего преобразования:

■ `Complex(FloatSignal)`

Waves 3 позволял передавать в параметры функции типа `BinarySignal` сигнал типа `FloatSignal`, при этом последний автоматически приводился к типу `BinarySignal` с использованием следующего преобразования:

■ `Binary(FloatSignal)`

Таким образом, достаточно добавить описанные выше преобразования-обертки в нужных местах, чтобы сделать скрипт работоспособным.

16 КОНВЕРТЕР etc

В составе *Waves* поставляется утилита **etc**, предназначенная для конвертации файлов осциллограмм в форматах ЭКРА в файлы формата Comtrade, с использованием командной строки:

```
etc "<Path>" [-o "<Dir>"] [-r 1991|1999|2013] [-a] [-c] [-e 0|1|2]
```

Параметр	Описание	Если отсутствует
<Path>	Путь к входному файлу осциллограммы в формате ЭКРА	
-o	Каталог (<Dir>), в котором будет сформирован выходной файл в формате Comtrade	1)
-r	Год ревизии стандарта 1991, 1999, 2013	1999
-a	Генерировать DAT секцию в формате ASCII	BINARY
-c ²⁾	Генерировать CFF файл	CFG + DAT
-e ³⁾	Кодировка CFG секции WIN(0), DOS(1), UTF8(2)	0

1) – При отсутствии параметра файл формируется в каталоге входного файла

2) – Игнорируется, если параметр **-r** имеет значение отличное от **2013**

3) – Значение **2** игнорируется, если параметр **-r** имеет значение отличное от **2013**

Выходной файл имеет имя входного файла с добавленным расширением формата Comtrade:

```
etc 390D5021.zfr --> 390D5021.zfr.cfg / 390D5021.zfr.dat
etc 390D5021.zfr -r 2013 -c --> 390D5021.zfr.cff
```

Осциллограммы в форматах ЭКРА могут состоять из нескольких фрагментов. Если все фрагменты имеют стандартные имена и находятся в одном каталоге, то **etc** автоматически соберет полную осциллограмму из них, и лишь затем произведет конвертацию. Поэтому не нужно вызывать **etc** для всех фрагментов по очереди, достаточно сделать это для одного из них (любого).

Запуск **etc** без командной строки, выводит подсказку по параметрам.

Установочная версия **Waves** прописывает свой каталог установки в переменную среды *PATH*, поэтому вызов утилиты **etc** возможен без указания полного пути к ней.

17 РАБОТА ПОД УПРАВЛЕНИЕМ LINUX

Перед началом установки *Waves* на *Linux* необходимо:

– Открыть документацию на ваш дистрибутив *Linux*, потому что часть действий, описанных ниже, выполняется по-разному в зависимости от используемого дистрибутива, и даже в зависимости от версии одного и того же дистрибутива. В случаях, когда выполняемые действия зависят от используемого дистрибутива, мы не будем приводить их в данном документе. Вместо этого мы будем просить вас обратиться к вашей документации или технической поддержке производителя вашего дистрибутива *Linux*.

– Узнать, как в вашем дистрибутиве *Linux* запускается *Терминал*. Терминал *Linux* это программа, которая позволяет вводить команды операционной системе.

17.1 УСТАНОВКА WINE

Установка и использование *Waves* под управлением *Linux* возможны только посредством пакета *Wine*. Данный пакет предназначен для запуска в *Linux* программ, написанных для *Microsoft Windows*. *Wine* имитирует программный интерфейс *Windows API* и окружение *Windows*, поэтому приложения запущенные под его управлением работая в *Linux*, "думают", что они запущены в *Windows*.

Информацию о том, как установить *Wine*, ищите в документации на ваш дистрибутив *Linux*.

Имитация *Windows* в *Wine* не всеобъемлющая, поэтому в некоторых аспектах, о которых будет отдельно упомянуто ниже, поведение *Waves* будет отличаться от такового в привычном окружении настоящей *Windows*. Поддержка *Windows API* постепенно наращивается и, вероятно, в будущих версиях *Wine* часть из отмеченных ниже особенностей будет устранена. Поэтому очень важно использовать как можно более свежую версию *Wine*, и регулярно обновлять ее.

17.1.1 ОКРУЖЕНИЕ WINDOWS

После установки *Wine* в домашнем каталоге пользователя появляется каталог `$HOME/.wine`, в котором присутствуют каталоги:

– `/drive_c` – здесь воссоздана привычная структура папок системного диска *Windows*.

– `/dosdevices` – здесь собраны символьные ссылки, представляющие собой виртуальные диски, видимые приложениям, работающим под управлением *Wine*. Диск `C:\` - ссылается на упомянутый выше каталог `/drive_c`. Диск `Z:\` - ссылается на корень файловой системы *Linux*. При подключении внешних накопителей по интерфейсу *USB*, здесь также могут появляться дополнительные виртуальные диски, для доступа к содержимому накопителей из приложений, работающих под управлением *Wine*.

17.1.2 ОТКЛЮЧЕНИЕ ДЕКОРИРОВАНИЯ ОКОН

По умолчанию *Wine* декорирует рамку окна приложения, запущенного под его управлением, делая ее визуально идентичной окнам программ, работающих под *Linux*

непосредственно. Это приводит к различным артефактам и неожиданностям в поведении приложений, поэтому данную функцию *Wine* необходимо сразу отключить.

1. Запустите *Редактор реестра Windows*, выполнив в *Терминале* команду:

```
wine regedit
```

2. Создайте узел HKEY_CURRENT_USER\Software\Wine\X11 Driver
3. В созданном узле создайте *строковый* параметр *Decorated* со значением **N**.

17.2 УСТАНОВКА/РАСПАКОВКА WAVES

Запустите установочный файл/портативный архив *Waves* из файлового менеджера вашего дистрибутива *Linux* и произведите установку/распаковку привычным образом.

17.2.1 ЯРЛЫК ЗАПУСКА WAVES

В *Windows* вы привыкли к тому, что в процессе установки *Waves* в меню **Пуск** создается ярлык быстрого запуска приложения. Несмотря на то, что практически во всех современных дистрибутивах *Linux* есть нечто подобное меню **Пуск Windows**, ярлык для быстрого запуска в процессе установки *Waves* в нем не создается и его, если нужно, придется создавать вручную.

Информацию о том, как создавать ярлыки ищите в документации на ваш дистрибутив *Linux*.

17.3 ОТЛИЧИЯ В ПОВЕДЕНИИ

В целом, с точки зрения пользователя, *Waves* в *Linux* сохраняет полную функциональность за одним исключением. Не работает стыковка плавающих инструментов анализа в основное окно, вследствие отсутствия поддержки необходимых для этого механизмов в *Wine*.