

# Вычисления в MasterSCADA™

*Обзор возможностей*

© InSAT Company

## Оглавление

Введение .....	2
Расчет и Событие.....	2
Формулы .....	2
Типовые формулы.....	4
Редактор формул.....	4
Внешнее управление значением Расчета.....	8
Событие.....	9
Другие способы обработки данных в Master SCADA .....	9
Функциональные блоки.....	9
Пользовательские ФБ .....	10
ФБ Скрипт (пользовательская программа).....	10
Обработка данных в клиентских модулях Master SCADA.....	11
Обработка в MasterGraph.....	11
Обработка в MasterTrend .....	11
Обработка в MasterReport.....	12
Заключение .....	14



## Введение

Данные, поступающие в Master SCADA с объекта контроля и управления обычно нуждаются в дополнительной обработке перед их использованием для выработки автоматического управляющего воздействия, либо информирования оператора. Простейшие обработки мы уже рассмотрели в статье об измерениях в Master SCADA. Это – масштабирование (перевод измеренного значения из кодов аналого-цифрового преобразователя в шкалу физического параметра) и линеаризация (приведение нелинейной шкалы, связанной с исходной нелинейностью чувствительного элемента датчика, например, давления или температуры, к линейной). В этой статье мы рассмотрим дальнейшую обработку параметра, в разных системах применяемую для самых разнообразных целей, например, для вычисления на основе нескольких измеренных величин значения параметра, недоступного непосредственному измерению.

Будем считать, что читатель уже ознакомился со статьями [«Методика разработки проектов в Master SCADA»](#) и [«Измерения в Master SCADA»](#) и самой программой, а, следовательно знает, как получить данные с нижнего уровня, от контроллера или модуля ввода-вывода, в дерево системы Master SCADA. В нашем случае, будем ли мы использовать для этого OPC-сервер или исполнительную систему MasterPLC – не имеет для темы этой статьи ровно никакого значения. Сейчас нас интересует: что же мы можем сделать с этими данными?

## Расчет и Событие

### Формулы

В Master SCADA имеется два типа переменных, для реализации формульных вычислений практически неограниченной сложности: Расчет () и Событие (). Эти переменные можно использовать для выполнения подавляющего большинства типичных действий по обработке данных. Отличие между ними в части рассматриваемых нами возможностей обработки данных только одно: если результатом Расчета всегда будет значение вещественного типа, то переменная Событие формирует только дискретное значение.

Вычислительные возможности этих переменных реализуются на закладке «Формула» в панели их свойств (Рисунок 1).

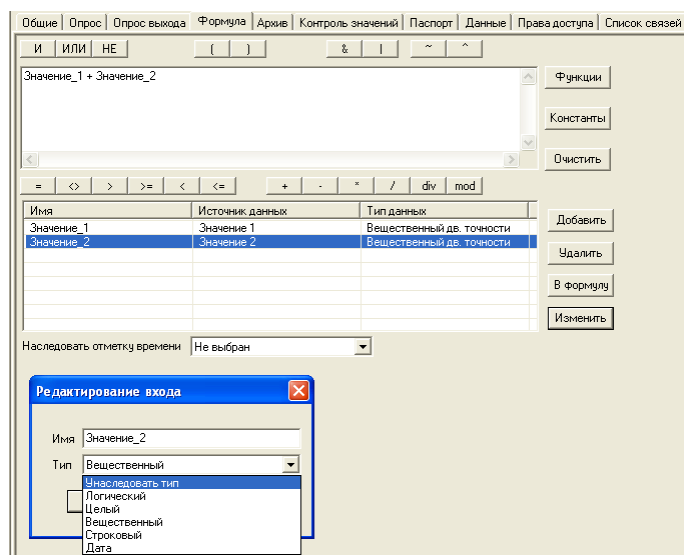


Рисунок 1

Установка связей между требующими обработки данными и входами Расчета производится стандартным для Master SCADA способом – простым перетаскиванием мышью источника данных (переменной или выхода функционального блока) из дерева проекта сначала в таблицу источников данных закладки «Формула», а уже оттуда (кнопкой «В формулу», либо двойным щелчком) в окно редактора текста формулы.

Переменные в таблице источников данных будем далее называть входами формулы.

Если выбрать вход в таблице и нажать на кнопку «Изменить» (все действия с источниками можно производить как кнопками, так и из контекстного меню по правой клавише мыши), то мы можем не только переименовать источник, но и изменить его тип данных. При перетаскивании связи на источник данных контролируется соответствие типа данных исходного значения и источника, а при несоответствии перед вычислением формулы производится приведение типа.

Результат вычисления окажется на выходе переменной. Тип значения результата автоматически формируется на основании типов значений входов (с приведением к более старшему типу) и не может быть изменен. Увидеть выбранный системой тип можно, как обычно, на закладке свойств «Общие». Результат при соответствующих типах входов может приобретать любой существующий в Master SCADA тип, включая даже тип «время». Добиться нужного типа значения результата можно только выбрав соответствующие типы значений входов формулы.

**Важно отметить**, что при изменении типа значения переменной, являющейся источником данных входа, тип значения входа не изменится, хотя в диагностических целях эта строка будет выделена красным цветом. Изменение типа входа в этой ситуации должно быть сделано повторным перетаскиванием источника, либо ручной сменой типа.

Полученное значение результата, если оно имеет аналоговый тип, как и значение любой другой аналоговой переменной может контролироваться на соблюдение предупредительных и аварийных границ.

**Обратите внимание**, что результат формулы может наследовать от входов значения их признаков качества (недоверности) в соответствии со стандартом ОРС. Чтобы это происходило, надо установить флаг «Формировать признаки качества выходов ФБ по признакам качества входов» в диалоге «Сервис/Настройка/Режим исполнения».

Назвав аргументы формулы ее входами, мы поступили осознанно – дело в том, что наряду с переменной Расчет можно использовать и функциональный блок (ФБ) Расчет (категория «Вычисления» палитры ФБ) с точно такими же вычислительными возможностями. В этом случае все эти входы будут видны в качестве входов ФБ (выход у него один – результат расчета), что безусловно повышает наглядность применения расчетов в проекте или схеме обработки данных (см. далее). Ведь переменная Расчет в схемах и проекте представлена, только как простой источник данных, наподобие переменной Значение.

Для архивирования результата расчета важно определить, с какой меткой времени должно сохраняться значение. Дело в том, что между его измерением в контроллере и проведением вычислений проходит некоторый промежуток времени, складывающийся из периодов опроса, цикла вычислений и т.п. В то же время, важно, чтобы значение обработанной переменной записывалось в архив с той же меткой времени, что и «сырой». Разработчику проекта виднее, какая из переменных, участвующих в расчете, является «ведущей», поэтому

мы и предоставляем ему возможность выбора, от какой из переменных наследовать метку времени.

**Важная рекомендация** – не использовать в качестве источников данных переменных дерева системы. Казалось бы, никто нам не мешает поместить вход контроллера сразу в формулу, однако такой подход нарушает принцип самодостаточности (инкапсуляции) объектов в дереве объекта. В результате осложняется тиражирование таких объектов, их копирование между проектами или использование в качестве библиотечных. Дело в том, что «внутренняя» связь формулы с переменными того объекта, которому она принадлежит (в состав которого входит переменная Расчет), не приводит к разрыву при перемещении объекта куда-либо. Разрываются только внешние связи самого объекта, которые в отличие от источников данных формулы находятся на виду. Предназначенные для внешних связей переменные объекта могут быть источниками данных для разных целей в рамках данного объекта, а их восстановление может быть произведено в полуавтоматическом режиме с помощью мастера восстановления связей.

### **Типовые формулы**

**Обратите внимание** на то, что список входов формулы можно создать не только перетаскиванием связей, но и простым добавлением, именованием и выбором типа значения. Такие «несвязанные» входы тоже можно использовать для создания формулы, в этом случае мы получим «абстрактную формулу» - пока нет связей, она не может быть вычислена, но зато может быть помещена в библиотеку, как типовая, для последующего использования в других проектах. В библиотеку помещается не сама формула, а переменная Расчет или Событие через контекстное меню этой переменной проекта. При помещении в библиотеку сохраняется как формула, так и все остальные свойства и настройки.

В библиотеку можно поместить и конкретную формулу, при этом все связи входов потеряют источник данных. Чтобы преобразовать типовую формулу, вставленную из библиотеки, в конкретную необходимо каждый вход формулы связать с источником данных в проекте, для чего переменную проекта перетащить из дерева в строку таблицы с выбранным входом.

Если вы планируете сохранить в дальнейшем эту формулу в качестве библиотечной (см. далее), то лучше переименовать (двойным щелчком на имени или кнопкой «Изменить») входы формулы так, чтобы у них были «говорящие» названия, а в поле комментария на закладке «Общие» переменной напишите пояснения. Это упростит понимание работы формулы при ее использовании в будущих проектах.

### **Редактор формул**

Теперь рассмотрим подробнее возможности редактора формул. Надеемся, что непосредственно редактирование не вызовет трудностей, и остановимся на особенностях вычислений.

Поле редактора (Рисунок 2) обрамлено группами клавиш, вставляющих в формулу необходимые операции арифметики, логики, побитовой логики, а также сравнения. В формуле могут использоваться различные функции, числа и типовые константы.

Сложность формулы ничем не ограничена. Скобочная структура может иметь любую глубину вложения. Вычисления могут быть как безусловными, так и с предварительным вычислением условия. В формуле могут производиться расчеты с целыми, вещественными, логическими значениями и значениями, имеющими тип «время».

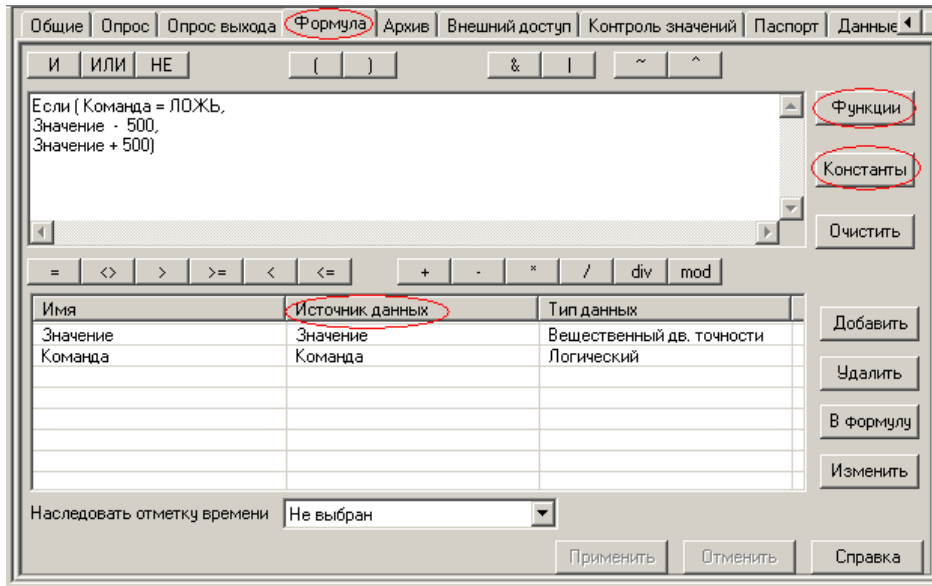


Рисунок 2

Контроль корректности ввода информации в формуле производится автоматически и при обнаружении ошибки выдается соответствующее предупреждение.

Доступные для использования функции тематически разделены на четыре группы (Рисунок 3): непосредственно вычислительные функции, функции работы с архивом, функции работы со значениями времени и функции работы с признаками качества значения.

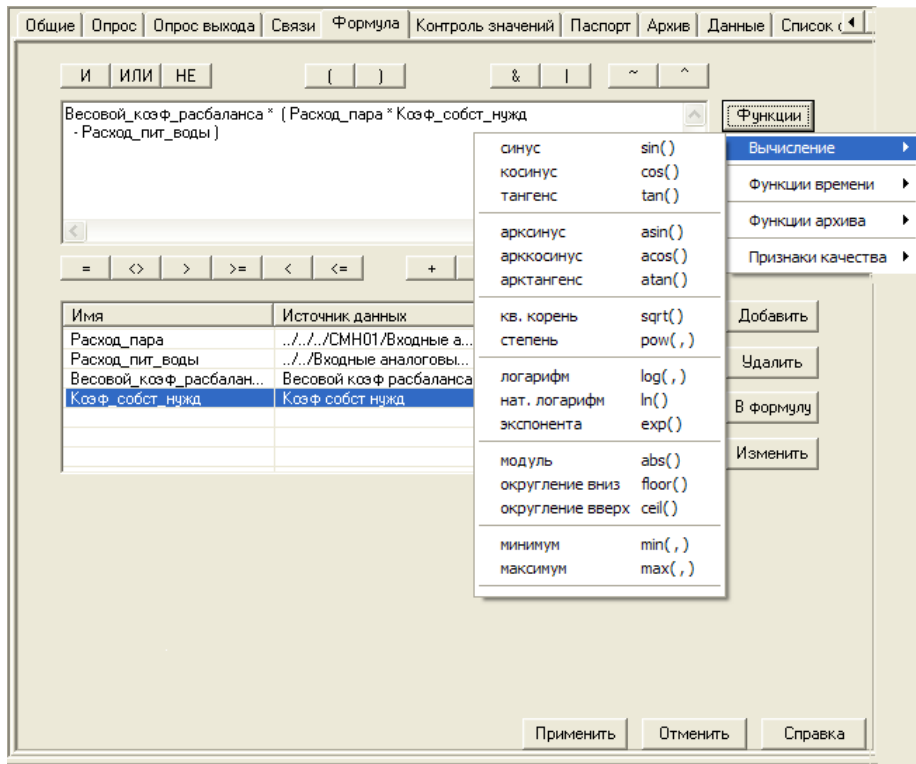


Рисунок 3

Для удобства мы представили каждую из групп в виде таблицы.

### Вычислительные функции

Синус	sin()
Косинус	cos()
Тангенс	tan()
Арксинус	asin()
Арккосинус	acos()
Арктангенс	atan()
Квадратный корень	sqrt()
Степень	pow(,)
Логарифм	log(,)
Натуральный логарифм	ln()
Экспонента	exp()
Модуль	abs()
Минимум	min(,)
Максимум	max(,)
Округление вверх	ceil()
Округление вниз	floor()
Логический выбор	ЕСЛИ(,,)

### Функции работы с архивом

Интеграл по архиву за текущие сутки	ArchIntegral (Переменная, Время1, Время2, Период приведения)
Среднее значение по архиву за текущие сутки	ArchAverage (Переменная, Время1, Время2)
Счетчик дискретных импульсов в архиве за текущие сутки	ArchDiscreteCounter (Дискретная переменная, Время1, Время2 )
Подсчет суммарного времени, когда переменная была в состоянии Вкл за текущие сутки	ArchDiscreteCounterTime (Дискря_Переменная, Время1, Время2 )
Мгновенное значение по архиву за текущие сутки	ArchMomentValue (Переменная, Время1, Тип поиска)
Интеграл по архиву	AbsArchIntegral (Переменная, Время1, Время2, Период приведения)
Среднее значение по архиву	AbsArchAverage (Переменная, Время1, Время2)
Счетчик дискретных импульсов в архиве	AbsArchDiscreteCounter (Дискретная переменная, Время1, Время2 )
Подсчет суммарного времени, когда переменная была в состоянии Вкл	AbsArchDiscreteCounterTime (Дискр_Переменная, Время1, Время2 )
Мгновенное значение по архиву	AbsArchMomentValue (Переменная, Время1, Тип поиска)

### Функции работы со значениями времени

Текущее время	Time ()
Текущая дата	Date ()
Текущие дата и время	DateTime ()
Преобразование ЧМС во время	Int2Time (,,)
Преобразование ДМГ в дату	Int2Date (,,)
Преобразование ДМГ ЧМС в дату_время	Int2DateTime(,,,,)
Сложение дат и времени	AddDateTime ( , )
Секунды во время	Sec2Time()
Выделение метки времени переменной	TimeStamp( )
Начало текущего часа	CurHour()
Начало текущей смены	CurShift()
Начало текущих суток	CurDay()
Начало текущей недели	CurWeek()
Начало текущего месяца	CurMonth()
Начало текущего года	CurYear()



### Функции работы с признаками качества значений

OPC признаки	OPCQuality( )
Признаки нарушения границ	OPCBounds( )
Снять признак качества	ClearQuality( )

Аргументом функций служат входы формулы, а в ряде случаев другие функции или константы.

Если математические функции обычно не вызывает вопросов, то специализированным функциям следует уделить некоторое внимание.

Инструменты для работы с архивными данными позволяют с помощью всего одной операции суммировать или усреднять данные за любой доступный в архиве период времени, производить подсчет времени наработки или числа включений оборудования. Чтобы указать период, за который должны производиться эти вычисления, используется широкий набор функций работы со временем. Важно отметить, что многие функции работы со временем оперируют не абсолютными значениями, а относительными, типа «начало смены». Это позволяет не менять ничего в формулах при изменении расписания смен.

Функция для работы с OPC-признаками качества используется, прежде всего, для учета достоверности исходных данных в процессе вычислений, обычно в качестве условия формирования специального значения в случае недостоверности.

Функция снятия признаков качества позволяет отменить наследование признаков недостоверности результатом вычислений при недостоверных входах формулы.

Аналогично и вычисление признака нарушения границ (заданных на закладке «Контроль значений» переменной, являющейся источником значений входа) обычно используется для учета аварийного состояния параметра при формировании результата вычисления.

**Обратите внимание**, что основное назначение списка констант – это предоставить вам удобное средство вставки в формулу значений признаков качества или границ для сравнения с результатом вычисления соответствующей функции. Например, так:

OPCQuality( Выход ) <> ОШИБКА\_УСТРОЙСТВА

или

OPCBounds( Выход ) <> ВАГ, здесь ВАГ – это верхняя аварийная граница.

### Внешнее управление значением Расчета

Важно отметить, что, кроме формирования значения переменной Расчет или Событие в результате вычисления формулы, это значение может быть принудительно изменено присвоением константы или значения другой переменной. Эта возможность заложена в списке действий на закладке «Расписание» свойств объекта, а также на закладке «Действия» свойств переменной типа События, либо Команды. В целях отладки также всегда остается возможность ручного изменения значения Расчета (как и для любой другой переменной – с закладки свойств Расчета «Данные», либо в столбце значений рядом с деревом проекта). Чтобы введенное значение осталось неизменным после пересчета формулы, переменную необходимо предварительно заблокировать.

## **Событие**

Переменная Событие служит, прежде всего, для сигнализации о тех или иных событиях, происходящих на объекте или в системе. Она вычисляется по формуле, так же, как и расчет, но ее значение будет в любом случае приведено к логическому типу. Моментом наступления События в Master SCADA принято считать переход значения этой переменной из состояния ЛОЖЬ в состояние ИСТИНА.

Событие – единственная переменная Master SCADA, способная формировать пользовательские системные сообщения. Эти сообщения могут содержать текущие (на момент возникновения события) значения переменных. Описание работы с сообщениями смотрите в статье, посвященной этой подсистеме.

Событие похоже не только на Расчет, но и на другой тип переменной – дискретную Команду. Как и Команда событие может быть использована для управления. Для этого на закладке Действия формируется список необходимых действий, а на закладке Права доступа ограничивается список должностей, которым эти действия разрешены. Но, если Команда, будучи перетащенной на поле мнемосхемы, создает кнопку, то Событие формирует «Зону выбора» - невидимую прямоугольную область, при щелчке по которой формируется событие, инициирующее действие (а при необходимости и выдающее пользовательское сообщение).

Этим «эклетичный» характер События не исчерпывается – если одни свойства оно унаследовало от Расчета, другие от Команды, то третьи – от дискретных Значений. Как и они Событие может быть перетащено на поле мнемосхемы не только в виде значения, но и в виде графического сигнализатора («правильного»: круглого или прямоугольного, либо произвольного: любого графического изображения).

## **Другие способы обработки данных в Master SCADA**

### **Функциональные блоки**

При создании проектов в Master SCADA широко используются функциональные блоки (ФБ). Мы уже коротко упоминали о них в предыдущих статьях, будем говорить и в дальнейшем. Здесь отметим только, что возможность создания сложных алгоритмов в специализированном редакторе схем ФБ (закладка «Общие» любого объекта проекта) открывает неограниченные перспективы. Стандартные библиотеки Master SCADA (Рисунок 4) содержат огромное число (около 100) ФБ, специально предназначенных для обработки значений. Из них можно собрать схемы любого размера.

**Обратите внимание**, что в случае простых обработок совершенно необязательно создавать схемы, можно просто вставить ФБ в объект, данные которого обрабатываются.

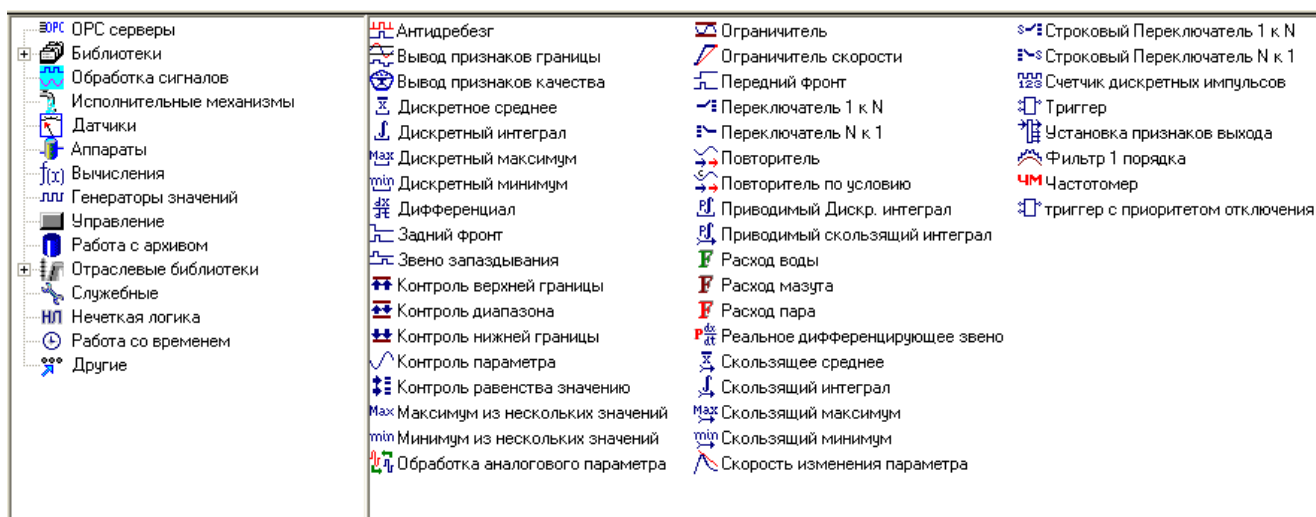


Рисунок 4

Основные библиотечные категории – «Вычисления» (свыше 30 ФБ, включая операции со строками), «Обработка сигналов» (свыше 40 блоков), «Обработка времени», «Нечеткая логика», «Генераторы значений». И если ФБ категории «Вычисления» на 80% пересекаются по возможностям с формульными вычислениями, то содержимое других категорий предоставляет инструментарий для готового решения подавляющего большинства стандартных для систем автоматизации операций над полученным от датчиков значениями. Ведь, например, реализация динамической (с учетом времени) обработки значений с помощью формульных вычислений затруднительна – нужна полноценная программа с использованием значений, запомненных на предыдущих итерациях.

### **Пользовательские ФБ**

В ряде случаев необходима сложная обработка данных с привлечением дифференциальных вычислений, спектрального анализа и т.п. В этом случае возможно создание собственных профессиональных программ, которые могут после создания быть помещены в библиотеки Master SCADA. Руководство по написанию таких ФБ находится в составе любого комплекта Master SCADA, включая демонстрационные, а также на сайте: [Общая информация о Master SCADA/Разработка модулей и программ расширения.](#)

### **ФБ Скрипт (пользовательская программа)**

Не всегда обработка данных бывает сложной, но тем не менее, выразительные средства схем ФБ или формульных вычислений оказываются недостаточны. Типичный пример – любые итерационные вычисления. В этом случае оказывается удобно использовать специализированный ФБ «Скрипт» (категория «Служебные») для упрощенного программирования. Ничего, кроме первичных знаний синтаксисам языка C#, не требуется. Все окружение программы формируется автоматически, а сама программа может быть полностью проверена в процессе разработки, чтобы исключить проблемы в режиме реального времени.

В дереве объекта «Скрипт» выглядит, как обычный ФБ с заданным числом входов и выходов (

Рисунок 5). Текст программы вводится в специализированном редакторе на закладке «Код» страницы свойств. На той же закладке создаются (добавлением через контекстное меню) Входы и Выходы. Параметр вставляется в код программы двойным щелчком по нему.

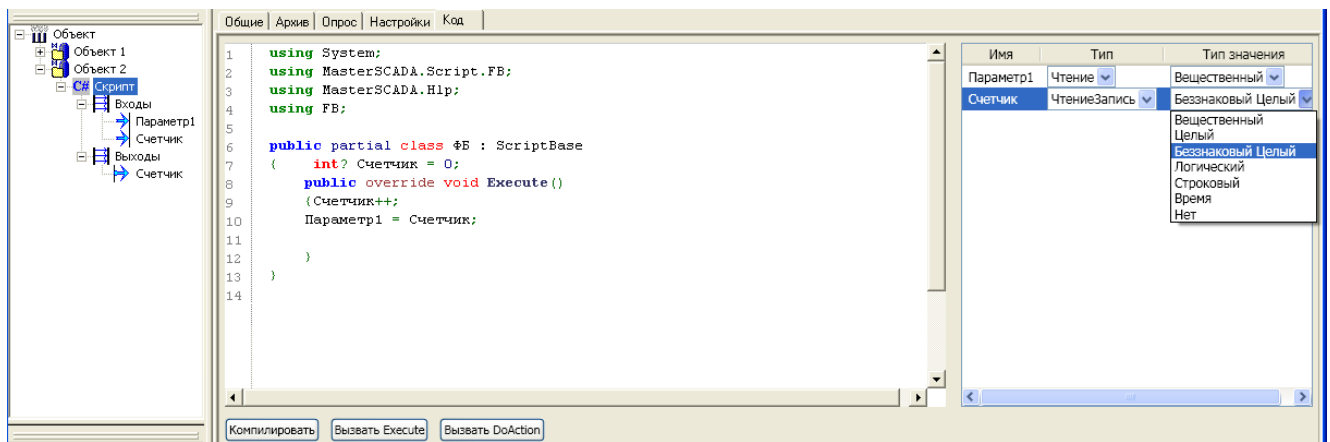


Рисунок 5

## Обработка данных в клиентских модулях Master SCADA

Практически все клиентские модули Master SCADA – графическая подсистема, подсистема трендов, генератор отчетов содержат те виды обработки входных значений, которые удобно делать «по месту».

### Обработка в MasterGraph

В графической подсистеме возможно осуществлять пропорциональный пересчет аналоговых параметров (входов мнемосхемы) для вычисления того, насколько должно

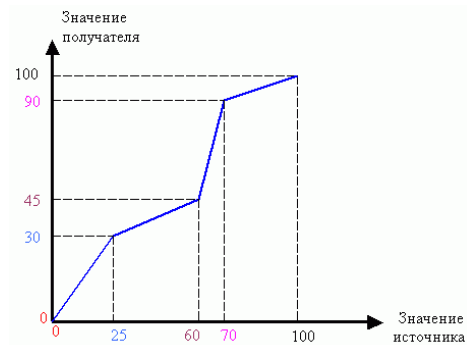


Рисунок 6

измениться числовое значение динамизируемого свойства при изменении исходного значения параметра на определенную величину. Это можно делать для каждого интервала изменения индивидуально (Ошибка! Источник ссылки не найден.). Для дискретных параметров при необходимости применяется инверсия значения.

### Обработка в MasterTrend

Тренд (график зависимости параметра от времени) может использоваться не только для просмотра, но и для анализа. С этой целью в него включены средства обработки просматриваемых графиков.

Обработка графика - это трансформация по определенному закону исходных значений, на основании которых отрисовывается график. Обработки делятся на однопараметрические (трансформация значений одного графика) и двухпараметрические (операция между значениями двух графиков).

Для выполнения однопараметрической обработки (Рисунок 7) обычно требуется задать интервал, который определяет количество значений, участвующих в групповой обработке. Например, количество значений, по которым вычисляется среднее. Некоторые обработки имеют смысл только для наладчиков системы (прежде всего, вычисления), в то время, как другие – в основном статистические функции (дисперсия, среднеквадратичное, корреляция, фильтрация, усреднения и экстремумы) могут использоваться технологами для анализа технологического процесса.

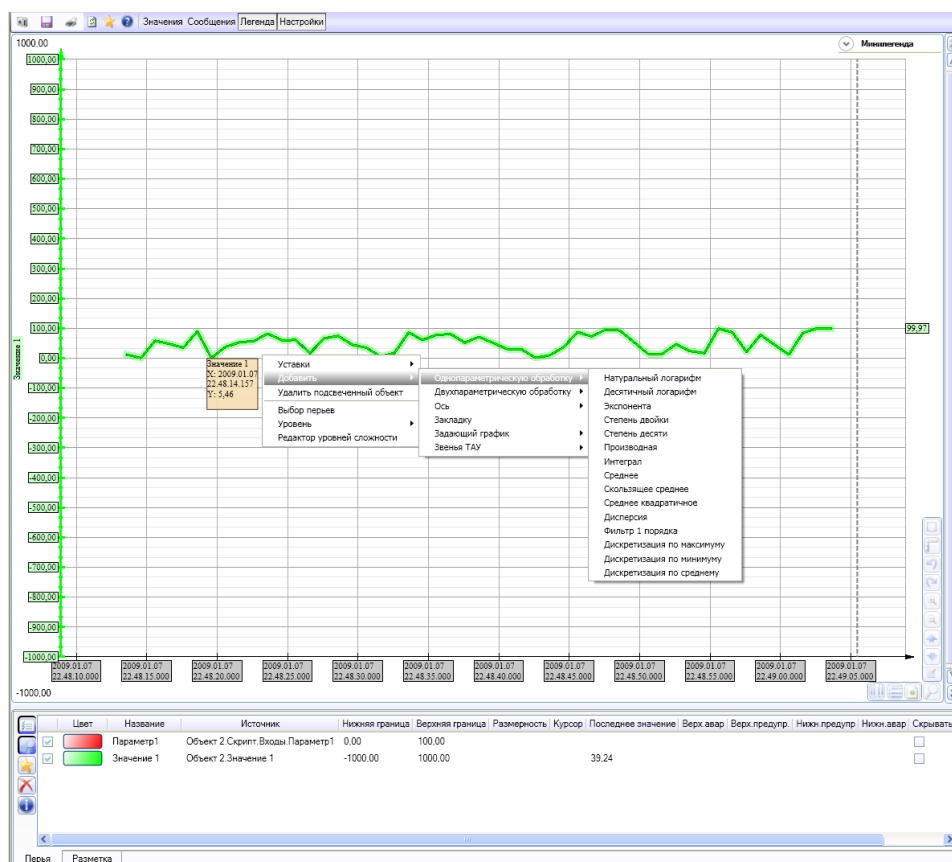


Рисунок 7

## Обработка в MasterReport

Требования к отчетам в последнее время выросли многократно. Это связано прежде всего с развитием систем коммерческого и технического учета различных ресурсов.

Многие виды вычислений для формирования отчетов неудобно, а то и невозможно сделать заранее, до начала постраничного формирования отчета (таковы, например, итоговые суммы столбцов на каждой странице). Для реализации многообразных и сложных потребностей в редактор отчета заложены мощные средства обработки исходных данных: вычисление формульных выражений, встроенные средства создания программ (C# или Visual Basic). Эти вычисления могут выполняться до или после формирования отчета, а также непосредственно в процессе интерпретации его внутренних полей. Возможно также использование или вычисление свойств визуальных элементов отчетов, прежде всего, столбцов таблиц для

реализации форматирования таблиц, зависящего от данных. Формирование (для просмотра или печати) отчета возможно, как с фиксацией значений на указанный момент времени, так и с вычислением текущего состояния. Может возникнуть вопрос – зачем нужны вычисления после построения отчета? Приведем пример: во время построения отчета производится расчет какой-то переменной. Результат этого расчета будет известен только в конце построения отчета, а вывести его нужно, например, на каждой странице.

Выражения допустимы в любом поле отчета прямо в тексте (Рисунок 8) и выделяются символами {}.

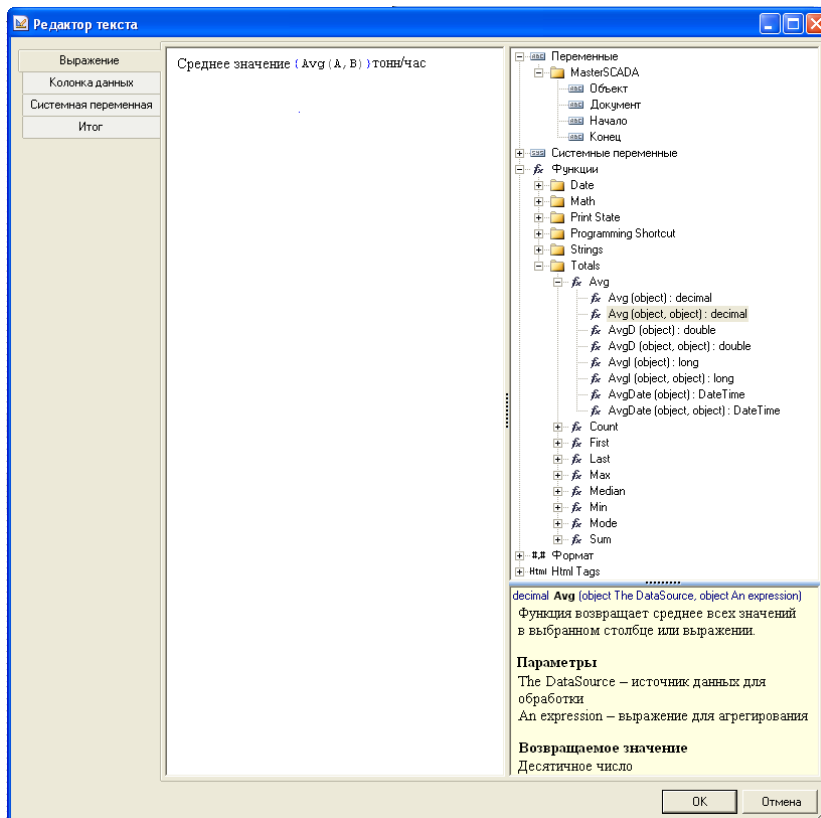


Рисунок 8

Формульные выражения позволяют использовать функции (включая стандартный набор агрегатных функций, типа суммы), внешние или локальные переменные, значения из SQL-совместимой базы данных, системные переменные типа дата, время, номер строки или страницы, а также свойства визуальных элементов, из которых составляется отчет, например, заголовка таблицы. Важно отметить, что выражения позволяют вычислять любую часть любой текстовой строки отчета.

Среди функций вычисления итогов есть такие специфические для отчетов возможности (Рисунок 9), как вычисление постраничных, поколоночных, построчных, нарастающих итогов, условного итога (например, с условием не считать строки с отрицательным значением) и т.п.

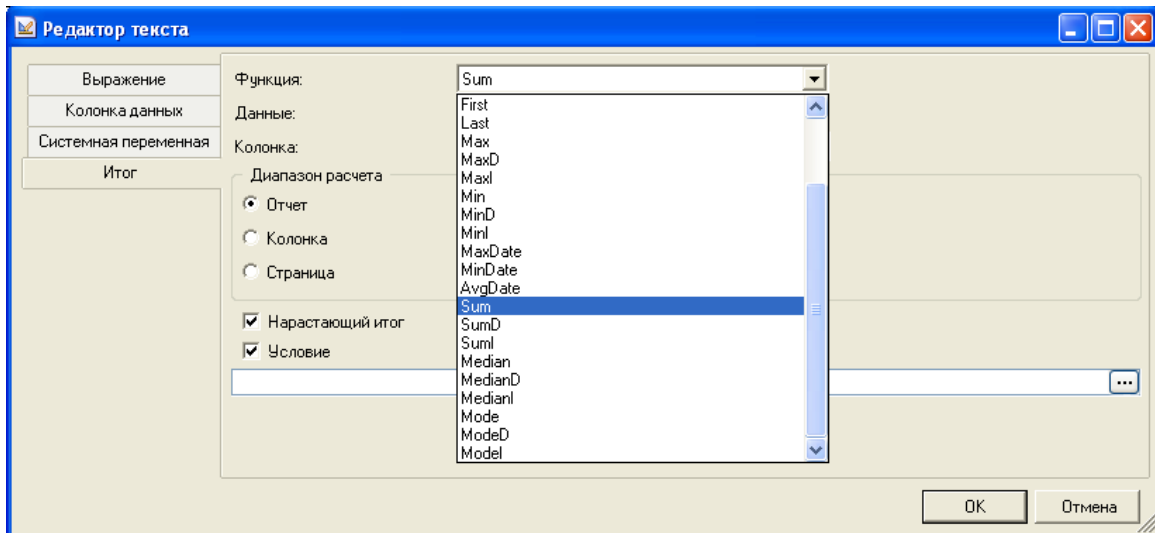


Рисунок 9

## Заключение

Широкие возможности вычислений в Master SCADA ориентированы именно на удобство обработки специализированных SCADA-данных и предоставляют разработчику проектов уникальный потенциал для их творческого использования. Среди нескольких разнообразных путей решения задачи важно выбрать тот вариант, который наиболее адекватен постановке задачи с точки зрения удобства реализации, наглядности и быстродействия.