

Формулы

Для вычисления значений параметров могут быть использованы формулы. Формула может начинаться с любого арифметического операнда или знака "=",

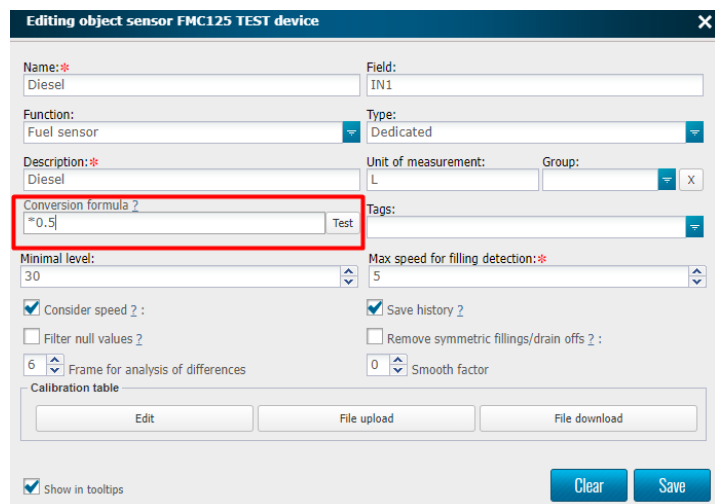


Рисунок 1

Если формула начинается с арифметического операнда, то значение параметра вычисляется.

На рисунке 1 выше формула применяется для настройки параметра IN1. Финальное значение параметра будет вычисленно по формуле $IN1 * 0,5$.

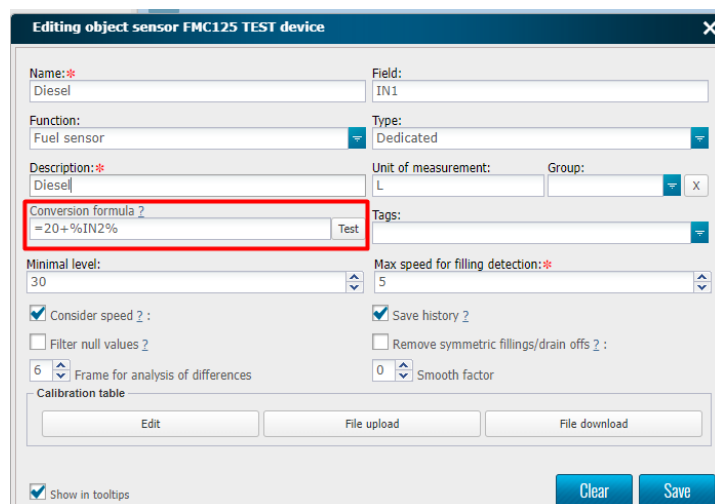


Рисунок 2

На рисунке 2 при получении параметра IN1, описанному датчику будет присвоено значение вычисленное по формуле $IN2 * + 20$.

Простые формулы

Формулы в обработчиках данных строятся на базе синтаксиса языка PHP и исполняются на сервере при обработке данных для конвертации их в нужный формат.

Формула может быть простой и начинаться непосредственно со знака арифметической операции. Поддерживаются операции прибавить (+), отнять (-), умножить (*), разделить (/) , так же поддерживаются круглые скобки для обеспечения правильной последовательности выполнения операции.

Например:

***10** - исходное значение будет умножено на 10

/1000 - исходное значение будет разделено на 1000

/100+7 - исходное значение будет разделено на 100 и к получившемуся результату будет прибавлено 7

/100+(7*3-2)*4

В формулу могут подставляться данные из других полей данных , скорость и количество спутников. Для подстановки значений в формулу их нужно поместить в экранирующие символы **%%**.

Для скорости используется переменная **%S%** , для количества спутников **%P%**, или **%satsinview%**. Для unixtimestamp текущего пакета используйте переменную **%ts%**, для unixtimestamp предыдущего пакета переменную **%ts1%**. Значение текущего датчика может быть подставлено переменной **%value%** .

Чтобы получить разницу в секундах относительно предыдущего пакета нужно вычислить **%ts1%-%ts%**.

Для всех остальных значений из пакета устройства берутся названия так как они присылаются драйвером.

Например: **%DIN1%** **%DIN2%** **%HDOP%**

Пример формулы с переменными:

***%DIN1%** - умножить исходное значение на значение присылаемое в поле DIN1

/%HDOP% - разделить исходное значение на значение из поля HDOP

Сложные формулы

Более сложные формулы начинаются со знака **=** и не используют исходное значение. Но его можно подставить в саму формулу с помощью переменной **%value%** либо переменной с именем поля.

Например:

=(1+2) - конечное значение будет равно 3

=(7/%value%) -конечное значение будет равно 7 деленному на исходное значение

Условные операторы

В формулах могут использоваться условия которые обеспечивают ветвление логики расчета зависящей от значений каких либо параметров или результатов операций.

Нотация условных операторов выглядит так: **условие ? значение_если_да : значение_если_нет**

Обычно такую запись называют — тернарный оператор.

Логически это выражение идентично выражению:

если условие вернуть значение_если_да иначе вернуть значение_если_нет

В условиях могут использоваться операторы сравнения равно (==), больше (>), меньше (<), больше или равно (>=), меньше или равно (<=), не равно (!=)

Например:

- если значение поля DIN1 равно 1 то конечное значение будет 22 иначе конечное значение будет равно 33

=(%temp%>30)? %temp%*10:%temp%*20 - если значение поля temp больше 30 то конечное значение будет равно значению поля temp умноженному на 10 иначе конечное значение будет равно значению поля temp умноженному на 20

Функции

В формулах могут использоваться некоторые PHP функции для преобразования данных. Это функции

unpack, pack, array_sum, substr, round, hex2bin, dechex, hexdec, decbin, number_format, max, min, abs, str_replace, time, strtotime, bytesrev, explode, sqrt, acos, asin, cos, sin, pow

передаваемые этим функциям значения и возвращаемые значения можно прочитать в документации PHP <https://www.php.net/>

Например:

=substr("%FIELD%", 1, 2) - конечное значение будет равно 2 символам из строки передаваемой в поле FIELD начиная с позиции 1

=round(%AIN1%/1000) - конечное значение будет равно округленному значению частного от деления значения поля AIN1 на 1000

Дополнительные функции

bytesrev

Например:

bytesrev('%value%')

строка **0123d848010000e5** преобразуется в **e500000148d82301** то есть разбивается по 2 символа и переворачивается

Деление на 0

Чтобы избежать непредсказуемых результатов при обработке значений датчиков следует избегать переменных в роли делителя или использовать условные операторы для проверки их на 0

Например:

=1000/%DIN1% - может породить ошибку вычисления в случае если значение поля DIN1 не определено или равно 0. В таком случае формулу нужно переписать так:

=(%DIN1%==0)? 0 : 1000/%DIN1% - если значение поля DIN1 равно 0 то конечное значение равно 0 иначе конечное значение равно частному от деления 1000 на значение поля DIN1