

ИМПОРТИРОВАНИЕ ТЕГОВ В MODBUS UNIVERSAL MASTER OPC

Импортирование тегов предназначено для быстрого создания конфигураций OPC сервера на основе CSV файлов. CSV файл – это простой, текстовый формат предназначенный для представления табличных файлов. Для формирования колонок используется специальный разделитель. Изначально в качестве разделителя использовалась запятая (аббревиатура CSV означает Comma-Separated Values — значения, разделённые запятыми), однако в настоящий момент чаще всего в качестве разделителя используют символ «;» - «точка с запятой». С CSV файлом может работать любой табличный редактор – Excel или Open Office Calc.

Чтобы произвести импорт CSV файла в OPC сервер необходимо чтобы файл был определенной структуры. По умолчанию требуется файл следующего формата:
Name;Address;DataType;Access;ByteOrder;Region;Comment

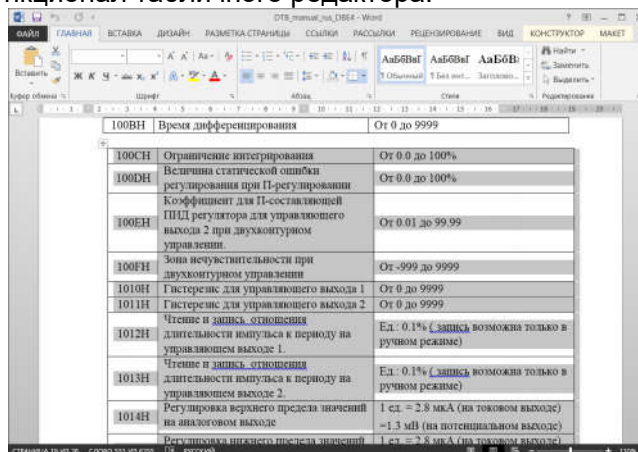
Статус;2;int16;ReadOnly;10325476;INPUT_REGISTERS;Статус измерения
значение;3;Float;ReadOnly;32105476;INPUT_REGISTERS;Измеренное значение
Статус;7;int16;ReadOnly;10325476;INPUT_REGISTERS;Статус измерения
значение;8;Float;ReadOnly;32105476;INPUT_REGISTERS;Измеренное значение
В табличном представлении данные записи будут выглядеть так:

Name	Address	DataType	Access	ByteOrder	Region	Comment
Статус	2	int16	ReadOnly	10325476	INPUT_REGISTERS	Статус измерения
значение	3	Float	ReadOnly	32105476	INPUT_REGISTERS	Измеренное значение
Статус	7	int16	ReadOnly	10325476	INPUT_REGISTERS	Статус измерения
значение	8	Float	ReadOnly	32105476	INPUT_REGISTERS	Измеренное значение

Таким образом, для импортирования нужно сформировать файл содержащий имя, адрес регистра, тип данных регистра, тип доступа, чередование байт, регион (функция которой нужно опрашивать регистр) и комментарий.

Каким образом можно получить CSV файл? Для конфигурируемых приборов (измерителей, регуляторов, модулей ввода-вывода) производитель предоставляет таблицу Modbus адресов в том или ином виде. Это может быть готовый Excel файл, однако чаще всего регистры описываются в документации представленной в pdf формате, но даже в этом случае извлечь данные не составит труда.

Сначала PDF файл нужно преобразовать в Word-формат – это можно сделать при помощи различных утилит (в том числе online-конвертеров) или воспользоваться Microsoft Word 2013 – он способен открывать и редактировать PDF формат. Открыв PDF в Word необходимо скопировать таблицу Modbus адресов в буфер, а затем вставить в Excel – теперь в вашем распоряжении весь функционал табличного редактора.



100BH	Время дифференцирования	От 0 до 9999
100CH	Ограничение интегрирования	От 0.0 до 100%
100DH	Величина статической ошибки регулирования при П-регулировании	От 0.0 до 100%
100EH	Коэффициент для П-составляющей ПИД регулятора для управляющего выхода 2 при двухконтурном управлении	От 0.01 до 99.99
100FH	Зона нечувствительности при двухконтурном управлении	От -999 до 9999
1010H	Гистерезис для управляющего выхода 1	От 0 до 9999
1011H	Гистерезис для управляющего выхода 2	От 0 до 9999
1012H	Чтение в записи, отношение длительности импульса к периоду на управляющем выходе 1.	Ед. 0.1% (запись возможна только в ручном режиме)
1013H	Чтение в записи, отношение длительности импульса к периоду на управляющем выходе 2.	Ед. 0.1% (запись возможна только в ручном режиме)
1014H	Регулировка верхнего предела значений на аналоговом выходе	1 ед. = 2.8 мА (на токовом выходе) =1.3 мВ (на потенциальном выходе) 1 ед. = 2.8 мА (на токовом выходе)

Адрес	Содержимое	Дополнение
1000	Текущее измеренное значение температуры PV (всперенная процесс)	Разрешение = 0.1. Обновление 1 раз в 0.4 сек.
1001	Значение уставки SV	Ед. измерения = 0.1 (DC или 0V)
1002	Верхний предел диапазона температуры	Ограничение значений уставки в верхнем пределе
1003	Нижний предел диапазона температуры	Ограничение значений уставки в нижнем пределе
1004	Тип используемого датчика температуры или аналогового сигнала	См. Тип температурного датчика или аналогового входа
1005	Метод регулирования	0: PID регулятор;
1006	Выбор режима работы (нагрев, охлаждение или двухзонный режим)	0: Нагрев; 1: Охлаждение; 2: Нагрев/охлаждение;
1007	Период следования импульсов на управляющем выходе 1	От 1 до 99 секунд
1008	Период следования импульсов на управляющем выходе 2	От 1 до 99 секунд
1009	Коэффициент пропорциональной составляющей PID регулятора	От 0.01 до 999.9
100A	Время интегрирования	От 0 до 9999
100B	Время дифференцирования	От 0 до 9999
100C	Ограничение интегрирования	От 0.0 до 100%
100D	Величина статической ошибки регулирования при П-регулировании	От 0.0 до 100%
100E	Коэффициент для П-составляющей PID регулятора для управляющего вы	От 0.01 до 99.99
100F	Зона нечувствительности при двухзонном управлении	От -999 до 9999
1010	Гистерезис для управляющего выхода 1	От 0 до 9999
1011	Гистерезис для управляющего выхода 2	От 0 до 9999
1012	Чтение и запись отношения длительности импульса к периоду на управле	Ед.: 0.1% (запись возможна только в ручном режиме)
1013	Чтение и запись отношения длительности импульса к периоду на управле	Ед.: 0.1% (запись возможна только в ручном режиме)
1014	Регулировка верхнего предела значений на аналоговом выходе	1 ед. = 2.8 мВ (на токовом выходе)

Используя штатный функционал Excel можно добавить новые столбцы и отредактировать имеющиеся. Например, в текущем файле адреса представлены в шестнадцатеричном виде, а при импорте в сервер адреса должны быть в десятичном. Преобразовать из одного формата в другой можно используя стандартную Excel команду – DEC().

DEC

Число: A2

Описание: 10

Преобразует текстовое представление числа в десятичном формате в десятичное значение.

Описание: основание системы счисления преобразуемого числа.

Значение: 4096

ОК Отмена

Адрес	Содержимое	Дополнение
4096	Текущее измеренное значение температуры PV (всперенная процесс)	Разрешение = 0.1. Обновление 1 раз в 0.4 сек.
4097	Значение уставки SV	Ед. измерения = 0.1 (DC или 0V)
4098	Верхний предел диапазона температуры	Ограничение значений уставки в верхнем пределе
4099	Нижний предел диапазона температуры	Ограничение значений уставки в нижнем пределе
4100	Тип используемого датчика температуры или аналогового сигнала	См. Тип температурного датчика или аналогового входа
4101	Метод регулирования	0: PID регулятор;
4102	Выбор режима работы (нагрев, охлаждение или двухзонный режим)	0: Нагрев; 1: Охлаждение; 2: Нагрев/охлаждение;
4103	Период следования импульсов на управляющем выходе 1	От 1 до 99 секунд
4104	Период следования импульсов на управляющем выходе 2	От 1 до 99 секунд
4105	Коэффициент пропорциональной составляющей PID регулятора	От 0.1 до 999.9
4106	Время интегрирования	От 0 до 9999
4107	Время дифференцирования	От 0 до 9999
4108	Ограничение интегрирования	От 0.0 до 100%
4109	Величина статической ошибки регулирования при П-регулировании	От 0.0 до 100%
4110	Коэффициент для П-составляющей PID регулятора для управляющего вых	От 0.01 до 99.99
4111	Зона нечувствительности при двухзонном управлении	От -999 до 9999
4112	Гистерезис для управляющего выхода 1	От 0 до 9999
4113	Гистерезис для управляющего выхода 2	От 0 до 9999
4114	Чтение и запись отношения длительности импульса к периоду на управле	Ед.: 0.1% (запись возможна только в ручном режиме)
4115	Чтение и запись отношения длительности импульса к периоду на управле	Ед.: 0.1% (запись возможна только в ручном режиме)
4116	Регулировка верхнего предела значений на аналоговом выходе	1 ед. = 2.8 мВ (на токовом выходе)

Используя средства Excel можно заполнить остальные поля таблицы.

Name	Address	DataType	Access	ByteOrder	Region	Comment
Значение	4096	int16	ReadWrite	10325476	HOLDING_REGISTERS	Текущее измеренное значение
Уставка	4097	int16	ReadWrite	10325476	HOLDING_REGISTERS	Значение уставки SV
Верхний предел диапазона	4098	int16	ReadWrite	10325476	HOLDING_REGISTERS	Верхний предел диапазона тем
Нижний предел диапазона	4099	int16	ReadWrite	10325476	HOLDING_REGISTERS	Нижний предел диапазона тем
Тип используемого	4100	int16	ReadWrite	10325476	HOLDING_REGISTERS	Тип используемого датчика тем
Метод регулирования	4101	int16	ReadWrite	10325476	HOLDING_REGISTERS	Метод регулирования
Выбор режима	4102	int16	ReadWrite	10325476	HOLDING_REGISTERS	Выбор режима работы (нагрев
Период следования Вых1	4103	int16	ReadWrite	10325476	HOLDING_REGISTERS	Период следования импульсо
Период следования Вых2	4104	int16	ReadWrite	10325476	HOLDING_REGISTERS	Период следования импульсо
Коэффициент пропорциональн	4105	int16	ReadWrite	10325476	HOLDING_REGISTERS	Коэффициент пропорциональн
Время интегрирования	4106	int16	ReadWrite	10325476	HOLDING_REGISTERS	Время интегрирования
Время дифференцирования	4107	int16	ReadWrite	10325476	HOLDING_REGISTERS	Время дифференцирования
Ограничение интегрирования	4108	int16	ReadWrite	10325476	HOLDING_REGISTERS	Ограничение интегрирования
Величина статической ошибк	4109	int16	ReadWrite	10325476	HOLDING_REGISTERS	Величина статической ошибк
П-составляющая	4110	int16	ReadWrite	10325476	HOLDING_REGISTERS	Коэффициент для П-состав
Зона нечувствительности при	4111	int16	ReadWrite	10325476	HOLDING_REGISTERS	Зона нечувствительности при
Гистерезис Выхода1	4112	int16	ReadWrite	10325476	HOLDING_REGISTERS	Гистерезис для управляющего
Гистерезис Выхода2	4113	int16	ReadWrite	10325476	HOLDING_REGISTERS	Гистерезис для управляющего
ШИМ Выход2	4114	int16	ReadWrite	10325476	HOLDING_REGISTERS	Чтение и запись отношения д
ШИМ Выход1	4115	int16	ReadWrite	10325476	HOLDING_REGISTERS	Чтение и запись отношения д
Верхний предел выхода	4116	int16	ReadWrite	10325476	HOLDING_REGISTERS	Регулировка верхнего предела

После того как CSV файл будет приведен к нужному формату, можно приступить к импорту CSV файла в OPC сервер. Для большей наглядности процедуры импорта мы подготовили видеопример. Его можно посмотреть по [ссылке](#).

Таким образом можно быстро формировать объемные конфигурации.

Шаблоны импорта Как быть если существует готовый CSV файл, но его формат не совсем соответствует требуемому формату для импорта? В этом случае на помощь придут шаблоны импорта.

Ранее мы использовали шаблон по умолчанию – у него все свойства тега получаются из полей CSV файла и никак дополнительно не обрабатываются. Есть возможность сделать собственный шаблон – который будет иметь нужные названия и количество полей, а также может обработать исходные данные по заданному алгоритму.

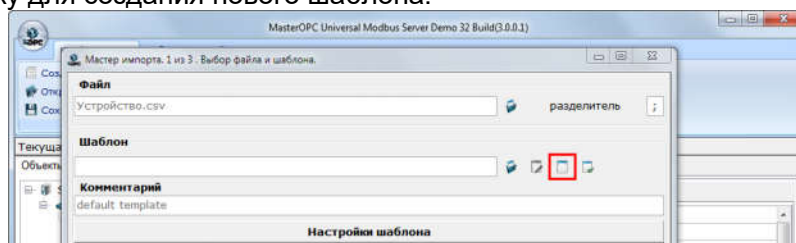
В качестве примера рассмотрим следующий CSV файл:

Name	Address	DataType	Access	Comment
Значение	8	Float	R/O	Измеренное значение
Уставка	9	Float	R/W	Значение уставки
Выход	A	Word	R/O	Выходное значение
ВП Авария	B	Word	R/W	Верхний аварийный предел
НП Авария	C	Word	R/W	Нижний аварийный предел

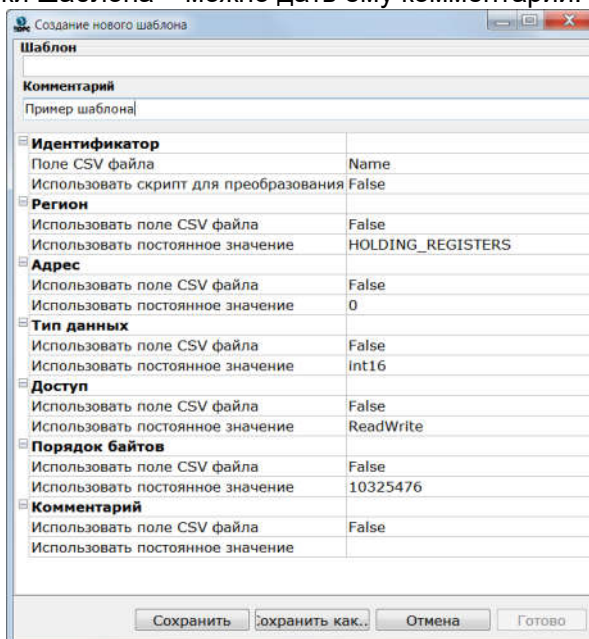
У данного CSV файла есть несколько отличий от шаблона:

1. Столбец с адресом регистра называется NumReg
2. Адреса регистров представлены в шестнадцатеричном формате.
3. Столбец DataType двухбайтовые числа обозначает как Word, который сервером не воспринимается (в сервере типу Word соответствует тип uint16).
4. Тип доступа имеет обозначения R/O, R/W вместо необходимых ReadOnly и ReadWrite.
5. Столбцы ByteOrder и Region отсутствуют. Чередование байт определяется типом данных, а Region – всегда Holding Registers.

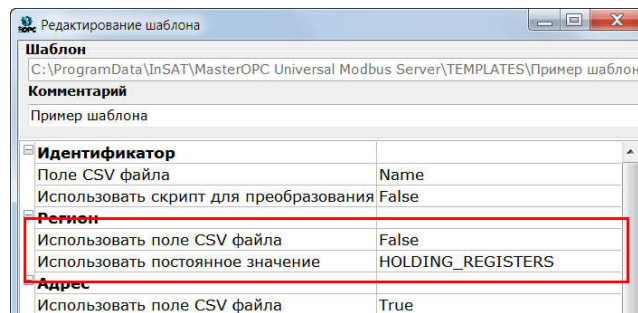
Рассмотрим, как сделать шаблон импорта для подобного файла. Запустим импорт файла, и нажмем на кнопку для создания нового шаблона.



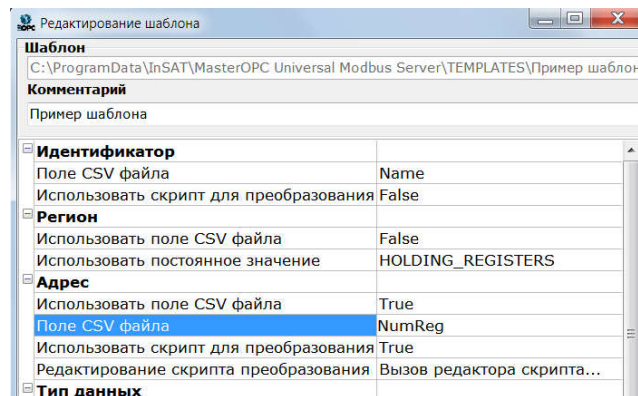
Появилось окно настройки шаблона – можно дать ему комментарий.



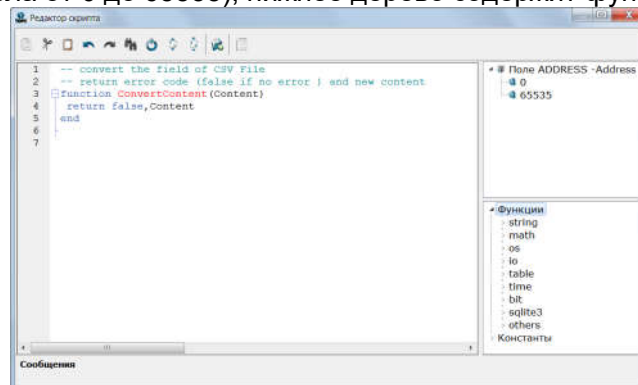
Адаптируем наш шаблон под CSV файл. Начнем с Региона. У нас регион всегда Holding_Registers, поэтому можно просто задать его в виде константы – отключить настройку «Использовать поле CSV файла» и указать регион.



Адаптируем адреса регистра – включим получение адреса из поля CSV файла и изменим название столбца на NumReg. У нас поля представлены в шестнадцатеричном формате, для перевода в необходимый десятичный формат воспользуемся скриптом – включим настройку «Использовать скрипт для преобразования» и нажмем на кнопку «Редактирование скрипта преобразования».



Открылось стандартное окно редактора скрипта. В левой части окна находится непосредственное поля для ввода кода скрипта. В правой части окна находится два дерева: верхнее дерево содержит константы с корректным именем возвращаемого свойства (в данном случае это числа от 0 до 65535), нижнее дерево содержит функции языка Lua.



Исходный код функции имеет следующий вид:

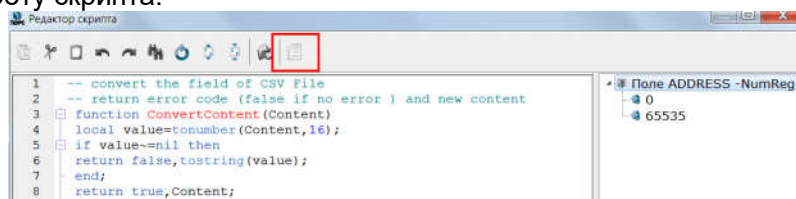
```
function ConvertContent(Content)
    return false,Content
end
```

Content – это исходное значение, считанное из CSV файла, именно его нам и нужно обработать. После обработки, нужно выполнить команду return и вернуть два значения – флаг ошибки (false если ошибки нет) и обработанное значение. Для перевода числа воспользуемся стандартной Lua функцией tonumber. Первым аргументом в функцию необходимо передать преобразуемую строку, вторым – основание системы счисления (в нашем случае – 16). В итоге код будет выглядеть следующим образом:

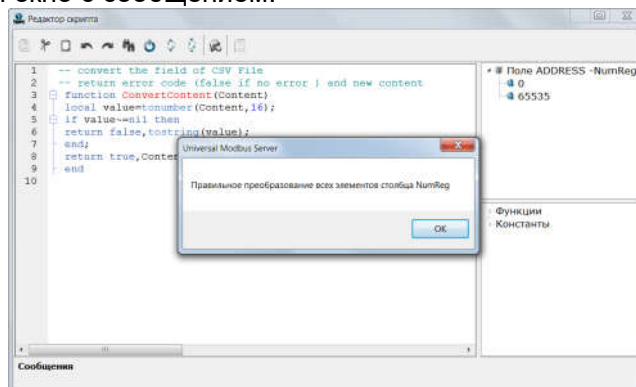
```
function ConvertContent(Content)
    local value=tonumber(Content,16);
    if value~=nil then
        return false,tostring(value);
    end;
    return true,Content;
end
```

Если значение будет преобразовано корректно, то будет возвращено отсутствие ошибки и преобразованное в десятичный формат значение, при этом значение нужно преобразовать в строку – функцией `tostring`. Если число преобразовать не удалось - функция `tonumber` вернула `nil`, то возвращается флаг ошибки и исходное значение.

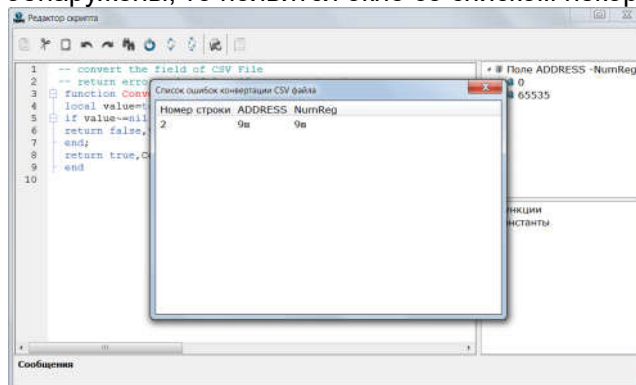
Можно в этом же окне проверить работу скрипта. Для этого в панели есть специальная кнопка – Проверить работу скрипта.



Нажмем на нее, и откроем файл, который мы хотим импортировать. Если ошибок не возникнет, то появится окно с сообщением.



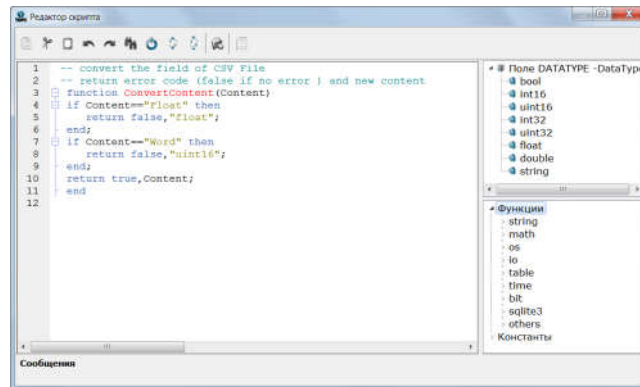
Если же ошибки будут обнаружены, то появится окно со списком некорректных ячеек.



Теперь разберемся с типом данных. Тип данных мы будем получать из поля CSV файла - `DataType`, но нужно преобразовать тип к нужному виду. Для этого напишем следующий код:

```
function ConvertContent(Content)
    if Content=="Float" then
        return false,"float";
    end;
    if Content=="Word" then
        return false,"uint16";
    end;
    return true,Content;
end
```

Поочередно проверяем содержимое поля `Content`, и возвращаем нужный тип. Если доходим до конца и ни один из типов не совпал, то возвращаем ошибку. В нашем случае преобразование достаточно простое – всего 2 типа, но, разумеется их может быть и больше.

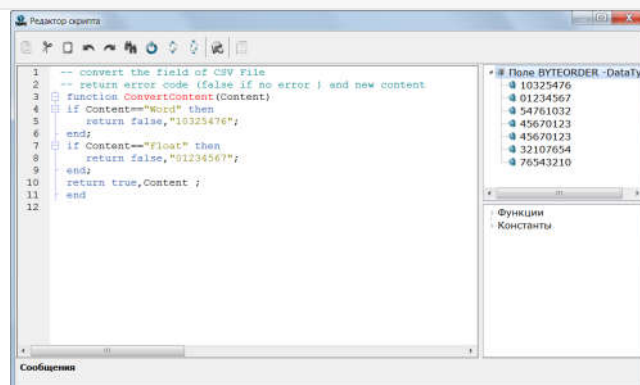


Теперь рассмотрим получение чередования байт. Мы будем определять чередование байт, по типу данных. Согласно стандарту Modbus переменные занимаемые 1 регистр (то есть тип Word) имеют чередование байт «Старшим байтом вперед» (10325476), а занимаемые 2 регистра (то есть float) имеют чередование байт «Старшим словом вперед» (01234567). Настроим данное свойство на получение значения поля DataTyре и включим скрипт. Код скрипта будет иметь следующий вид:

```

function ConvertContent(Content)
if Content=="Word" then
    return false,"10325476";
end;
if Content=="Float" then
    return false,"32104567";
end;
return true,Content ;
end

```

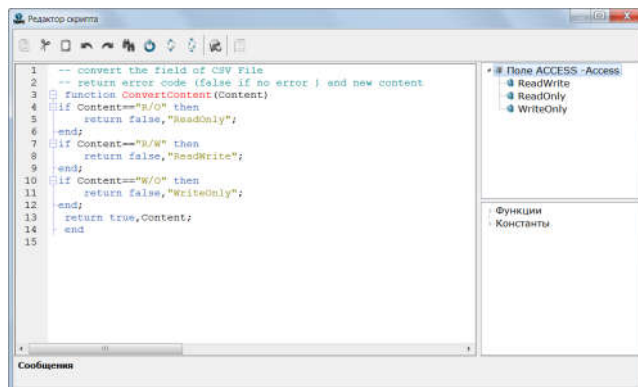


Аналогично сделаем получение типа доступа. Код будет иметь следующий вид:

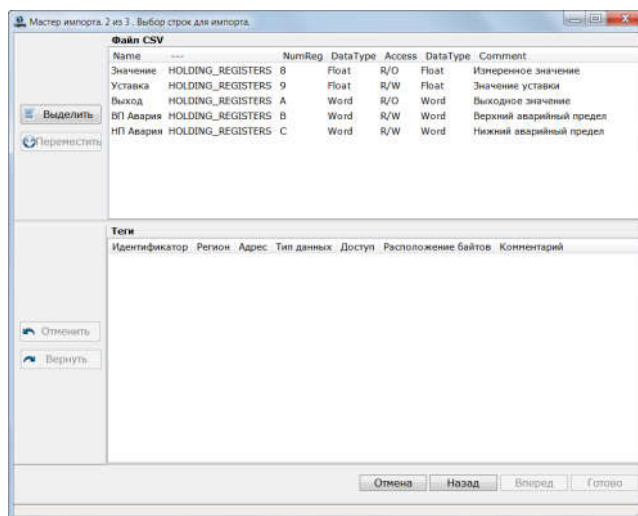
```

function ConvertContent(Content)
if Content=="R/O" then
    return false,"ReadOnly";
end;
if Content=="R/W" then
    return false,"ReadWrite";
end;
if Content=="W/O" then
    return false,"WriteOnly";
end;
return true,Content;
end

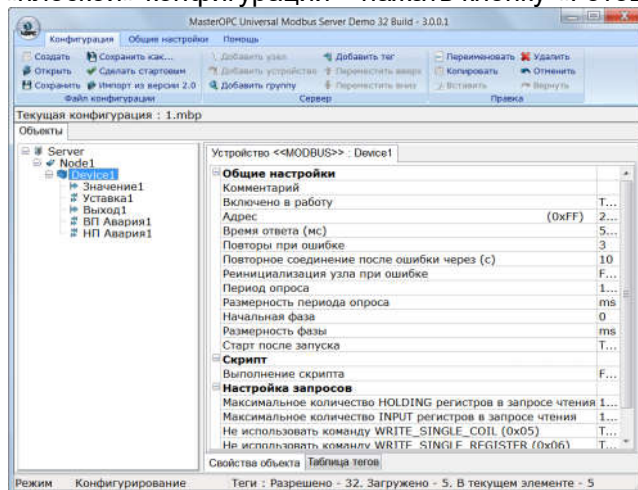
```



Теперь сформированный отчет можно сохранить в папку шаблонов и выполнить импортирование CSV файла. Импортирование прошло успешно – никаких ошибок не выдалось.



Можно выбрать необходимые поля для импорта, и перейти на шаг для создания иерархии, либо если достаточно «плоской» конфигурации – нажать кнопку «Готово».



Теги добавятся в устройство.

Таким образом, используя шаблоны импорта можно адаптировать мастер импорта под нужный тип CSV файлов.