



Das Ganze sehen



ibaPDA

Редактор выражений

Руководство часть 4

Версия 7.0.0

Системы измерения для
промышленности и энергетики

Производитель

iba AG
Königswarterstr. 44
90762 Фюрт
Германия

Контактные данные

Центральный офис +49 911 97282-0
Факс +49 911 97282-33
Служба поддержки +49 911 97282-14
Технический отдел +49 911 97282-13
E-Mail iba@iba-ag.com
Веб-страница www.iba-ag.com

Распространение и воспроизведение данного документа, а также использование и передача его содержания без согласия автора запрещены. Нарушение авторских прав преследуется по закону.

© iba AG 2019, все права защищены.

Содержание данной публикации было проверено на предмет соответствия описанному аппаратному и программному обеспечению. Отклонения, однако, не могут быть исключены, поэтому гарантия на полное совпадение не предоставляется. Информация, содержащаяся в данной публикации, регулярно актуализируется. Необходимые исправления содержатся в последующих изданиях или могут быть загружены из Интернета.

Актуальную версию можно всегда найти на нашей веб-странице www.iba-ag.com

Версия	Дата	Ревизия - глава / страница	Автор	Версия ПО
7.0.0	10/2019	Überarbeitete Ausgabe gem. ibaPDA-V7	rm	7.0.0

Windows® является маркой и зарегистрированной торговой маркой корпорации Microsoft. Другие упомянутые в настоящем руководстве названия продуктов и компаний могут являться зарегистрированными торговыми марками и принадлежать соответствующим лицам.

Содержание

1	О данном руководстве	7
1.1	Целевая аудитория и базовые знания	7
1.2	Условные обозначения.....	7
1.3	Используемые символы	8
1.4	Структура документации.....	9
2	Редактор выражений (виртуальные сигналы)	11
2.1	Логические функции.....	11
2.1.1	Функции сравнения >, >=, <, <=, <>, =	11
2.1.2	Булевы функции	12
2.1.3	ЛОЖЬ.....	12
2.1.4	If (Если).....	13
2.1.5	OneShot.....	14
2.1.6	SetReset.....	16
2.1.7	TOF.....	17
2.1.8	TON.....	18
2.1.9	TP	19
2.1.10	ИСТИНА.....	19
2.2	Математические функции.....	20
2.2.1	Основные арифметические операции +, -, *, /.....	20
2.2.2	Abs	21
2.2.3	Ceiling	22
2.2.4	Diff	23
2.2.5	Exp	24
2.2.6	Floor.....	24
2.2.7	Int	25
2.2.8	Log	28
2.2.9	Log10	28
2.2.10	Mod	29
2.2.11	Pow	30
2.2.12	Round	31
2.2.13	Sqrt	32

2.2.14	Truncate	33
2.3	Тригонометрические функции	34
2.4	Статистические функции	36
2.4.1	Avg	36
2.4.2	AvgInTime	36
2.4.3	KurtosisInTime	38
2.4.4	MAvg	39
2.4.5	Max	40
2.4.6	Max2	42
2.4.7	MaxInTime	43
2.4.8	Min	45
2.4.9	Min2	46
2.4.10	MinInTime	47
2.4.11	MKurtosis	49
2.4.12	MMax	50
2.4.13	MMin	51
2.4.14	MSkewness	52
2.4.15	SkewnessInTime	52
2.4.16	StddevInTime	53
2.5	Триггерные функции	55
2.5.1	PeriodicTrigger	55
2.5.2	TriggerChangeRate	56
2.5.3	TriggerConstant	57
2.5.4	TriggerEdge	59
2.5.5	TriggerLevel	60
2.5.6	TriggerHarmonicLevel	62
2.6	Текстовые функции	65
2.6.1	CharValue	65
2.6.2	ConvertFromText	66
2.6.3	CountText	68
2.6.4	CompareText	69
2.6.5	FindText	70

2.7	Различные функции	72
2.7.1	Count	72
2.7.2	Delay	74
2.7.3	DelayLengthL	76
2.7.4	DelayLengthV.....	77
2.7.5	DWORD.....	78
2.7.6	Eff.....	79
2.7.7	ElapsedTime	80
2.7.8	ExecuteCommand.....	80
2.7.9	ExtendPulse	83
2.7.10	GenerateSignal	84
2.7.11	GetFloatBit.....	85
2.7.12	GetIntBit	87
2.7.13	GetSystemTime.....	89
2.7.14	LimitAlarm	90
2.7.15	PulseFreq	92
2.7.16	SampleAndHold	92
2.7.17	Sign	93
2.7.18	T	94
2.7.19	VarDelay.....	95
2.7.20	WindowAlarm.....	97
2.8	Функции диагностики.....	99
2.8.1	CameraStatus.....	99
2.8.2	DataStoreInfo.....	100
2.8.3	DataStoreInfoHD.....	101
2.8.4	DongleInfo	102
2.8.5	FobDLinkStatus	103
2.8.6	FobFastLinkStatus	103
2.8.7	FobFlexDeviceStatus.....	104
2.8.8	FobFLinkStatus.....	104
2.8.9	FobMLinkStatus	105
2.8.10	FobPlusControlLinkStatus	106

2.8.11	FobSDLinkStatus, FobSDexpLinkStatus	107
2.8.12	FobTDCLinkStatus, FobTDCexpLinkStatus.....	107
2.8.13	ICPSensorStatus.....	108
2.8.14	InterruptCycleTime	108
2.8.15	InterruptTime	108
2.8.16	IsMeasuring	109
2.8.17	MultiStationStatus.....	110
2.8.18	PerformanceCounter	110
2.8.19	Ping.....	113
2.8.20	TimeSinceLastSync.....	114
2.8.21	TimeSyncStatus.....	115
2.9	Функция фильтра	116
2.9.1	BP.....	116
2.9.2	HP	116
2.9.3	LP	117
2.9.4	EnvelopeSpectral	118
2.9.5	Preprocess	119
2.10	Функции запоминания	120
2.11	Плагины	120
3	Техподдержка и контакты.....	121

1 О данном руководстве

Данное руководство описывает функцию и применение программного обеспечения *ibaPDA*.

1.1 Целевая аудитория и базовые знания

Данная документация предназначено для квалифицированных специалистов по работе с электрическими и электронными модулями, которые обладают необходимыми знаниями в области коммуникационных и измерительных технологий. Такими специалистами считаются лица, которые на основании своей профессиональной подготовки, специальных знаний и опыта, а также знаний соответствующих предписаний могут оценить возможные последствия и риски.

1.2 Условные обозначения

В данном руководстве используются следующие условные обозначения:

Действие	Условное обозначение
Команды меню	Меню <i>Функциональная схема</i>
Вызов команды меню	"Шаг 1 - шаг 2 - шаг 3 - шаг x" Пример: Выбрать меню <i>Функциональная схема</i> - <i>Добавить</i> - <i>Новый функциональный блок</i>
Клавиши клавиатуры	<Название клавиши> Пример: <Alt>; <F1>
Одновременное нажатие клавиш	<Название клавиши> + <Название клавиши> Пример: <Alt> + <Strg>
Графические клавиши (кнопки)	<Название клавиши> Пример: <OK>; <Отмена>
Имя файла, путь	"Имя файла", "Путь" Пример: „Test.doc“

1.3 Используемые символы

В данной документации используются символы техники безопасности, которые имеют следующее значение:

Опасно!



Несоблюдение данного предписания по технике безопасности грозит летальным исходом или тяжкими телесными повреждениями!

- Соблюдайте технику безопасности

Внимание!



Несоблюдение данного предписания по технике безопасности может привести к летальному исходу или тяжким телесным повреждениям!

- Соблюдайте технику безопасности.

Осторожно!



Несоблюдение данного предписания по технике безопасности может привести к травмам или причинить материальный ущерб!

- Соблюдайте технику безопасности.

Важно



Особые указания, например, исключения из правил и т.д.

Совет



Советы, наглядные примеры и маленькие хитрости, позволяющие облегчить работу.

Дополнительная документация



Ссылка на дополнительную документацию или специальную литературу.

1.4 Структура документации

Данная документация полностью описывает функции системы *ibaPDA*. Она создана как руководство для введения в эксплуатацию, а также как справочный документ. Части и разделы расположены в последовательности, соответствующей порядку конфигурирования системы.

Дополнительно к данной документации для получения последней информации об обновленной версии программы Вы можете обратиться к журналу версий в главном меню Справка - Изменения (Файл [versions.htm](#)). В данном файле рядом с перечисленными устраненными программными ошибками есть краткие ссылки на изменения в системе.

Кроме того, с каждым обновлением ПО, содержащим существенные новые свойства, выходит специальная документация «Новые свойства...» с подробным описанием новых функций.

Версия программного обеспечения, к которой относится соответствующий текст данной документации, приведен соответственно в таблице ревизий на странице 2.

Документация системы *ibaPDA*-(PDF и печатная версия) поделена на семь отдельных частей. Каждая часть имеет свою собственную нумерацию глав и страниц и актуализируется независимо.

Часть 1	Введение и установка	Общие указания, лицензионная политика, аддоны Установка и запуск программы Пользовательский интерфейс, системная архитектура, клиент-сервер управление пользователями, печать
Часть 2	Диспетчер ввода/вывода	Основная информация по диспетчеру вв/выв, общие настройки группы и векторные сигналы, текстовые сигналы, выводы, файлы конфигурации
Часть 3	Интерфейсы и модули	Интерфейсы для сбора измеренных данных Стандартные интерфейсы, <i>ibaFOB</i> , интерфейсы на базе Ethernet и т.д. Для получения информации об интерфейсах, для которых есть отдельные руководства, обратитесь, пожалуйста, к ним.
Часть 4	Редактор выражений	Все функции для расчета виртуальных сигналов
Часть 5	Запись данных	Виды записи данных, профилей записи, выбора сигналов
Часть 6	Визуализация данных	Все режимы отображения данных в режиме реального времени, управление ими и настройка

Часть 7	Приложение	Различные дополнения, списки ошибок и т.д.
----------------	------------	--

2 Редактор выражений (виртуальные сигналы)

При помощи математических функций и булевых логик могут образовываться «виртуальные сигналы». Данные виртуальные сигналы могут записываться как другие измеренные сигналы и/или использоваться для простой реализации комплексных триггерных условий. При помощи виртуальных сигналов Вы можете проводить уже при измерении вычисления, например, сложение и вычитание, проверку на нарушение допусков и т.д. Или Вы генерируете реверенсные сигналы и признаки, чтобы проводить дополнительные сравнения при анализе.

Важно



Указания по условным обозначениям параметров функции (аргументы) во всплывающих подсказках и текстах-справках:

Если в списке параметров функции параметру присвоено значение, то оно является предварительно заданным значением, которое принимается, если параметр пропущен. Пропущены могут быть только параметры, имеющие предварительно заданное значение.

В функции `GenerateSignal ()`, например, могут быть пропущены аргументы 2, 3 и 4. `GenerateSignal (3,10,1,1)` равняется `GenerateSignal (3)`.

2.1 Логические функции

2.1.1 Функции сравнения `>`, `>=`, `<`, `<=`, `<>`, `=`

Операции сравнения `>` (больше), `>=` (больше / равно), `<` (меньше), `<=` (меньше / равно), `<>` (неравно) и `=` (равно) позволяют сравнивать значения двух выражений (операндов) друг с другом. Операции возвращают как результат соответственно булево значение ИСТИНА или ЛОЖЬ. В качестве операндов могут быть внесены оригинальные сигналы, расчетные выражения или просто постоянные значения. Результат может изображаться и обрабатываться в качестве нового выражения как сигнал. Таким образом легко образуются новые сигналы, которые в свою очередь могут снова использоваться как условия для других функций.

Важно



Если точка пересечения двух кривых находится между двумя точками измерения, то результат сравнения последних двух измерений сохраняется до следующей точки измерения. Т.е. любые изменения с ИСТИНЫ на ЛОЖЬ (или наоборот) всегда вводятся во вкладке точек измерений. Соединительная линия между двумя точками измерения при изображении аналоговых значений является только графическим приближением

2.1.2 Булевы функции

например ('Expression1') AND ('Expression2')

AND	Логическое И
OR	Логическое ИЛИ
XOR	Логическое исключающее ИЛИ
NOT	Логическое НЕТ, отрицание

Булевы функции

Описание

Булевы функции AND (логическое И), OR (логическое ИЛИ), NOT (логическое НЕТ, отрицание) или XOR (логическое исключающее ИЛИ) позволяют связывать бинарные выражения, например, цифровые сигналы. В соответствии с правилами булевых функций, функции возвращают как результат соответственно значение ИСТИНА и ЛОЖЬ. Цифровые сигналы, вычисляемые (бинарные) выражения и числовые значения 0 или 1 могут вводиться как параметры.

Результат может изображаться и обрабатываться в качестве нового выражения как сигнал. Таким образом легко образуются новые сигналы, которые в свою очередь могут снова использоваться как условия для других функций.

AND			OR			XOR			NOT	
A	B	f (A,B)	A	B	f (A,B)	A	B	f (A,B)	A	f (A)
0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	1	1	0	1	1	0
0	1	0	0	1	1	0	1	1		
1	1	1	1	1	1	1	1	0		

Table 1: Логические функции, таблицы истинности

Пример

Графическое представление булевых функций

Решение

Рис. 1: Обзор булевых функций

2.1.3 ЛОЖЬ

FALSE ()

Описание

Возвращает логическое выражение ЛОЖЬ или ноль (0).

2.1.4 If (Если)

```
If('Condition', 'Expression1', 'Expression2')
```

Аргументы

'Condition'	Условие в виде действия с булевыми результатами ИСТИНА или ЛОЖЬ	
'Expression1'	Действие выполняется, если 'Condition' ИСТИНА	
'Expression2'	Действие выполняется, если 'Condition' ЛОЖЬ	

Описание

Функция If (ЕСЛИ) служит для условного выполнения дополнительных вычислений. В зависимости от булева результата условия ('Condition'), которое само может быть действием, при результате ИСТИНА выполняется действие 'Expression1', при результате ЛОЖЬ соответственно действие 'Expression2'.

Таким образом можно проводить различные вычисления с управлением процессом. Функция может использоваться для иерархического построения с реализацией дополнительных разветвлений.

Совет



Если для 'Condition' введено только значение, то как условие считывается, больше (ИСТИНА) или меньше (ЛОЖЬ) ли значение, чем 0,5.

Пример

Распознать, когда измеренный сигнал находится над установленной границей.

Решение

Предел сформулирован в 'Condition' как условие с логическими операндами. Если 'Condition' ИСТИНА, то возвращается значение 4, при ЛОЖЬ значение 1.

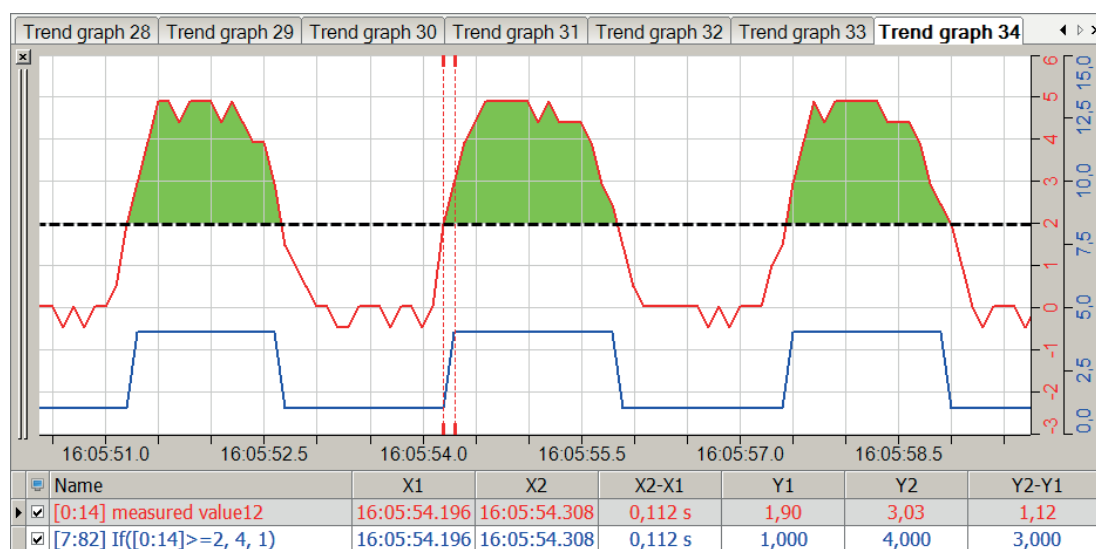


Рис. 2: Над пунктирной линией (зеленая): условие выполнено; под линией: условие не выполнено

2.1.5 OneShot

`OneShot('Expression')`

Описание

Данная функция возвращает результат ИСТИНА, если текущее измеренное значение 'Expression' не равно предыдущему. Она возвращает результат ЛОЖЬ, если текущее измеренное значение равно предыдущему.

Пример 1

Обнаружение изменение значений

Постановка задачи

Для действительного графика сигнала должны отображаться изменения значения.

Решение

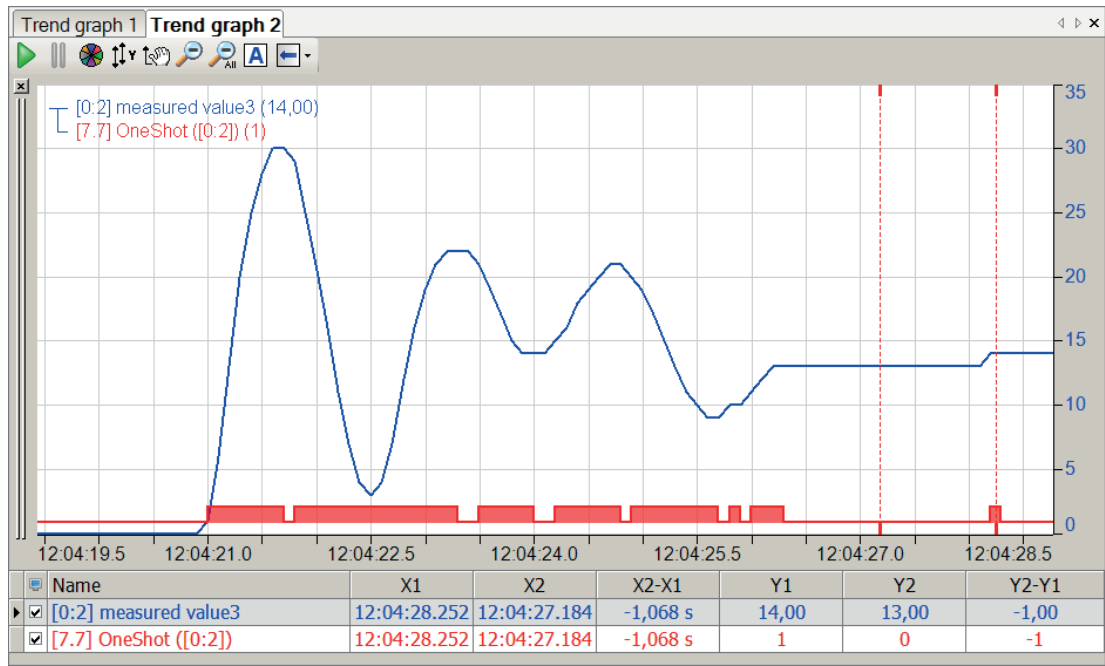


Рис. 3: Синий график: исходный сигнал; красная диаграмма: Диапазоны с изменениями значений сигналов

Пример 2

Обнаружение фронтов

Постановка задачи

Требуется сформировать положительные фронты цифрового сигнала.

Решение

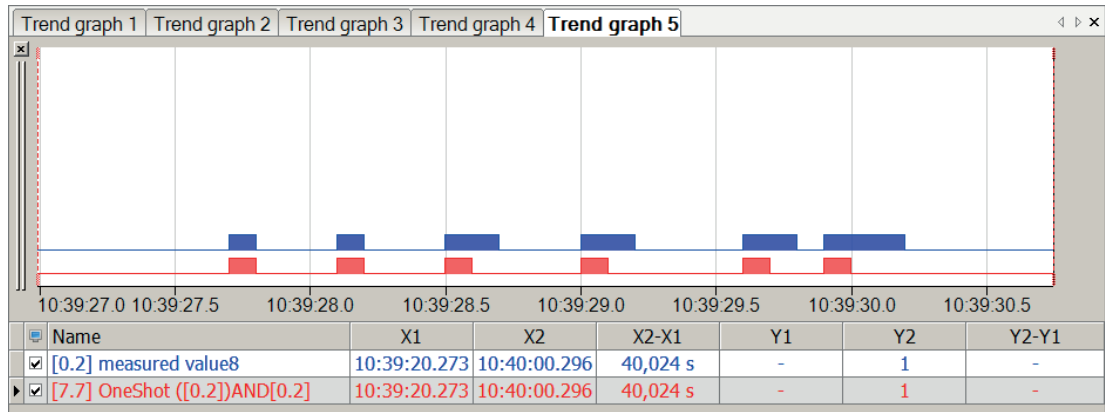


Рис. 4: Синяя диаграмма: исходный сигнал; красная диаграмма: положительные фронты

2.1.6 SetReset

```
SetReset('Set', 'Reset', 'SetDominant=1*')
```

Аргументы

'Set'	Передний фронт устанавливает функцию на ИСТИНА	
'Reset'	Передний фронт устанавливает функцию на ЛОЖЬ	
	Опциональный параметр (по умолчанию = 1), который управляет тем, какой входной аргумент является доминантным, если оба аргумента получают одновременно положительный фронт.	
	'Setdominant' = 1	Set имеет преимущество по отношению к Reset
	'Setdominant' = 0	Reset имеет преимущество по отношению к Set

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция используется для управления цифровым результатом (ИСТИНА/ЛОЖЬ) при помощи передних фронтов (переход с 0 на 1) аргументов 'Set' и 'Reset'.

Передний фронт операнда 'Set' возвращает статическое ИСТИНА как результат. Передний фронт операнда „Reset“ возвращает результат на ЛОЖЬ. Аргумент 'SetDominant' является опциональным и определяет приоритетность 'Set' или 'Reset'.

Совет



При аналоговом сигнале превышение значения 0,5 соответствует переднему фронту.

Пример

Данная функция может использоваться для активирования условных вычислений с одним сигналом и деактивирования с другим сигналом (например, в сочетании с If-функцией).

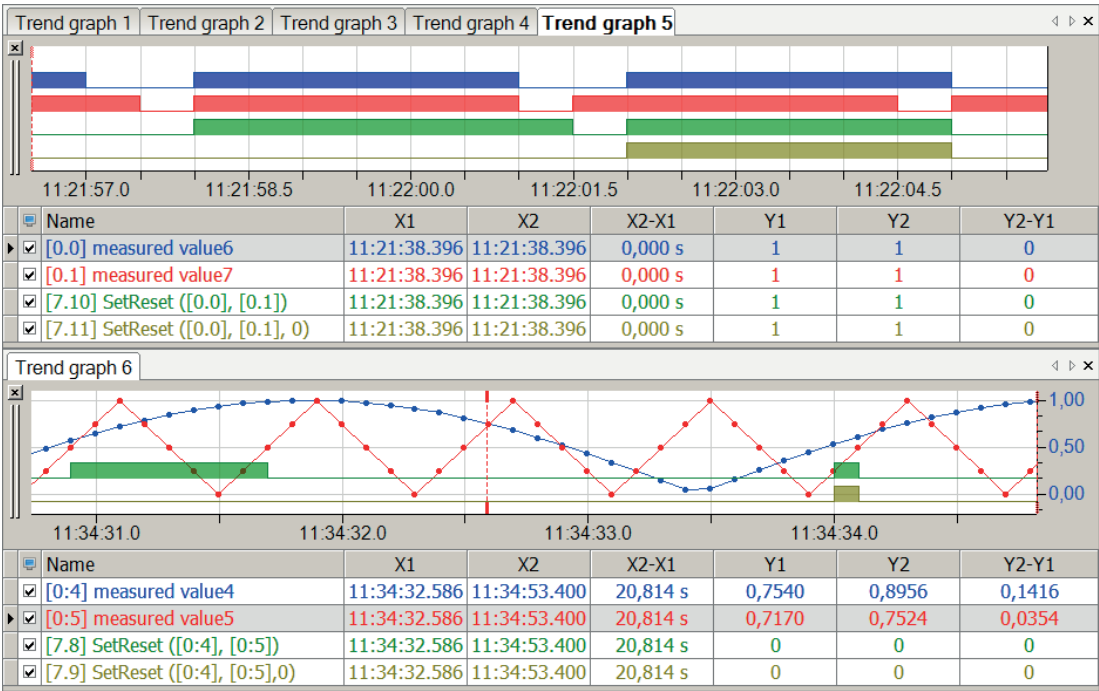


Рис. 5: Функция SetReset с цифровым и аналоговыми сигналами; для одновременных фронтов устанавливается параметр, заданный в 'SetDominant'; при смещенных по времени фронтах параметр 'SetDominant' не действует

2.1.7 TOF

TOF('IN', 'PT*')

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Задержка выключения. Выключение выхода осуществляется 'pt' секунд после выключения входа 'in'.

Пример

Задержка выключения на одну секунду

Решение

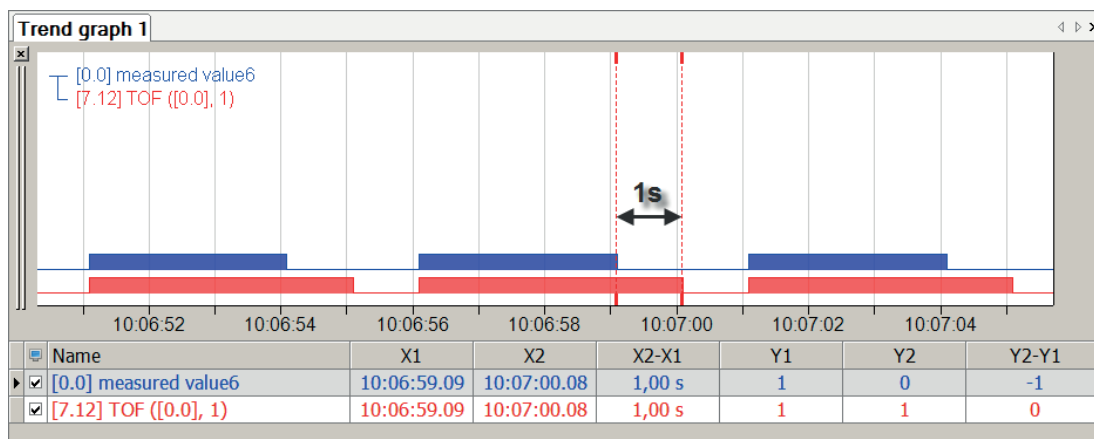


Рис. 6: Синяя диаграмма: измеренное значение; красная диаграмма: Выходное значение с выходом с задержкой на одну секунду

2.1.8 TON

TON ('IN', 'PT*')

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Задержка включения. Включение выхода осуществляется 'pt' секунд после включения входа 'in'.

Пример

Задержка включения на одну секунду

Решение

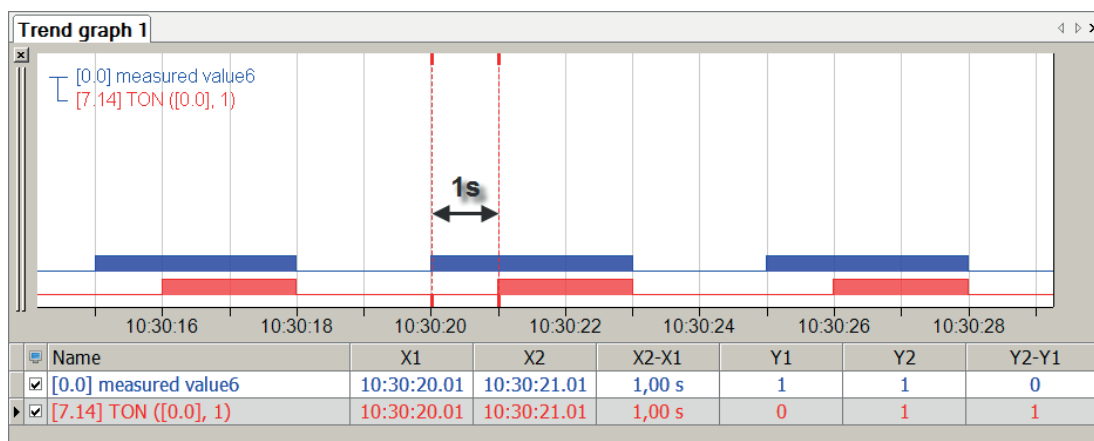


Рис. 7: Синяя диаграмма: измеренное значение; красная диаграмма: Выходное значение с выходом с задержкой в одну секунду

2.1.9 TP

TP('in','pt*')

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Функция импульса. Выход переключается на 'PT' секунд после переднего фронта на вход 'IN'.

Совет



Дополнительный передний фронт во время выходного импульса не продлит переключение и не перезапускает импульс.

Пример

Генерирование 0,5-секундного импульса из случайного сигнала

Решение

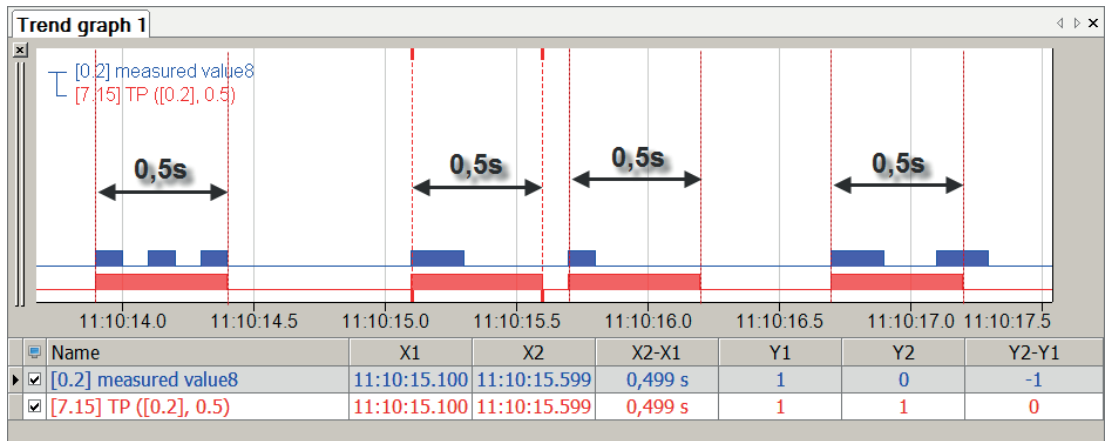


Рис. 8: Синяя диаграмма: измеренное значение; красная диаграмма: Выходное значение в 0,5-секундных импульсах

2.1.10 ИСТИНА

TRUE ()

Описание

Возвращает логическое выражение ИСТИНА или 1.

2.2 Математические функции

2.2.1 Основные арифметические операции +, -, *, /

```
('Expression1') + ('Expression2')
```

Описание

Все сигналы и выражения могут обрабатываться при помощи основных арифметических действий (сложение, вычитание, умножение и деление). Если в качестве операндов используются цифровые сигналы или выражения с основными арифметическими действиями, то программа переводит значения ИСТИНА как 1.0 и ЛОЖЬ как 0.0. Результатом основного арифметического действия является аналоговое значение.

Пример

Графическое изображение основных арифметических действий

Решение

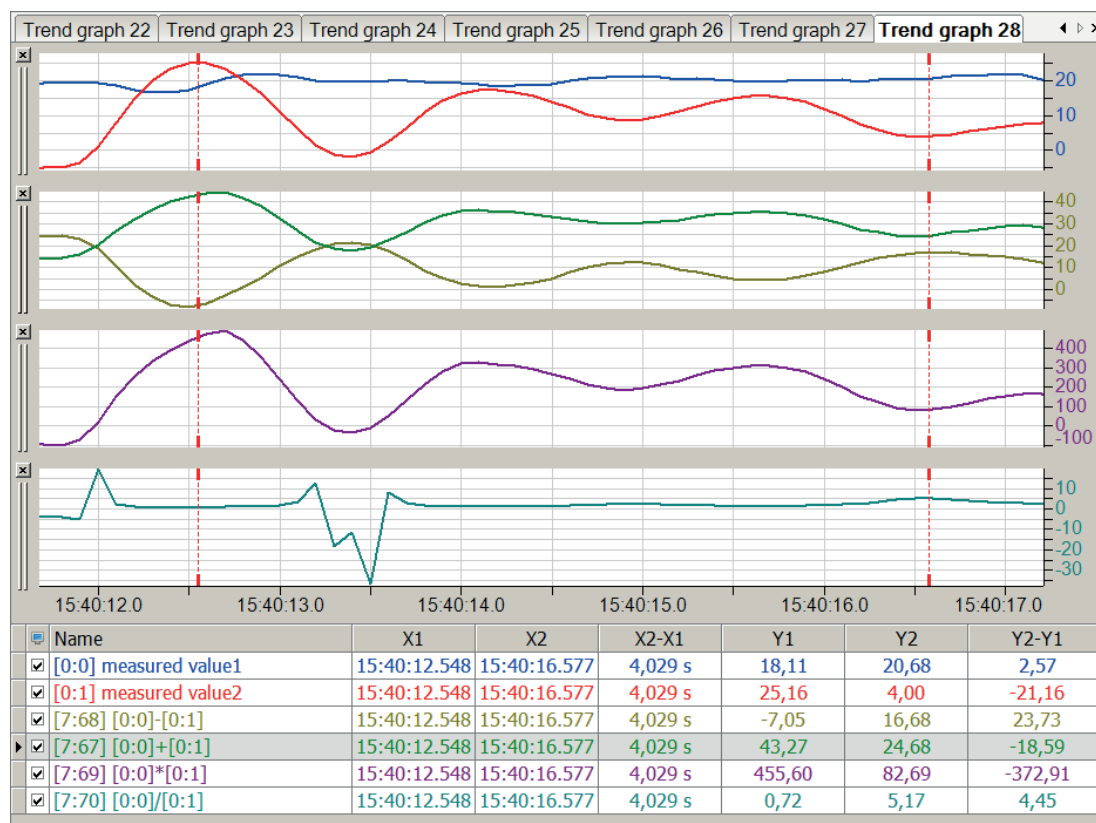


Рис. 9: Синий график: Измеренное значение 1; красный график: Измеренное значение 2; зеленый график: Сложение; желтый график: Вычитание; розовый график: Умножение; голубой график: Деление;

2.2.2 Abs

`Abs('Expression')`

Описание

Функция абсолютного значения возвращает абсолютное значение ($= |\text{значение}|$) аргумента 'Expression'.

Пример

Релевантно только абсолютное значение измеренного сигнала

Решение



Рис. 10: Синий график: исходный сигнал; красный график: Абсолютные значения сигнала

Совет



Интерполированные значения при смене знака между двумя точками измерения могут отличаться по величине.

2.2.3 Ceiling

`Ceiling('Expression')`

Описание

Данная функция возвращает наименьшее целочисленное значение, которое больше или равно 'Expression'.

Пример

Округление в большую сторону значений графика сигналов до целых чисел

Задача

В графике сигналов измеренное значение при каждом измерении необходимо округлить в большую сторону до ближайшего целого значения.

Решение

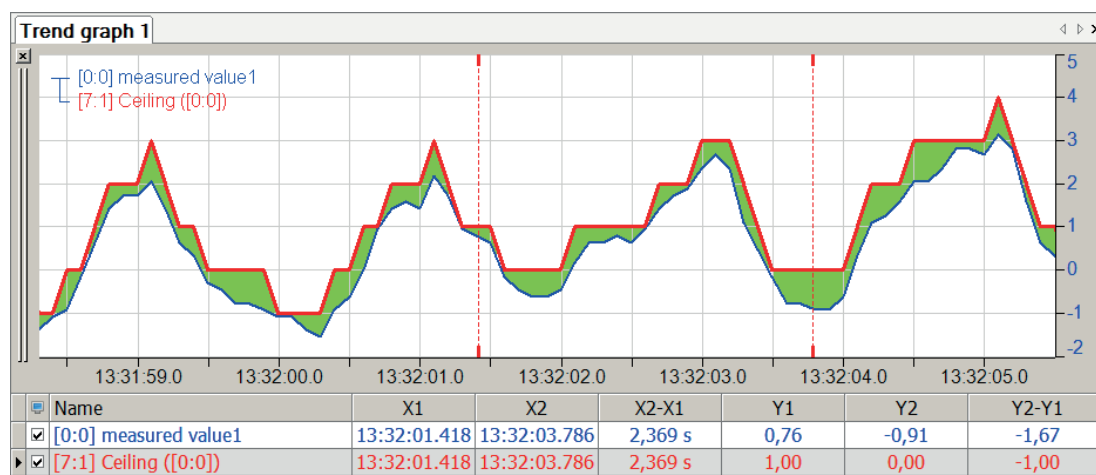


Рис. 11: Синий график: исходный сигнал; красный график: Характеристика сигнала с округленными в большую сторону значениями

2.2.4 Diff

Diff('Expression')

Описание

Данная функция возвращает как результат дифференциал dx/dt 'Expression'.

Пример

Если 'Expression' является сигналом измерения длины, то из него при помощи функции Diff может быть рассчитан график скорости.

Задача

Из измеренного сигнала длины необходимо рассчитать график скорости и из него график ускорения.

Решение

Повторное (итеративное) выполнение функции Diff при использовании значений первой как 'Expression' получают график ускорения.

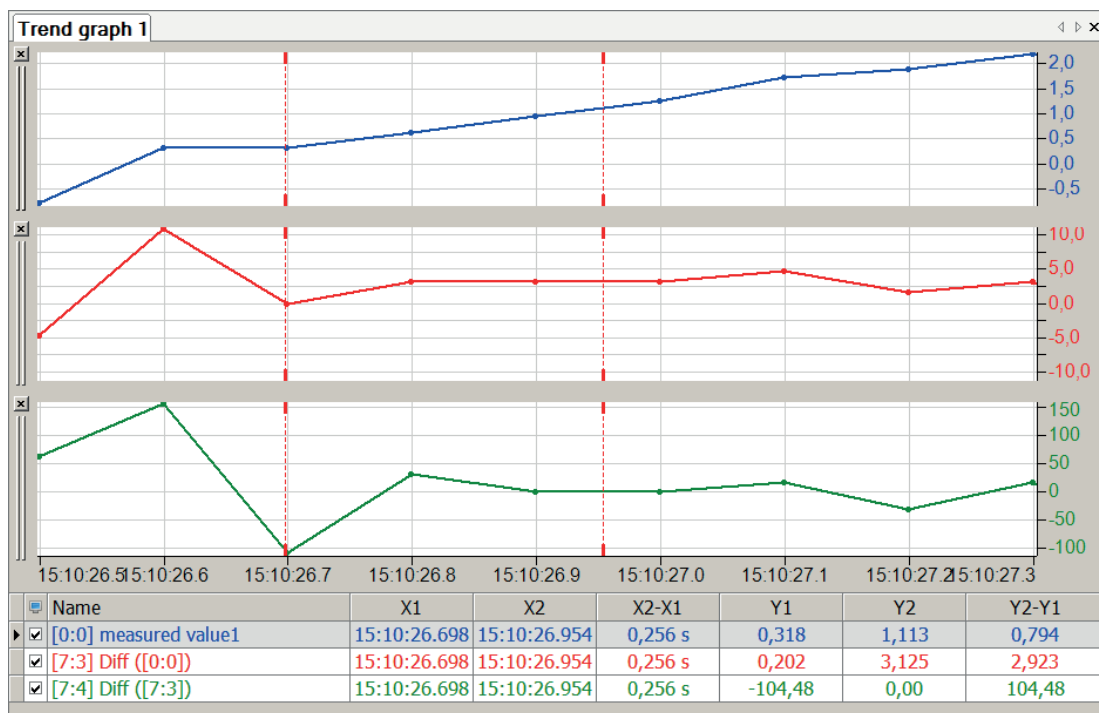


Рис. 12: Синий график: пройденный участок; красный график: расчетный график скорости; зеленый график: расчетный график ускорения

Совет



Если интерес представляет только график ускорения и скорость нерелевантна, то он может быть рассчитан напрямую рекурсивным вызовом функции $Diff(Diff([0:0]))$.

2.2.5 Exp

`Exp('Expression')`

Описание

Данная функция вычисляет результат от $(e)^{\text{'Expression'}}$

Пример

Графическое изображение функции Exp

Решение

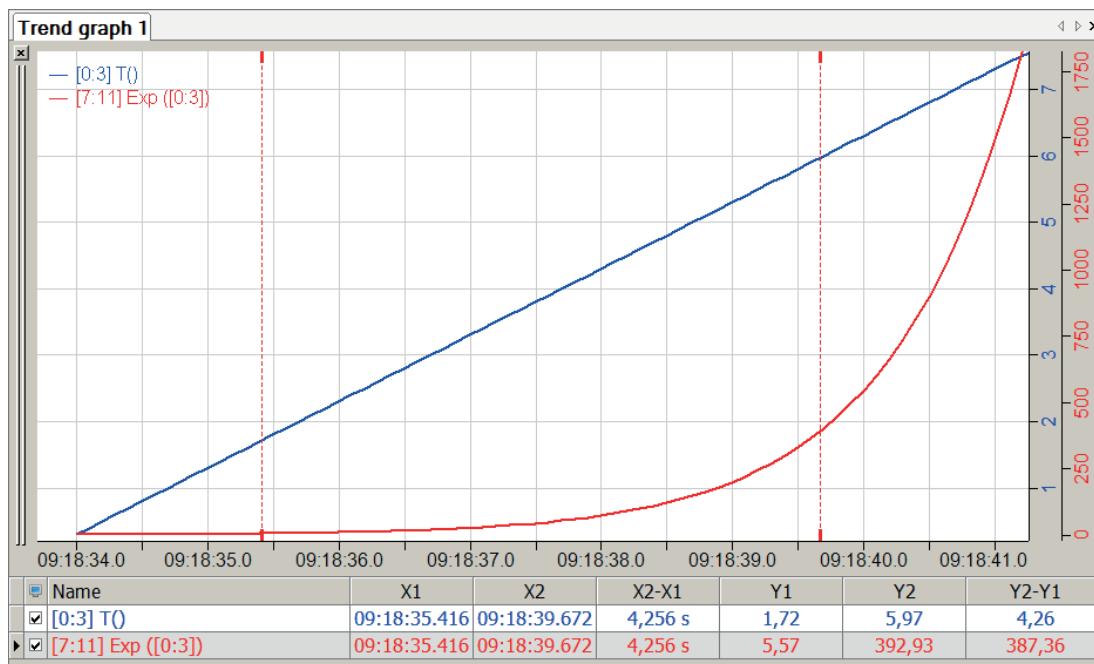


Рис. 13: Синий график: Функция времени; красный график: Экспоненциальная функция времени

2.2.6 Floor

`Floor('Expression')`

Описание

Данная функция возвращает наибольшее целочисленное значение, которое меньше или равно 'Expression'.

Пример

Округление в меньшую сторону значений графика сигналов до целых чисел

Задача

В графике сигналов измеренное значение при каждом измерении необходимо округлить в меньшую сторону до ближайшего целого значения.

Решение

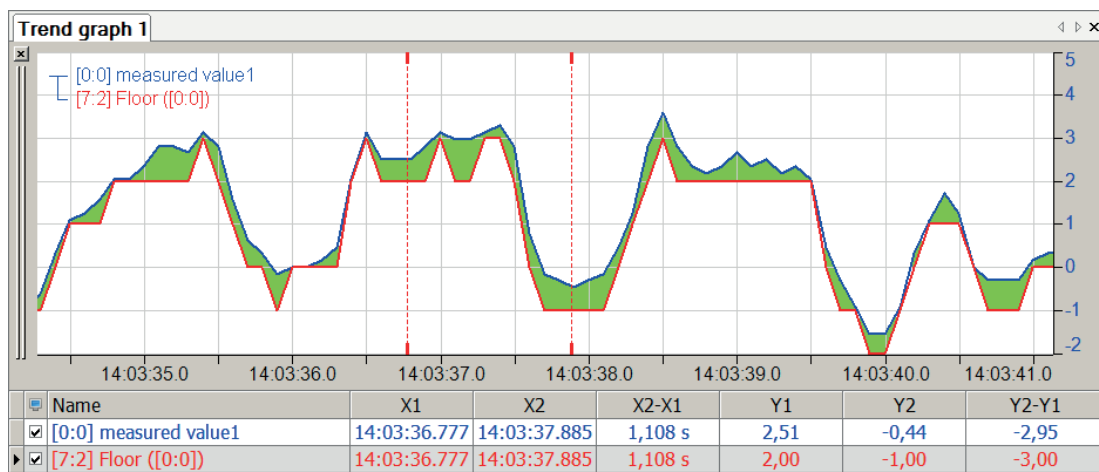


Рис. 14: Синий график: исходный сигнал; красный график: Характеристика сигнала с округленными в меньшую сторону значениями

2.2.7 Int

Int 'Expression', 'Reset'

Аргументы

'Expression'	Измеренное значение	
'Reset'	Опциональный цифровой параметр, который может использоваться для сброса интеграла или подавления процесса интегрирования. 'Reset' может быть сам выражением	
	'Reset' > 0	Интеграл сбрасывается.
	'Reset' = 0	Активируется интегрирование (уставка)

Описание

Данная функция возвращает как результат интеграл ($y * dt$) аргумента 'Expression'. Параметр 'Reset' может использоваться для сброса интеграла или подавления процесса интегрирования, например, для многократного интегрирования одного и того же сигнала при периодических или реверсивных процессах. 'Reset' может также сам быть выражением.

Примеры:

Int([0:0])	Сброс не осуществляется ('Reset' пропущен)
Int([0:0], If(Mod(T(), 20)=0, TRUE(), FALSE()))	Интеграл сбрасывается каждые 20 секунд.
Int([0:0], [3.1])	Например, с [3.1] = If([0:0]>10, 1, 0) Интеграл сбрасывается, если выражение [3.1] возвращает ИСТИНА, т.е. если выражение [0:0] превышает предельное значение 10.

Пример 1

Совет



Данная функция может использоваться в виртуальном модуле с запоминанием. Ее результирующие значения можно получить, остановив или перезапустив измерение.

Если 'Expression' является сигналом ускорения, то при итеративном выполнении функции Int из него можно рассчитать пройденный путь.

Задача

При помощи датчика ускорения необходимо задать скорость, а также пройденный путь.

Решение

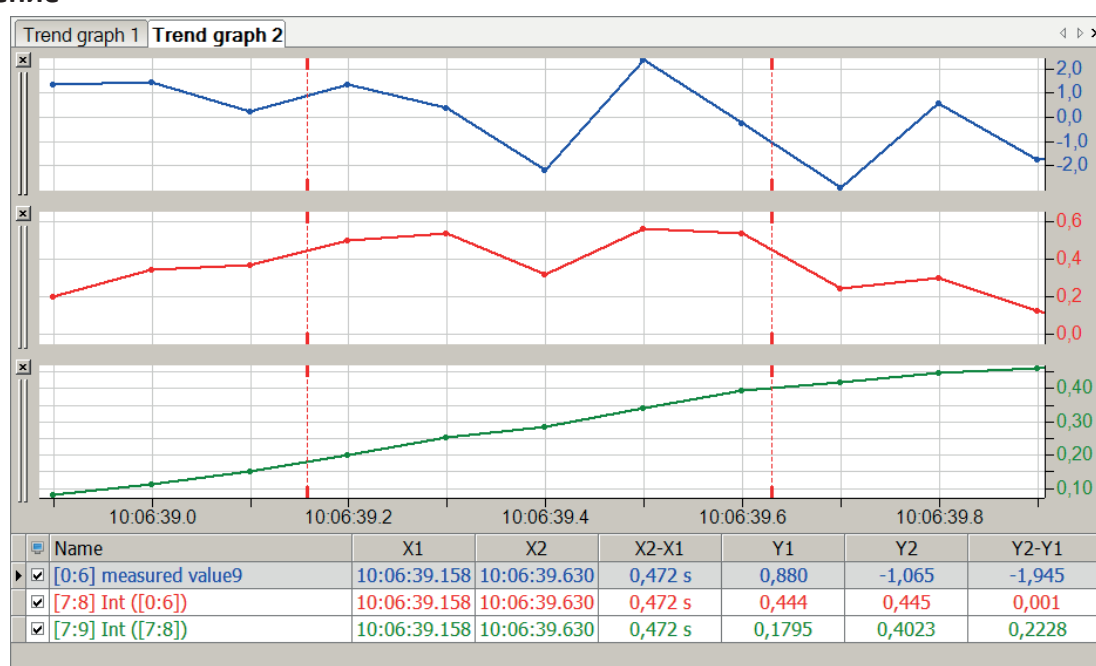


Рис. 15: Синий график: измеренное ускорение; красный график: расчетный график скорости; зеленый график: расчетный пройденный путь

Совет



Если интерес представляет только пройденный путь, то его можно рассчитать при рекурсивном вызове функции `Int(Int([0:6]))`.

Пример 2

Интеграл следует повторно запускать с интервалом в 20 секунд.

Задача

Функция модуля может использоваться для сброса интеграла, т.к. каждые 20 секунд $T() \bmod 20 = 0$

Решение

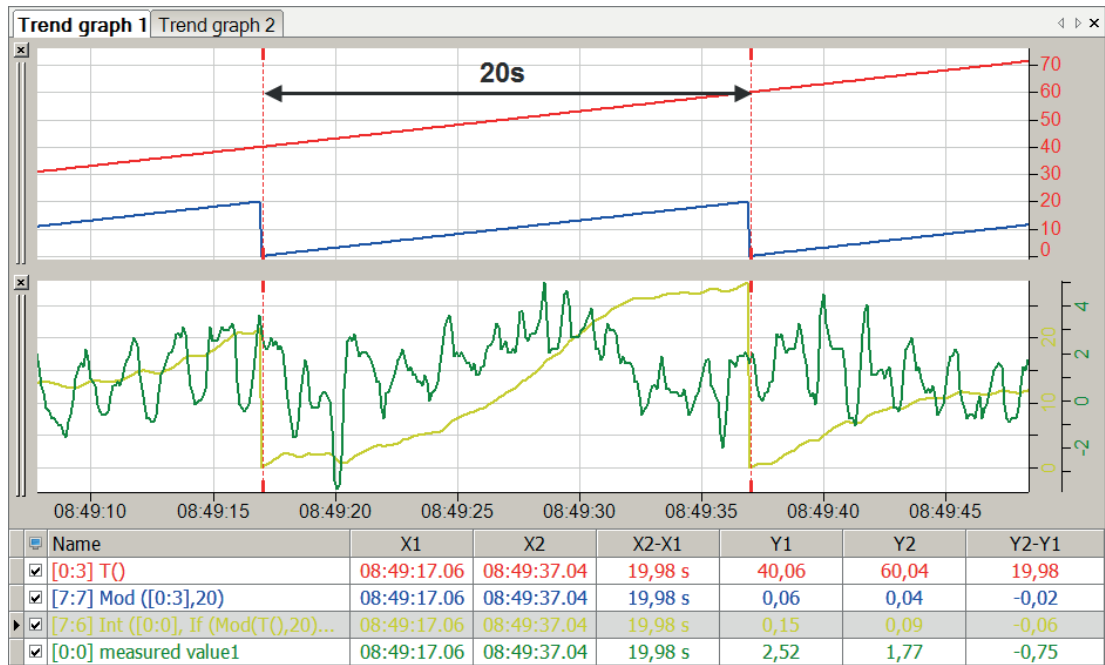


Рис. 16: Красный график: Функция времени; синий график: mod20 функции времени; зеленый: Измеренное значение; желтый: Интеграл измеренного значения со сбросом 20 секунд

2.2.8 Log

`Log('Expression')`

Описание

Данная функция возвращает как результат натуральный логарифм ($\lg x$) от 'Expression'.

Пример

Вычисление известного натурального логарифма

Решение

Логарифм от числа Эйлера e всегда равен 1: $\ln e = 1$

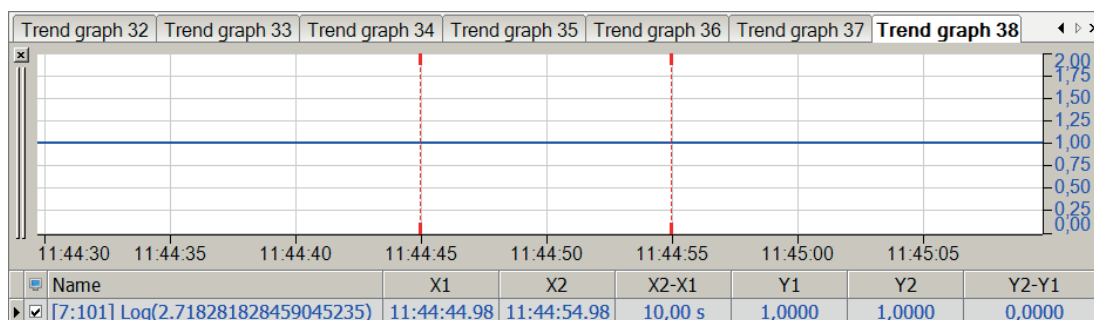


Рис. 17: Натуральный логарифм от числа Эйлера e

Совет



Отрицательные значения для 'Expression' не выводят сообщение об ошибке, но и не выводят результат.

2.2.9 Log10

`Log10('Expression')`

Описание

Данная функция возвращает как результат десятичный логарифм ($\lg x$) выражения 'Expression'.

Пример

Вычисление известных десятичных логарифмов.

Решение

$$\lg 1 = \text{Log10}(1) = 0$$

$$\lg 10 = \text{Log10}(10) = 1$$

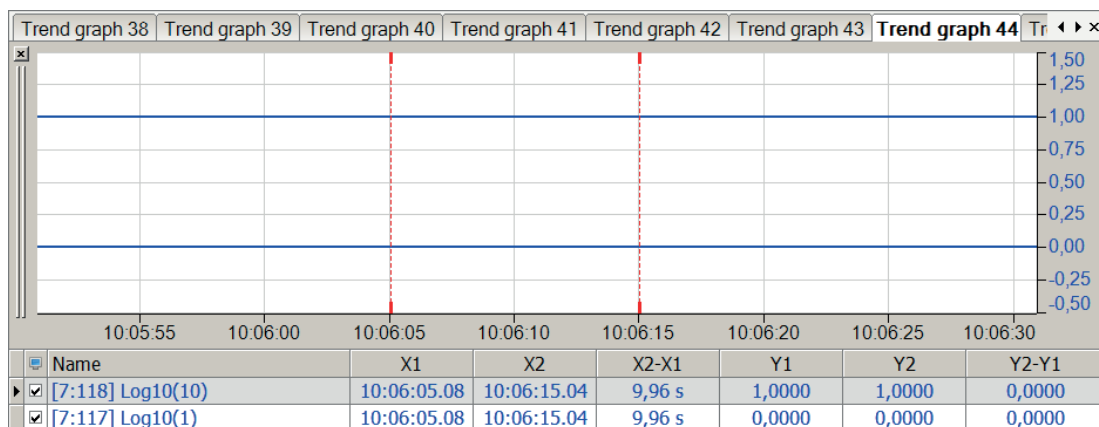


Рис. 18: Десятичный логарифм 1 и 10

Совет

Отрицательные значения для 'Expression' не выводят сообщение об ошибке, но и не выводят результат.

2.2.10 Mod

`Mod('Expression1', 'Expression2')`

Описание

Данная функция возвращает как результат модуль 'Expression1' и 'Expression2'. Функция использует C-функцию fmod, которая позволяет использовать значения с плавающей запятой для 'Expression1' и 'Expression2'.

r по модулю - это остаток от деления выражение1 / выражение2, где обратное равенство: выражение1 = выражение2 * i + r , где i - целое число (Integer).

r по модулю имеет всегда один и тот же знак, что и 'Expression1' и абсолютное значение r всегда меньше, чем абсолютное значение 'Expression2'.

Если 'Expression1' < 'Expression2', то функция возвращает значение 'Expression1' как результат. Остаток может быть обозначен на математическом языке также как «выражение1 по модулю выражение2».

Примеры: $\text{Mod}(7, 3) = 1$; семь разделить на три равно двум, остаток 1.

$\text{Mod}(20.0, 10.5) = 9.5$

Пример

Из функции времени релевантны только секунды, часы и минуты отсекаются.

Решение

С Modulo60 функции времени как остаток остаются только секунды.

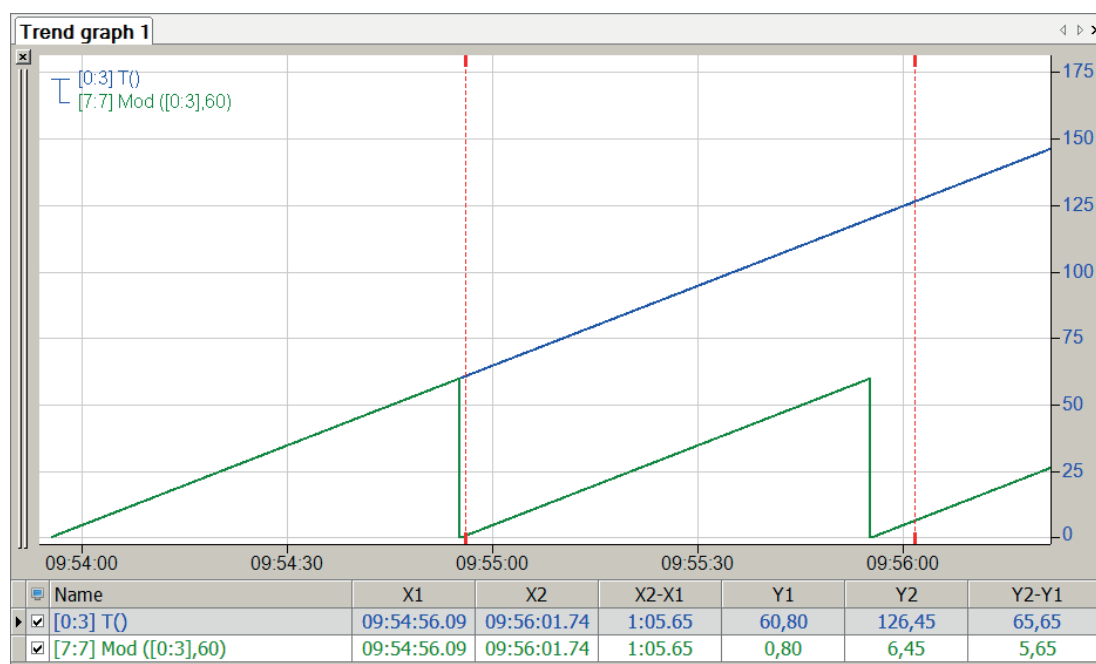


Рис. 19: Синий график: Функция времени; зеленый график: Modulo60 функции времени

2.2.11 Pow

`Pow('Expression1', 'Expression2')`

Аргументы

'Expression1'	Базовое число
'Expression2'	Показатель степени

Описание

Данная функция возводит 'Expression1' (базовое число) в степень 'Expression2' (показатель степени): $(\text{'Expression1'})^{\text{'Expression2'}}$

Пример

Вычисление некоторых степеней

Решение

$$(2)^0 = \text{Pow}(2, 0) = 1$$

$$(2)^{-2} = \text{Pow}(2, -2) = 0,25$$

$$(-2)^2 = \text{Pow}(-2, 2) = 4$$

$$(10)^{\lg 2} = \text{Pow}(10, \lg 2) = 2$$

$$(0)^{-1} = \text{Pow}(0, -1) = +\infty \text{ (бесконечность)}$$

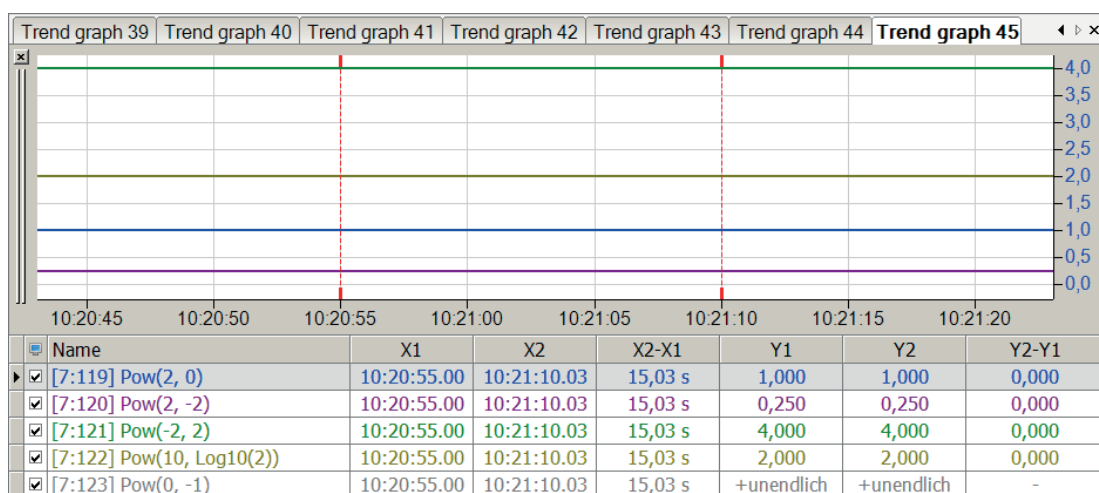


Рис. 20: Вычисление изображенных степеней

Совет

Возведение числа 0 в степень -1, что соответствует делению на 0, возвращает не сообщение об ошибке, а предельное значение $+\infty$ (бесконечность).

2.2.12 Round

Round('Expression')

Описание

Данная функция округляет 'Expression' до следующего целого числа (целочисленного значения) большую или меньшую сторону.

Пример

Округление значений графика сигналов до целых чисел

Задача

В графике сигналов измеренное значение при каждом измерении необходимо округлить до следующего целого значения.

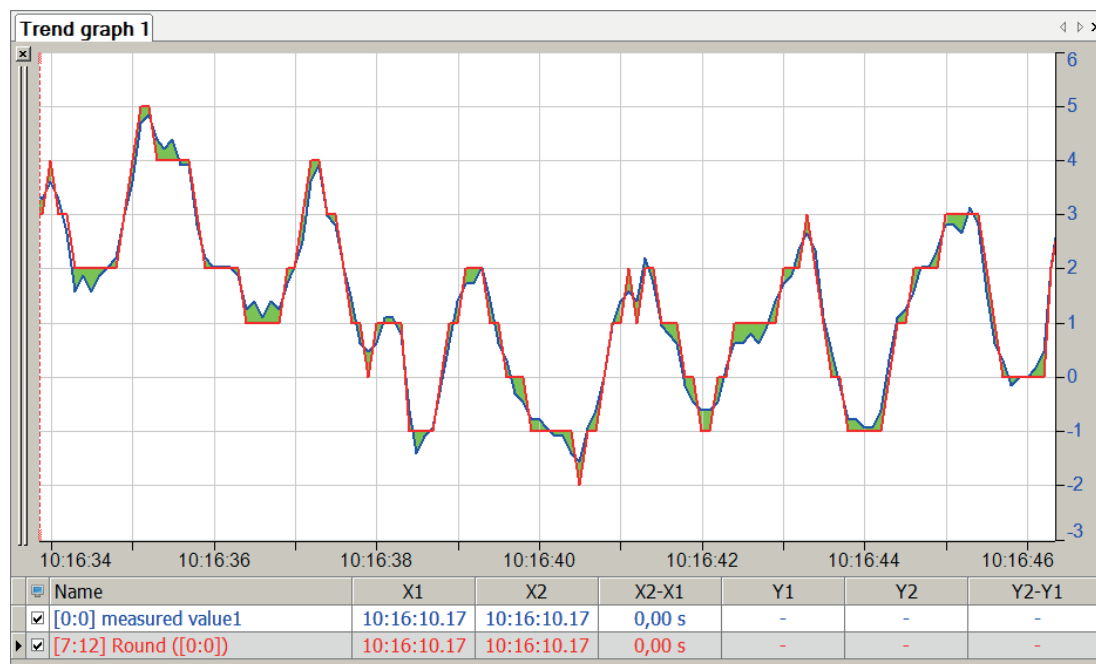
Решение

Рис. 21: Синий график: исходный сигнал; красный график: Характеристика сигнала с округленными значениями

2.2.13 Sqrt

`Sqrt('Expression')`

Описание

Данная функция возвращает значение квадратного корня аргумента 'Expression'.

Пример

Вычисление некоторых известных квадратных корней

Решение

$$\sqrt{4} = \text{Sqrt}(4) = 2$$

$$\sqrt{2} = \text{Sqrt}(2) = 1,41421356...$$

$$\sqrt{-1} = \text{Sqrt}(-1) = i \text{ (необходим комплексный расчет)}$$

Совет

Отрицательные значения для 'Expression' не выводят сообщение об ошибке, но и не дают результат.

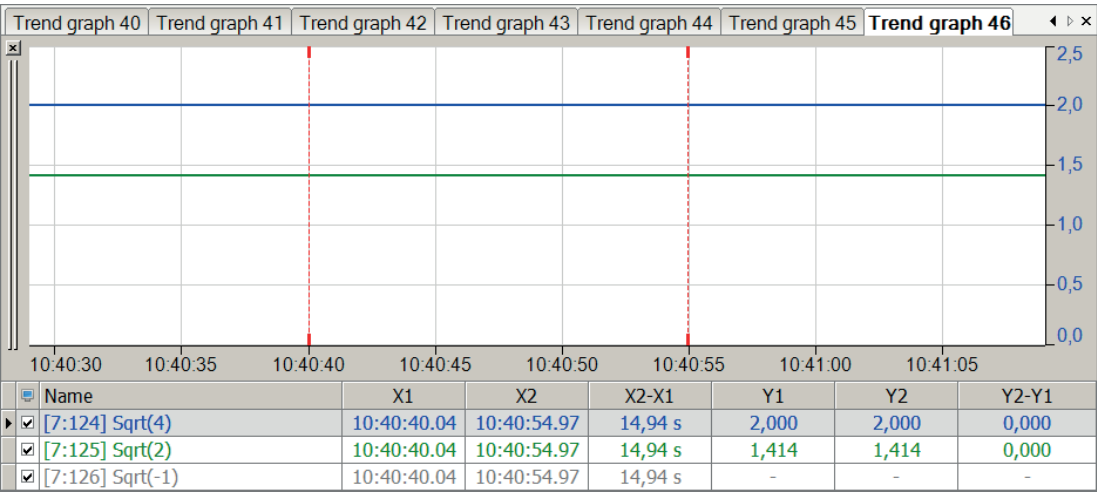


Рис. 22: Вычисление изображенных квадратных корней

2.2.14 Truncate

Truncate('Expression')

Описание

Функция Truncate отсекает знаки после запятой в значениях с плавающей запятой. Отрицательные значения округляются таким образом в большую сторону до следующего целочисленного значения, положительные значения - в меньшую сторону.

Пример

Отсечение знаков после запятой в значениях графика сигналов

Задача

В графике сигналов измеренное значение при каждом измерении необходимо выводить без знаков после запятой.

Решение

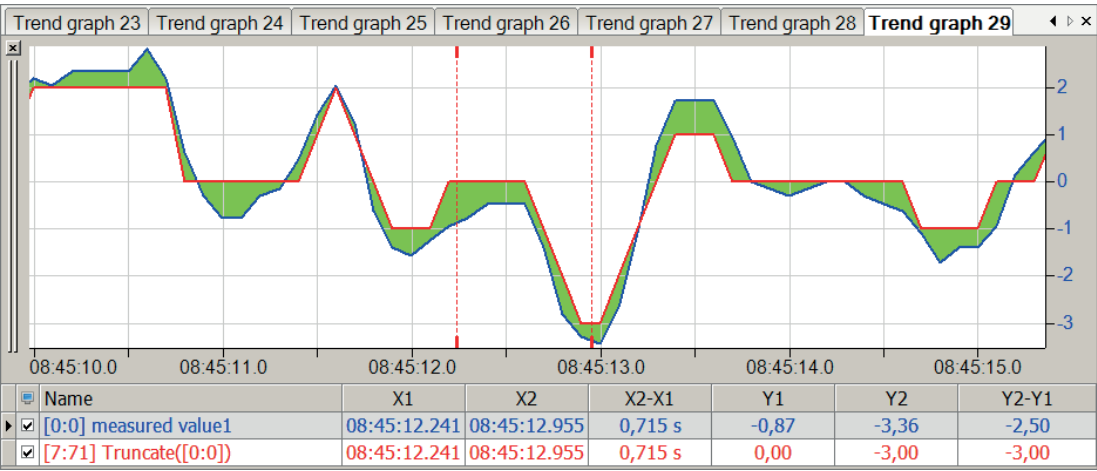


Рис. 23: Синий график: исходный сигнал; красный график: График сигнала с отсеченными знаками после запятой

2.3 Тригонометрические функции

`Funktion('Expression')`

Описание

Для самых разных вычислений, в которых требуются тригонометрические функции, например, расчет мощности в системах переменного тока, доступны стандартные функции и соответствующие обратные функции.

Функция	Описание
<code>Sin('Expression')</code>	Данная функция возвращает как результат синус 'Expression' в радианах.
<code>Cos('Expression')</code>	Данная функция возвращает как результат косинус 'Expression' в радианах.
<code>Tan('Expression')</code>	Данная функция возвращает как результат тангенс 'Expression' в радианах.
<code>Asin('Expression')</code>	Данная функция возвращает как результат арксинус 'Expression' в радианах.
<code>Acos('Expression')</code>	Данная функция возвращает как результат арккосинус 'Expression' в радианах.
<code>Atan('expression')</code>	Данная функция возвращает как результат арктангенс 'Expression' в радианах.

Совет



Для генерирования сигналов может также использоваться функция `GenerateSignal`.

Пример

Изображение тригонометрических функций

Решение

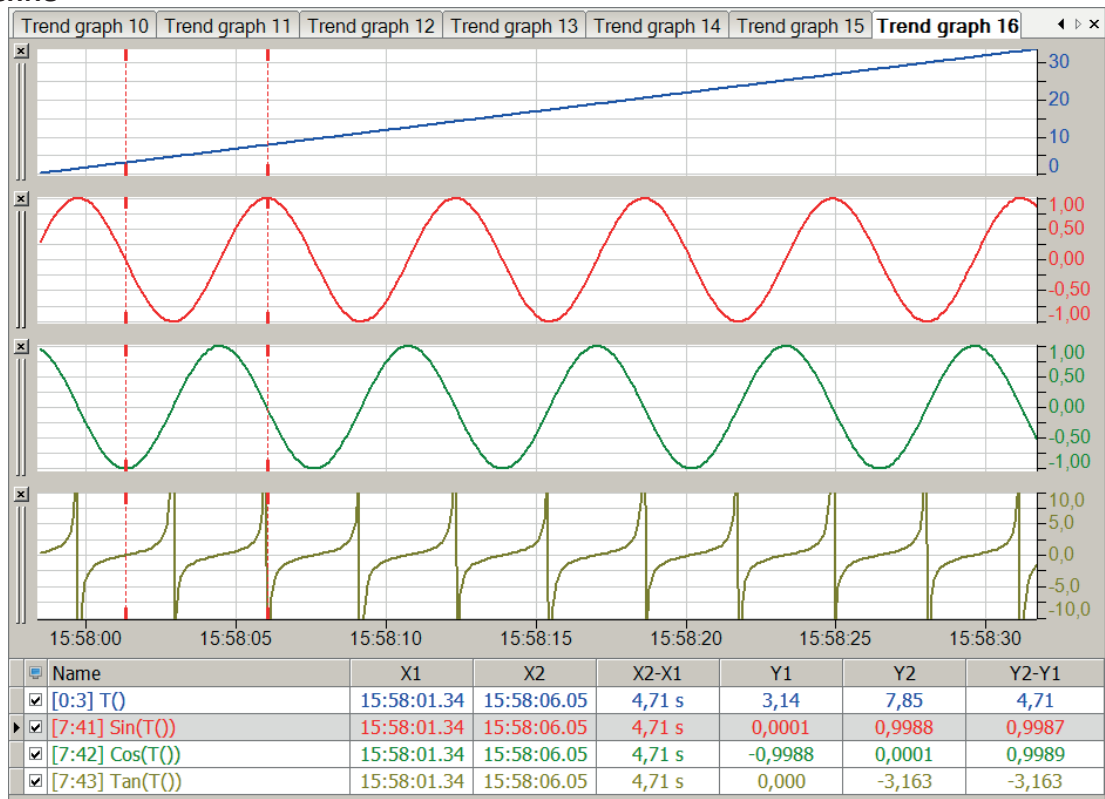


Рис. 24: Синий: Функция времени T() как основа для тригонометрических функций; красный: Синусоидальная функция времени; зеленый: Косинусоидальная функция времени; желтый: Функция тангенс времени

Pi ()

Описание

Число Pi заложено в системе как постоянная величина (p = 3.1415927...) для различных вычислений. Данная функция позволяет вставить число Pi в Ваше вычисление.

2.4 Статистические функции

2.4.1 Avg

`Avg('Expression', 'Reset=0')`

Аргументы

'Expression'	Измеренное значение, для которого образуется среднее значение	
'Reset' (сброс)	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления:	
	'Reset' = 0	Проведение вычисления
	'Reset' = 1	Остановить вычисление и установить результат на мгновенное значение 'Expression'

Описание

Данная функция возвращает как результат среднее значение 'Expression' с момента начала измерения или последнего сброса. Сигнал записывается в оперативную память. Арифметическое среднее значение вычисляется непрерывно. Если 'Reset' = 1 (ИСТИНА), то результат равен фактическому значению аргумента 'Expression'.

Hinweis



Результат функции Avg выводится только в последующем интервале.

2.4.2 AvgInTime

`AvgInTime('Expression', 'Interval', 'Reset=0')`


Аргументы

'Expression'	Измеренное значение, для которого образуется среднее значение	
'Interval'	Длина интервала в секундах	
'Reset'	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления:	
	'Reset' = 0	Выполнить вычисление
	'Reset' = 1	Остановить вычисление и установить результат на 0
	'Reset' = 2	Остановить вычисление и сохранить результат
	'Reset' = 3	Вычислить сейчас и затем остановить вычисление

Описание

Данная функция возвращает как результат среднее значение каждого временного интервала длины 'interval' выражения 'Expression'. Сигнал записывается в оперативную память компьютера. После прохождения интервала рассчитывается среднее арифметическое значение для данного интервала.

Совет



Результат функции AvgInTime будет отображен только в последующем интервале.

Пример

Влияние параметра 'Reset'

Постановка задачи

Параметр 'Reset' должен переключаться с 0 на 1, 2, или 3 через каждые 5 секунд, чтобы отображать результаты параметра.

Решение

Три if-запроса передают через функцию модуля времени значение 0 как ИСТИНА и значения 1, 2, 3 как ЛОЖЬ с интервалом 5 секунд параметру 'Reset' функции AvgInTime.

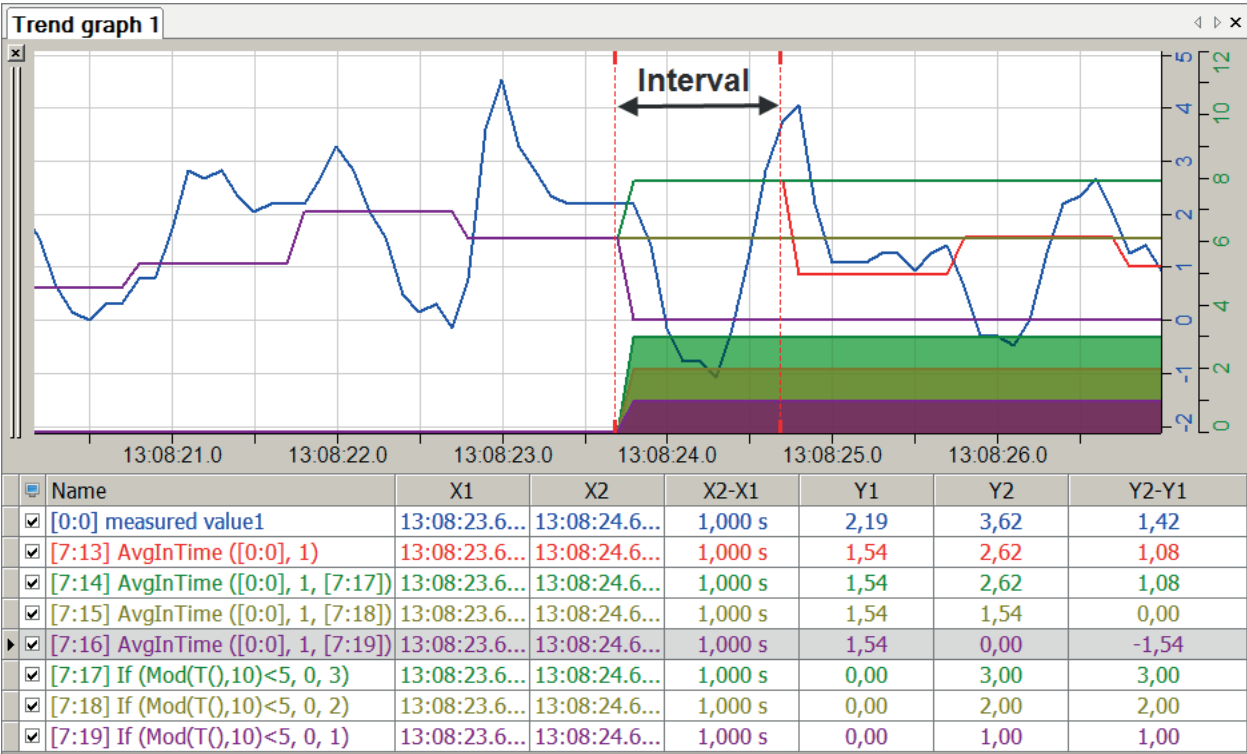


Рис. 25: Красный график: непрерывное вычисление без указания 'Reset'; зеленый график: ('Reset' = 3) значение вычисляется перед переключением; желтый график: ('Reset' = 2) значение не изменяется во время переключения; лиловый график: ('Reset' = 1) значение = 0 во время переключения

2.4.3 KurtosisInTime

KurtosisInTime('Expression','Interval',' Reset=0')

Аргументы

'Expression'	Измеренное значение, для которого образуется куртозис	
'Interval'	Длина интервала в секундах, на базе которого рассчитывается куртозис.	
'Reset'	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления	
	'Reset'=0	Выполнить вычисление
	'Reset'=1	Остановить вычисление и установить результат на 0
	'Reset'=2	Остановить вычисление и сохранить результат
	'Reset'=3	Вычислить сейчас и затем остановить вычисление

Описание

Вычисление куртозиса используется, например, для оценки и анализа колебаний. Он служит для определения количества отклонений в пределах сигнала вибрации.

С математической точки зрения, куртозис является показателем относительной «плоскостности» распределения (по сравнению с обычным распределением, имеющим нулевой куртозис). Положительный куртозис отображает конусообразное распределение (так называемое островершинное распределение), а отрицательный куртозис отображает плоское распределение (плосковершинное распределение).

Данный статистический метод особенно подходит для анализа случайных или стохастических сигналов, например, в случае техобслуживания по состоянию (мониторинг состояния) при виброанализе. Для описания графика сигнала используются методы плотности или частотности распределения вероятностей. Предполагается, что сигнал шума с амплитудным распределением Гаусса может быть измерен в исправных установках после отфильтровывания, например, частотных составляющих колебаний вибраций. В случае возникновения неисправностей отдельные импульсные сигналы смешиваются с данным сигналом, изменяя функцию распределения. Образование подходящих показателей, например, коэффициента амплитуды и коэффициент куртозиса, позволяет проводить оценку состояния установки.

При регулярных измерениях данные методы обеспечивают обзор состояния установки. Однако, недостаток заключается в том, что показатели после повышения снова снижаются. Причина в том, что количество импульсных сигналов возрастает при прогрессирующей неисправности. Это, в свою очередь влияет, на эффективное значение, но не на пиковое значение.

Изменения сигнала времени, вызванные ударными импульсами, влияют на изменения результирующей функции распределения. Неисправности очевидно дискретного характера могут вызвать резкое повышение коэффициента куртозиса. Таким образом его абсолютное значение позволяет делать выводы о неисправностях.

2.4.4 MAvg

`MAvg('Expression', 'WindowInterval', ' UpdateInterval=timebase', ' Reset=0')`

Аргументы

'Expression'	Измеренное значение, для которого образуется среднее значение	
	Длина интервала в секундах, с помощью которой образуется среднее значение; должна быть кратной 'UpdateInterval'.	
	Опциональный параметр (уставка = опорное время); задает, с каким так-ком проводится вычисление.	
'Reset'	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления.	
	'Reset'=0	Выполнить вычисление
	'Reset'=1	Остановить вычисление и установить результат на 0
	'Reset'=2	Остановить вычисление и сохранить результат

Описание

Данная функция возвращает как результат скользящее арифметическое среднее значение 'Expression', вычисляемое за 'WindowInterval' в секундах. Вычисление выполняется с периодом 'UpdateInterval'. 'UpdateInterval' является опциональным параметром и указывается в секундах. Если 'UpdateInterval' не указан, то он устанавливается равным опорному времени (по умолчанию), т.е. наименьшее возможное. Тогда вычисление осуществляется поступательно по одной точке измерения соответственно. Вычисление может осуществляться также и через большие интервалы, если для 'UpdateInterval' вводится 9кратное) значение опорного времени.

'WindowInterval' задает период, для которого каждый раз рассчитывается среднее значение. 'WindowInterval' должен быть кратным интервалу 'UpdateInterval'. В противном случае, 'WindowInterval' автоматически изменяется на первое кратное значение интервала 'UpdateInterval', большее или равное интервалу 'WindowInterval'.

Совет



При помощи данных функций можно также обрабатывать сигналы и выражения, но не на базе времени, а на базе длины, частоты или 1/длина. Вместо секунд необходимо указать диапазон оси X в м, Гц или 1/м в зависимости от соответствующей базы.

Пример

Для измеренного значения релевантны средние значения предыдущих 2 и 20 секунд.

Постановка задачи

Функция MAvg используется однократно с интервалом в 2 секунды и однократно с интервалом в 20 секунд.

Решение

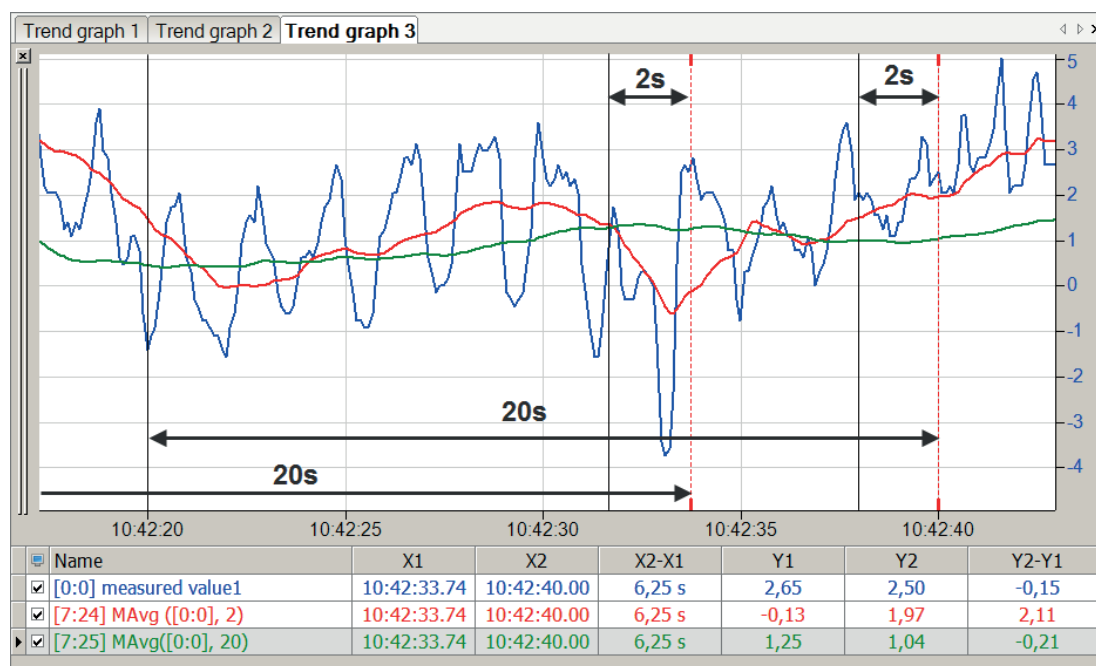


Рис. 26: Синий график: Измеренное значение, красный график: Среднее значение за интервал в 2 секунды; зеленый график: Среднее значение за интервал в 20 секунды

2.4.5 Max

Max('Expression', 'Reset=0')

Аргументы

'Expression'	Измеренное значение, для которого рассчитывается максимальное значение	
'Reset'	Опциональный цифровой параметр, который может использоваться для сброса максимального значения, например, для игнорирования процессов стабилизации измеренного сигнала в фазе пуска. 'Reset' может также сам быть выражением.	
	'Reset' = TRUE	Остановить вычисление и установить результат на мгновенное значение 'Expression'
	'Reset' = 0	Выполнить вычисление; выводится последнее распознанное максимальное значение. (уставка)

Описание

Данная функция возвращает как результат максимальное значение 'Expression'. Оно отображается как постоянное значение (горизонтальная линия) в полосе сигналов. Каждое значение сравнивается с предыдущим. Если новое значение больше старого, то большее значение вносится в график. Если новое значение равно или меньше старого, то вводится старое значение. Цифровой сигнал 'Reset' может выборочно останавливать вычисление максимального значения и сбрасывать результат снова на фактическое значение входного сигнала. Без сигнала сброса нет возможности сбрасывать индикацию, за исключением

случаев, когда измерение останавливается и запускается заново. 'Reset' может быть также выражением.

Примеры:

Max([0:0])	Сброс не осуществляется.
Max([0:0],If(Mod(T(),20)=0,TRUE(),FALSE()))	Максимальное значение сбрасывается каждые 20 секунд.
Max([0:0],[3.1])	например, с [3.1] = If([0:0]<10, 1, 0) Максимальное значение сбрасывается, если выражение [3.1] возвращает ИСТИНА, т.е. если выражение [0:0] не достигает предельное значение 10.

Пример

Совет



Данная функция может использоваться в виртуальном модуле с запоминанием. Ее результирующие значения можно получить, остановив или перезапустив измерение.

Необходимо рассчитать максимум для сигнала. Фаза пуска не должна учитываться в расчете.

Описание задачи

Чтобы удалить всплески во время фазы пуска, максимальное значение сбрасывается при использовании функции 'Reset'. Это можно сделать при помощи функции TriggerConstant, которая ожидает выравнивание сигнала. Настройка параметра 'Reset' осуществляется через распознавание фронтов ('OneShot') триггера.

Важно



Параметр 'Reset' не может постоянно быть установлен на ИСТИНА, т.к. максимальное значение будет постоянно сбрасываться и таким образом соответствовало бы измеренному значению.

Решение

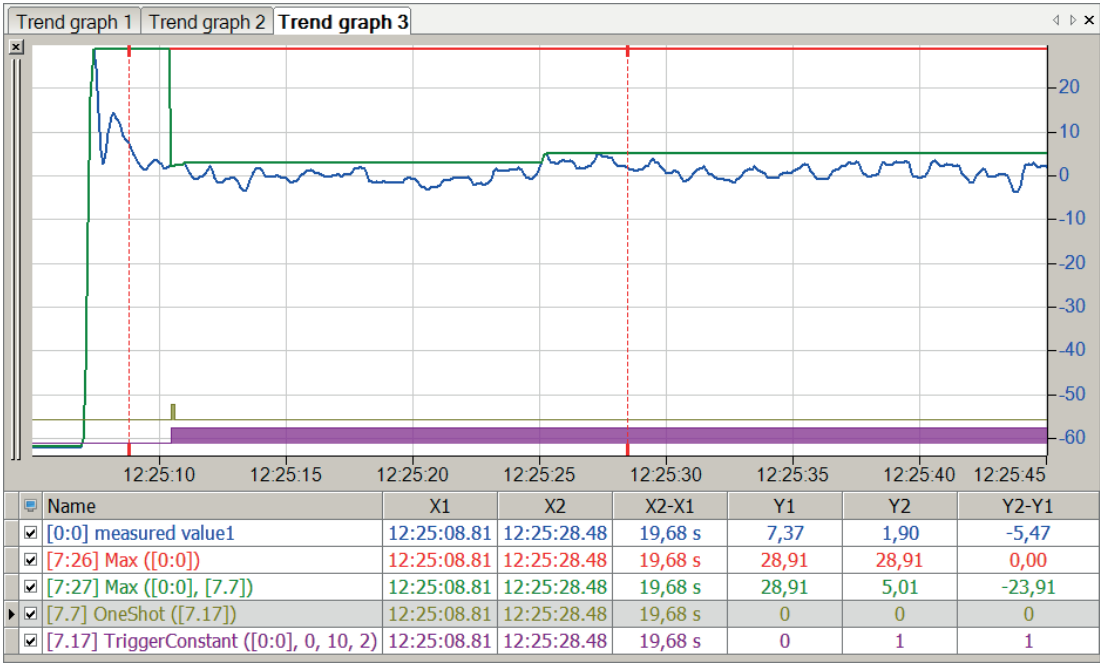


Рис. 27: Синий график: Измеренное значение, красный график: искаженное максимальное значение без 'Reset'; зеленый график: Максимальное значение со сбросом после фазы пуска; желтая диаграмма: Функция One-Shot для срабатывания 'Reset'; лиловая диаграмма: Триггер после фазы пуска

2.4.6 Max2

Max2 ('Expression1', 'Expression2')

Описание

Данная функция возвращает как результат максимум из двух сигналов 'Expression1' и 'Expression2'. Оба сигнала сравниваются значение за значением и соответствующее большее значение передается как результат.

Пример

Из двух измеренных значений релевантно только большее значение.

Решение

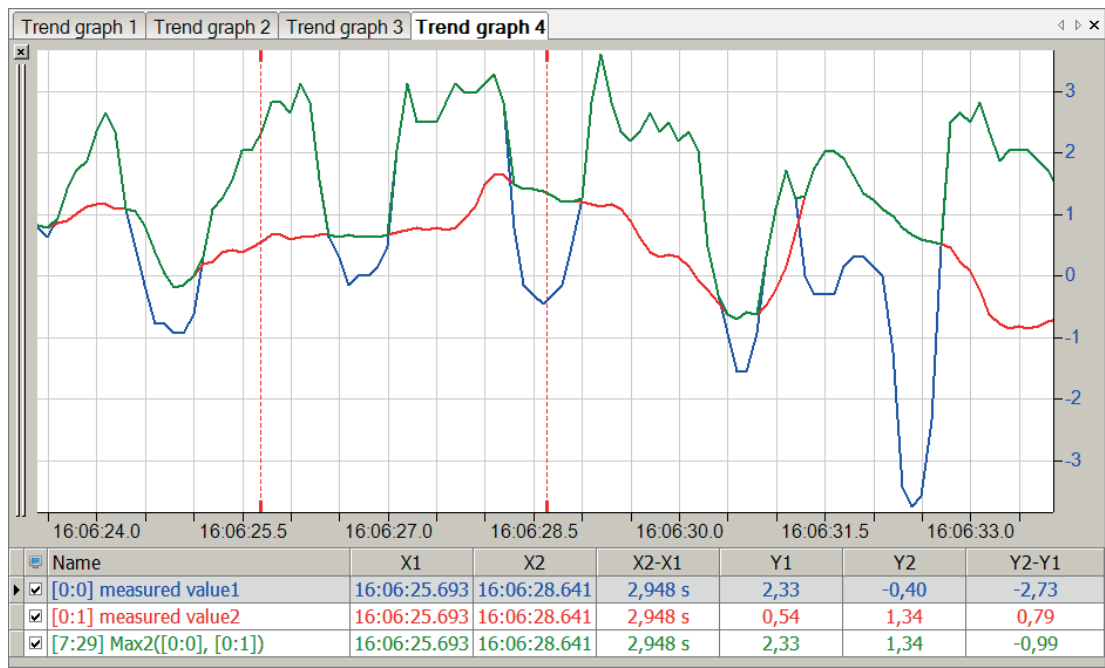


Рис. 28: Синий график: измеренное значение1; красный график: измеренное значение2; зеленый график: график максимальных значений

2.4.7 MaxInTime

```
MaxInTime('Expression','Interval',' Reset=0')
```

Аргументы

'Expression'	Измеренное значение, для которого образуется максимальное значение	
'Interval'	Длина интервала в секундах, за который рассчитывается максимальное значение.	
'Reset'	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления	
	'Reset'=0	Выполнить вычисление
	'Reset'=1	Остановить вычисление и установить результат на 0
	'Reset'=2	Остановить вычисление и сохранить результат
	'Reset'=3	Вычислить сейчас и затем остановить вычисление

Описание

Данная функция возвращает как результат максимальное значение 'Expression' каждого интервала длины 'Interval' (в с).

Важно

Результат функции MaxInTime отображается только в последующем интервале.

Пример

Влияние параметра 'Reset'

Постановка задачи

Параметр 'Reset' должен переключаться с 0 на 1, 2 или 3 через каждые 5 секунд, чтобы отображать результаты параметра.

Решение

Три if-запросы передают через функцию модуля времени значение 0 как ИСТИНА и значения 1, 2, 3 как ЛОЖЬ с интервалом 5 секунд параметру 'Reset' функции MaxInTime.

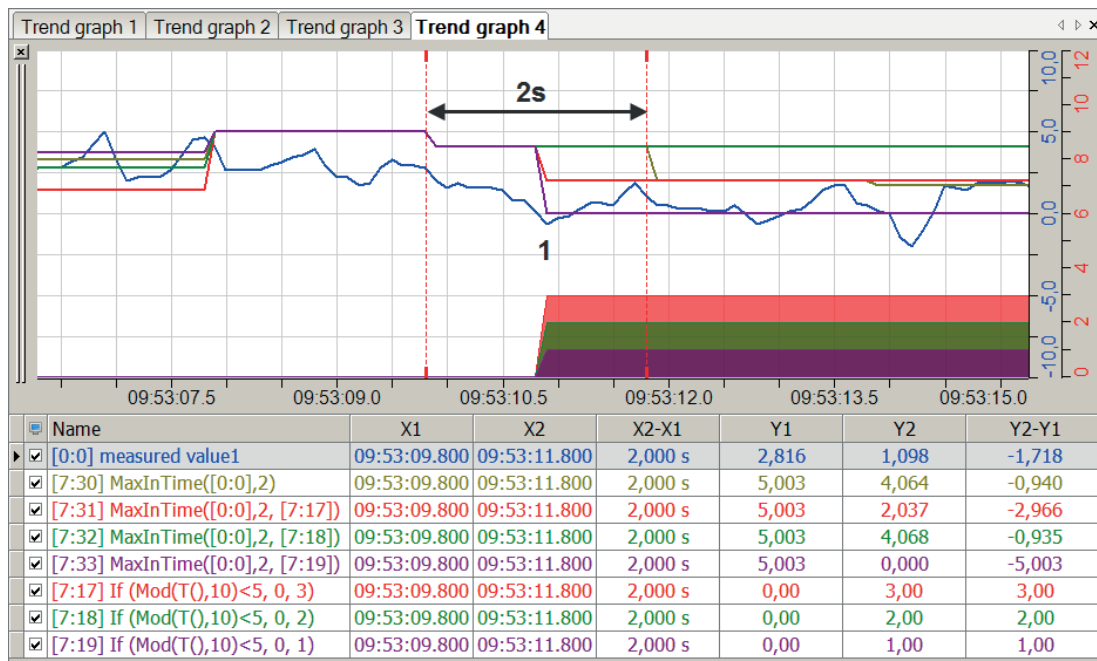


Рис. 29: Синий график: измеренное значение; желтый график: непрерывное вычисление без указания 'Reset'; красный график: ('Reset'=3) значение рассчитывается перед прерыванием, в пределах интервала (1); зеленый график: ('Reset'='') значение остается постоянным во время прерывания; лиловый график: ('Reset'=1) значение=0 во время прерывания, также в пределах интервала (1)

2.4.8 Min

```
Min('Expression', 'Reset=0')
```

Аргументы

'Expression'	Измеренное значение, для которого рассчитывается минимальное значение	
'Reset' (сброс)	Опциональный цифровой параметр, который может использоваться для сброса минимума, например, для игнорирования процессов выравнивания измеренного сигнала во время фазы пуска. 'Reset' может сам быть выражением.	
	'Reset' = TRUE	Остановить вычисление и установить результат на мгновенное значение 'Expression'
	'Reset' = 0	Провести вычисление; выводится последнее распознанное минимальное значение. (уставка)

Описание

Данная функция выводит как результат минимальное значение сигнала 'Expression'. Оно отображается как постоянное значение (горизонтальная линия) в полосе сигналов. Каждое значение сравнивается с предыдущим. Если новое значение меньше старого, Min вводит в график меньшее значение. Если новое значение равно или больше старого, то вводится старое значение. Цифровой сигнал 'Reset' может выборочно останавливать вычисление минимального значения и сбрасывать результат снова на фактическое значение входного сигнала. Без сигнала сброса нет возможности сбрасывать индикацию, за исключением случаев, когда измерение останавливается и запускается заново. 'Reset' может быть также выражением.

Примеры:

Min([0:0])	Сброс не осуществляется.
Min([0:0],If(Mod(T(),20)=0,TRUE(),FALSE()))	Минимальное значение сбрасывается каждые 20 секунд.
Min([0:0],[3.1])	например, с [3.1] = If([0:0]<10, 1, 0) Минимальное значение сбрасывается, если выражение [3.1] возвращает ИСТИНА, т.е. если выражение [0:0] не достигает предельное значение 10.

Пример

Совет



Данная функция может использоваться в виртуальном модуле с запоминанием. Ее результирующие значения можно получить, остановив или перезапустив измерение.

Необходимо задать минимум для сигнала. При этом фаза пуска не должна учитываться в расчете.

Описание задачи

Чтобы удалить всплески во время фазы пуска, максимальное значение сбрасывается при помощи функции 'Reset'. Это можно сделать при помощи функции TriggerConstant, которая ожидает выравнивание сигнала. Настройка параметра 'Reset' осуществляется посредством распознавание фронтов ('OneShot') триггера.

Важно



Параметр 'Reset' не может быть постоянно настроен на ИСТИНА, так как минимальное значение будет тогда сбрасываться постоянно и, следовательно, соответствовать измеренному значению.

Решение

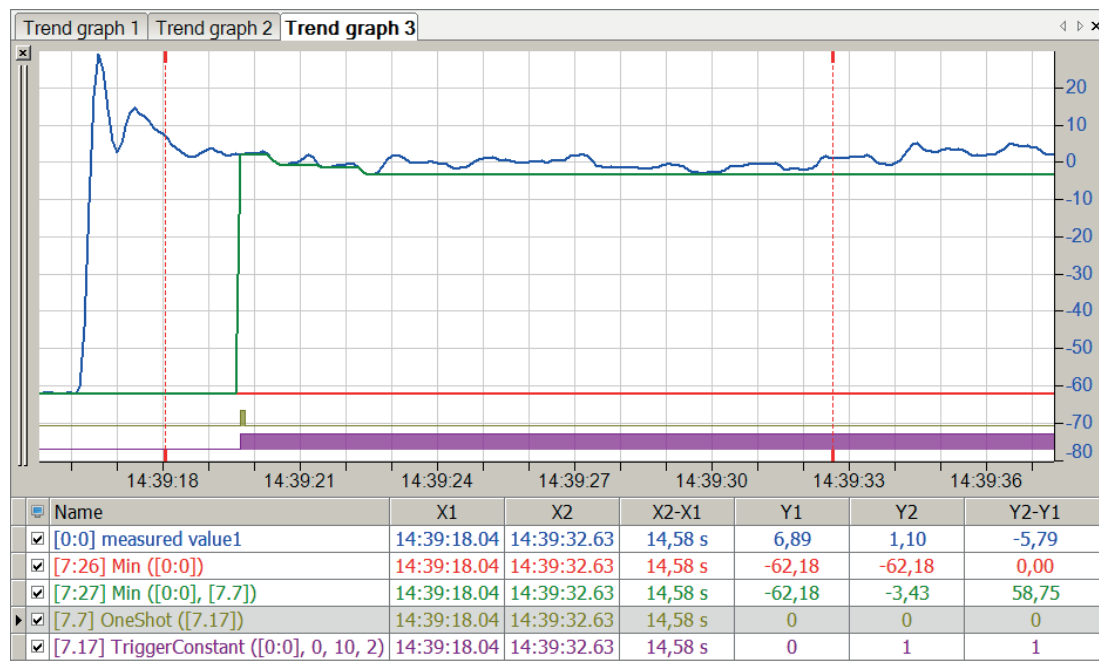


Рис. 30: Синий график: Измеренное значение, красный график: искаженное минимальное значение без 'Reset'; зеленый график: Минимальное значение со сбросом после фазы пуска; желтая диаграмма: Функция One-Shot для срабатывания 'Reset'; лиловая диаграмма: Триггер после фазы пуска

2.4.9 Min2

Min2('Expression1', 'Expression2')

Описание

Данная функция возвращает как результат минимум из двух сигналов 'Expression1' и 'Expression2'. Оба сигнала сравниваются значение за значением и соответствующее меньшее значение передается как результат.

Пример

Из двух измеренных значений релевантно только меньшее значение.

Решение

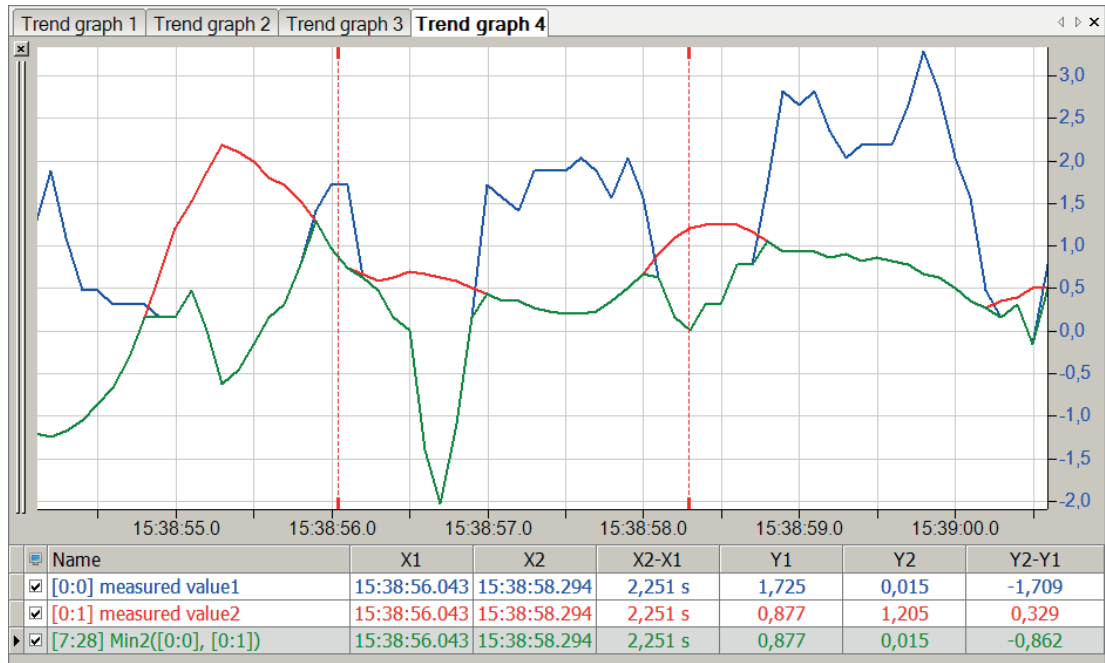


Рис. 31: Синий график: измеренное значение1; красный график: измеренное значение2; зеленый график: график минимальных значений

2.4.10 MinInTime

MinInTime('Expression','Interval',' Reset=0')

Аргументы

'Expression'	Измеренное значение, для которого образуется минимальное значение	
'Interval'	Длина интервала в секундах, при помощи которого рассчитывается минимальное значение.	
'Reset'	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления	
	'Reset'=0	Выполнить вычисление
	'Reset'=1	Остановить вычисление и установить результат на 0
	'Reset'=2	Остановить вычисление и сохранить результат
	'Reset'=3	Вычислить сейчас и затем остановить вычисление

Описание

Данная функция возвращает как результат минимальное значение 'Expression' каждого интервала длины 'Interval' (в с).

Важно



Результат функции MinInTime выводится только в последующем интервале.

Пример

Влияние параметра 'Reset'

Постановка задачи

Параметр 'Reset' должен переключаться с 0 на 1, 2 или 3 через каждые 5 секунд, чтобы отображать результаты ввода параметра.

Решение

Три if-запросы передают через функцию модуля времени с интервалом 5 секунд значение 0 как ИСТИНА и значения 1, 2, 3 как ЛОЖЬ параметру 'Reset' функции MinInTime.

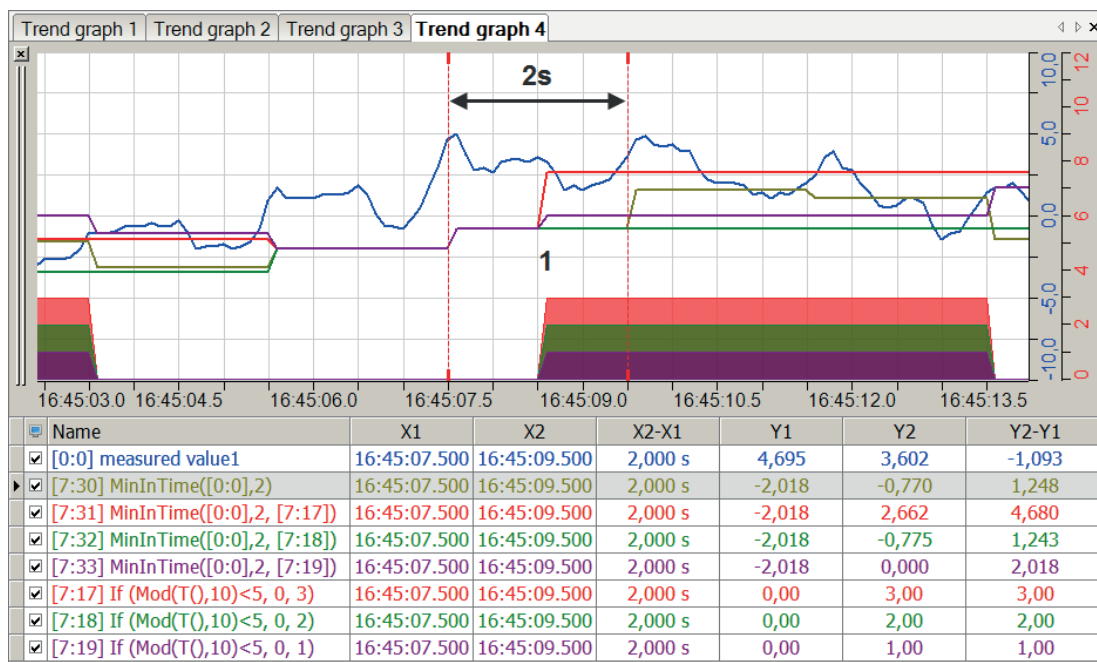


Рис. 32: Синий график: измеренное значение; желтый график: непрерывное вычисление без указания 'Reset'; красный график: ('Reset'=3) значение рассчитывается перед прерыванием, в пределах интервала (1); зеленый график: ('Reset'='') значение остается постоянным во время прерывания; лиловый график: ('Reset'=1) значение=0 во время прерывания, также в пределах интервала (1)

2.4.11 MKurtosis

`MKurtosis('Expression', 'WindowInterval', 'UpdateInterval=timebase', 'Reset=0')`

Аргументы

'Expression'	Измеренное значение, для которого образуется куртозис	
'WindowInterval'	Длина интервала в секундах, с помощью которой образуется куртозис; должна быть кратной 'UpdateInterval'.	
'UpdateInterval'	Опциональный параметр (уставка = опорное время); задает, с каким таким проводится вычисление.	
'Reset'	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления.	
	'Reset'=0	Выполнить вычисление
	'Reset'=1	Остановить вычисление и установить результат на 0
	'Reset'=2	Остановить вычисление и сохранить результат

Описание

Возвращает значение куртозиса 'Expression' каждые 'UpdateInterval' секунд, на базе скользящего окна 'WindowInterval' секунд.

Параметр 'UpdateInterval' является опциональным. Если не указано, то он равен опорному времени функции (т.е. наименьшему возможному значению).

'WindowInterval' должен быть кратным 'UpdateInterval'. Если нет, то 'WindowInterval' автоматически изменяется, на первое краткое 'UpdateInterval' большее или равное 'WindowInterval'.

Для получения информации о куртозисе см. раздел [↗ KurtosisInTime](#), страница 38

2.4.12 MMax

`MMax('Expression', 'WindowInterval', 'UpdateInterval=timebase', 'Reset=0')`

Аргументы

'Expression'	Измеренное значение, для которого образуется максимальное значение	
'WindowInterval'	Длина интервала в секундах, с помощью которой вычисляется максимальное значение; должен быть кратным 'UpdateInterval'.	
'UpdateInterval'	Опциональный параметр (уставка = опорное время); задает, с каким таким проводится вычисление.	
'Reset'	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления	
	'Reset'=0	Выполнить вычисление
	'Reset'=1	Остановить, сбросить вычисление и установить результат на 0
	'Reset'=2	Остановить, сбросить вычисление и сохранить результат

Описание

Данная функция возвращает максимум аргумента 'Expression' каждые 'UpdateInterval' секунд, на базе скользящего окна 'WindowInterval' секунд.

Параметр 'UpdateInterval' является опциональным. Если не указано, то он равен опорному времени функции (т.е. наименьшему возможному значению).

'WindowInterval' должен быть кратным 'UpdateInterval'. Если нет, то 'WindowInterval' автоматически изменяется, на первое краткое 'UpdateInterval' большее или равное 'WindowInterval'.

2.4.13 MMin

`MMin('Expression', 'WindowInterval', 'UpdateInterval=timebase', 'Reset=0')`

Аргументы

'Expression'	Измеренное значение, для которого образуется минимальное значение	
'WindowInterval'	Длина интервала в секундах, с помощью которой рассчитывается минимальное значение; должен быть кратным 'UpdateInterval'.	
'UpdateInterval'	Опциональный параметр (уставка = опорное время); задает, с каким таким проводится вычисление.	
'Reset'	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления	
	'Reset'=0	Выполнить вычисление
	'Reset'=1	Остановить, сбросить вычисление и установить результат на 0
	'Reset'=2	Остановить, сбросить вычисление и сохранить результат

Описание

Данная функция возвращает минимум аргумента 'Expression' каждые 'UpdateInterval' секунд, на базе скользящего окна 'WindowInterval' секунд.

Параметр 'UpdateInterval' является опциональным. Если не указано, то он равен опорному времени функции (т.е. наименьшему возможному значению).

'WindowInterval' должен быть кратным 'UpdateInterval'. Если нет, то 'WindowInterval' автоматически изменяется, на первое краткое 'UpdateInterval' большее или равное 'WindowInterval'.

2.4.14 MSkewness

`MSkewness('Expression', 'WindowInterval', 'UpdateInterval=timebase', 'Reset=0')`

Аргументы

'Expression'	Измеренное значение, для которого образуется асимметрия	
	Длина интервала в секундах, с помощью которой образуется асимметрия; должна быть кратной 'UpdateInterval'.	
	Опциональный параметр (уставка = опорное время); задает, с каким тактом проводится вычисление.	
'Reset'	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления.	
	'Reset'=0	Выполнить вычисление
	'Reset'=1	Остановить вычисление и установить результат на 0
	'Reset'=2	Остановить вычисление и сохранить результат

Описание

Возвращает значение асимметрии 'Expression' каждые 'UpdateInterval' секунд, на базе скользящего окна 'WindowInterval' секунд.

Параметр 'UpdateInterval' является опциональным. Если не указано, то он равен опорному времени функции (т.е. наименьшему возможному значению).

'WindowInterval' должен быть кратным 'UpdateInterval'. Если нет, то 'WindowInterval' автоматически изменяется, на первое краткое 'UpdateInterval' большее или равное 'WindowInterval'.

Для получения информации об асимметрии см. раздел [↗ SkewnessInTime](#), страница 52

2.4.15 SkewnessInTime

`SkewnessInTime('Expression', 'Interval', 'Reset=0')`

Аргументы

'Expression'	Измеренное значение, для которого образуется асимметрия	
'Interval'	Длина интервала в секундах, при помощи которого рассчитывается асимметрия.	
'Reset'	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления	
	'Reset' = 0	Выполнить вычисление
	'Reset' = 1	Остановить вычисление и установить результат на 0
	'Reset' = 2	Остановить вычисление и сохранить результат
	'Reset' = 3	Вычислить сейчас и затем остановить вычисление

Описание

Аналогично коэффициенту куртозиса коэффициент асимметрии предназначен для оценки и анализа колебаний. Коэффициент асимметрии может применяться в случае, если должны проверяться свойства симметрии сигнала колебания (напр. сигнал ускорения).

При данной функции выбранное выражение подразделяется на равные интервалы величины 'Interval'. Для данных интервалов будет затем осуществляться вычисление асимметрии.

С математической точки зрения речь идет при этом об оценке асимметрии функции распределения. Распределение называется правосимметричным (положительная асимметрия), если основная часть распределения сконцентрирована в левой части. Распределение называется левоасимметричным (отрицательная асимметрия), если большая часть распределения сконцентрирована в правой части. Уровень асимметрии определяется третьим моментом распределения.

Процесс расчета асимметрии аналогичен функции KurtosisInTime.

2.4.16 StddevInTime

```
StddevInTime('Expression','Interval',' Reset=0')
```

Аргументы

'Expression'	Измеренное значение, для которого образуется стандартное отклонение	
'Interval'	Длина интервала в секундах, при помощи которого образуется стандартное отклонение.	
'Reset' (сброс)	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления	
	'Reset' = 0	Выполнить вычисление
	'Reset' = 1	Остановить вычисление и установить результат на 0
	'Reset' = 2	Остановить вычисление и сохранить результат
	'Reset' = 3	Вычислить сейчас и затем остановить вычисление

Описание

Данная функция возвращает как результат стандартное отклонение 'Expression' через каждый интервал времени длины. Опциональный параметр 'Reset' позволяет остановить вычисление.

Вычисление стандартного отклонения осуществляется по формуле:

$$s_x = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

s_x = Standardabweichung

\bar{x} = Mittelwert

n = Anzahl Messungen

Пример

Для графика сигналов необходимо задать стандартное отклонение с интервалом времени в одну секунду. Спрос не требуется.

Решение

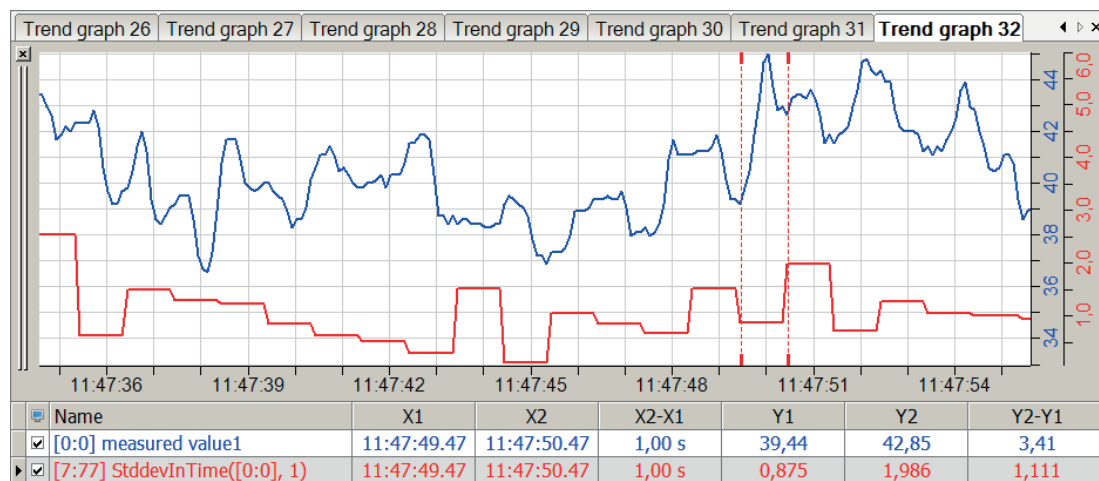


Рис. 33: Синий график: Измеренные значения, красный график: Стандартное отклонение с интервалом длины в одну секунду;

Важно



Стандартное отклонение всегда указывается для предыдущего интервала.

2.5 Триггерные функции

2.5.1 PeriodicTrigger

```
PeriodicTrigger('Interval*', 'StartTime*', 'UseSystemTime*')
```

Аргументы

'Interval*'	Интервал в минутах, в течение которого возвращается значение ИСТИНА	
'StartTime*'	Указание в минутах; выводит начальное время ('StartTime' modulo 'Interval') после использования функции с модулем.	
	Определяет, используется ли системное время или внутренний таймер высокого разрешения.	
	'UseSystemTime' = 0	Используется не системное время, а внутренний таймер.
	'UseSystemTime' > 0	Используется системное время.

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает значение ИСТИНА каждые 'Interval' минут, начиная с 'StartTime' Modulo 'Interval' минут. Меркер 'UseSystemTime' задает, должно ли использоваться системное время или внутренний таймер высокого разрешения.

Пример

Изображение триггера каждые 10 секунд.

Решение

Изображение следующих функций:

- PeriodicTrigger(10/60, 10/60, 1)
- PeriodicTrigger(10/60, 11/60, 1)
- PeriodicTrigger(10/60, 15/60, 1)

с 10/60 мин (соответствует 10 с) для 'Interval', снова 10/60 мин или 11/60 мин (11 с) и 15/60 мин (15 с) для 'StartTime' и 1 (системное время) для 'UseSystemTime'.

Как представлено на рисунке, триггеры отличаются по их времени начала. По выше представленной схеме вычисления интервал триггера красного графика начинается в 0 с, интервал зеленого графика в 1 с и интервал синего графика в 5 с.

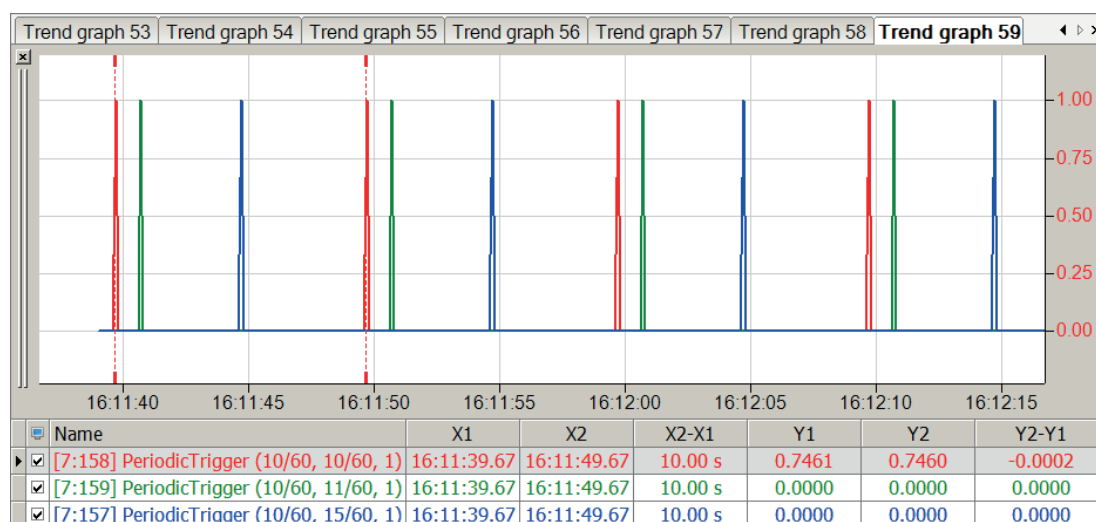


Рис. 34: Триггер каждые 10 секунд с различным временем начала

Важно

Отрицательные значения для 'StartTime' недействительны и ведут к сообщению об ошибке.

2.5.2 TriggerChangeRate

`TriggerChangeRate('Expression', 'DeltaY*', 'DeltaT*', 'DeadTime*')`

Аргументы

'Expression'	Измеренное значение
'DeltaY*'	Требуемая разница между значениями для срабатывания триггера
'DeltaT*'	Рассматриваемый интервал времени
'DeadTime*'	Время в секундах, в течение которого должно выполняться триггерное условие до срабатывания. Если задержка не требуется, то необходимо ввести значение 0.

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Функция возвращает ИСТИНА, пока изменение измеренного значения 'Expression' (dy) в пределах интервала 'DeltaT' больше 'DeltaY'.

Важно

i

Функция сравнивает разницу значений с интервалом времени DeltaT с DeltaY, градиент кривой может также отличаться между значениями.

Пример
Различия в срабатывании триггера и влияния времени нечувствительности 'DeadTime'

Решение

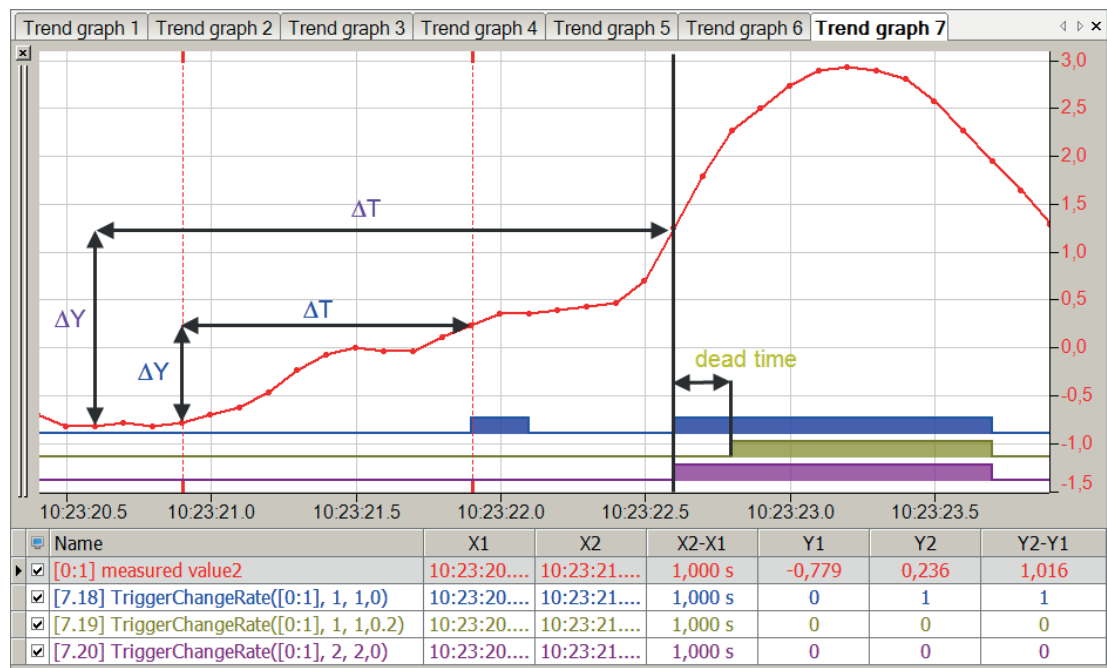


Рис. 35: Красный график: Измеренное значение; синяя диаграмма: срабатывание триггера с $\Delta Y=1$ и $\Delta T=1$; желтая: срабатывание триггера с $\Delta Y=1$ и $\Delta T=1$; фиолетовая: срабатывание триггера с $\Delta Y=2$ и $\Delta T=2$.

2.5.3 TriggerConstant

TriggerConstant('Expression', 'Level*', 'Epsilon*', 'DeadTime*')

Аргументы

'Expression'	Измеренное значение
'Level*' (уровень)	Указание значения центральной линии диапазона, в котором должен сработать триггер.
'Epsilon*'	Значение, обозначающее расстояние обоих пределов диапазонов до центральной линии
'DeadTime*'	Время в секундах, в течение которого должно выполняться триггерное условие до срабатывания. Если задержка не требуется, то необходимо ввести значение 0.

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Функция возвращает ИСТИНА, пока 'Expression' остается в диапазоне ['Level' - 'Epsilon', 'Level' + 'Epsilon'] как минимум в течение 'DeadTime'.

Важно



В отличие от функций TriggerEdge и TriggerLevel триггерный сигнал выводится в течение всего периода превышения или недостижения уровня, а не одним импульсом.

Пример

Триггер должен срабатывать, если измеренное значение дольше, чем одну секунду находится в диапазоне между 0,5 и 1,5.

Решение

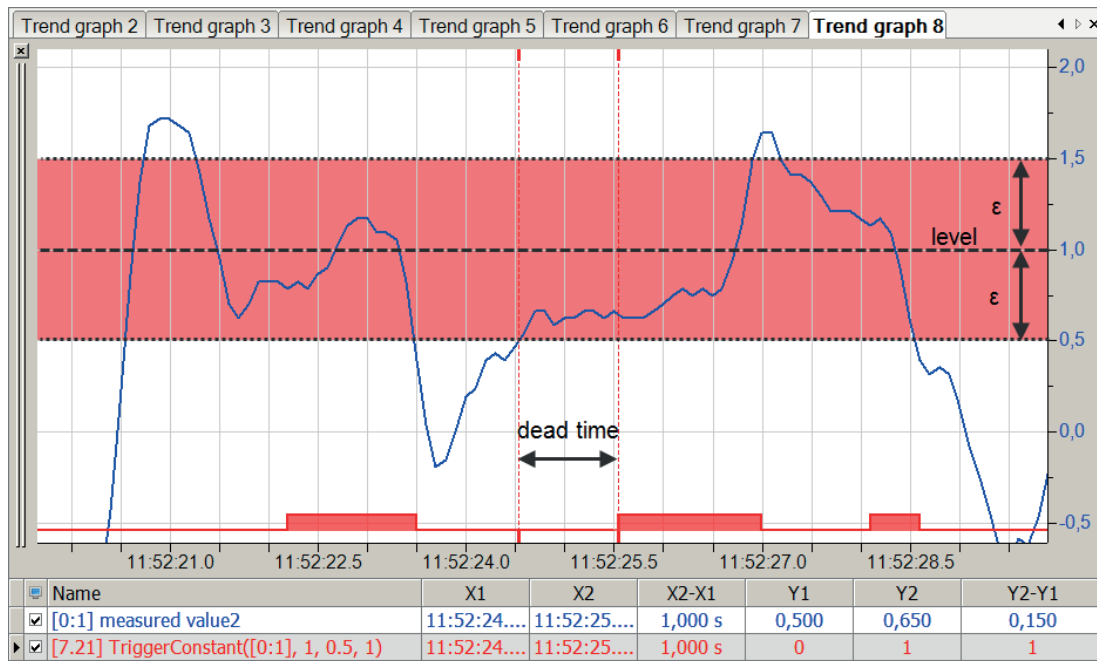


Рис. 36: Синий график: Измеренное значение; красная диаграмма: триггер сработал

2.5.4 TriggerEdge

TriggerEdge('Expression', 'Level*', 'EdgeType*', 'DeadTime*')

Аргументы

'Expression'	Измеренное значение	
'Level*' (уровень)	Значение уровня	
'EdgeType*'	Указание типа фронта (передний, задний или оба), которые учитываются при подсчете пересечения значения уровня	
	'EdgeType' < 0	только задние фронты (пересечение в отрицательном направлении)
	'EdgeType' > 0	только задние фронты (пересечение в положительном направлении)
	'EdgeType' = 0	передние и задние фронты
'DeadTime*'	Время в секундах, в течение которого измеренное значение должно превышать или не достигать значения уровня, чтобы сработал триггер.	

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Срабатывает, если 'Expression' превышает или не достигает 'Level' и остается на этой стороне уровня хотя бы в течение 'DeadTime' секунд. Если 'Expression' является цифровым сигналом, то 'Level' установлен на 0.5. 'EdgeType' задает, какие фронты и пересечения считаются:

Важно



В отличие от функции TriggerLevel и TriggerConstant при пересечении значения уровня выводится только один импульс.

Пример

Триггер должен срабатывать в случае, когда измеренное значение находится выше 1,5 или ниже -0,5 в течение 0,5 секунд.

Решение

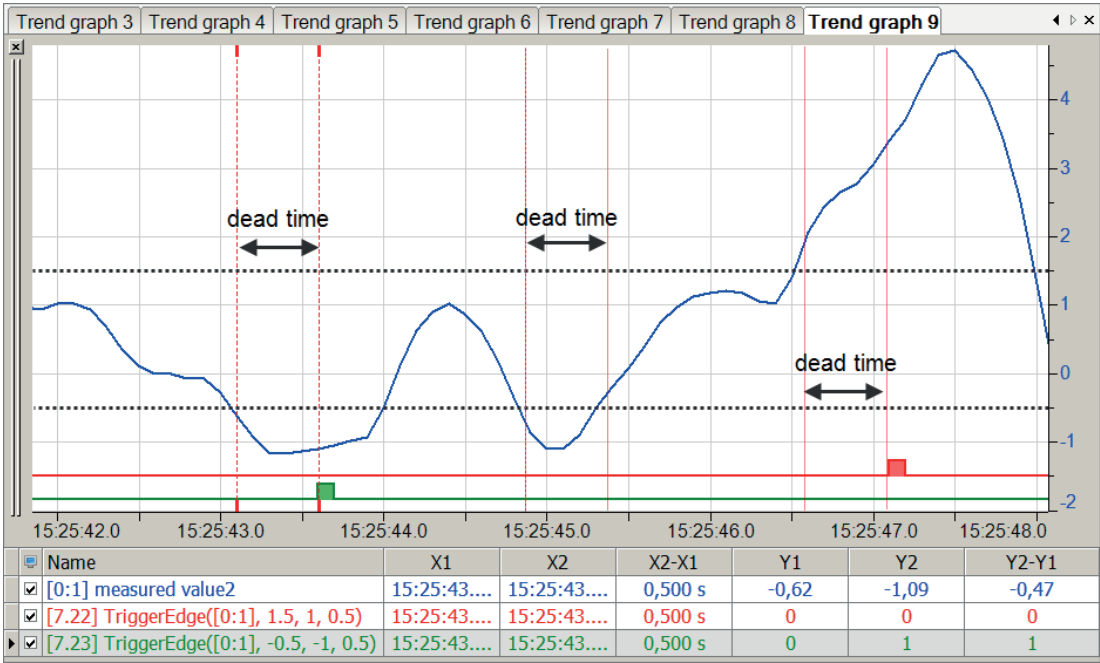


Рис. 37: Синий график: Измеренное значение; красный график: Триггер, когда измеренное значение больше 1,5 в течение более 0,5 секунд; зеленый: Триггер, когда измеренное значение меньше -0,5 в течение более 0,5 секунд

2.5.5 TriggerLevel

TriggerLevel('Expression', 'Level*', 'LevelType*', 'DeadTime*')

Аргументы

'Expression'	Измеренное значение	
'Level*' (уровень)	Значение уровня	
'LevelType*'	Указание, какая сторона 'Level' рассматривается	
	'LevelType' = 0	под уровнем
	'LevelType' = 1	над уровнем
'DeadTime*'	Время в секундах, в течение которого измеренное значение должно сохраняться на рассматриваемой стороне значения уровня, чтобы сработал триггер.	

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Срабатывает, если 'Expression' в течение миним. 'DeadTime' секунд находится выше или ниже 'Level'. 'LevelType' задает, какая сторона 'Level' контролируется.

Важно

i

В отличие от функций TriggerEdge и TriggerConstant в течение превышения или недостижения периодически выводятся триггерные импульсы через интервалы 'DeadTime'.

Пример

Триггер должен срабатывать в случае, когда измеренное значение находится выше 0,8 и ниже -0,2 в течение 0,4 секунд. При этом триггерный импульс должен выводиться в течение всего периода времени превышения или недостижения.

Решение

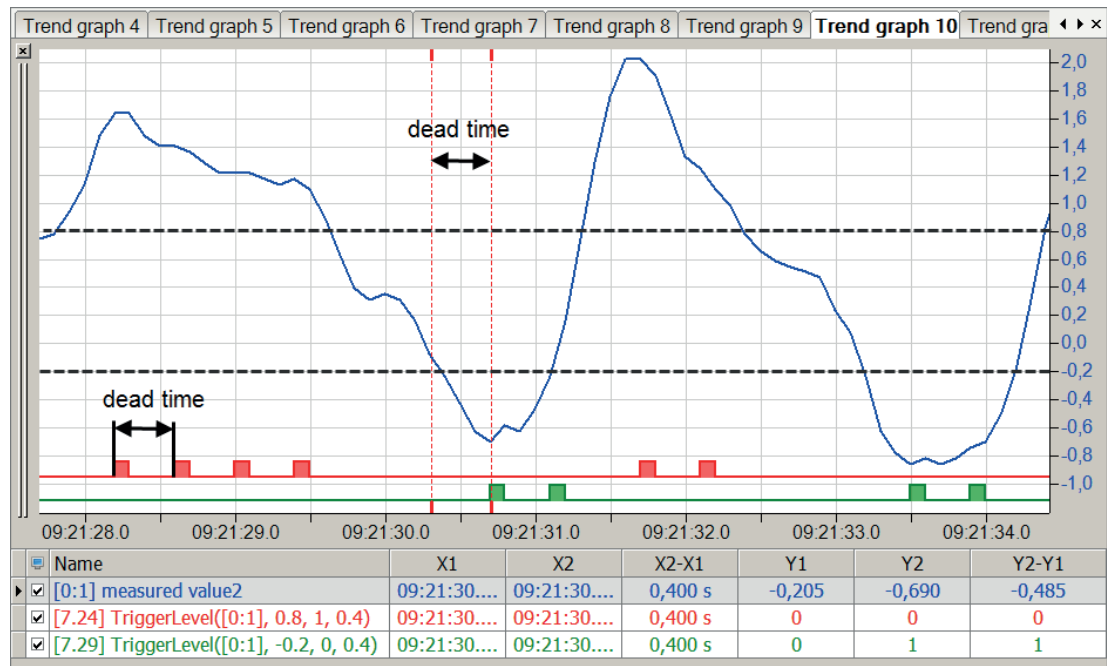


Рис. 38: Синий график: Измеренное значение; красная диаграмма: Триггер каждые 0,4 секунды, пока измеренное значение будет выше 0,8; зеленый: Триггер каждые 0,4 секунды, пока измеренное значение будет ниже -0,2

2.5.6 TriggerHarmonicLevel

```
TriggerHarmonicLevel('Expression', "'LimitProfile*'", 'Harmonic*')
```

Аргументы

'Expression'	Измеренный сигнал
""LimitProfile*""	Имя профиля предельного значения, заданного в PQU
'Harmonic*'	Порядок (гармоника), предельное значение которого должно контролироваться, значения от 0 до 50

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Триггер срабатывает (результат ИСТИНА), если значение сигнала выше заданного в профиле ""LimitProfile"" PQU гармонического предела. Параметр 'Harmonic' определяет, какое предельное значение используется из профиля предельного значения.

Пример

При измерении на распределительном устройстве среднего напряжения линейное напряжение U12 должно контролироваться на наличие высших гармоник. При превышении предписанного в соответствии с EN50160 предельного значения для 3. гармоники в 10-минутном интервале должен сработать триггерный сигнал.

Профиль предельного значения сконфигурирован в диспетчере вв/выв в модулях спектра. Имя профиля: "EN50160 - Medium voltage - Harmonics".

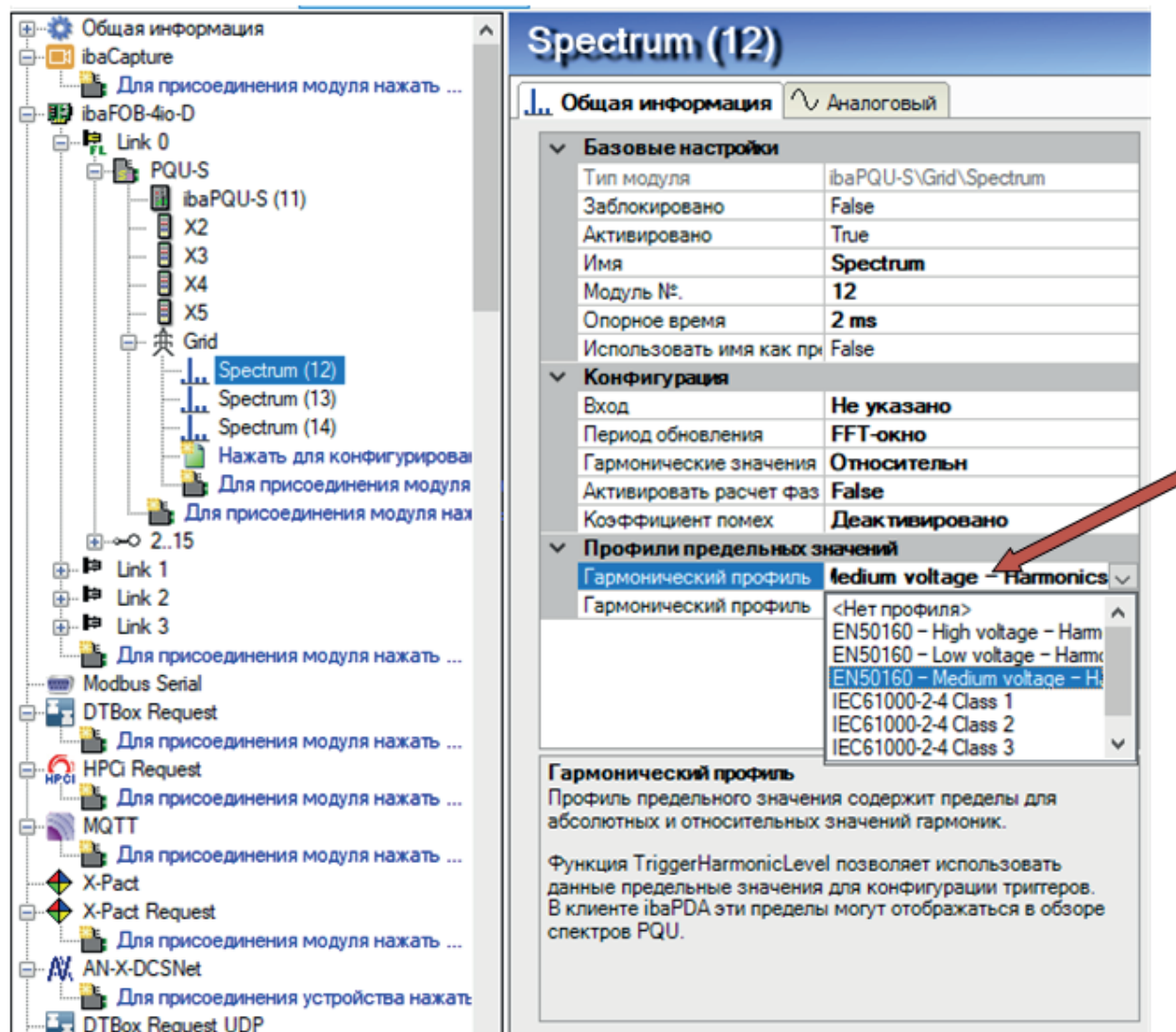


Рис. 39: Настроенный профиль предельного значения: EN50160 - Medium voltage - Harmonics

Предельные значения могут быть просмотрены через *Конфигурировать профили*. В примере ниже предельное значение для 3-ей гармоники находится на 5 %.

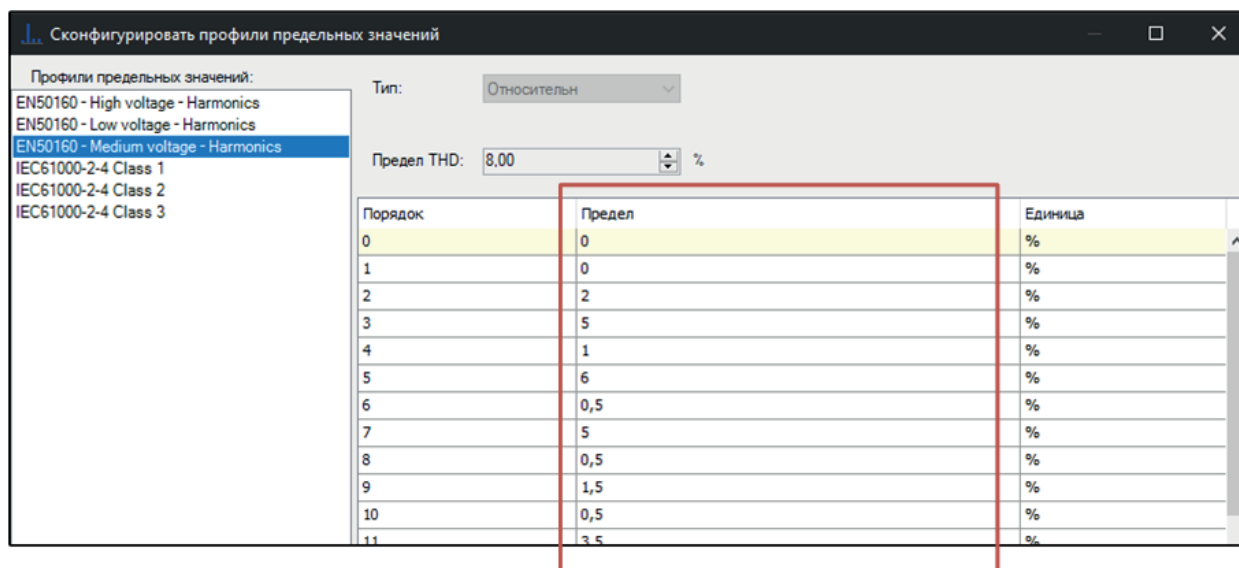


Рис. 40: Профиль предельного значения EN50160 - Medium voltage - Harmonics

Решение

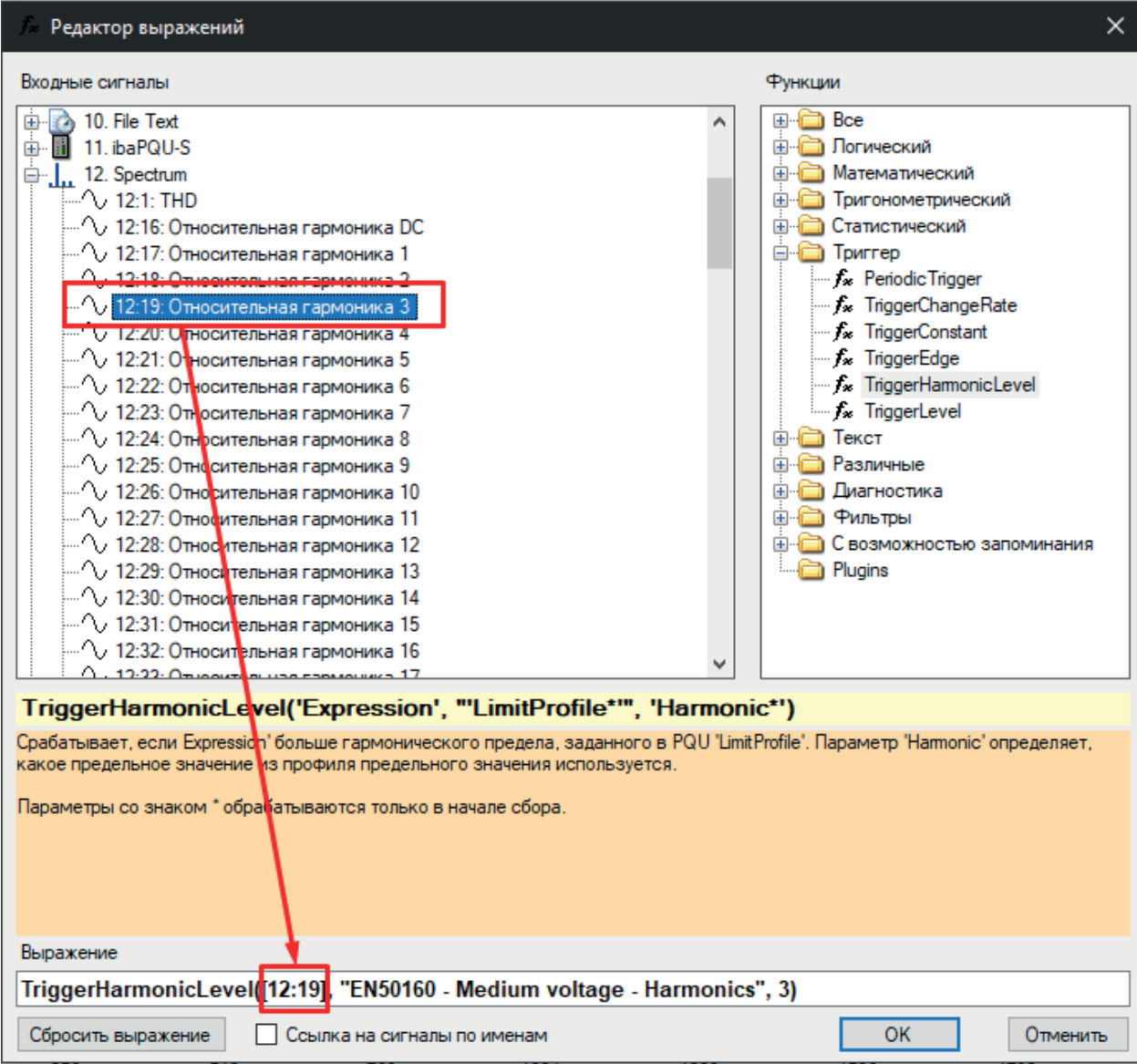


Рис. 41: Конфигурация выражения TriggerHarmonicLevel

Триггер срабатывает, если 10 минутное-значение 3-ей гармоники U12 находится выше 5%.

2.6 Текстовые функции

2.6.1 CharValue

CharValue('Text', 'CharNumber=0')

Аргументы

'Text'	Текстовый сигнал	
'CharNumber'	Позиция знака в тексте, по умолчанию = 0	

Описание

Данная функция возвращает десятичное значение ASCII знака в положении 'CharNumber' в тексте. Предварительная настройка возвращает первый знак (позиция = 0).

Пример: Если 'Text' имеет значение "A_Text für Test", то результатом CharValue('Text', 0) является значение ASCII 65 для буквы A.

Пример

Изображение названного в описании примера

Решение

Считывание десятичного значения ASCII первого знака в тексте "A_Text für Test". A имеет в таблице ASCII значение 65.



Рис. 42: Индикация результата CharValue в графике тренда

Совет



См. таблицу ASCII ниже <http://www.ascii-code.com/>

2.6.2 ConvertFromText

ConvertFromText('Expression', 'DecimalPoint*=0', 'Begin=0', 'End'=-1 (end of text)')

Аргументы

'Expression'	Имя текстового сигнала
--------------	------------------------

'DecimalPoint*'	Десятичный знак	
	DecimalPoint = 0	Точка
	DecimalPoint = 1	Запятая
'Beginn'	Индекс первого знака текста, уставка = 0	
'End'	Индекс последнего знака текста, уставка = -1 (конец текста)	

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Функция интерпретирует число с плавающей запятой из текста и возвращает числовое значение в виде аналогового сигнала. Если на первом месте заданного аргументами 'Beginn' и 'End' диапазона не стоит число, выводится 0. Исключение: если только есть пробелы до первого числового знака. В противном случае считывается текст до первого нечислового разряда или максимально до 'End'. Ведущие нули перед числом не должны прерываться пробелами или нечисловыми знаками.

Пример

Считывание заданного текста

Решение

Функция: ConvertFrom Tex('Text', 0,18,-1)

Содержание текстового сигнала: `Spannungssollwert:6.9 V`

Значение напряжения начинается в позиции 18. В качестве индекса конца используется -1, чтобы собирались также значения с несколькими разрядами до и/или после запятой.

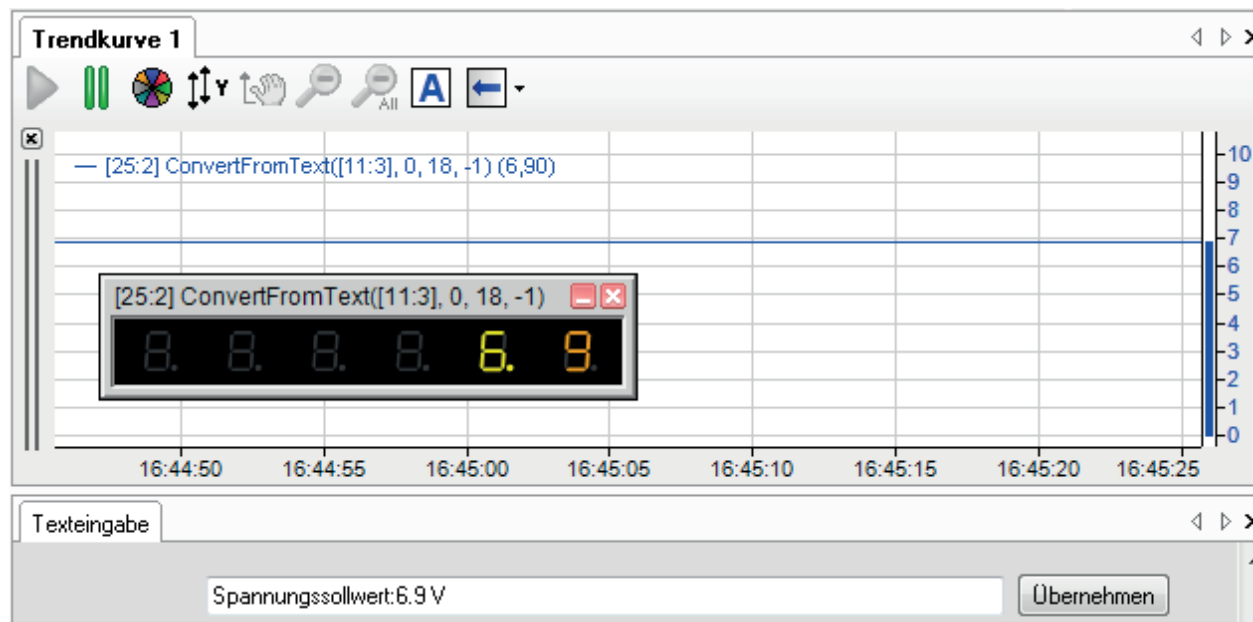


Рис. 43: Считывание числового значения (здесь: 6.9) из текстового сигнала

2.6.3 CountText

```
CountText('Text', 'CountOnlyDifferent*=0', 'Reset=0')
```

Аргументы

'Text'	Имя текстового сигнала	
	Опциональный параметр (уставка=0)	
	'CountOnlyDifferent' <> 0	Новый текст считается только, если он отличается от предыдущего.
	'CountOnlyDifferent' = 0	Считается каждый новый текст.
'Reset' (сброс)	Опциональный параметр, который может использоваться для сброса показаний счетчика. 'Reset' может также сам быть выражением.	
	'Reset' > 0	Счетчик сбрасывается
	'Reset' = 0	Активируется счет (уставка)

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция считает прием или изменение текстового сигнала и возвращает показание счетчика.

2.6.4 CompareText

```
CompareText('Text1', 'Text2', ' CaseSensitive=1*')
```

Примечание:

Изначально имя данной функции было TextCompare. Позднее оно было переименовано в CompareText. Условия совместимости позволяют использовать функцию TextCompare и далее, аргументы идентичны.

Аргументы

'Text1'	Первый текст для сравнения	
'Text2'	Второй текст для сравнения	
'CaseSensitive*'	Опциональный параметр (по умолчанию = 1) для определения, должно ли проводиться сравнение с учетом регистра (учет верхнего и нижнего регистра)	
	'CaseSensitive' = 1	Учитывать верхний и нижний регистр
	'CaseSensitive' <> 1	Игнорировать верхний и нижний регистр

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция сравнивает 2 текста друг с другом по алфавиту. Это значит, что знаки двух строк сравниваются друг с другом по-отдельности, начиная с левого знака. Функция использует текущие параметры культуры (Windows) для получения информации об орфографии и алфавитном порядке. Сравнение не учитывает ни многозначные числа, ни значение слов или длину строки. Пробел учитывается при сравнении. Опциональный параметр 'CaseSensitive=1' позволяет задать, следует ли учитывать верхний и нижний регистр (=1 или не указано), или нет (<>1).

Динамический текст может использоваться при помощи текстового сигнала. Текстовый сигнал указывается в прямоугольных скобках []. Возможно также вводить постоянный текст, просто указывая текст в кавычках.

Как результат функция выводит аналоговое значение.

Результаты

-1	Если знаки первого текста по алфавиту стоят перед знаками второго текста
0	если оба текста идентичны
1	Если знаки первого текста по алфавиту стоят после знаков второго текста

Пример

В таблице ниже приведены некоторые примеры:

Text1	Text2	Результат		Примечание
		CompareText ("Text1", "Text2", 0)	CompareText	

1234 abcd	1234 abcd	0	0	1 = 2
1234 abcd	1234 bcde	-1	-1	1 < 2 "a" стоит перед "b"
1234 Abcd	1234 abcd	0	1	1 = 2 (верхний/нижний регистр не учитываются) 1 > 2 (верхний/нижний регистр не учитывается) "A" стоит после "a"
12340 abcd	1234 _ abcd	1	1	1 > 2 "0" стоит после " _ "
1234 0abcd	1234 abcd	-1	-1	1 < 2 "0" стоит перед "a"
12034 abcd	1234 abcd	-1	-1	1 < 2 "0" стоит перед "3"
1234 abcd	1y34 abcd	-1	-1	1 < 2 "2" стоит перед "y"
1z34 abcd	1Y34 abcd	1	1	1 > 1 "z" стоит после "Y"

2.6.5 FindText

```
FindText('Text1', 'Text2', ' CaseSensitive=1*')
```

Аргументы

'Text1'	Первый текст для сравнения	
'Text2'	Второй текст для сравнения	
'CaseSensitive*'	Опциональный параметр (по умолчанию = 1) для определения, должно ли проводиться сравнение с учетом регистра (учет верхнего и нижнего регистра)	
	'CaseSensitive' <> 0	Учитывать верхний и нижний регистр
	'CaseSensitive' = 0	Игнорировать верхний и нижний регистр

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция проверяет, содержится ли 'Text2' в 'Text1' и возвращает как результат индекс (положение в 'Text1, первый знак = index 0), где 'Text2' появился в первый раз. Пробелы учитываются при поиске. С опциональным параметром 'CaseSensitive=1', можно определить, должен ли учитываться верхний или нижний регистр (=1 или не задано), или нет (<>1).

Динамический текст может использоваться при помощи текстового сигнала. Текстовый сигнал указывается в прямоугольных скобках []. Возможно также вводить постоянный текст, просто указывая текст в кавычках.

Как результат функция выводит аналоговое значение.

Результаты

-1	если второй текст не содержится в первом тексте
0....n	место, где 'Text2' встречается в 'Text1' первый раз

Пример

В таблице ниже приведены некоторые примеры:

Text1	Text2	Результат		Примечание
		FindText ("Text1","Text2",0)	FindText	
Светит Солнце	Солнце	4	4	
Светит луна	Солнце	-1	-1	
Светит солнце	Солнце	4	-1	

Примеры функции FindText

2.7 Различные функции

2.7.1 Count

Count('Expression', 'Level*', 'Hysteresis*', 'EdgeType*', 'Reset=0')

Аргументы

'Expression'	Измеренное значение	
'Level*' (уровень)	Значение уровня	
	Значение полосы гистерезиса	
'EdgeType*' (тип фронта)	Указание, должны ли учитываться передний или задний или передний и задний фронты	
	'EdgeType' <0	Только задние фронты (выход за пределы полосы гистерезиса в отрицательном направлении)
	'EdgeType' >0	Только передние фронты (выход за пределы полосы гистерезиса в положительном направлении)
	'EdgeType' = 0	передние и задние фронты
'Reset' (сброс)	Опциональный цифровой параметр, который может использоваться для сброса счетчика. 'Reset' может также быть выражением.	
	'Reset' > 0	Счетчик сбрасывается
	'Reset' = 0	Значение счетчика сохраняется/счетчик считает дальше (параметр)

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Важно



Условие 'Reset' не должно относиться к самой функции Count.

Описание

Данная функция считает, сколько раз 'Expression' пересекает уровень 'Level'.

При помощи параметра 'Hysteresis' можно указать полосу допуска, расположенную в равных частях над и под 'Level'. Считаются только полные проходы через полосу допуска.

Параметр 'EdgeType' определяет, какие фронты считаются.

Параметр 'Reset' служит для сброса значения счетчик на 0. 'Reset' может быть также выражением.

Примеры:

Count([0:0],10,1,1)	Сброс не осуществляется ('Reset' пропущен)
Count([0:0],10,1,1,If(Mod(T(),20)=0,TRUE(),FALSE()))	Счетчик сбрасывается с периодичностью в 20 с.
Count([0:0], [3.1])	например, с [3.1] = If([0:0]<1, 1, 0) Счетчик сбрасывается, если выражение [3.1] возвращает ИСТИНА, т.е. Если выражение [0:0] не достигает предельное значение 1

Совет

Функция COUNT может также использоваться для бинарных сигналов. Для этого ввести как уровень 0.5 и как гистерезис, например, 0.1. При этом учитывается и считается каждая смена с ЛОЖЬ на ИСТИНА и наоборот.

Пример**Совет**

Данная функция может использоваться в виртуальном модуле с запоминанием. Ее результирующие значения можно получить, остановив или перезапустив измерение.

Необходимо посчитать количество отклонений измеренного сигнала от значения 10 в течение 20 секунд в обоих направлениях, но только после полного пересечения полосы гистерезиса ширины 2. При этом максимальное значение каждого интервала должно сохраняться без фазы пуска.

Задача

Параметр 'Reset' функции Count состоит из If-запроса функции Modulo времени. Максимальное значение сбрасывается однократно с функцией OneShot, как только измеренное значение стабилизируется в начале измерения.

Решение

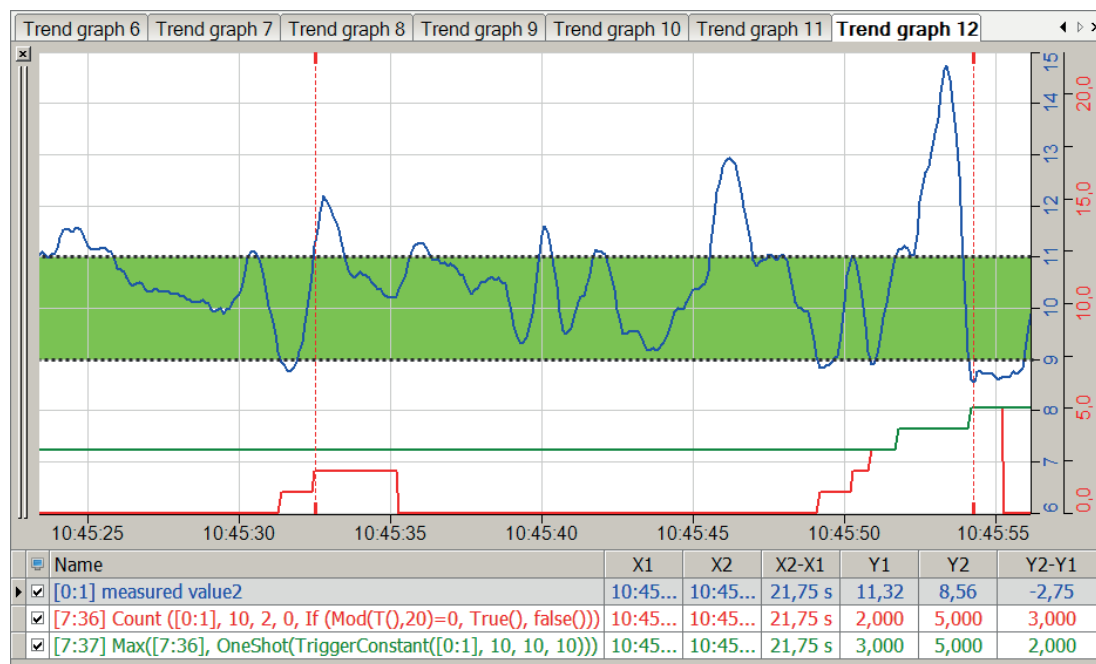


Рис. 44: Синий график: измеренное значение; зеленый диапазон: Полоса гистерезиса; красный сигнал: Функция Count; зеленый сигнал: Максимальное значение интервалов функции Count

Совет



При указании полосы гистерезиса 2 вокруг 'Level' 10, пересечения в возрастающем направлении считаются сначала для 'Expression' > 11 и в нисходящем направлении для 'Expression' < 9.

2.7.2 Delay

`Delay('Expression', 'NumberSamples*')`

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает копию сигнала 'Expression' с задержкой. Задержка указывается в количестве измерений ('NumberSamples'). Результатом является график сигналов со значениями оригинального сигнала в 'NumberSamples' перед текущим временем.

Чтобы избежать перегрузки памяти, 'NumberSamples' следует ограничить до 10 000.

Совет



При различном опорном времени измеренного значения и функции опорное время функции релевантно для количества измерений.

Пример

Измеренное значение должно выводиться с задержкой на 2 измеренных значения и на 10 измеренных значений.

Задача

При выбранном опорном времени 100мс задержка на 2 измеренных значения соответствует времени 0,2 секунды и задержка на 10 измеренных значений - 1 секунде.

Решение

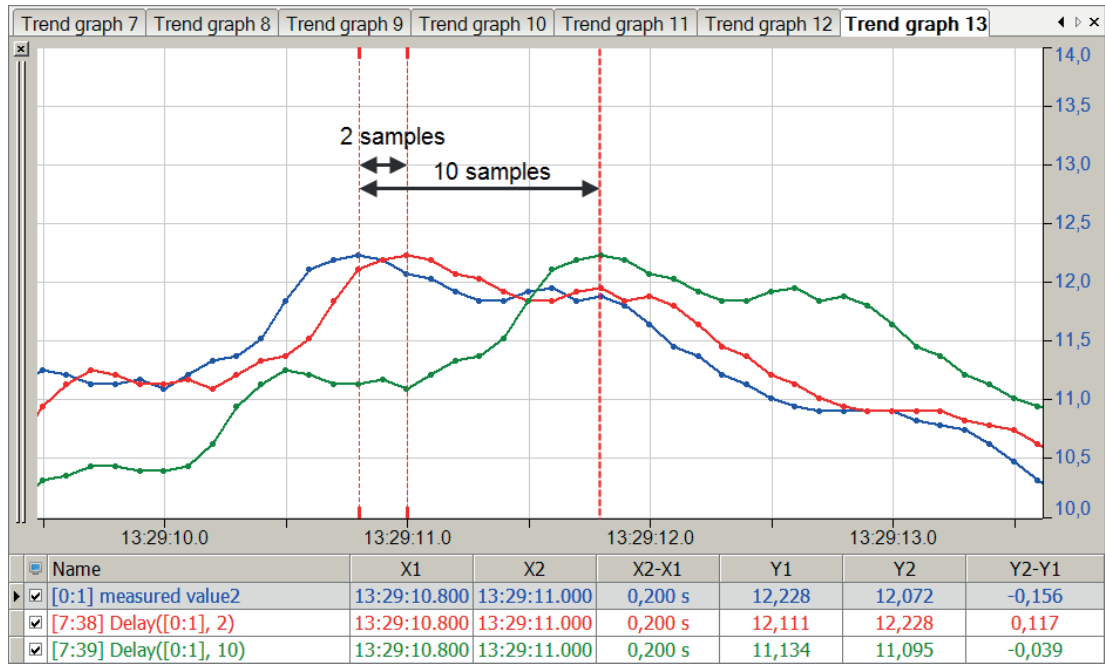


Рис. 45: Синий график: исходный сигнал; красный график: задержка сигнала на 2 измеренных значения (=0,2секунды); зеленый график: задержка сигнала на 10 измеренных значений (=1 секунда)

2.7.3 DelayLengthL

```
DelayLengthL('Expression', 'Length', 'MaxLengthDelta', 'DelayInMeter', 'Resolution*', 'Filter=0*')
```

Аргументы

'Expression'	Входной сигнал	
'Length'	Сигнал длины (в м)	
	Верхний предел для учета изменений сигнала длины	
'DelayInMeter'	Замедление (в м)	
'Resolution*'	Разрешение База длины результата (в м)	
'Filter*'	Опциональный параметр (уставка=0) для настройки фильтра при преобразовании время-длина	
	'Filter' = 1	Минимальный фильтр
	'Filter' = 2	Максимальный фильтр
	'Filter' = 0 и другие	Без фильтра

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция создает через сигнал скорости 'Длина' (в м/с) версию аргумента 'Expression' на базе длины с задержкой 'DelayInMeter' метров. Изменения в сигнале длины, превышающие 'MaxLengthDelta', игнорируются. Разрешение - это опорная длина результата (в м).

'Filter' задает фильтр при преобразовании время-длина.

Прежде всего для мониторинга данных о качестве в сочетании с ibaQDR измеренные сигналы удаленных источников можно сравнивать относительно длины продукта. Расстояние между источниками сигнала определяет значение 'DelayInMeter'.

2.7.4 DelayLengthV

`DelayLengthV('Expression', 'Speed', 'DelayInMeter', 'Resolution*', 'Filter=0*')`

Аргументы

'Expression'	Входной сигнал	
'Speed'	Сигнал скорости (в м/с)	
'DelayInMeter'	Замедление (в м)	
'Resolution*'	Разрешение База длины результата (в м)	
'Filter*'	Опциональный параметр (уставка=0) для настройки фильтра при преобразовании время-длина	
	'Filter' = 1	Минимальный фильтр
	'Filter' = 2	Максимальный фильтр
	'Filter' = 0 и другие	Без фильтра

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция создает через сигнал скорости 'Speed' (в м/с) версию аргумента 'Expression' на базе длины с замедлением 'DelayInMeter' метров. Разрешение - это опорная длина результата (в м).

'Filter' задает фильтр при преобразовании время-длина.

2.7.5 DWORD

DWORD('Low', 'High')

Аргументы

'Low'	16 bit Integer: Word	
'High'	16 bit Integer: Word	

Описание

Данная функция возвращает как результат тип 32 bit Integer DWORD, состоящий из int16 WORDS 'Low' и 'High'.

Пример

Вычисление двух DWORDS из простой комбинации 0 и 1 для 'Low' и 'High'.

	DWORD, состоящий из int16 WORDS			
DWORD('Low','High')	int16 'High'	int16 'Low'	Пе- ресчет	Десят.
DWORD(0, 1)	0000 0000 0000 0001	0000 0000 0000 0000	216	65536
DWORD(1, 0)	0000 0000 0000 0000	0000 0000 0000 0001	20	1

Решение

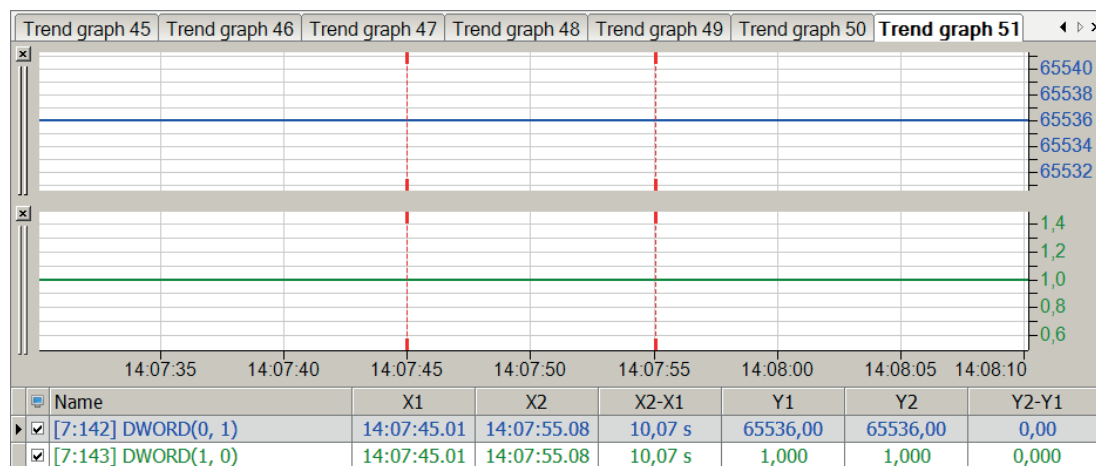


Рис. 46: Вычисление двух простых DWORDS

2.7.6 Eff

Eff('Expression', 'Frequency')

Описание

Данная функция вычисляет эффективное значение 'Expression' с основной частотой 'Frequency'.

Для вычисления эффективного значения используется следующая формула.

$$E_{\text{eff}} = \sqrt{\frac{1}{N} \sum_{n=1}^N e^2(n)}$$

$e(n)$: Messpunkt n von Signal e ('expr')

N : Anzahl der Messwerte pro Periode

Важно



Эффективное значение выводится только в следующем такте основной частоты.

Пример

Для графика переменного напряжения с частотой 0,1 кГц с наложением второго переменного напряжения 0,5 кГц требуется рассчитать эффективное значение напряжения для обоих частот.

Решение

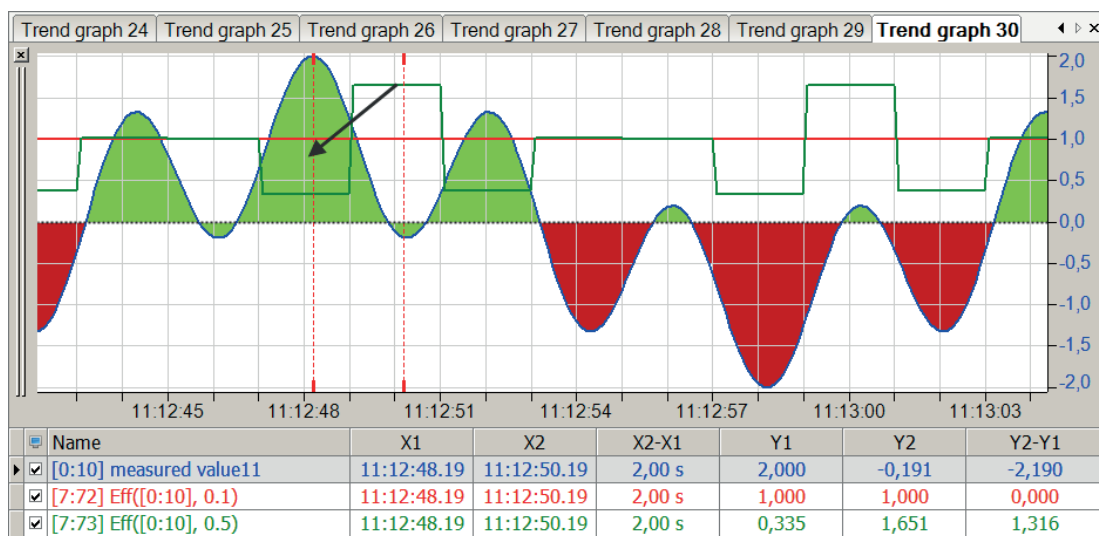


Рис. 47: Синий график: Сигнал переменного напряжения; зеленый график: Эффективное напряжение с расчетной частотой 0,5 кГц; красная прямая: Эффективное напряжение с расчетной частотой 0,1 кГц

2.7.7 ElapsedTime

`ElapsedTime('Start','Stop')`

Описание

Данная функция возвращает время, прошедшее с момента последнего переднего фронта в 'Start'. Врем останавливается при переднем фронте в 'Stop'.

Пример

Измерение интервала времени между последним передним фронтом измеренного значения и следующим передним фронтом другого измеренного значения.

Решение

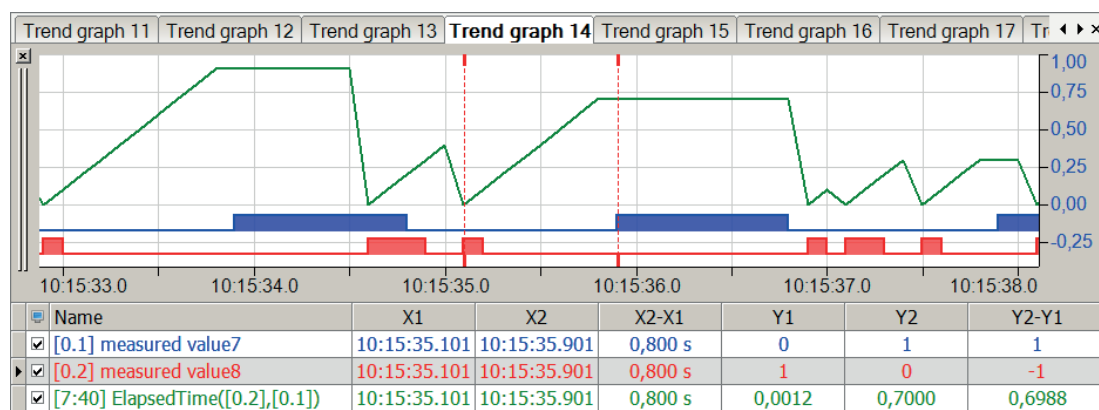


Рис. 48: Синяя диаграмма: Измеренное значение 'Stop'; красная диаграмма: Измеренное значение 'Start'; зеленый график: временной отрезок между передним фронтом измеренного значения 'Start' до измеренного значения 'Stop'

Важно



За счет переключения фронтов в области точек измерений графическое изображение и расчет истекшего времени может отклоняться на одну точку измерения.

2.7.8 ExecuteCommand

`ExecuteCommand('Trigger','"Command"', '"Arguments"')`

Аргументы

'Trigger'	Бинарный сигнал или выражение, запускающие выполнение команды
""Command""	Командная строка выполняемой команды/файла (.exe, .bat и т.д.)
""Arguments""	Требуемые для выполнения команды аргументы/параметры

Описание

Данная функция выполняет командную строку 'Command' 'Arguments' при переднем фронте триггера 'Trigger'. Возвращаемое значение - 1, пока идет запущенный процесс. Другие передние фронты игнорируются, пока идет процесс.

В "'Command'" необходимо указать полный, абсолютный путь к выполняемому файлу.

Важно



Команда выполняется службой *ibaPDA*. Поскольку служба *ibaPDA* работает, как правило, под системной учетной записью (пользователь: SYSTEM), команда также выполняется под системной учетной записью. Тогда на пульте текущего пользователя пользовательский интерфейс запущенной программы (если есть) не отображается. Если Вы хотите при помощи данной функции запустить программу с отображаемым интерфейсом, например *ibaAnalyzer*, то Вам необходимо присвоить программу текущему сеансу пользователя. Для этого доступны вспомогательные средства, такие как PsExec, см. пример 2 ниже.

Пример 1

Завершите работу компьютера

Задача

В зависимости от сигнала из системы управления в связи с переходом на источник бесперебойного питания при сбое в подаче питания должно произойти контролируемое отключение компьютера с *ibaPDA*- (безопасное завершение работы).

Решение

Для завершения работы компьютера используется команда «shutdown» из командной оболочки Windows. Т.к. требуются несколько команд, такие как останов службы *ibaPDA*, используется пакетный файл.

```
ExecuteCommand([4.14], "D:\Schulung\ibaPDA_ExecuteCommand\Shutdown_PC.bat", "")
```

- [4.14]: Цифровой триггерный сигнал для выполнения команды
- D:\Schulung\ibaPDA_ExecuteCommand\Shutdown_PC.bat: Путь к программе для пакетного файла

Содержимое пакетного файла Shutdown_PC.bat:

1	SETLOCAL
2	sc stop ibaPDAService
3	shutdown /f /s /t 10
4	ENDLOCAL

Пример 2

Запустить ibaAnalyzer с отображением на экране пользователя

Задача

При переднем фронте бинарного сигнала [4.17] должна завестись программа ibaAnalyzer.

Решение

Т.к. программа ibaAnalyzer выполняется под системной учетной записью и запускается на пульте текущего пользователя без визуального отображения, то ее необходимо перевести на пульт пользователя.

Для этого может использоваться программа *Psexec*.

<https://docs.microsoft.com/de-de/sysinternals/downloads/psexec>

В данном случае тогда PsExec является выполняемой командой и программа *ibaAnalyzer* относится к аргументам команды.

```
ExecuteCommand([4.17], "d:\software\Pstools\Psexec.exe", "-i 1  
-accepteula c:\Programme\iba\ibaAnalyzer\ibaAnalyzer.exe")
```

- [4.17]: Цифровой триггерный сигнал для выполнения команды
- d:\software\Pstools\Psexec.exe: Программный путь для PsExec
- -i 1: Перевод запускаемой программы на сеанс пользователя № 1
(в Windows 7 номер сеанса пользователя можно найти в диспетчере задач, вкладка «Пользователь».)
- -accepteula: Переход к лицензионному соглашению
- c:\Programme\iba\ibaAnalyzer\ibaAnalyzer.exe: Путь к программе ibaAnalyzer

2.7.9 ExtendPulse

`ExtendPulse('Input', 'Time*')`

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция удлиняет импульс на 'Input' на минимальную длину 'Time' секунд. Последующий передний фронт в 'Input' перезапускает время.

Пример

Удлинение импульсов измеренного значения на минимальную длину 0,3.

Решение

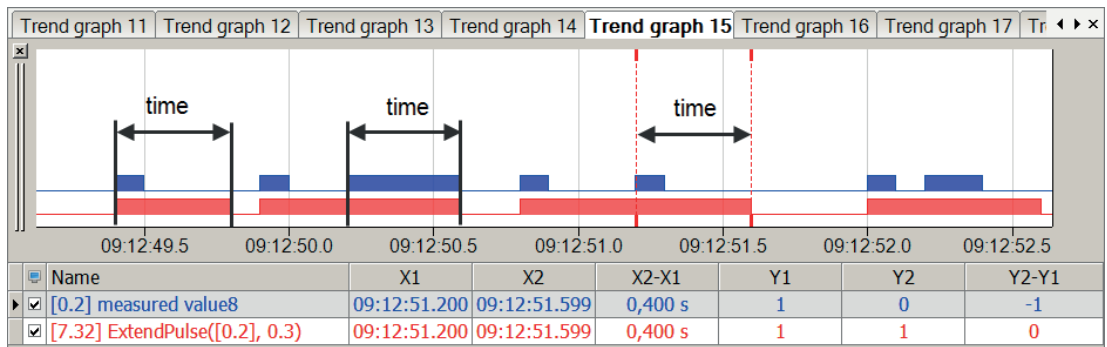


Рис. 49: Синяя диаграмма: Измеренное значение; красная диаграмма: удлиненные импульсы измеренного значения

Важно

i При переключении фронтов в сетке измеренных значений графическое изображение и расчет минимальной длины могут находиться на одну точку измерения над минимальной длиной, указанной в 'Time'.

2.7.10 GenerateSignal

```
GenerateSignal('Type', 'Amplitude=10', 'T1=1', 'T2=1')
```

Аргументы

'Type'	Тип сигнала	
	'Type' = 0	Функция синус, A=амплитуда и T1=длительность периода
	'Type' = 1	Функция косинус, A=амплитуда и T1=длительность периода
	'Type' = 2	Треугольная функция, A=амплитуда, T1 = время переднего фронта, T2 = время заднего фронта
	'Type' = 3	Прямоугольная функция, A=амплитуда, T1 = время верхнего фронта, T2 = время нижнего фронта
	'Type' = 4	Случайный сигнал с A=максимальной амплитудой
'Amplitude'	Опциональный параметр, указание амплитуды; уставка = 10	
'T1'	Опциональный параметр, время 1 (кроме типа 4); уставка=1	
'T2'	Опциональный параметр, время 2 (только для типа 2 и типа 3); уставка=1	

Описание

Данная функция генерирует тестовый сигнал с амплитудой 'Amplitude' и промежутками времени 'T1' и 'T2'. Могут быть сгенерированы следующие типы сигналов:

Пример

Различные примеры для сгенерированных сигналов

Решение

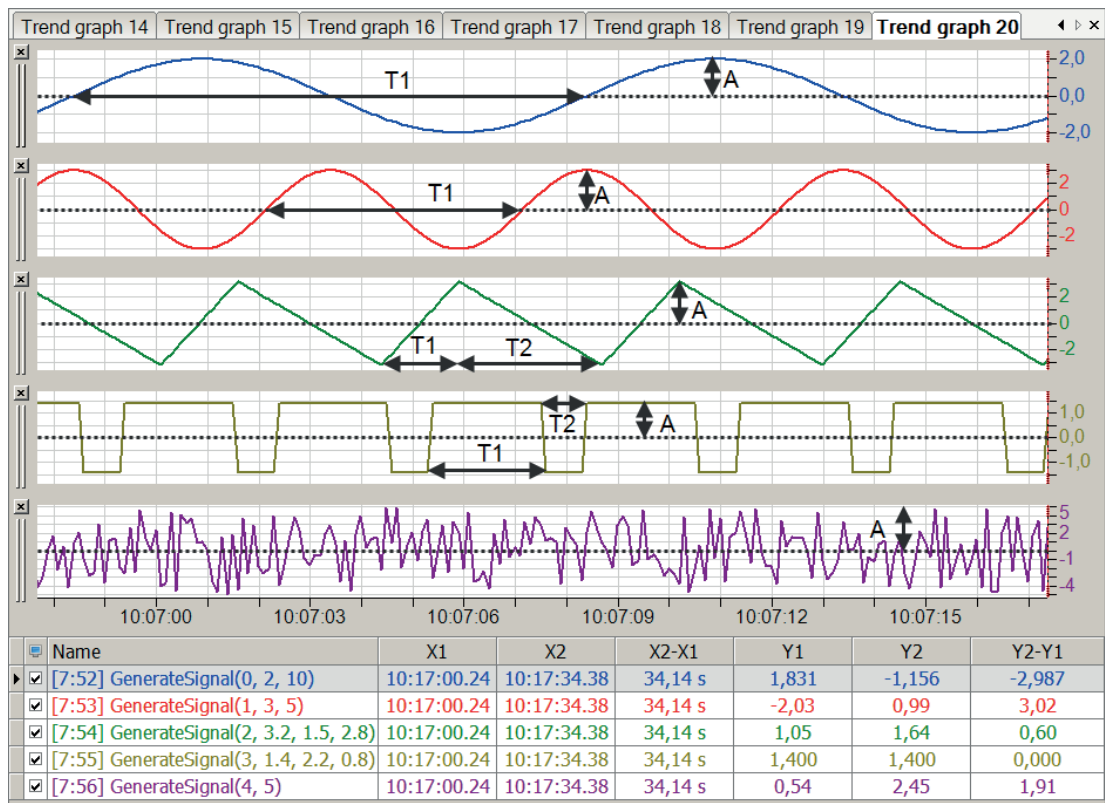


Рис. 50: Синий: Функция синус тип 0; красная: Функция косинус тип 1; зеленая: Треугольная функция тип 2; желтая: Прямоугольная функция тип 3; лиловая: Случайная функция тип 4

2.7.11 GetFloatBit

```
GetFloatBit('Expression','BitNo')
```

Аргументы

'Expression'	Любое число
'BitNo'	Данное число, от 0 до 15, или от 0 до 31 указывает положение желаемого цифрового сигнала в 16-битовом или 32-битовом блоке в потоке данных, относительно соответствующего адреса смещения. Повышение номера бита от 1 до 15 (31), то повышение адреса на 2 (4).

Описание

Данная функция интерпретирует 'Expression' как битовую маску значения float и выдает как результат Булево значение бита 'BitNumber' выражения 'Expression'. Действительная последовательность битового номера: от 0 (LSB) до 31 (MSB).

Данная функция была специально разработана на случай, если 32 бита упаковываются как данные float для передачи/записи. Функция GetBitFloat оценивает только валентность указанного бита 'BitNo', независимо от того, является ли он частью мантиссы или порядка. В отличие от функции GetBit, значения не округляются.

Пример

Указанное в 'Expression' значение преобразуется в формат с плавающей запятой согласно IEEE 754 и извлекаются разряды, которые для этого переходят в ИСТИНА.

Решение

Числа с плавающей запятой в соответствии с форматом IEEE 754 могут записываться следующим образом: (знак)*мантисса*10^порядок

Число 0,15625 как число с плавающей запятой в соответствии с IEEE 754 выглядит следующим образом:

знак (1 бит)	порядок (8 бит)	мантисса (23 бит)
0	0111 1100	0100 0000 0000 0000 0000 000

При этом LSB (Least Significant Bit - младший двоичный разряд) - в крайнем правом положении.

Совет



Числа с плавающей запятой в соответствии с форматом IEEE 754 записываются изображенным образом. При этом в отдельных компонентах необходимо учитывать следующие пункты:

Знак: 0 - для положительных; 1 - для отрицательных

Порядок: Т.к. при пересчете в число с плавающей запятой значение порядка смещается на 127, 127 нужно опять вычесть из изображенного здесь порядка при пересчете в десятичное значение.

Мантисса: так как в мантиссе стоит обычно 1 перед запятой, этот разряд больше не отображается. Поэтому все знаки мантиссы - знаки после запятой и при пересчете должны учитываться с отрицательными порядками к базе 2.

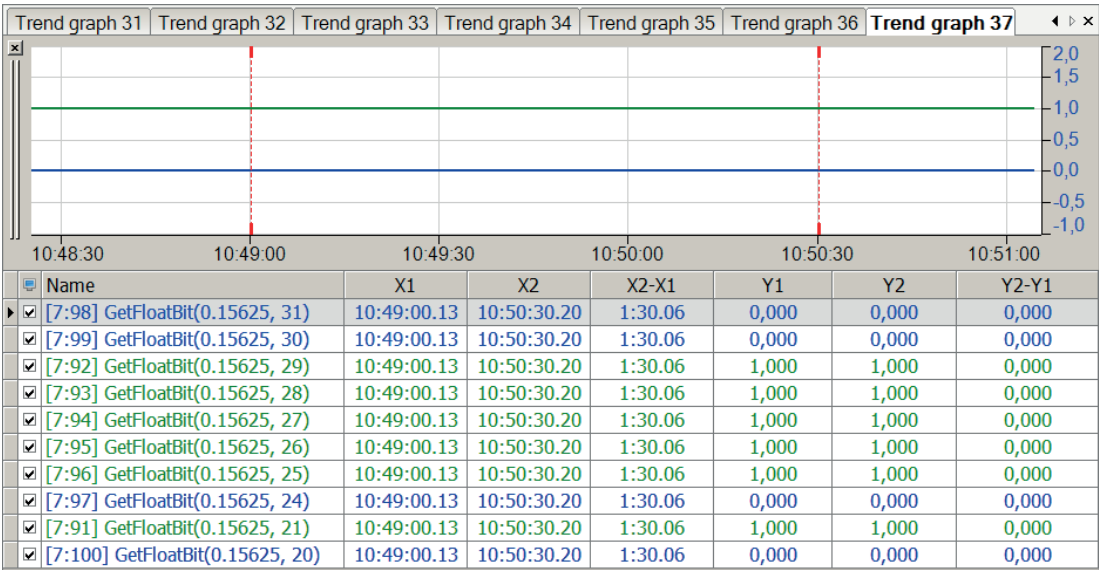


Рис. 51: Извлечение валентности выбранных битов значения float 0,15625

2.7.12 GetIntBit

```
GetIntBit('Expression', 'BitNo')
```

Описание

Данная функция возвращает как результат булево значение бита 'BitNo' выражения 'Expression' после округления 'Expression' до ближайшего целого значения. Округление осуществляется соответственно с шагом 0,5. (2,49 → 2; 2,50 → 3).
Действительная последовательность номеров бита: от 0 (LSB) до 31 (MSB)

Пример

Изображение времени в булевых значениях

Решение

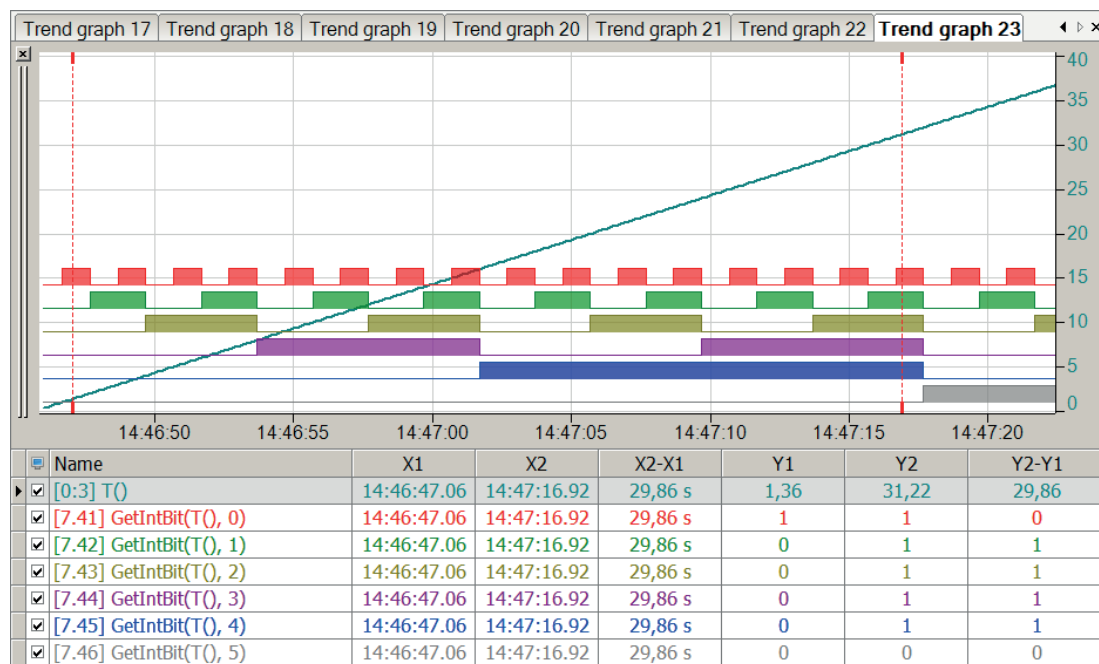


Рис. 52: Функция GetIntBit первых шести битов функции времени T()

2.7.13 GetSystemTime

```
GetSystemTime('Part*')
```

Аргументы

'Part*'	Отображаемая часть системного времени	
	'Part' = 0	Милисекунды
	'Part' = 1	Секунды
	'Part' = 2	Минуты
	'Part' = 3	Часы
	'Part' = 4	День месяца
	'Part' = 5	Месяц
	'Part' = 6	Год
	'Part' = 7	День года
	'Part' = 8	День недели (понедельник=1, воскресенье=7)

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает указанную в 'Part' часть системного времени.

Совет



В *ibaQPanel* день недели (часть 8) может быть изображен в виде названия при помощи текстовой индикации нескольких состояний. Для других частей обычного обозначения вполне достаточно. Используйте индикатор в виде циферблата для отображения в *ibaQPanel* времени как на аналоговых часах.

Пример

Вывод минут, секунд и миллисекунд

Решение

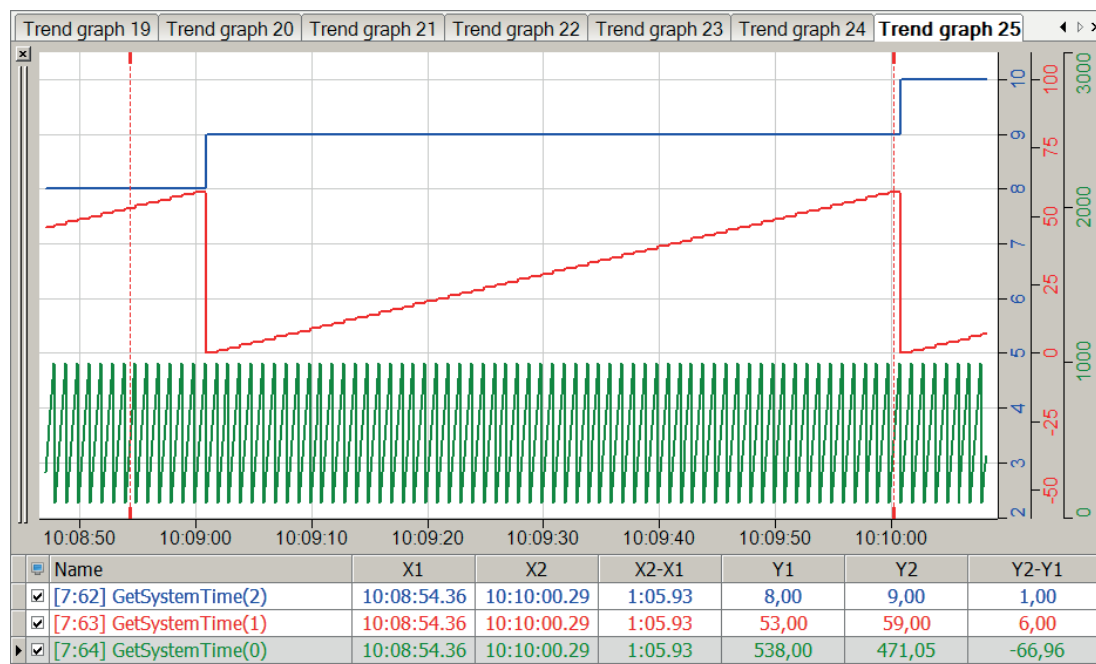


Рис. 53: Синий: минуты системного времени ('part'=2); красный: секунды системного времени ('part'=1); зеленый: миллисекунды системного времени ('part'=0)

2.7.14 LimitAlarm

LimitAlarm('Expression', 'Limit', 'DeadBand', 'Time', ' Reset=0')

Аргументы


'Expression'	Измеренное значение	
'Limit'	Предельное значение, с которого функция возвращает ИСТИНА	
'DeadBand'	Указание зоны нечувствительности ниже верхнего предельного значения, в пределах которого функция не сбрасывается на ЛОЖЬ.	
'Time'	Время, в течение которого измеренное значение должно быть выше предельного значения, пока функция станет ИСТИНОЙ	
'Reset'	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления	
	'Reset' = 0	Выполнить вычисление
	'Reset' = 1	Остановить, сбросить вычисление и установить результат на 0
	'Reset' = 2	Остановить, сбросить вычисление и сохранить результат

Описание

Данная функция контролирует измеренное значение ('Expression') и устанавливает результат на ИСТИНА, если измеренное значение больше указанного времени ('Time') выше предельного значения ('Limit'). Результат функции будет снова ЛОЖЬ, если измеренное

значение не достигает предельное значение на указанное в зоне нечувствительности ('DeadBand') значение.

Совет



Функция LimitAlarm может использоваться и для нижнего предельного значения. Для этого должны зеркалироваться только измеренное значение и предельное значение, т.е. умножаться на (-1).

Например, `LimitAlarm([0:1] *(-1), 9 *(-1), 0.5, 0.4)`

Пример

Функция должна возвращать ИСТИНА, если измеренное значение дольше 0,4 секунд выше значения 11. Функция должна возвращать снова ЛОЖЬ, когда измеренное значение опускается ниже 10,5.

Решение

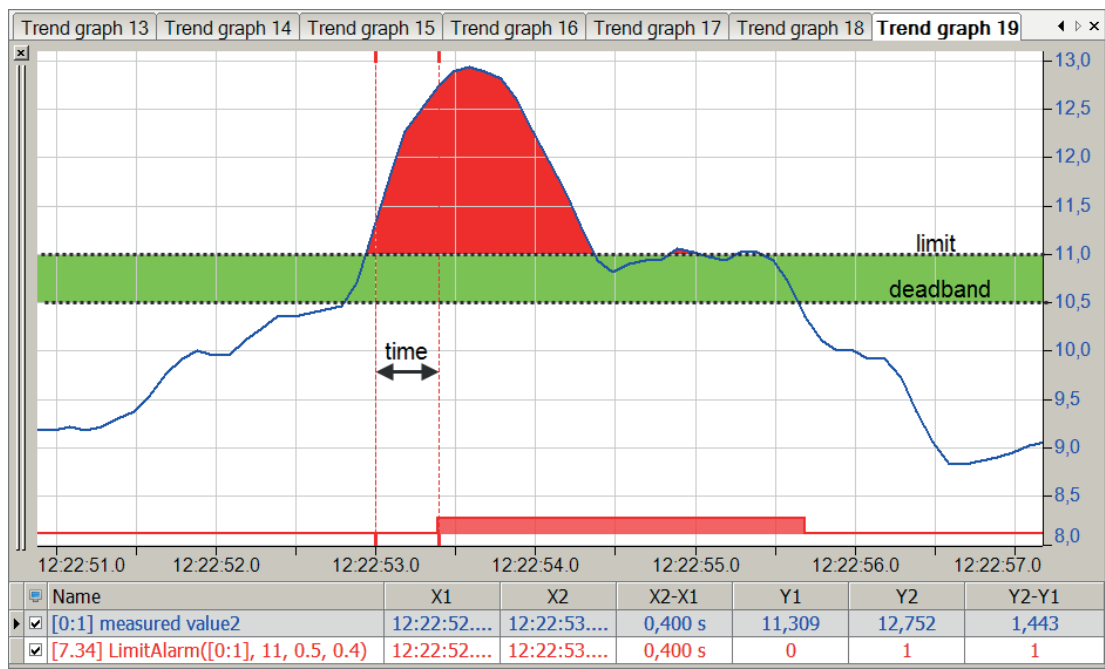


Рис. 54: Синий график: измеренное значение; красная диаграмма: Превышение предельного значения; зеленая полоса: зона нечувствительности, которая препятствует прямой установке функции на ЛОЖЬ.

2.7.15 PulseFreq

```
PulseFreq('Expression', 'Omega=0*', 'EdgeType=2*' )
```

Аргументы

'Expression'	Сигнал счетчика импульсов	
'Omega*'	Частота фильтра	
'EdgeType*'	Считаемый тип фронта	
	'EdgeType' = -1	Только задние фронты
	'EdgeType' = 0	Передний и задний фронты
	'EdgeType' = 1	Только передние фронты
	'EdgeType' = 2	'Expression' является счетчиком импульсов

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция вычисляет частоту импульсов или счетчиков импульсов 'Expression'. Единица измерения результата - импульсы/с или Гц.

Фильтр нижних частот с предельной угловой скоростью 'Omega' применяется к результату. Если 'Omega' 0, то фильтр нижних частот деактивирован. 'EdgeType' определяет, какие фронты импульсов должны учитываться.

Как вычисленная частота возвращается нуль, если за 1000 считывания не появляется ни одного импульса.

Данная функция создана специально для анализа инкрементального датчика WAGO 750-631. Данная функция может использоваться для вычисления скорости по состоянию счетчика импульсов. Состояние счетчика импульсов дифференцируется с учетом возможного переполнения. Т.к. результат данной дифференциации может быть с частотами помех или шумами, к нему применяется фильтр нижних частот. Настраиваемая частота фильтров должна быть немного выше максимальной частоты импульсов.

2.7.16 SampleAndHold

```
SampleAndHold('Expression', 'Sample', 'Initial=0)
```

Аргументы

'Expression'	Измеренное значение
'Sample'	Параметр, который определяет, вытекает ли функция из измеренного значения (1) или сохраняет последнее измеренное значение (0). 'Sample' может быть сам условием или задаваться другой функцией.
'Initial'	Опциональный параметр (уставка=0), который задает исходное значение функции, если в начале измерения 'Sample' неактивен.

Описание

Данная функция является функцией выборки/хранения. Вывод вытекает из 'Expression', если 'Sample' = TRUE. Он остается без изменений, если 'Sample' = FALSE. Опциональный параметр 'Initial' позволяет указывать исходное значение выхода, если при вызове функция стоит на «Сохранить».

Пример

Функция должна вытекать из измеренного значения, если оно положительное и удерживаться, если измеренное значение отрицательное.

Задача

If-запрос как параметр 'Sample' выводит значение 1 для положительных измеренных значений значение 0 для отрицательных измеренных значений.

Решение

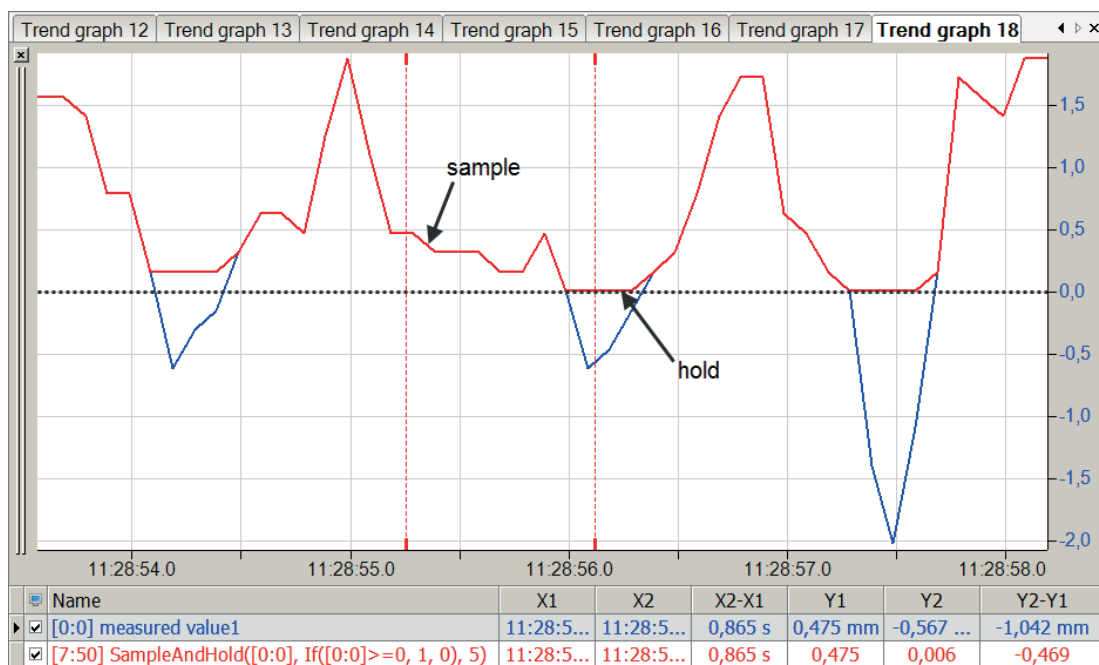


Рис. 55: Синий график: Измеренное значение, красный график: Выходное значение с сохраненными измеренными значениями

2.7.17 Sign

Sign('Expression')

Описание

Данная функция выводит как результат знак перед выражением 'Expression'.

'Expression' > 0 --> +1

'Expression' = 0 --> 0

'Expression' < 0 --> -1

Пример

Релевантен только знак измеренного значения.

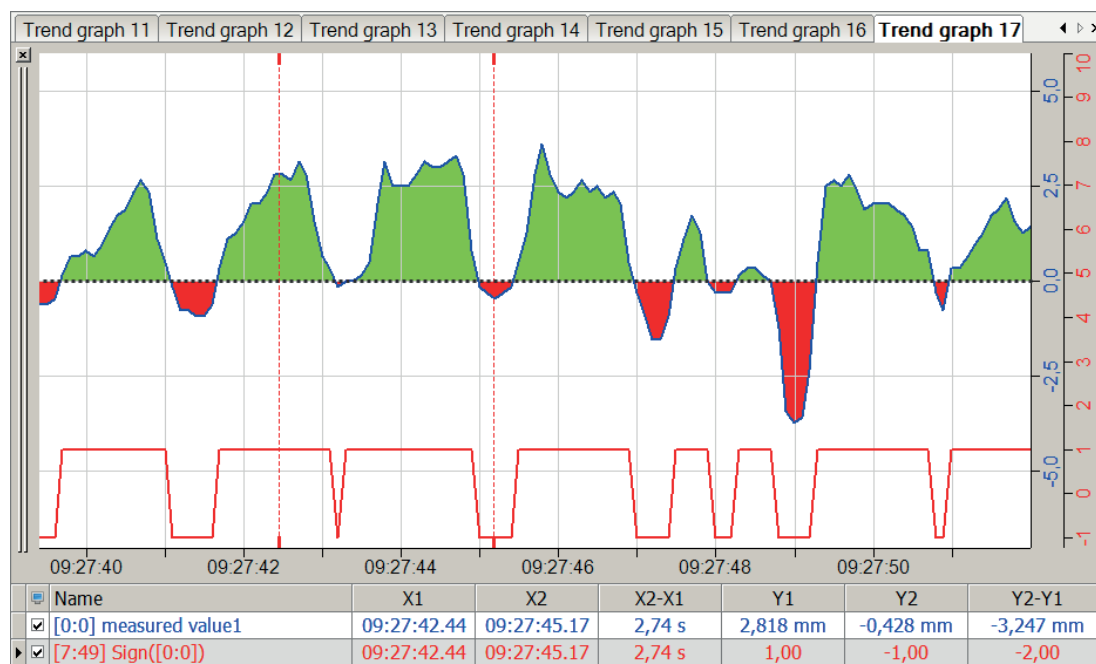
Решение

Рис. 56: Синий график: Измеренное значение, красный график: Знак перед измеренным значением

2.7.18 T

T ()

Описание

Данная функция возвращает прошедшее с момента начала измерения (в секундах).

Функция времени позволяет вычислять зависимые от времени, виртуальные величины, например синусоиды.

Пример синусоиды при 0,5 Гц: $\sin(2 \cdot \pi() \cdot 0,5 \cdot T())$

Совет

Функция времени может быть добавлены в другие выражения просто двойным щелчком, как в сигнале. Скобки после буквы T обязательны в связи с синтаксисом, точно как с числом Pi.

Пример

Изображение функции времени и зависимой от времени виртуальной величины

Решение

Вычисление функции времени и зависимой от времени синусоиды

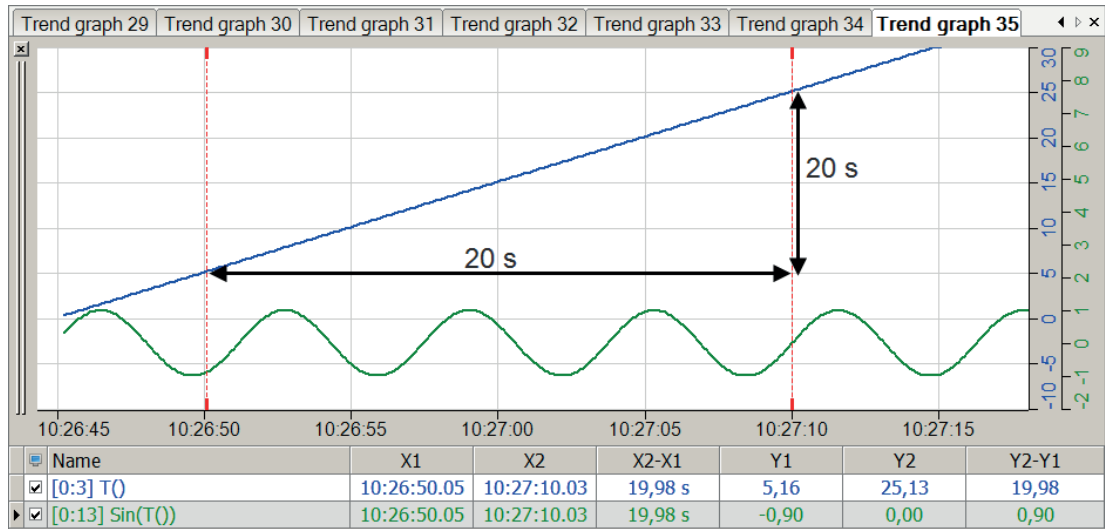


Рис. 57: Изображение функции времени и зависимой от времени синусоиды

2.7.19 VarDelay

```
VarDelay('Expression','Delay', 'MaxDelay=30*')
```

Аргументы

'Expression'	Входной сигнал
'Delay'	Время задержки в секундах
'MaxDelay*'	Опциональный параметр (уставка=30) для определения максимально допустимой задержки в секундах

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция задерживает сигнал 'Expression' на время задержки 'Delay'. В отличие от функции Delay время задержки может со временем изменяться. 'MaxDelay' указывает максимально допустимую задержку и настроена по умолчанию на 30 с.

Пример

Переменная по времени задержка сигнала при достижении максимально допустимой задержки.

Решение

Как изменяемая задержка используется функция времени T(). Она сама и синусоидальный сигнал задаются с данной переменной задержкой, и записываются графики результатов. Как только достигнута максимально допустимая задержка, задержка остается постоянной, и графики результатов отображают только лишь выходные сигналы.

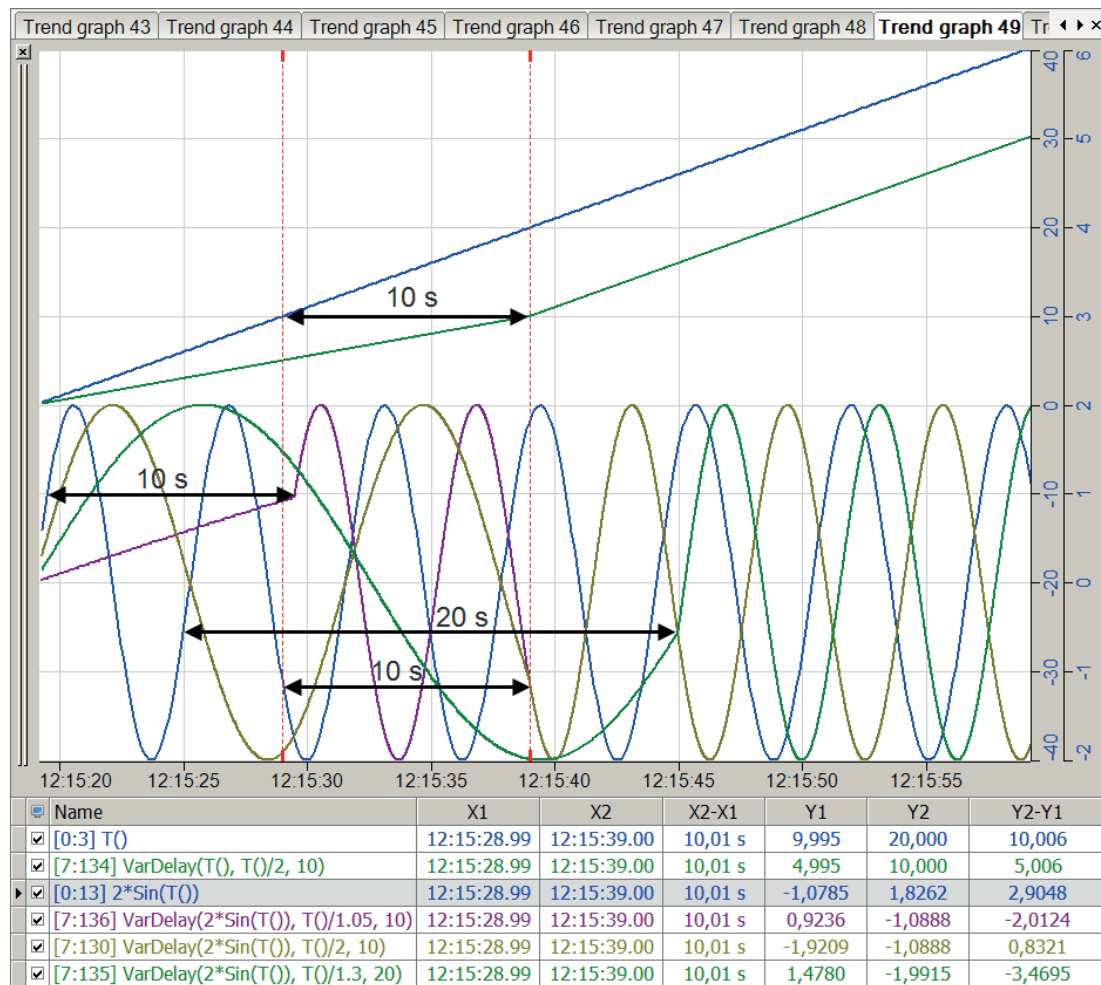


Рис. 58: Вверху: Задержка сигнала времени на переменное смещение с максимальным значением 10 секунд; внизу: Задержка синусоидальной функции с тремя различными значениями задержки и максимальными значениями задержки.

Совет



Отрицательные значения для 'Delay' не выводят сообщение об ошибке, а обрабатываются как 0, т.е. задержки нет.

2.7.20 WindowAlarm

```
WindowAlarm('Expression', 'Limit1', 'DeadBand1', 'Limit2', 'DeadBand2', 'Time', 'Reset=0')
```

Аргументы

'Expression'	Измеренное значение	
'Limit1'	Верхнее предельное значение, с которого функция возвращает ИСТИНА	
'DeadBand1'	Указание зоны нечувствительности ниже верхнего предельного значения ('Limit1'), в пределах которого функция не сбрасывается на ЛОЖЬ.	
'Limit2'	Нижнее предельное значение, с которого функция возвращает ИСТИНА	
'DeadBand2'	Указание зоны нечувствительности выше нижнего предельного значения ('Limit1'), в пределах которого функция не сбрасывается на ЛОЖЬ.	
'Time'	Время, в течение которого измеренное значение должно быть больше верхнего предельного значения и меньше нижнего предельного значения	
'Reset'	Опциональный параметр (уставка=0) для остановки и повторного запуска вычисления	
	'Reset'=0	Выполнить вычисление
	'Reset'=1	Остановить, сбросить вычисление и установить результат на 0
	'Reset'=2	Остановить, сбросить вычисление и сохранить результат

Описание

Данная функция контролирует измеренное значение ('Expression') и устанавливает результат на ИСТИНА, если измеренное значение больше указанного времени ('Time') находится за пределами диапазона между верхним предельным значением ('Limit1') и нижним предельным значением ('Limit2'). Результат функции становится снова ЛОЖЬ, если измеренное значение не достигает верхнее предельное значение на указанное в зоне нечувствительности1 ('DeadBand1') значение, или превышает нижнее предельное значение на указанное в зоне нечувствительности2 ('DeadBand2') значение.

Пример

Функция должна срабатывать, если измеренное значение дольше 0,4 с за пределами диапазона значений между 8 и 12. Верхняя зона нечувствительности должна составлять при этом 0,56 нижняя зона нечувствительность 0,3.

Решение

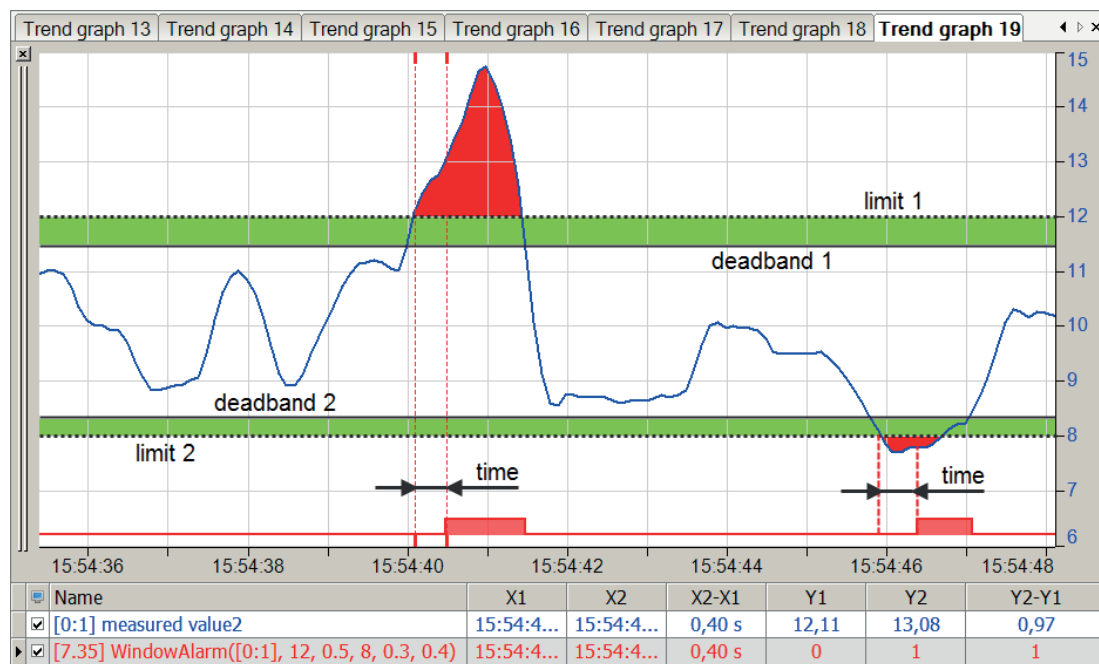


Рис. 59: Синий график: измеренное значение; красная диаграмма: Срабатывание функции WindowAlarm; зеленый: Зоны нечувствительности соответствующих предельных значений

2.8 Функции диагностики

2.8.1 CameraStatus

```
CameraStatus('ModuleNo*', 'SignalNo*', 'Timeout=2*')
```

Аргументы

'ModuleNo*'	Номер модуля сервера ibaCapture в дереве сигналов	
'SignalNo*'	Номер сигнала (камера)	
'Timeout*'	Время в секундах, в течение которого должен измениться сигнал синхронизации.	

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Жанная функция возвращает состояние камеры ibaCapture.

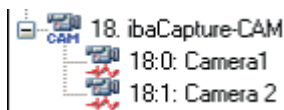
Результаты

0	Камера не в порядке
1	Камера ОК

Камера не в порядке если значение сигнала синхронизации камеры не изменяется в течение 'Timeout' секунд.

'ModuleNumber' - это номер модуля сервера ibaCapture в дереве сигналов.

Пример:



Состояние камеры с таймаутом, равным 2 с: CameraStatus(18, 0, 2)

2.8.2 DataStoreInfo

```
DataStoreInfo('DatastoreIndex*', 'InfoType*')
```

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция дает информацию о выбранной записи данных. Данная информация может использоваться для управления другими функциями или для индикации и диагностики.

Для обычной записи данных (PDA) используйте 'DatastoreIndex' >= 0.

Для записи данных QDR используйте 'DatastoreIndex' < 0.

Индекс можно посмотреть в дереве в диалоговом окне конфигурации записи данных. Индекс увеличивается сверху вниз.

Укажите тип информации ('InfoType'), который Вы хотите получить.

Поддерживаются следующие типы информации:

Типы информации	Возможные результаты
0: Состояние записи	0 = остановлена 1 = ожидание триггера 2 = запись 3 = посттриггерная запись
1: Сохранение в каталоге резервной копии:	0 = используется базовый каталог 1 = используется каталог резервной копии
2: Записанное время в текущем файле, указанное в секундах	
3: Свободное место на текущем жестком диске, указанное в МБ	Значение обновляется каждую минуту.
4: ibaQDR синхронизирован?	0 = ibaQDR HE синхронизирован 1 = ibaQDR синхронизирован

Table 2: Типы информации и возможные результаты функции DataStoreInfo

2.8.3 DataStoreInfoHD

```
DataStoreInfoHD('DatastoreIndex*', 'InfoType*')
```

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция дает информацию о выбранных записях HD. Данная информация может использоваться для управления другими функциями или для индикации и диагностики.

Желаемую запись данных HD Вы можете идентифицировать по 'DatastoreIndex*' >= 0.

Индекс можно посмотреть в дереве в диалоговом окне конфигурации записи данных. Индекс увеличивается сверху вниз.

Укажите тип информации ('InfoType'), который Вы хотите получить.

Поддерживаются следующие типы информации:

Типы информации	Возможные результаты
0: Состояние записи	0 = остановлена 1 = автономная 2 = запись идет
1: Скорость передачи данных	Значение в кбит/с

Table 3: Типы информации и возможные результаты функции DataStoreInfoHD

2.8.4 DongleInfo

`DongleInfo('InfoType*')`

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная информация дает информацию о различных свойствах, касаемых донгла. Данная информация может использоваться в целях индикации и диагностики.

Укажите тип информации ('InfoType'), который Вы хотите получать. Данный тип информации рассчитывается и выводится в начале измерения. Если Вы хотите получить из донгла больше информации, то Вам нужно несколько раз сконфигурировать функцию соответственно с другим типом информации.

Поддерживаются следующие типы информации:

Типы информации	Возможные результаты
0: Донгл доступен	TRUE = Донгл присутствует FALSE = Нет донгла или донгл поврежден
1: Лимит времени донгла в днях	Срок действия донгла
2: Лимит времени демо-версии в днях	Оставшееся время действия донгла в случае испытательного периода/демо-версий
3: ibaQDR лимит времени для сбора в секундах	Оставшееся время работы системы ibaQDR после извлечения донгла.
10: Счетчик установок донгла	Сколько раз был вставлен донгл
11: Счетчик извлечений донгла	Сколько раз был извлечен донгл
12: Счетчик замен донгла	Сколько раз был заменен донгл

Table 4: Типы информации и возможные результаты функции DataStoreInfo

Пример применения

Анализ типа информации 0 (доступность донгла) позволяет контролировать наличие донгла и его работу.

Данную информацию Вы можете использовать для триггерной отправки электронного сообщения с информацией для получателя о том, что донгл больше недоступен. Система, во время работы которой был потерян донгл, останавливается после периода ожидания. Этого следует избегать в случае работы систем ibaQDR на производстве.

2.8.5 FobDLinkStatus

`FobDLinkStatus('BoardNo*', 'LinkNo*')`

Аргументы

'BoardNo*'	Номер карты (от 0 до 7)	
'LinkNo*'	Номер соединения (указан в дереве сигналов в диспетчере вв/выв)	

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает состояние соединения карты ibaFOB-D.

Номер карты (от 0 до 7) можно взять из 7-значной индикации на карте или графического изображения. Номер соединения можно найти в дереве сигналов в диспетчере вв/выв.

Результаты

0	Соединение неактивно
1	Соединение ОК
2	Соединение прервано
3	Соединение RX в порядке, но кольцо Flex прервано

Условия

Данная функция возвращает только одно значение, если в диспетчере вв/выв созданы и сконфигурированы активные входные модули.

При использовании устройства мониторинга шин (например, ibaBM-DPM-S) в диспетчере вв/выв должен быть сконфигурирован как минимум один активный ведомый.

2.8.6 FobFastLinkStatus

`FobFastLinkStatus('BoardNo*', 'LinkNo*', 'Filtered*')`

Аргументы

'BoardNo*'	Номер карты (от 0 до 3)	
'LinkNo*'	Номер соединения (указан в дереве сигналов в диспетчере вв/выв)	
'Filtered*'	Настройка фильтра (ИСТИНА или ЛОЖЬ)	

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает состояние соединения карты ibaFOB-X в режиме 32 Mbit (32 Mbit/s).

Номер карты (от 0 до 3) можно взять из 7-значной индикации на карте или графического изображения. Номер соединения можно найти в дереве сигналов в диспетчере вв/выв.

'Filtered' может быть либо ИСТИНА либо ЛОЖЬ. Состояние соединения с фильтром игнорирует любое изменение состояния соединения, которое быстрее 40 мс.

Результаты

0	Соединение неактивно
1	Соединение ОК
2	Соединение прервано

Условия

Данная функция возвращает только одно значение, если в диспетчере вв/выв созданы и сконфигурированы активные входные модули.

При использовании устройства мониторинга шин (например, ibaBM-DPM-S) в диспетчере вв/выв должен быть сконфигурирован как минимум один активный ведомый.

2.8.7 FobFlexDeviceStatus

`FobFlexDeviceStatus('BoardNo*', 'LinkNo*', 'Address*')`

Аргументы

'BoardNo*'	Номер карты (от 0 до 7)	
'LinkNo*'	Номер соединения (указан в дереве сигналов в диспетчере вв/выв)	
'Address*'	Адрес устройства в кольце Flex (от 1 до 15)	

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает состояние устройства Flex с адресом 'Adresse' на соединении 'LinkNo' карты ibaFOB-D 'BoardNo'.

Номер карты (от 0 до 7) можно взять из 7-значной индикации на карте или графического изображения. Номер соединения можно найти в дереве сигналов в диспетчере вв/выв.

Адрес устройства настроен на поворотном переключателе устройства. Возможные значения адреса от 1 до 15.

Результаты

0	Устройство не сконфигурировано
1	Устройство ОК
2	Устройство не подключено

2.8.8 FobFLinkStatus

`FobFLinkStatus('BoardNo*', 'LinkNo*')`

Аргументы

'BoardNo*'	Номер карты (от 0 до 3)	
'LinkNo*'	Номер соединения (указан в дереве сигналов в диспетчере вв/выв)	

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает состояние соединения карты FOB-S или FOB-X в режиме 3 Mbit (3,3 Mbit/s).

Номер карты (от 0 до 3) можно взять из 7-значной индикации на карте или графического изображения. Номер соединения можно найти в дереве сигналов в диспетчере вв/выв.

Результаты

0	Соединение неактивно
1	Соединение ОК
2	Соединение прервано

Условия

Данная функция возвращает только одно значение, если в диспетчере вв/выв созданы и сконфигурированы активные входные модули.

При использовании устройства мониторинга шин (например, ibaBM-DPM-S) в диспетчере вв/выв должен быть сконфигурирован как минимум один активный ведомый.

2.8.9 FobMLinkStatus

```
FobMLinkStatus('BoardNo*', 'LinkNo*')
```

Аргументы

'BoardNo*'	Номер карты (от 0 до 3)	
'LinkNo*'	Номер соединения (указан в дереве сигналов в диспетчере вв/выв)	

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает состояние соединения карты FOB-S в режиме 5 Mbit (5 Mbit/s).

Номер карты (от 0 до 3) можно взять из 7-значной индикации на карте или графического изображения. Номер соединения можно найти в дереве сигналов в диспетчере вв/выв.

Результаты

0	Соединение неактивно
1	Соединение ОК

2	Соединение прервано
---	---------------------

Условия

Данная функция возвращает только одно значение, если в диспетчере вв/выв созданы и сконфигурированы активные входные модули.

2.8.10 FobPlusControlLinkStatus

```
FobPlusControlLinkStatus('BoardNo*')
```

Аргументы

'BoardNo*'	Номер карты (от 0 до 3)	
------------	-------------------------	--

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает состояние соединения карты ibaFOB-PlusControl

Номер карты (от 0 до 3) можно взять из 7-значной индикации на карте или графического изображения Карта имеет только одно соединение.

Результаты

0	Соединение неактивно
1	Соединение ОК
2	Соединение прервано

Условия

Данная функция возвращает только одно значение, если в диспетчере вв/выв созданы и сконфигурированы активные входные модули.

2.8.11 FobSDLinkStatus, FobSDexpLinkStatus

```
FobSDLinkStatus('BoardNo*')
```

```
FobSDexpLinkStatus('BoardNo*')
```

Аргументы

'BoardNo*'	Номер карты (от 0 до 3)	
------------	-------------------------	--

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает состояние соединения карты ibaFOB-SD (PCI) или карты ibaFOB-SDexp (PCIe).

Номер карты (от 0 до 3) можно взять из 7-значной индикации на карте или графического изображения Карта имеет только одно соединение.

Результаты

0	Соединение неактивно
1	Соединение ОК
2	Соединение прервано

Условия

Данная функция возвращает только одно значение, если в диспетчере вв/выв созданы и сконфигурированы активные входные модули.

2.8.12 FobTDCLinkStatus, FobTDCexpLinkStatus

```
FobTDCLinkStatus('BoardNo*')
```

```
FobTDCexpLinkStatus('BoardNo*')
```

Аргументы

'BoardNo*'	Номер карты (от 0 до 3)	
------------	-------------------------	--

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает состояние соединения карты ibaFOB-TDC (PCI) или карты ibaFOB-TDCexp (PCIe).

Номер карты (от 0 до 3) можно взять из 7-значной индикации на карте или графического изображения Карта имеет только одно соединение.

Результаты

0	Соединение неактивно
1	Соединение ОК
2	Соединение прервано

Условия

Данная функция возвращает только одно значение, если в диспетчере вв/выв созданы и сконфигурированы активные входные модули.

2.8.13 ICPSensorStatus

```
ICPSensorStatus('ModuleNo*', 'SensorNo*')
```

Аргументы

	Номер модуля Padu-8-ICP
'SensorNo*'	Номер сенсора (от 0 до 7)

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

При помощи данной функции контролируется состояние сенсора ICP. Первый параметр 'ModuleNo' указывает номер модуля ibaPADU-8-ICP или модуля ibaMS8xICP-/IEPE в дереве сигналов. Второй параметр 'SensorNo' задает, какой сенсор должен контролироваться. Номер сенсора может быть от 0 до 7.

Результаты

0	Датчик ОК
1	Обнаружен разомкнутый контур

2.8.14 InterruptCycleTime

```
InterruptCycleTime('Type=0*')
```

Аргументы

'Type*'	Опциональный параметр (уставка=0), который задает тип значения результата.	
	'Type' = 0	Выводится фактическое значение.
	'Type' = 1	Выводится минимум (наименьшее измеренное время цикла).
	'Type' = 2	Выводится максимум (наибольшее измеренное время цикла).

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает время цикла прерывания в микросекундах. Значение обновляется не при каждом прерывании, а при меньшем цикле.

2.8.15 InterruptTime

```
InterruptTime('Type=0*')
```

Аргументы

'Type*'	Опциональный параметр (уставка=0), который задает тип значения результата.	
	'Type' = 0	Выводится фактическое значение.
	'Type' = 1	Выводится минимум (наименьший измеренный период).
	'Type' = 2	Выводится максимум (наибольший измеренный период).

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает время прерывания (период прерывания) в микросекундах. Значение обновляется не при каждом прерывании, а при меньшем цикле.

2.8.16 IsMeasuring

IsMeasuring()

Описание

При текущем измерении результат данной функции ИСТИНА, при остановленном измерении - ЛОЖЬ.

Совет



Функция "Pause" (||) в ibaPDA останавливает только индикацию, а не измерение.

2.8.17 MultiStationStatus

MultiStationStatus()

Описание

Возвращает текущий режим нескольких станций.

Результаты

0	Автономный
1	Ведомый
2	Мастер

2.8.18 PerformanceCounter

PerformanceCounter('Category*', 'CounterName*', 'InstanceName*')

Аргументы

'Category*'	Текстовый ввод из столбца «Объект» обзора производительности Windows
	Текстовый ввод из столбца «Индикатор производительности» обзора производительности Windows
	Текстовый ввод из столбца «Экземпляр» обзора производительности Windows

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Функция PerformanceCounter может считывать, отображать определенные характеристики производительности и записывать график. Она возвращает значение счетчика производительности или 0, если счетчик производительности не существует. Функция PerformanceCounter позволяет считывать соответственно один индикатор производительности. Если Вы хотите отображать несколько индикаторов производительности, то Вам необходимо соответственно конфигурировать функцию.

Пример

Мониторинг производительности системы

Решение

Для выбора параметров производительности в Windows 7, которые подлежат мониторингу, щелкните на значок Windows и найдите Управление компьютером. Затем Вам необходимо выбрать слева узел *Мониторинг производительности* в *Служебные программы - Производительность - Средства наблюдения*. Вы можете добавить параметры производительности, которые должны контролироваться, при помощи зеленого символа + в верхней вкладке.

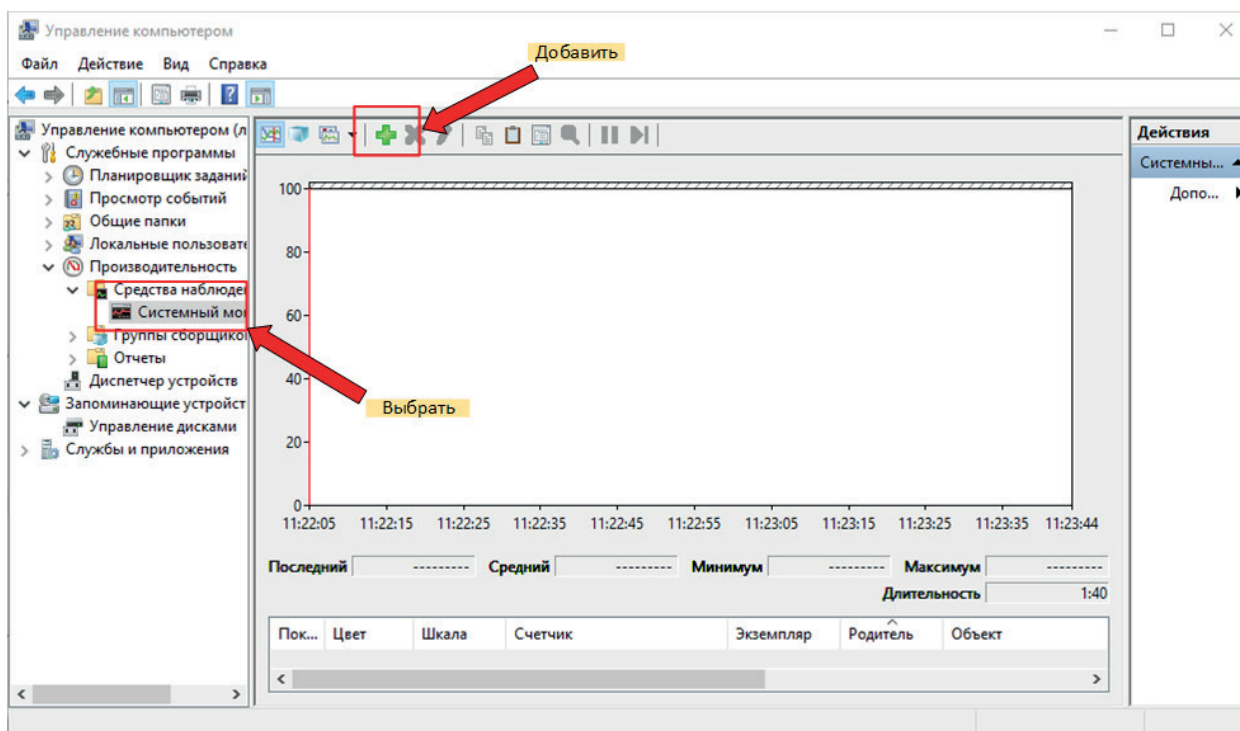


Рис. 60: Отображение новых контролируемых параметров производительности

Выберите индикаторы производительности и затем экземпляры. В данном случае выбираются и затем подтверждаются нажатием на <OK>: процесс, ibaPDA и Все элементы. На рисунке ниже выделены и пронумерованы шаги.

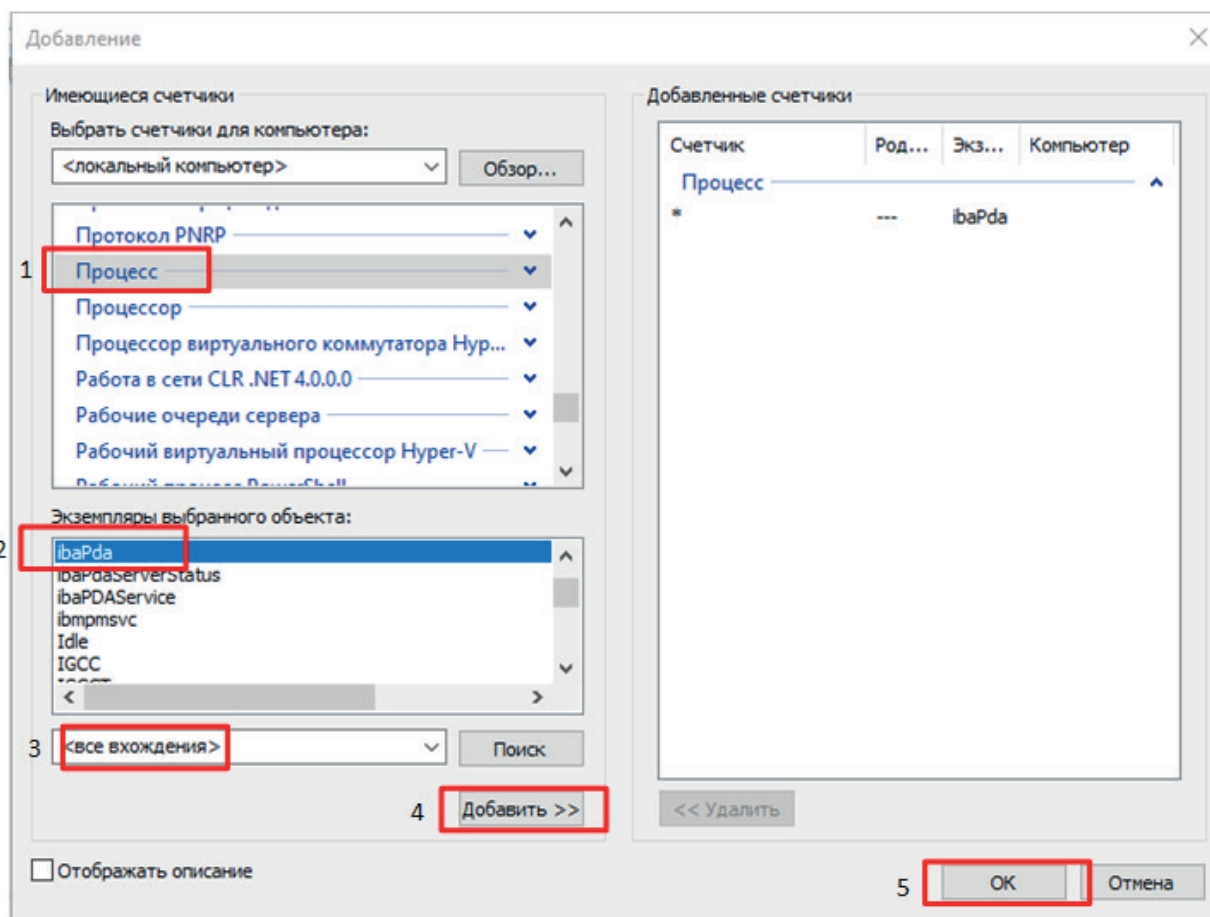


Рис. 61: Выбор аргументов функции

Присвоение отдельным аргументам функции PerformanceCounter объяснено на следующем рисунке. Для ввода в ibaPDA Вам необходимо ввести точные обозначения отдельных индикаторов производительности, объекты и экземпляры.

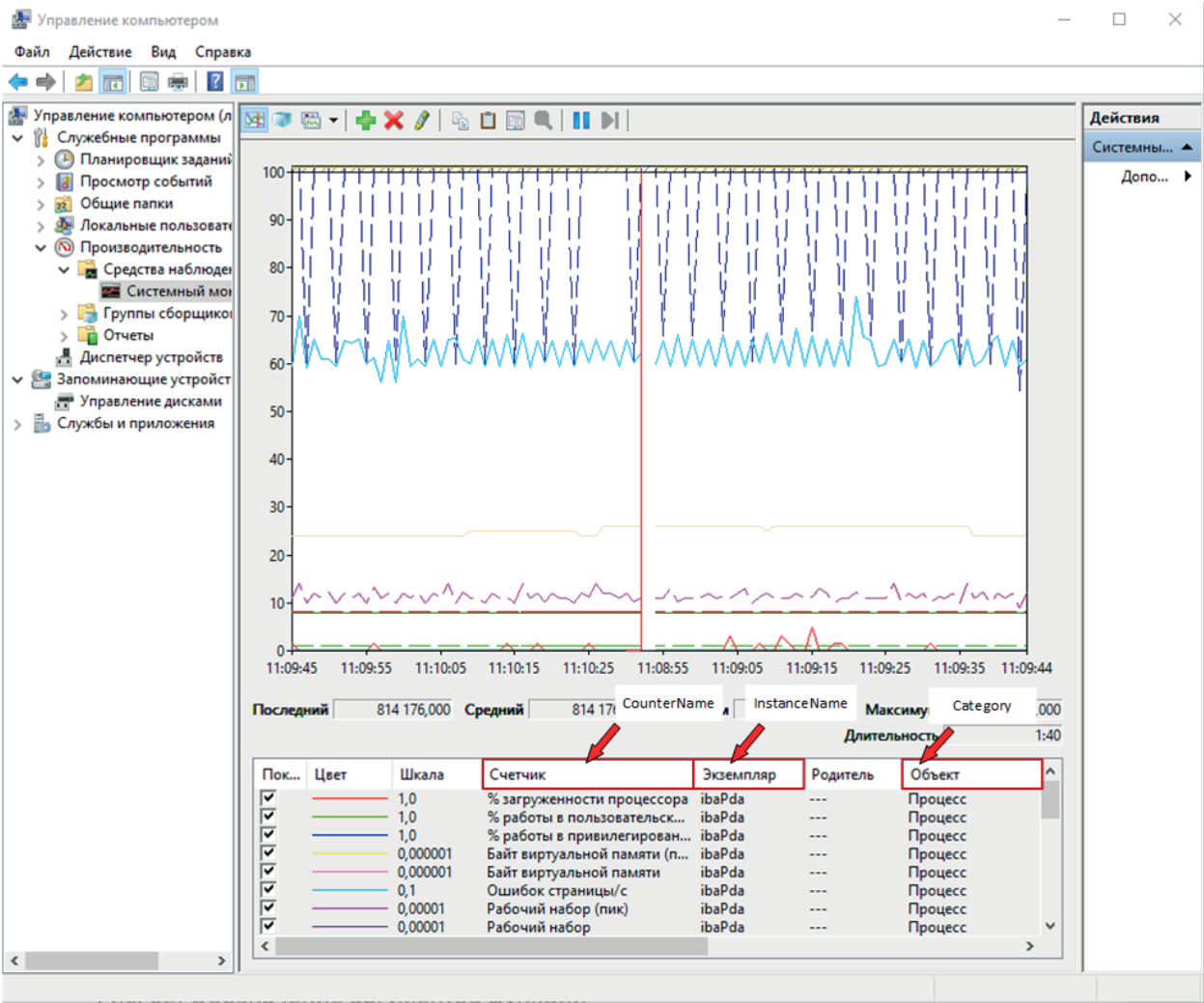


Рис. 62: Обозначение аргументов функции

Пример ввода в ibaPDA для определения времени процессора:

Выражение

PerformanceCounter("Процесс", "% загрузки процессора", "ibaPda")

Сбросить выражение

☐ Ссылка на сигналы по именам

OK

Отменить

2.8.19 Ping

Ping('Address', 'Trigger', ' Timeout=5', ' Size=32')

Аргументы

'Address'	IP-адрес целевого компьютера
'Trigger'	Триггерный сигнал (передний фронт), который должен отправлять пинг
'Timeout'	Опциональный параметр (уставка=5) для указания времени ожидания ответа, прежде чем будет выведен как результат "нет ответа".
'Size'	Опциональный параметр: Величина пинг-запроса в байтах (уставка=32)

Описание

Данная функция отправляет пинг-запрос в 'Address' при переднем фронте триггера 'Trigger'. Функция возвращает время в миллисекундах, необходимое до получения пинг-ответа. Если в течение 'Timeout' секунд ответ не приходит, то выводится -1. Результат - 0, пока не был отправлен пинг-запрос. 'Size' определяет размер пинг-запроса в байтах.

Как результат функция выводит аналоговое значение.

Результаты

-1	Нет ответа на пинг-запрос в течение 'Timeout'
0	Пинг-запрос не отправлен
n	Время ответа в мс

2.8.20 TimeSinceLastSync

TimeSinceLastSync()

Описание

Данная функция возвращает время в секундах, прошедшее с момента последней синхронизации времени. Если синхронизация времени не осуществляется, функция возвращает -1.

Совет



Синхронизация времени в ibaPDA настраивается в меню «Конфигурация» и затем «Диспетчер вв/выв» под вкладкой «Общее». Возможные источники для синхронизации времени DCF77, IEC1131 и DGM, или же синхронизация времени не настроена.

2.8.21 TimeSyncStatus

TimeSyncStatus('Source*')

Аргументы

'Source*'	Источник для синхронизации времени	
	'Source' = -1	Последний использованный источник для синхронизации времени
	'Source' = 0	DCF77 источник 1
	'Source' = 1	DCF77 источник 2
	'Source' = 2	IEC1131
	'Source' = 3	DGM200P
	'Source' = 4	Ведомый PTP
	'Source' = 5	ibaClock

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Данная функция возвращает состояние выбранного источника для синхронизации времени.

'Source' может содержать следующие значения:

Результаты

0	Источник неактивен
1	Источник активный и корректный
2	Источник активный, но некорректный

Совет



Синхронизация времени в *ibaPDA* настраивается в меню «Конфигурация» и затем «Диспетчер вв/выв» под вкладкой «Общее».

2.9 Функция фильтра

2.9.1 ВР

```
BP('Expression', 'Frequency1*', 'Frequency2*')
```

Аргументы

'Expression'	Измеренное значение
'Frequency1*'	Нижний предел полосы частот для фильтра
'Frequency2*'	Верхний предел полосы частот для фильтра

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Функция является полосно-пропускающим фильтром с полосой частот между частотами 'Frequency1' и 'Frequency2'. Фильтром является фильтр Баттерворта второго порядка.

2.9.2 НР

```
HP('Expression', 'Frequency*')
```

Аргументы

'Expression'	Измеренное значение
'Frequency*'	Значение предельной частоты фильтра

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Функция является фильтром верхних частот с предельной частотой 'Frequency'. Фильтром является фильтр Баттерворта второго порядка.

2.9.3 LP

LP('Expression', 'Omega*', 'Reset=0')

Аргументы

'Expression'	Измеренное значение	
'Omega*'	Угловая частота фильтра, из которой получается частота функции LP к частоте = 'Omega'/(2*PI)	
'Reset'	Опциональный цифровой параметр, который может использоваться для деактивации функции. 'Reset' может также сам быть выражением.	
	'Reset' > 0	Фильтр деактивируется, входной сигнал отображается без фильтра
	'Reset' = 0	Фильтр используется

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Функция является фильтром нижних частот с частотой = 'Omega'/(2*PI) к 'Expression'. Фильтр является однополюсным фильтром со скоростью спада 20дБ/дек. Если опциональный параметр 'Reset' ИСТИНА, то фильтр деактивируется и результатом является неотфильтрованный входной сигнал 'Expression'. 'Reset' может быть также сформулирован как выражение.

Примеры:

LP([0:0],10)	Фильтр активен ('Reset' пропущен).
LP([0:0],10,If([0:0]<0,1,0))	Функция фильтра применяется только к положительным значениям.
LP([0:0],10,[3.1])	Например, с [3.1] = If([0:0]>10, 1, 0) Фильтр деактивируется, как только выражение [3.1] возвращает ИСТИНА, т.е. выражение [0:0] превышает предельное значение 10.

Пример

Применение функции фильтра к синусоидальному колебанию с частотой 10 Гц, на которое накладывается дополнительное колебание в 50 Гц.

Задача

Функция фильтра должна применяться со значениями 5, 10 и 40 для 'Omega'.

Решение

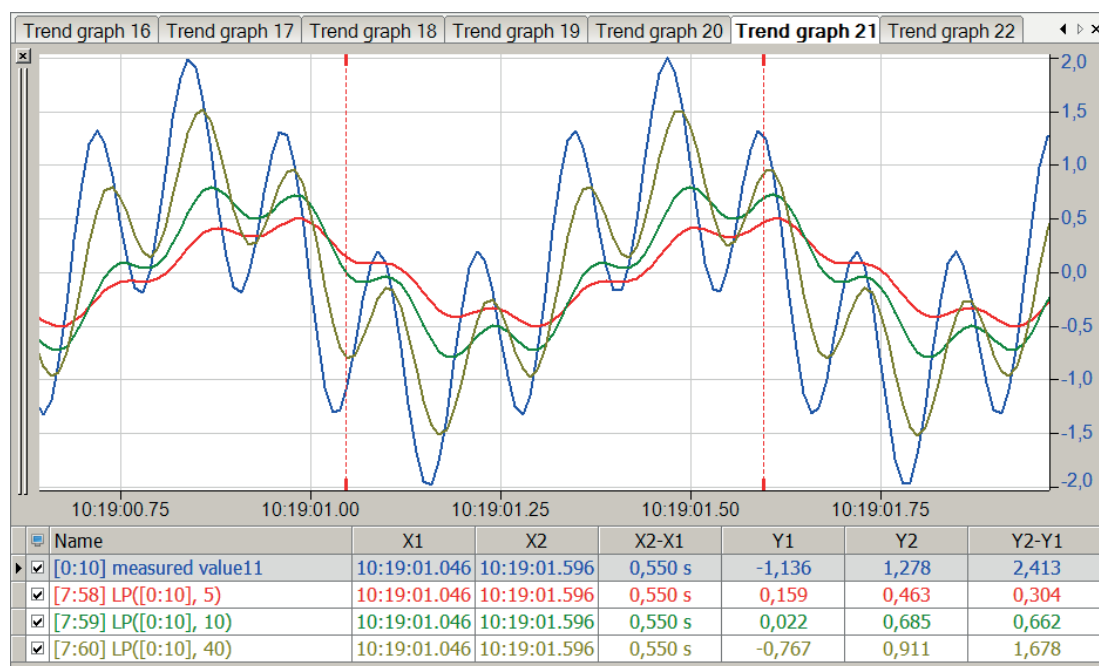


Рис. 63: Синий график: исходный сигнал наложенного синусоидального колебания; красный график: отфильтрованный сигнал с ' Ω '= 5; зеленый график: отфильтрованный сигнал с ' Ω ' = 10; желтый график: отфильтрованный сигнал с ' Ω ' = 40;

Чем меньше значение, выбранное для ' Ω ', тем сильнее ослаблен сигнал.

2.9.4 EnvelopeSpectral

`EnvelopeSpectral('Expression', 'Frequency1*', 'Frequency2*', 'Cutoff frequency')`

Аргументы

'Expression'	Измеренное значение
'Frequency1*'	Указание нижнего предела частотной полосы для расчета огибающей кривой Значение не должно быть меньше 10% частоты Найквиста.
'Frequency2*'	Указание верхнего предела частотной полосы для расчета огибающей кривой Значение не должно быть больше 90% частоты Найквиста
'Cutoff frequency'	Опциональное значение предельной частоты для конфигурации фильтра нижних частот, применяемого к огибающей кривой.

Параметры, отмеченные знаком «*», обрабатываются только один раз в начале сбора.

Описание

Функция рассчитывает спектральную огибающую кривую 'Expression' с постоянным полосно-пропускающим фильтром между частотами 'Frequency1' и 'Frequency2'. Наименьшая ширина полосы составляет 10% частоты Найквиста, т.е. 10% половины частоты дискретизации.

Сигнал 'Expression' повторно считывается с опорным временем виртуального сигнала. Фильтр сглаживания не используется.

2.9.5 Preprocess

```
Preprocess('Expression', "'Preprocess profile name*')
```

Аргументы

'Expression'	Выражение, которое должно быть предварительно обработано
'Preprocess profile name**'	Имя профиля предварительной обработки или препроцессора

Параметры, отмеченные знаком «*», принимаются только один раз в начале сбора.

Описание

Профиль предварительной обработки с именем 'Preprocess profile name' применяется к 'Expression'.

Профили предварительной обработки могут быть сконфигурированы в диспетчере препроцессора. Данный диспетчер можно открыть через редактор выражений модуля Inspectra Expert

2.10 Функции запоминания

Функции в данной группе могут использоваться в так называемом модуле запоминания. Функции со свойством запоминания всегда сохраняют их последнее значение, пока они целенаправленно не будут сброшены.

При использовании данных функций в модуле запоминания соответственно последнее вычисленное значение результата можно сохранить, остановив измерение. При повторном запуске измерения данное значение можно загрузить снова как исходное значение.

Это позволяет не потерять долгосрочные данные, такие как например, данные счетчика отработанных часов и расходы сред вследствие останова и запуска измерения.

Функции с данным свойством:

➤ *Count*, страница 72

➤ *Int*, страница 25

➤ *Max*, страница 40

➤ *Min*, страница 45

2.11 Плагины

Система плагина ibaPDA была разработана в целях обеспечения пользователям ibaPDA возможности создавать собственные функции, чтобы проводить при помощи ibaPDA заданные пользователем вычисления. ibaPDA выполняет данные функции в режиме реального времени. Данные функции могут использоваться точно так же как интегрированные функции в выражениях для виртуальных сигналов. Они появляются также в редакторе выражений (плагины).

Чтобы создавать заданные пользователем функции, Вам необходимо создать файл NET.dll. Данный файл может быть записан на любом языке .NET (C#, C++, VB.NET,...).

Дополнительная документация



Для получения более детального описания программирования плагина и конфигурации обратитесь к руководству «Плагин ibaPDA».

3 Техподдержка и контакты

Служба поддержки

Тел.: +49 911 97282-14
Факс +49 911 97282-33
E-Mail: support@iba-ag.com

Важно



При обращении в техподдержку указывайте серийный номер (iba-S/N) продукта.

Контактные данные

Фактический адрес

iba AG
Königswarterstraße 44
90762 Fürth
Deutschland иба АГ, Кёнигсвартерштрассе 44, 90762 Фюрт Германия

Тел.: +49 911 97282-0
Факс +49 911 97282-33
E-Mail: iba@iba-ag.com
Контактное Харальд Опель
лицо:

Адрес для корреспонденции

iba AG
Gebhardtstraße 10
90762 Fürth
Deutschland иба АГ, Гебхардтштрассе 10, 90762 Фюрт Германия

В Германии и по всему миру

Адреса других наших региональных филиалов и представительств можно найти на нашей веб-странице.

www.iba-ag.com.