

Modbus TCP/RTU эмулятор.

Modbus TCP/RTU эмулятор - программа имитирующая работу Modbus TCP/RTU ведомых устройств. Modbus TCP/RTU эмулятор, в дальнейшем программа, предназначена для разработчиков программного обеспечения, системных интеграторов, студентов и может быть использована для отладки Modbus клиентов, OPC/DDE серверов, отладки оборудования, отладки SCADA проектов, создания стендов имитации технологического оборудования, систем обучения оперативного персонала, диагностики каналов связи и для многих других задач.

Программа имитирует работу Modbus TCP и Modbus RTU устройств. Одной из отличительных особенностей программы является то, что пользователь способен реализовать практически любую функциональность. Достигается это с помощью скриптового модуля. Скрипты дают возможность реализовать сложные алгоритмы, такие как: ПИД регуляторы; имитация счетчиков газа, воды, работу технологического оборудования и т.д.; имитацию работы нормального режима промышленного оборудования; имитация возникновения аварийных и нештатных ситуаций промышленного оборудования; реализована возможность доступа к данным в пределах одного сервера, объединяющего несколько устройств, т.е. можно строить логические схемы и логику обрабатывающую в зависимости от возникшей ситуации и многое другое.

Работа с Modbus TCP/RTU эмулятором.

На рисунке 7 показана программа с окном вывода информации о состоянии серверов.

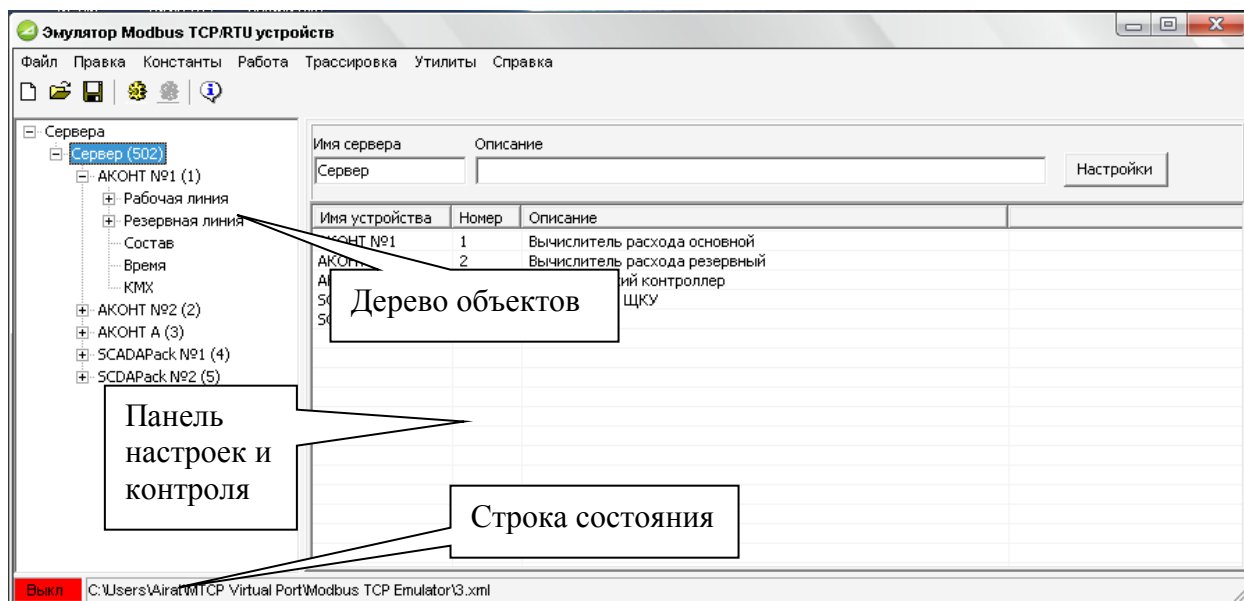


Рисунок 7

Окно программы разделено на пять зон:

1. Меню программы, состоящее из пунктов: Файл, Правка, Константы, Работа, Трассировка, Справка.
2. Панель инструментов, частично дублирующая функциональность меню программы.
3. Дерево объектов, дерево в котором отображаются сервера и устройства.
4. Панель настроек и контроля, предназначена для настройки программы и для контроля во время работы программы в режиме имитации.
5. Строка состояния, отображает текущее состояние программы, запущен режим имитации или нет, отображает путь к загруженному файлу с конфигурацией.

Для успешной работы необходимо разобраться с терминологией программы.

- Сервер – логический объект, предназначенный для организации канала связи между устройством и Modbus клиентом. Сервер поддерживает два типа каналов связи: Modbus TCP, для организации канала связи по сети Ethernet; Modbus RTU, для организации канала связи по последовательным портам.
- Устройство – объект, имитирующий работу Modbus Slave устройства. Устройство является контейнером для объектов типа «Группа», «Тег» и имеет сетевой адрес, в соответствии с которым оно отвечает на запросы клиентов.
- Группа – контейнер, содержащий в себе набор объектов типа «Группа» и «Тег». Группа может содержать в себе неограниченное количество групп и тегов. Группа не выполняет никаких функций кроме как разделения других групп и тегов на множества общей тематики или функционала.
- Тег – основной элемент программы, определяющий функциональность работы. Тег представляет собой указатель на выбранную область Modbus пространства. Тег работает со значением, располагающимся в указанной области. Тег извлекает содержимое указанного Modbus пространства, обрабатывает в соответствии с заложенной в него функциональностью и помещает результат назад в указанное пространство.

Работа Modbus TCP/RTU эмулятора в режиме настройки.

1. **Добавление и настройка Сервера.** Сервер добавляется выбором в меню пункта «Правка – Добавить сервер» или в контекстном меню дерева объектов пункт «Добавить сервер». На рисунке 8 показано окно программы с выбранным сервером.

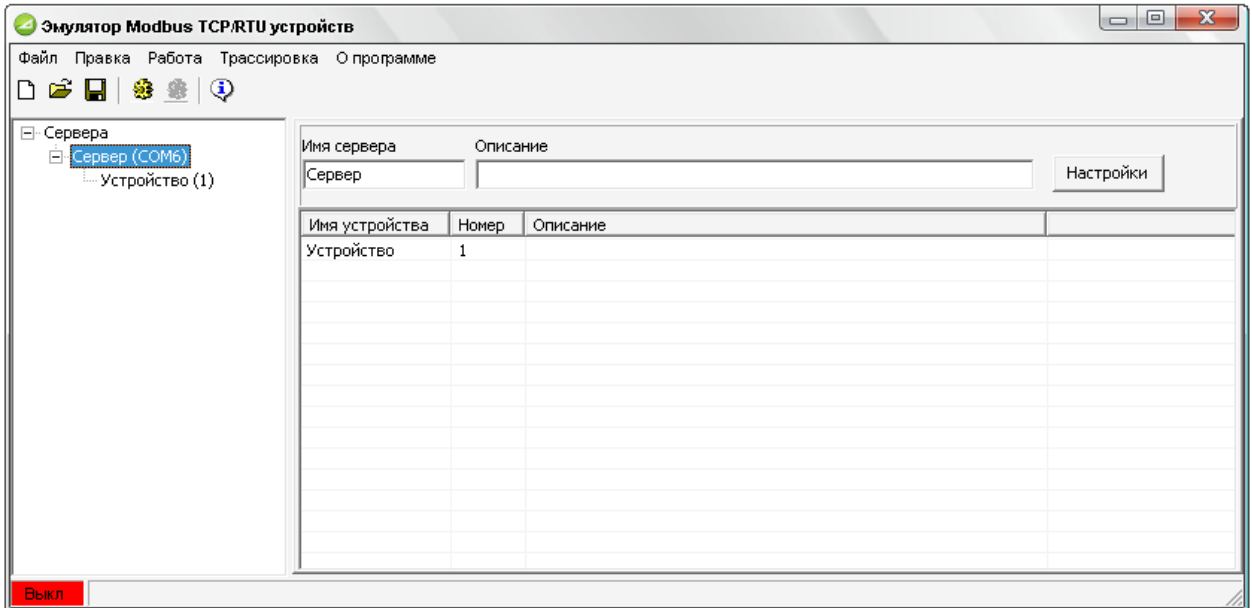


Рисунок 8

Как видно из рисунка 8 в программу добавлен сервер, «слушающий» com порт COM6, с одним устройством под сетевым номером №1. Панель настроек и контроля содержит информацию об имени сервера, описание сервера и список устройств принадлежащих серверу. Для настройки сервера используется кнопка «Настройки» по нажатию на которую появляется окно настроек сервера, показанное на рисунке 9.

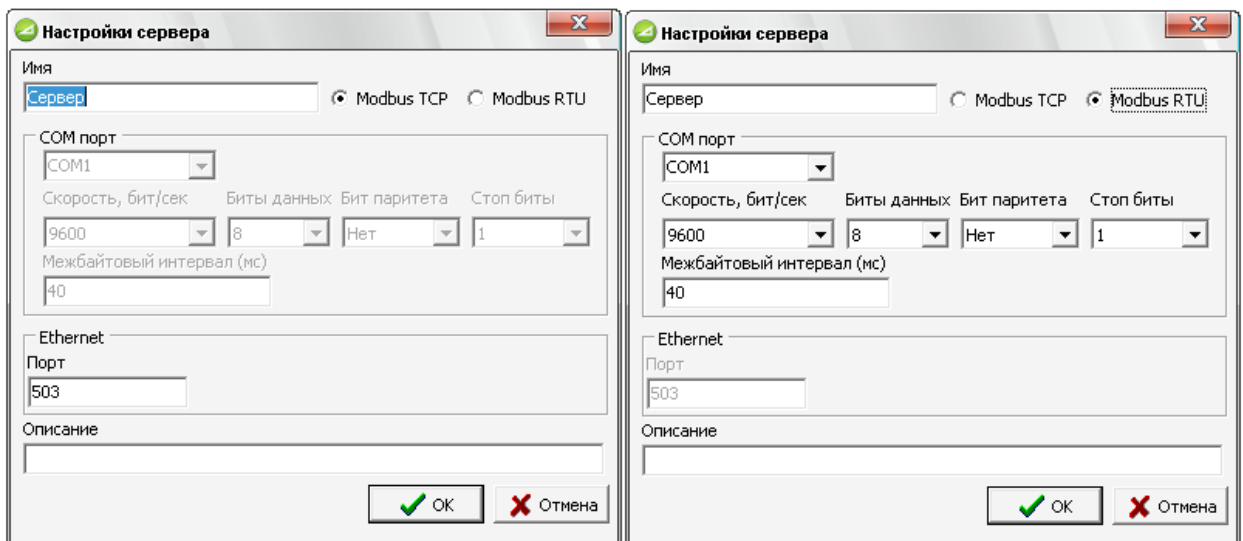


Рисунок 9

Сервер может работать в одном из двух режимов: Modbus TCP, Modbus RTU. Выбор между режимами осуществляется с помощью переключателей «Modbus TCP» и «Modbus RTU». Для каждого режима разблокируется свой набор элементов настройки.

В Modbus TCP режиме доступна только настройка TCP порта.

В Modbus RTU режиме доступны настройки COM порта: имя порта; скорость передачи данных; биты данных; бит паритета; стоп биты; межбайтовый интервал – допустимый интервал между приходящими байтами, по истечении которого считается что отправка клиентом информации завершена.

2. **Удаление Сервера.** Сервер удаляется выбором в меню пункта «Правка – Удалить сервер» или в контекстном меню дерева объектов пункт «Удалить сервер». При удалении сервера будут удалены все устройства, группы и теги находящиеся от него в зависимости.

3. **Добавление и настройка Устройства.** Для добавления устройства необходимо выбрать сервер, которому должно принадлежать устройство, выбрать в меню пункт «Правка – Добавить устройство» или в контекстном меню дерева объектов пункт «Добавить устройство». На рисунке 10 показано окно устройства.

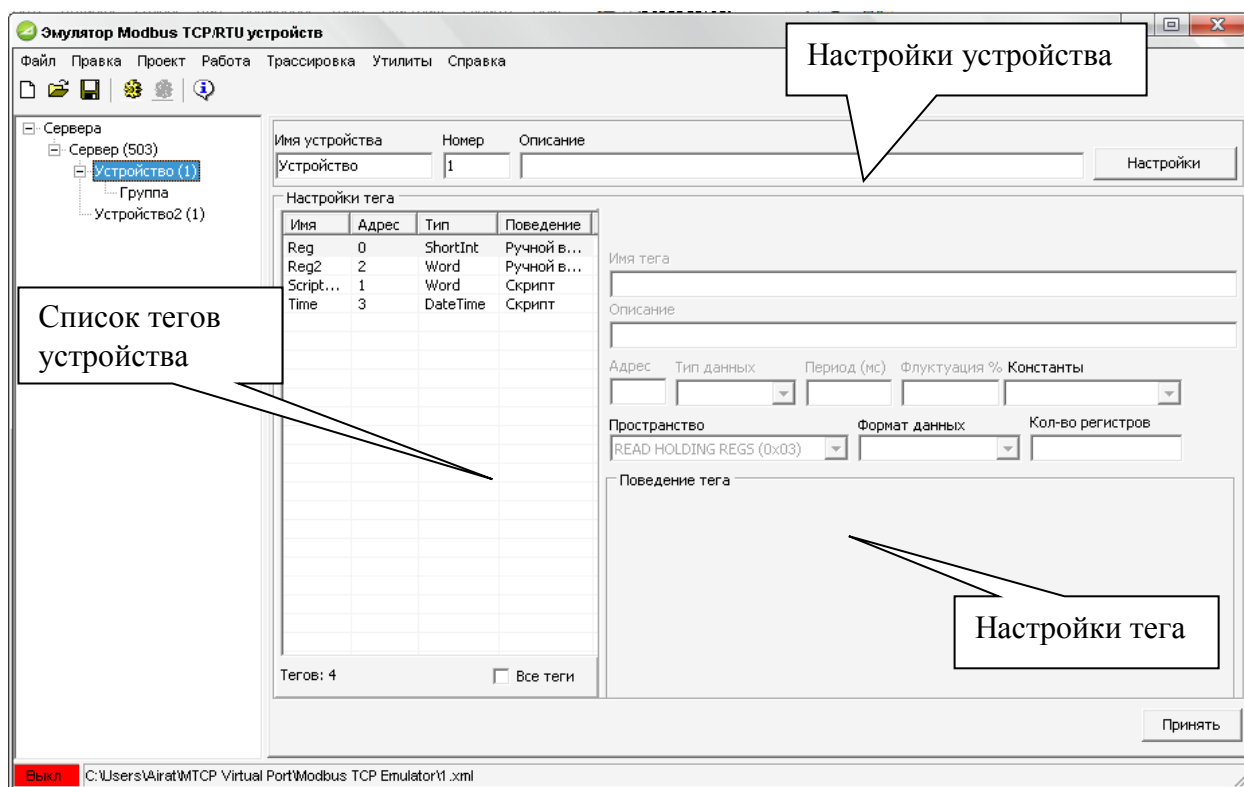


Рисунок 10

Окно настроек устройства состоит из двух частей: собственно настройки самого устройства и панель с тегами. Панель с тегами в свою очередь делится на панель со списком тегов устройства и панель с настройками единичного тега.

На рисунке 10 показана конфигурация устройства, имитирующая работу модуля аналогового ввода.

На рисунке 10 отображено 3 пункта настроек устройства:

- Имя устройства – наименование имитируемого устройства.
- Номер – сетевой номер устройства в сети Modbus.
- Описание – краткое описание устройства.

Для изменения настроек устройства необходимо нажать на кнопку «Настройки». После нажатия на кнопку появится окно настроек устройства показанное на рисунке 11.

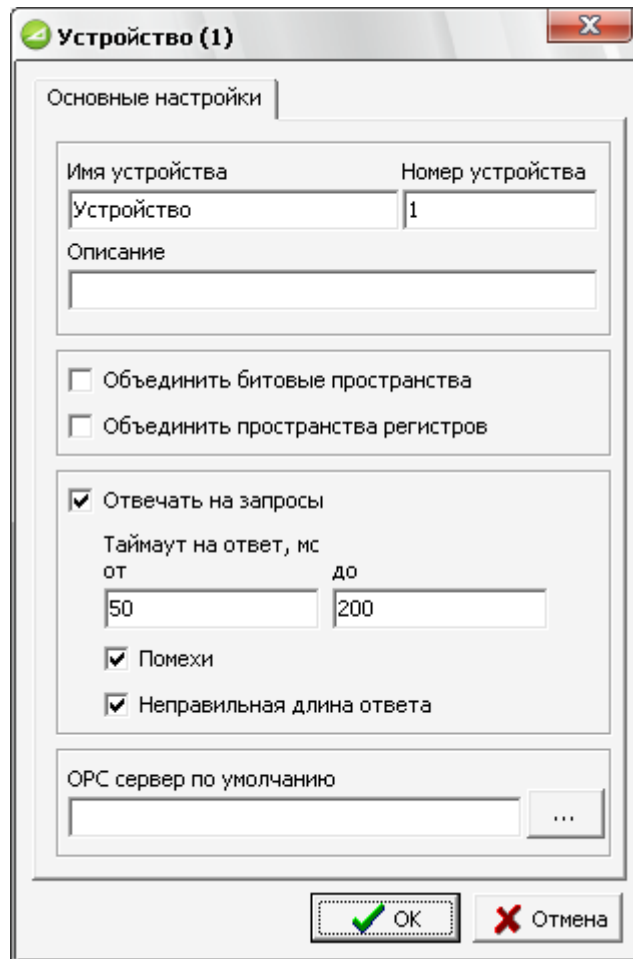


Рисунок 11

Помимо вышеперечисленных настроек устройства содержит еще несколько настроек:

- Отвечать на запросы – при снятии этого флага устройство не будет обрабатывать, и отвечать на запросы программ верхнего уровня. При снятии этого флага RTU устройство просто не отвечает, с TCP устройством соединение устанавливается, но запросы остаются без внимания.
- Таймаут на ответ – содержит два поля «от» и «до», если поля имеют разное значение то устройство будет выбирать случайное число из указанного диапазона и «молчать» в течении этого времени.
- Помехи – это поле активирует функцию выдачи в линию связи случайного набора данных на каждые 10 – 20 запросов, выбирается случайно, формируется один пакет с помехами.
- Неправильная длина ответа – на каждые 10 – 20 запросов, выбирается случайно, сформированный ответ искажается: для Modbus TCP искажается заголовок ответа, для Modbus RTU уменьшается длина пакета.
- Объединить битовые пространства – когда выставлен этот флаг, чтение командами Modbus 0x01 и 0x02 осуществляется из одного буфера.
- Объединить пространства регистров – когда выставлен этот флаг, чтение командами Modbus 0x03 и 0x04 осуществляется из одного буфера.
- OPC сервер по умолчанию – при добавлении тега работающего с данными из OPC этот сервер будет предлагаться по умолчанию.

Пункты «Отвечать на запросы» и «Таймаут на ответ» могут изменяться во время работы программы в режиме имитации.

4. **Удаление устройства.** Устройство удаляется выбором в меню пункта «Правка – Удалить устройство» или в контекстном меню дерева объектов пункт «Удалить устройство». При удалении устройства будут удалены все группы и теги принадлежащие ему. Устройство не может быть удалено, если на один из тегов устройства есть ссылка из скрипта. На рисунке 13 показано сообщение, возникающее при невозможности удаления устройства.

5. **Добавление группы.** Для добавления группы надо выбрать устройство или группу и выбрать в меню пункт «Правка – Добавить группу» или в контекстном меню дерева объектов пункт «Добавить группу». Вложенность групп неограниченна. На рисунке 12 показано окно добавления группы.

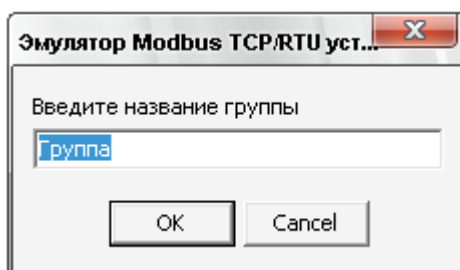


Рисунок 12

6. **Удаление группы.** Группа удаляется выбором в меню пункта «Правка – Удалить группу» или в контекстном меню дерева объектов пункт «Удалить группу». Группа удаляет все вложенные в нее группы и теги. Группа не может быть удалена, если на один из тегов группы есть ссылка из скрипта. На рисунке 13 показано сообщение, возникающее при невозможности удаления группы.

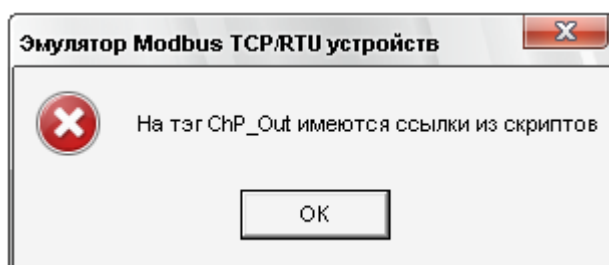


Рисунок 13

7. **Добавление и настройка тега.** Для того, что бы добавить тег необходимо в дереве объектов выбрать устройство и группу, выбрать в меню пункт «Правка – Добавить тег» или в контекстном меню списка тегов устройства (рисунок 10) пункт «Добавить тег» или использовать сочетание клавиш «Ctrl+A».

Каждый тег содержит следующие настройки:

- Имя тега – имя тега, по которому идентифицируется тег. Имя тега не уникально, может повторяться.
- Описание – краткое описание тега.
- Адрес – Modbus адрес, на который ссылается тег. Адресация ведется в диапазоне 0 ÷ 65535. Для небулевых тегов адрес задается в виде целого числа, для булевых переменных адрес задается в виде X.Y. Где X – адрес регистра в Modbus пространстве, Y – смещение в битах в регистре (0÷15). Пример задания адреса для булевого тега: 1.0.
- Тип данных – определяет с каким типом данных работает тег. Поддерживаемые на текущий момент типы данных:

- Word – 16 разрядное целое число в диапазоне $0 \div 65535$;
 - Float – 32 разрядное число с плавающей запятой в диапазоне $-1,5 \cdot 10^{45} \div 3,4 \cdot 10^{38}$;
 - ShortInt – 16 разрядное целое число в диапазоне $-32768 \div 32767$;
 - Integer – 32 разрядное целое число в диапазоне $-2147483648 \div 2147483647$;
 - DWord – 32 разрядное целое число в диапазоне $0 \div 4294967296$;
 - Int64 – 64 разрядное целое число в диапазоне $-9223372036854775808 \div 9223372036854775807$;
 - Bool – булева переменная способная принимать только два значения 0 или 1, True или False;
 - DateTime – 64 разрядное число с плавающей запятой для работы со временем, целая часть числа это количество дней прошедших с 30.12.1899, дробная часть числа при умножении на 100 показывает сколько процентов времени от 24 часов истекло за текущие сутки. Для работы со временем в скриптах заложены специальные функции, облегчающие с ним работу;
 - Double – 64 разрядное число с плавающей запятой в диапазоне $-5,0 \cdot 10^{324} \div 1,7 \cdot 10^{308}$;
 - String – символьная строка, размер строки задается в настройках тега.
- Период – период в мс с которым обрабатывается функциональность тега.
 - Пространство – выбор Modbus пространство, на которое указывает адрес. На текущий момент программа поддерживает два вида пространств: Read Holding Registers (с кодом чтения 3), поддерживает как чтение так и запись; Read Input Registers (с кодом чтения 4) поддерживает только чтение. Запись данных извне осуществляется только в Read Holding Registers пространство, командами записи Write Single Register (код записи 06), Write Multiple Registers (код записи 16), Mask Write Register (код записи 22).
 - Поведение тега – определяет функциональность работы тега в режиме имитации. Реализованные на текущий момент функциональности:
 - Ручной ввод – в этом режиме значение тега только отображается на экране, также имеется возможность ручного ввода данных или доступ на чтение и запись из скриптов.
 - Инкремент – при каждом проходе тег инкрементируется на число заданное в поле ввода, число инкремента может быть как положительным так и отрицательным.
 - Случайное число – формируется случайное число в диапазоне заданном в полях Min и Max.
 - Синусоида – формируется синусоида с амплитудой заданной в поле «Амплитуда», с периодом в мс заданным в поле «Период», со смещением относительно нуля заданным в поле «Смещение».
 - Пила – формируется пила с амплитудой заданной в поле «Амплитуда», с периодом в мс заданным в поле «Период», со смещением относительно нуля заданным в поле «Смещение», так же выбирается форма пилы в поле «Тип пилы».
 - Скрипт – функциональность определяется пользователем, путем написания pascal скрипта. Более подробное описание скриптового языка и примеры использования показаны в следующем разделе. Скрипт редактируется в отдельном окне, вызываемом кнопкой «Редактировать...», в настройках тега скрипт отображается только для чтения.
 - Флуктуация – случайная составляющая для функциональностей типа «Случайное число», «Инкремент», «Синусоида», «Пила». Для случайного числа изменение происходит на процент указанный в поле флуктуация. К результату добавляется случайная составляющая в % указанном от текущего значения, при 0 случайная составляющая не формируется.

Для принятия изменений в настройки тега необходимо нажать на кнопку «Принять» в правом нижнем углу панели контроля и управления.

Рисунок 15

9. **Дубликат тега.** Выбранный тег дублируется путем выбора пункта в меню пункта «Правка – Дубликат» или в контекстном меню списка тегов устройства «Дубликат» или использовать сочетание клавиш «Ctrl+D». Эта функция предназначена для быстрого добавления тегов с одинаковыми свойствами.

10. **Копировать, Вставить, Вырезать.** Функции осуществляются над выбранными тегами и группами. Теги или группы можно скопировать и вставить в другое устройство или группу. Функция вырезать работает так же, как и копировать, только оригинал тега или группы не сохраняется на прежнем месте.

11. **Добавить устройство из шаблона, Сохранить устройство в шаблон.** Начиная с версии 2.15.24.X добавилась возможность работы с шаблонами устройств. Для сохранения устройство в шаблон необходимо выбрать интересующее устройство в дереве объектов и выбрать пункт меню «Правка → Сохранить устройство в шаблон». В файл с расширением tmpl будет сохранено устройство со всеми его группами и тегами. Сохраненный шаблон можно в дальнейшем использовать в других конфигурациях, просто используя пункт меню «Правка → Добавить устройство из шаблона». При работе с шаблонами необходимо помнить, что при использовании скриптов сохраняются ссылки только на внутренние теги устройства, ссылки на внешние теги теряются.

12. **Настройки перехода программы в режим имитации, автозагрузка и автозапуск.** При переходе в режим имитации программа отображает окно, показанное на рисунке 16.

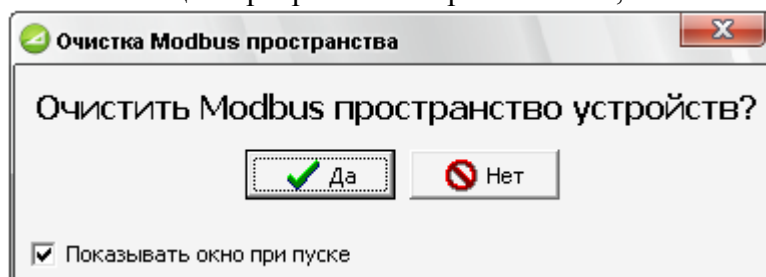


Рисунок 16

В этом окне запрашивается обнулять содержимое Modbus регистров перед переходом в режим имитации или оставить значения оставшиеся после предыдущего пуска. Если в поле «Значение по умолчанию» тега работающего в режиме «Ручной ввод» введено не нулевое значение, то в регистры будет записано именно оно.

При выборе пункта меню «Работа – Настройки...» появляется окно, показанное на рисунке 17. В этом окне можно настроить файл конфигурации, загружаемый автоматически при пуске программы, автоматический переход в режим имитации, отображать ли окно, показанное на рисунке 16 и что делать с Modbus пространством в случае если не выбрана настройка «Показывать окно обнуления при пуске».

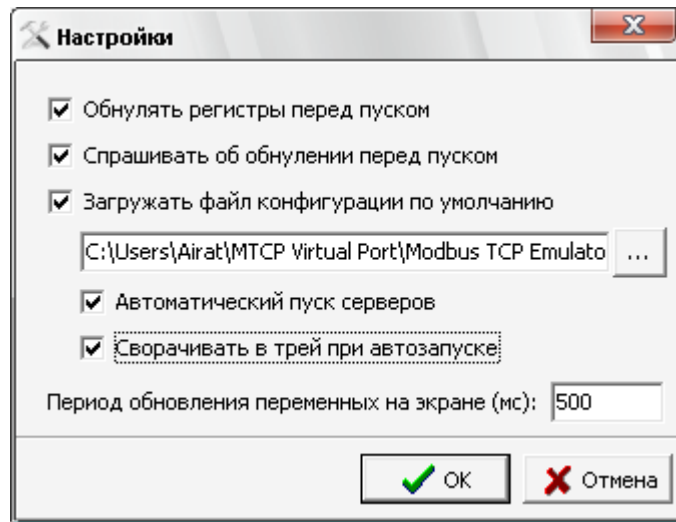


Рисунок 17

13. Утилиты.

Проверка контрольной суммы.

В программе имеется возможность проверки или расчета контрольной суммы Modbus пакета. Окно проверки показано на рисунке 18.

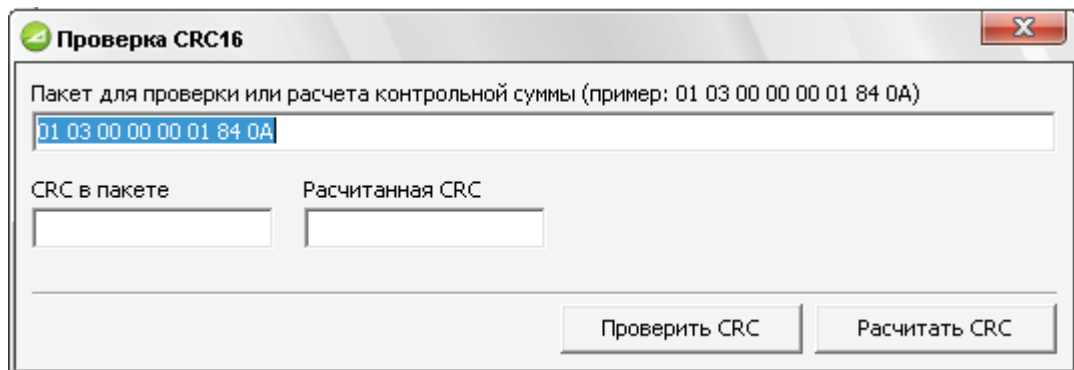


Рисунок 18

В поле «Пакет для проверки или расчета контрольной суммы» вводится пакет, для которого необходимо рассчитать или проверить контрольную сумму. После этого нажимается кнопка «Проверить CRC» для проверки контрольной суммы пакета или «Расчитать CRC» для расчета контрольной суммы пакета. В полях «CRC пакета» и «Расчитанная CRC» выводятся контрольные суммы присутствующая в пакете и рассчитанная соответственно.

Поиск тега

На рисунке 19 показано окно поиска тега. Поиск может осуществляться по двум параметрам: поиск по имени тега и поиск по адресу тега.

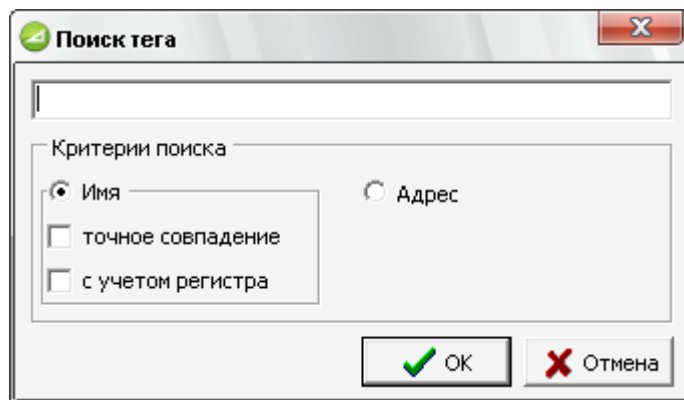


Рисунок 19

После ввода искомых данных и нажатия на кнопку «ОК» поиск тега по всем серверам и устройствам, после чего показывается окно изображенное на рисунке 20. В этом окне выводится полный путь к искомому тегу и в скобках дополнительно показан адрес по которому располагается тег.

№	Тег
1	Сервер.SCADAПак.АI.FT304_PV (413)

Рисунок 20

14. Константы

Для вызова окна редактирования констант (рисунок 21) необходимо выбрать пункт меню «Константы».

Как видно из рисунка константы подразделяются на группы соответствующие собственной тематике, например: двигатель, задвижка, насос и т.д.

Каждой группе можно назначить свой набор констант, соответствующих заданному значению.

При работе в режиме имитации вместо значения отображается его расшифровка с цифровым значением в скобках, рисунок 22.

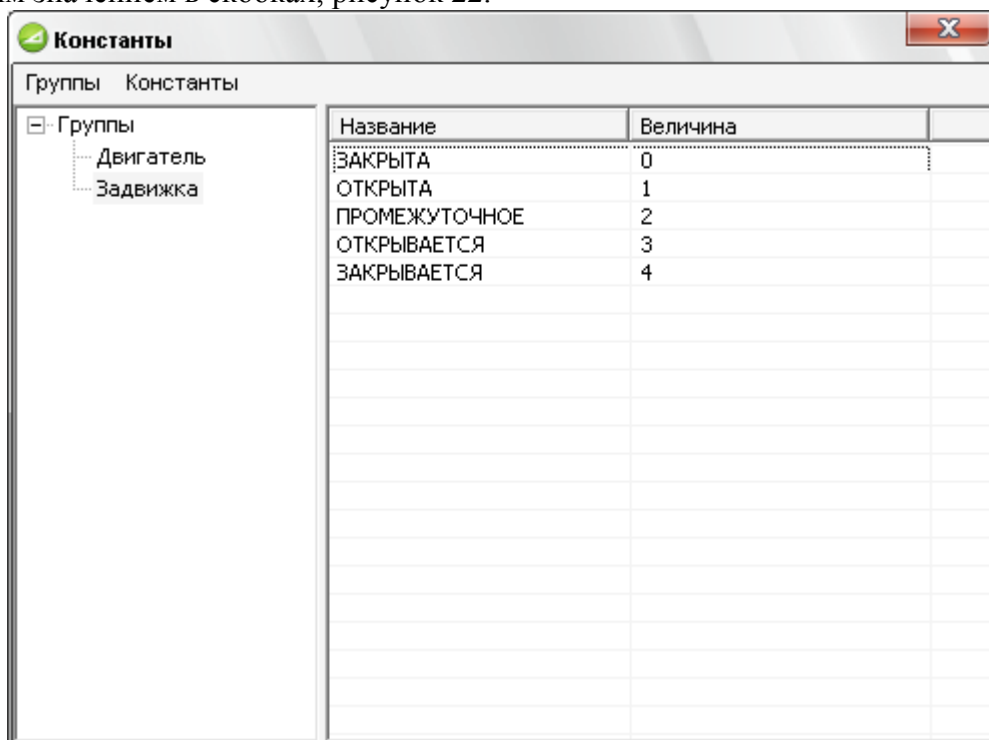


Рисунок 21

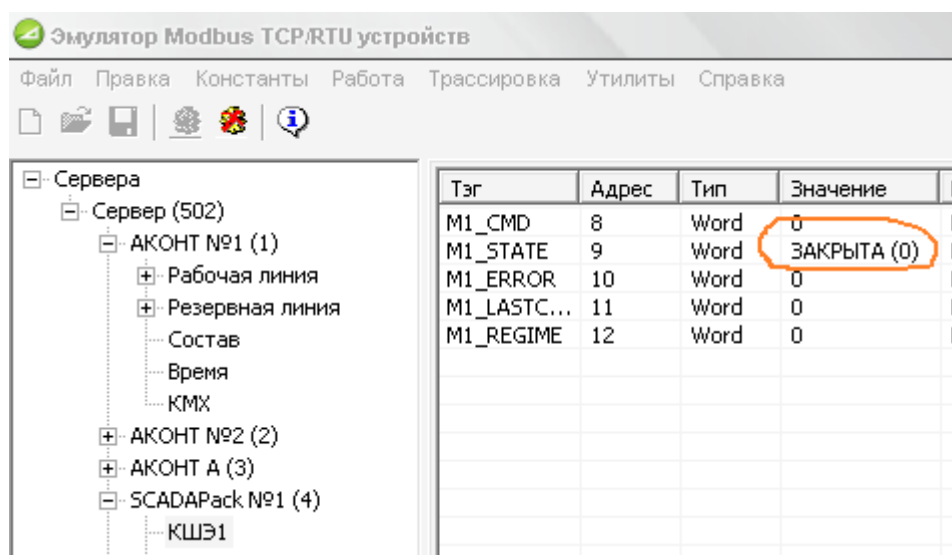


Рисунок 22

15. Функции.

Часто бывает необходимо выполнить в разных скриптах одни и те же вычисления с разными входными параметрами. Для того, что бы не реализовывать один и тот же программный код в разных местах, в Modbus эмулятор была введена возможность добавления собственных функций.

Добавление и редактирование функций осуществляется с помощью окна «Функции» по казанное на рисунке 23. Окно «Функции» вызывается из меню «Проект→Функции».

Окно редактирования функций разделено на две части:

- Список функций – эта часть предназначена для добавления, удаления и изменения настроек функций;
- Список параметров функции – эта часть предназначена для управления входными параметрами функции.

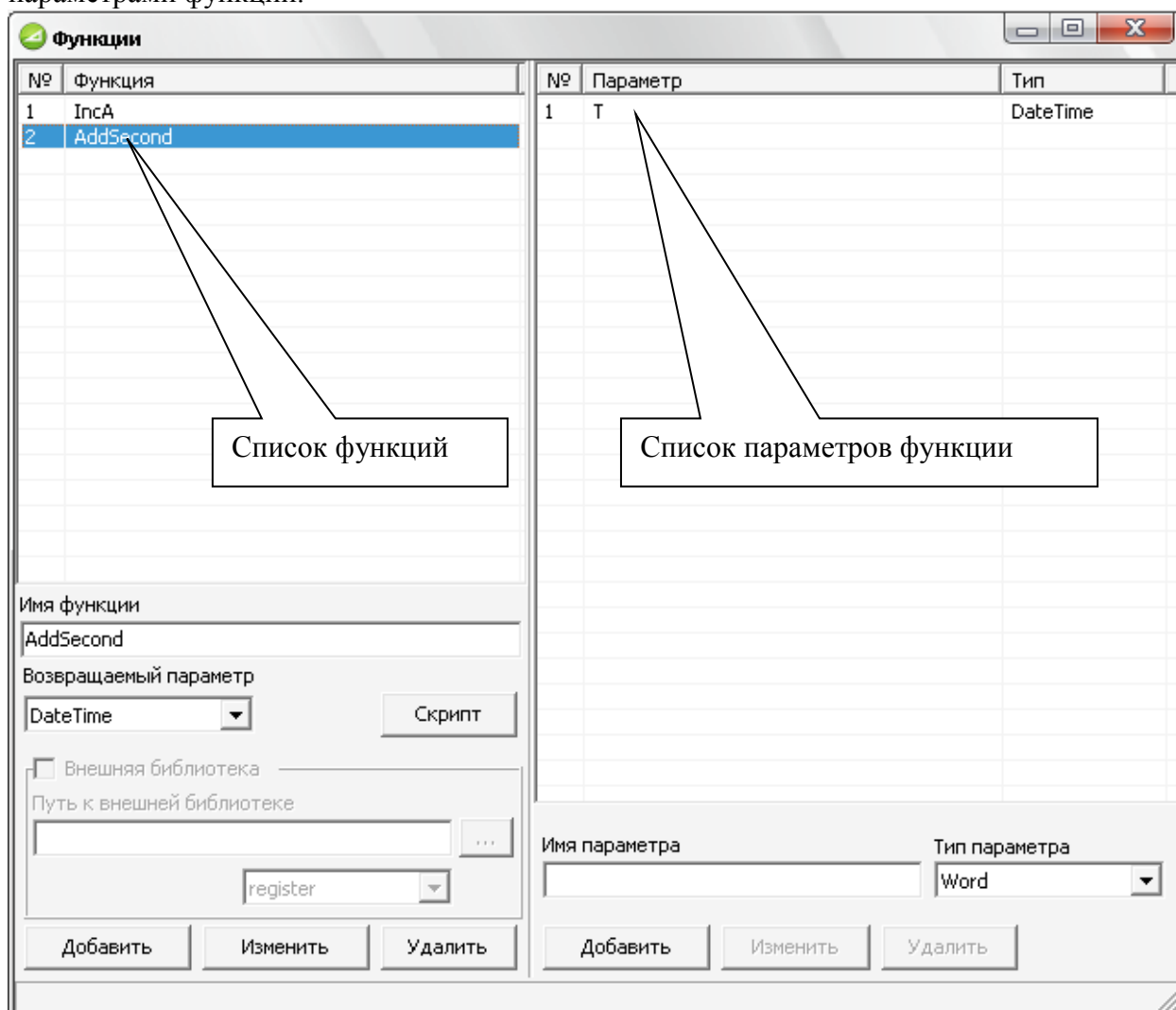


Рисунок 23 Окно управления функциями

Добавление функции.

Для того, что бы добавить новую функцию необходимо в поле «Имя функции» ввести ее название, выбрать тип возвращаемого функцией результата и нажать на кнопку «Добавить» расположенную в части списка функций.

Изменение параметров функции.

Для изменения параметров функции необходимо выбрать функции в списке функций, внести изменения в имя функции или сменить тип возвращаемого параметра и нажать на кнопку «Изменить».

Удаление функции.

Для удаления функции необходимо выбрать функцию из списка функций и нажать на кнопку «Удалить».

Добавление параметра функции.

Для добавления параметра функции необходимо ввести имя параметра в поле «Имя параметра», выбрать тип передаваемого параметра в функцию и нажать на кнопку добавить.

Удаление параметрам функции.

Для удаления параметра необходимо выбрать его из списка параметров и нажать на кнопку «Удалить».

Изменение параметра функции.

Для изменения параметра необходимо выбрать его из списка параметров, задать новое имя и тип параметра и нажать на кнопку «Изменить».

Редактирование скрипта функции.

Для редактирования скрипта функции, необходимо выбрать ее из списка функций и нажать на кнопку «Скрипт». После нажатия на кнопку «Скрипт» появляется окно редактирования скрипта функции показанное на рисунке 24. Подробнее о работе со скриптами смотрите в главе «Использование скриптов в Modbus TCP/RTU эмулятор», а так же «Приложение №1».

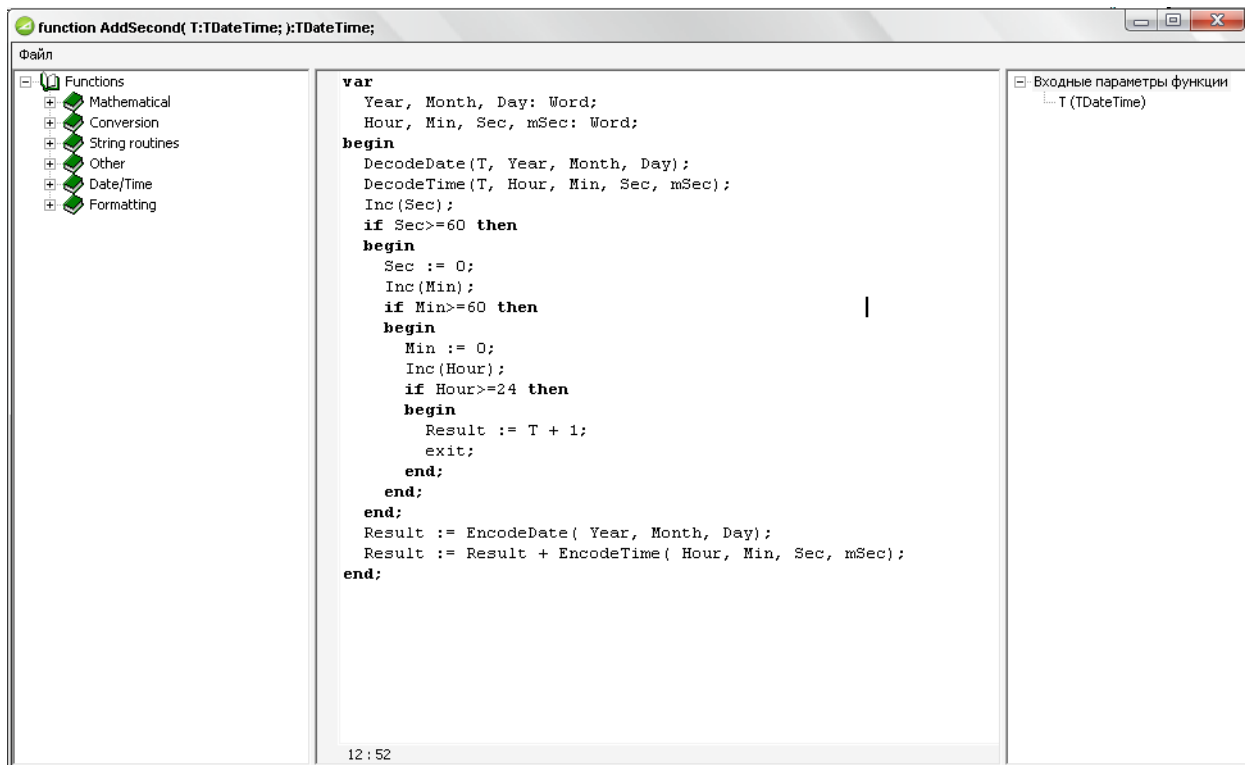


Рисунок 24 Окно редактирования скрипта функции

Работа Modbus TCP/RTU эмулятора в режиме имитации.

1. Переход в режим имитации. Для перехода в режим имитации необходимо выбрать пункт меню «Работа - Пуск» или нажать на клавишу «F9». На рисунке 23 показано окно программы в режиме имитации. Панель настроек и контроля в этом режиме поделена на две части:

- Окно мгновенных значений. В этом окне отображаются мгновенные значения тегов, а так же краткая сводка по каждому тегу.
- Окно ошибок. В этом окне отображаются ошибки, возникшие в ходе работы. Выводится время и дата возникновения ошибки и описание возникшей ошибки. На рисунке 25 показано, что в скрипте для тега FT305 возникла ошибка – деление на ноль.

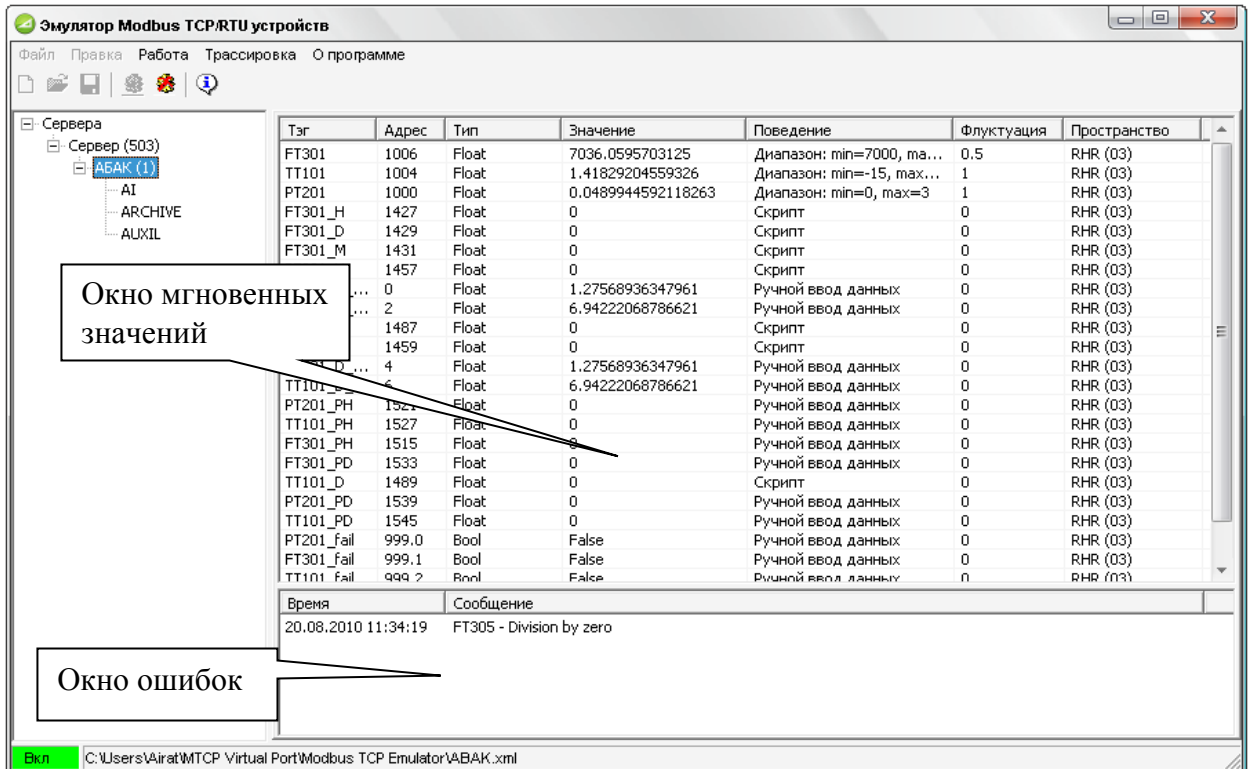


Рисунок 25

2. Переход в режим настройки. Для перехода в режим настройки необходимо в меню выбрать пункт «Работа - Останов» или нажать сочетание клавиш «Ctrl-F2». Программа перейдет в режим настройки.

3. Ручной ввод данных в режиме имитации. Ручной ввод данных возможен только в режиме имитации и только для тегов с функциональностью «Ручной ввод». Для ввода необходимо совершить двойное нажатие на выбранном теге. На экране возникнет окно, показанное на рисунке 26.

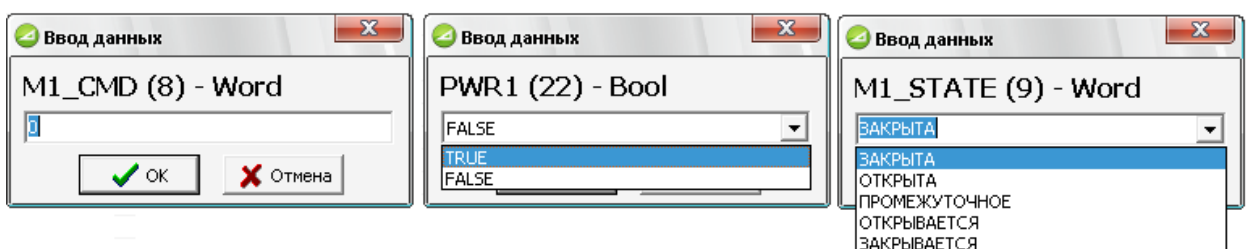


Рисунок 26

Верхняя строка окна отображает название тега, его адрес в скобках и тип данных. Окно содержит одно поле для ввода информации, принимающее в зависимости от типа данных разный вид. Для ввода чисел или строк используется простое поле ввода, для ввода булевых переменных используется список из двух значений TRUE (правда) и FALSE (ложь), для ввода констант показывается список констант входящих в группу констант используемую тегом, при этом допускается ввод чисел, для которых константа не указана.

4. **Трассировка.** На рисунке 27 показано окно трассировки. Трассировка необходима для отображения потока получаемых и отправляемых данных, для контроля корректности отправляемых данных и может быть полезна для отладки Modbus клиентов.

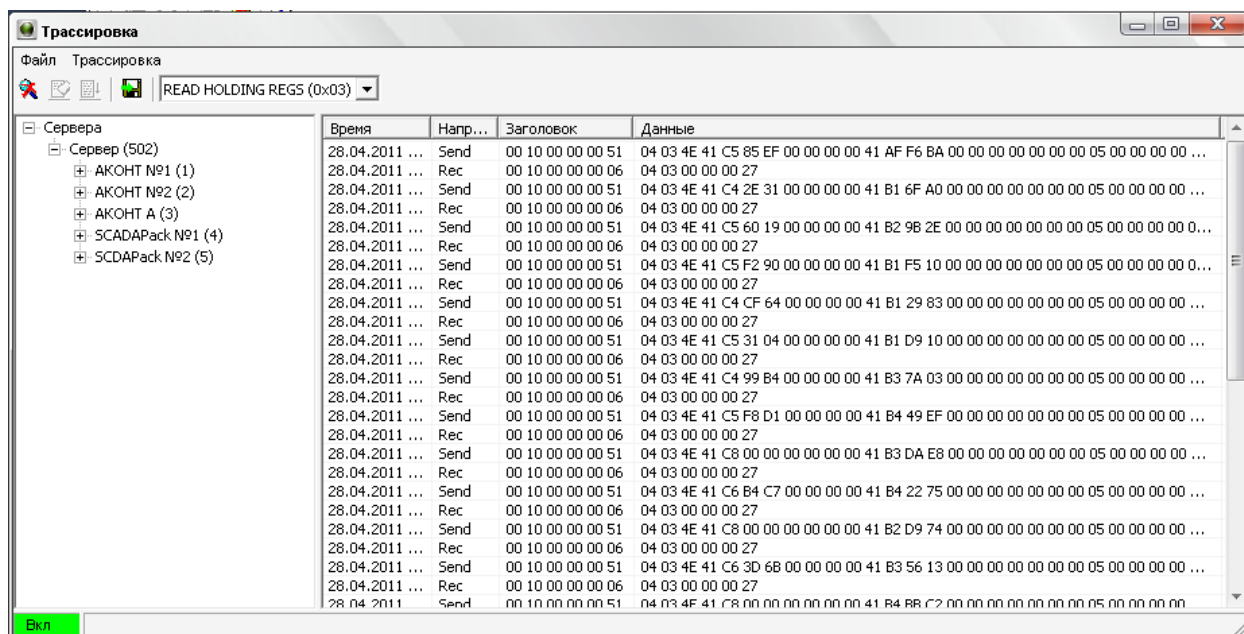


Рисунок 27

Окно трассировки содержит два пункта меню: Файл и Трассировка.

- Файл. Меню содержит один пункт «Путь к файлу трассировки...». При выборе этого пункта появится диалоговое окно позволяющее указать, куда будет сохраняться весь поток проходящей через программу информации.
- Трассировка. Содержит шесть пунктов:
 - Включить. Разрешает программе перехватывать и отображать в окне данные, проходящие через программу. Трассировка запускается.
 - Выключить. Запрещает программе перехватывать данные. Трассировка останавливается.
 - Глубина. Настройка определяет максимально разрешенное количество строк для отображения в окне трассировки. При глубине равной нулю ограничения нет.
 - Очистить. Очищает окно трассировки.
 - Включить сохранения в файл. Помимо отображения потоков информации в окне трассировки, все данные сохраняются в текстовый файл, указанный в «Файл – Путь к файлу трассировки...».
 - Выключить сохранение в файл. Отключается механизм сохранения потока в файл.

Проходящий через программу трафик может отфильтровываться по двум параметрам:

1. По Modbus номеру устройства – для фильтрации необходимо в дереве объектов выбрать интересующее устройство

2. По Modbus команде – для фильтрации необходимо в ниспадающем списке команд выбрать интересующую вас команду, фильтрация по команде работает только конкретного устройства, т.е. сначала необходимо выполнить пункт №1.

Использование скриптов в Modbus TCP/RTU эмулятор.

Как было написано выше одной из отличительных особенностей программы - наличие встроенного механизма исполнения скриптов. На текущий момент реализован механизм интерпретации скриптов написанных на языке pascal. Для работы со скриптом необходимо добавить тег и выставить ему функциональность «Скрипт». На рисунке 28 показаны настройки тега с функциональностью «Скрипт».

Сервер.АКОНТ №1.Резервная линия.FT302СН

Имя тега
FT302СН

Описание
Объем по основной линии за текущий час

Адрес Тип данных Период (мс) Флуктуация %
1232 Float 1000 0

Пространство Формат данных Кол-во регистров
READ HOLDING REGS (0x03) 1_0_3_2 2

Поведение тега
Ручной ввод Скрипт Инкремент Случайное число Sin Пила

```
//Скрипт на исполнение
begin
  if rezOn = 0 then
    begin
      //Накопление только при включенном учете
      self := self + FT302SU/3600;
    end;
  end;
end;
```

1:1

Редактировать...

Принять

Рисунок 28

На рисунке 28 показаны настройки тега счетчика рассчитывающего потребленный объем газа за текущий час.

Для редактирования скрипта используется отдельное окно, вызываемое кнопкой «Редактировать...». Окно редактирования скрипта показано на рисунке 29. Окно редактирования разделено на 3 части:

- Поле доступных функций – функции, которые можно использовать в скриптах. Это функции работы со временем, со строками, преобразования данных, математические и т.д.
- Поле редактирования – поле, в котором непосредственно вводится скрипт.
- Поле переменных – в этом поле отображается список доступных скрипту внешних переменных. В текущей реализации программы под внешней переменной понимаются теги устройства, которому принадлежит тег со скриптом.

Меню окна редактирования скрипта состоит из трех пунктов:

- Файл. Меню файл содержит четыре пункта:

- Новый. Скрипт содержащийся в окне будет удален и заменен заглушкой: `begin end;`.
- Открыть. В окно редактирования можно загрузить скрипт из файла.
- Сохранить. Скрипт, содержащийся в окне, можно сохранить в файл.
- Компилировать. После редактирования скрипта, его для проверки необходимо скомпилировать. Если в процессе компиляции произошла ошибка будет выведено окно с указанием ошибки и курсор редактирования переместится в поле редактирования на место с ошибкой.
- Выход. Закрывается окно редактирования скрипта.
- Скрипт. Меню скрипт содержит три пункта.
 - На пуск. Скрипт выполняемый единожды, в момент перехода в режим имитации.
 - Во время исполнения. Скрипт выполняемый в режиме имитации с периодичность указанной в поле «Период» настроек тега.
 - На останов. Скрипт выполняется единожды в момент перехода из режима имитации в режим редактирования.
- Переменные. Активен только один подпункт «Теги...». Выбрав этот подпункт, отображается окно, рисунок 30, с помощью которого скрипту можно разрешить доступ к другим тегам устройства. В панели тегов доступных скрипту можно используемому тегу назначить любое имя, в дальнейшем скрипт будет обращаться к этому тегу по заданному имени. Для задания имени необходимо левой кнопки мыши два раза нажать на строку в колонке «Переменная».

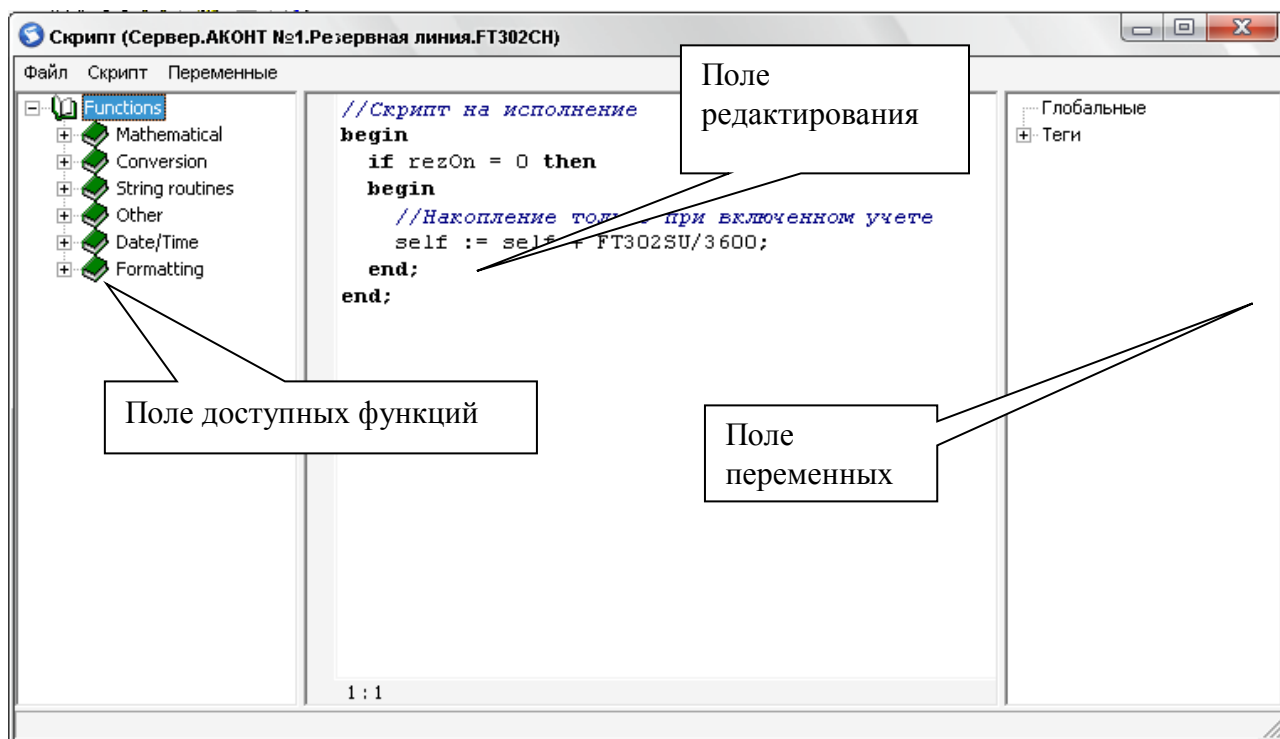


Рисунок 29.

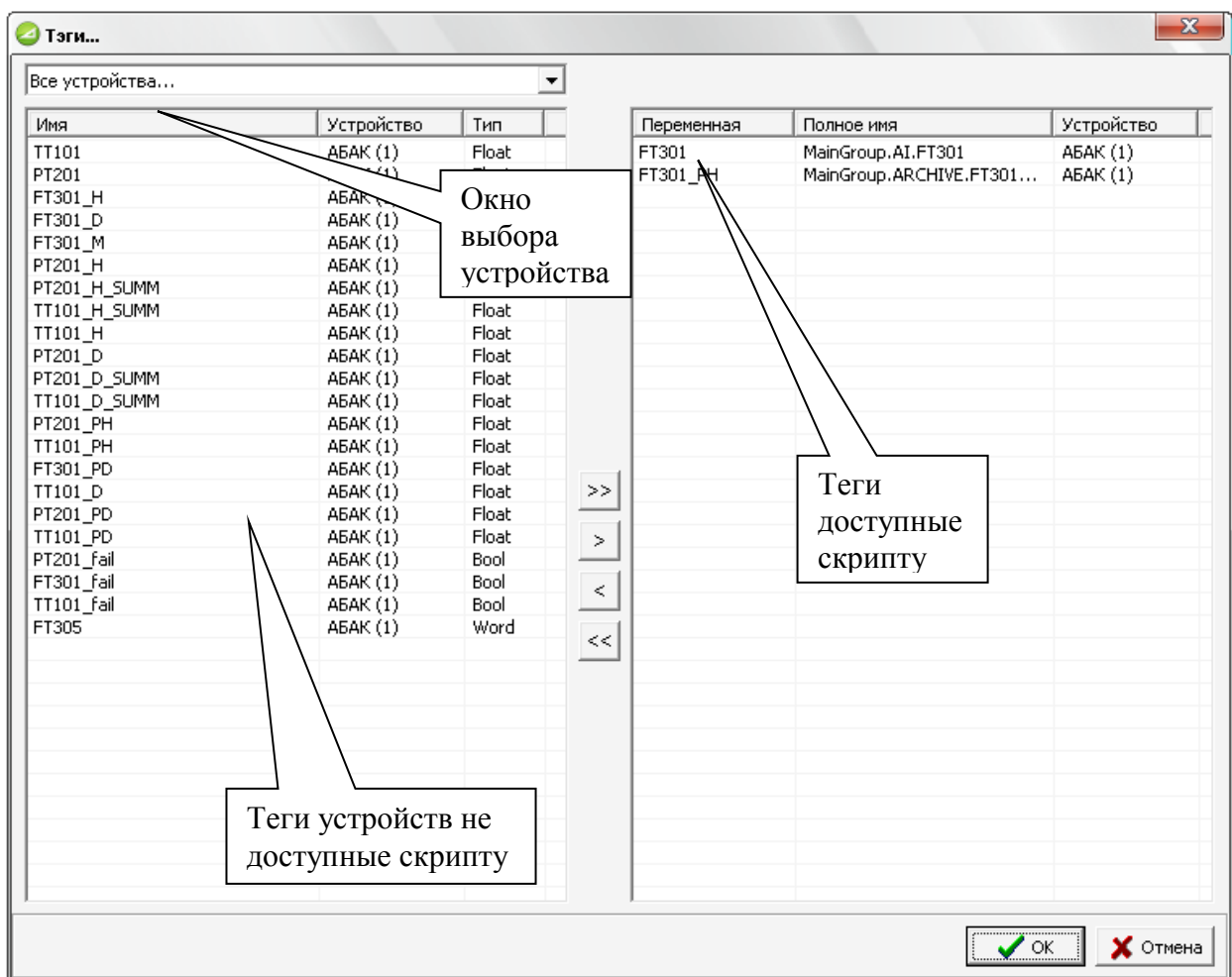


Рисунок 30

Как видно из рисунков, доступ к значениям других тегов осуществляется как к простым переменным. Доступ к значению тега, которому принадлежит скрипт, осуществляется с помощью зарезервированного слова «self». При добавлении тега с именем "self" он будет игнорироваться скриптом.

Приложение №1

Описание функций доступных для использования в скриптах.

Математические функции:

Функция	Описание	Пример
function Abs(e: Extended): Extended	возвращает абсолютное значение	var R: single; begin R := Abs(-1); end; Результат: R = 1
function ArcTan(X: Extended): Extended	возвращает арктангенс числа в радианах. π радиан = 180°	
function Cos(e: Extended): Extended	возвращает косинус числа в радианах. π радиан = 180°	
function Exp(X: Extended): Extended	возвращает экспоненту числа.	
function Frac(X: Extended): Extended	возвращает дробную часть числа	var R: single; begin R := Frac(1,485); end; Результат: R = 0.485
function Int(e: Extended): Extended	возвращает целую часть числа	var I: Single; J: Single; begin I := Int(1,485); J := Int(1,985); end; Результат: I = 1, J = 1
function Ln(X: Extended): Extended	возвращает натуральный логарифм числа	
function PI: Extended	возвращает $\pi = 3.14$	
function Round(e: Extended): Integer	возвращает значение округленное к ближайшему целому числу	var I: Integer; J: Integer; begin

		I := Round(1,485); J := Round(1,685) end; Результат: I = 1, J=2
function Sin(e: Extended): Extended	возвращает синус числа в радианах. π радиан = 180°	
function Sqrt(e: Extended): Extended	возвращает корень квадратный числа	
function Tan(X: Extended): Extended	возвращает тангенус числа в радианах. π радиан = 180°	
function Trunc(e: Extended): Integer	возвращает целую часть числа	var I: Integer; J: Integer; begin I := Trunc(1,485); J := Trunc(1,985); end; Результат: I = 1, J = 1

Функции работы со строками

Функция	Описание	Пример
function Chr(i: Integer): Char	Возвращает символ в ASCII формате преобразованный из целочисленного кода	<pre>var s: string; c: char; begin c := char(65); s := 'Символ - ' + c; end;</pre> <p>Результат: Символ - А</p>
function CompareText(s, s1: String): Integer	Сравнивает две строки без учета регистра. Если s и s1 равны – возвращает 0. Если s > s1 – возвращает число больше нуля. Если s < s1 – возвращает число меньше нуля.	
function Copy(s: String; from, count: Integer): String	возвращает часть строки с указанной позиции и указанного размера	<pre>var s: string; begin s:=Copy('Hello',1,2); end;</pre> <p>Результат: s = He</p>
procedure Delete(var s: String; from, count: Integer)	удаляет часть строки с указанной позиции и указанного размера	<pre>var s: string; begin s:='Hello'; Delete(s,1,2); end;</pre> <p>Результат: s=llo</p>
procedure Insert(s: String; var s2: String; pos: Integer)	вставляет строку s в строку s2 с указанной позиции	<pre>var s: string; begin s2:='He'; Insert('llo',s2,3); end;</pre> <p>Результат: s=Hello</p>
function Length(s: Variant): Integer	возвращает длину строки в байтах	<pre>var l: Integer; begin l:=Length('Hello'); end;</pre>

		Результат: I=5
function Lowercase(s: String): String	приводит все символы строки к нижнему регистру	<pre>var s: string; begin s:=Lowercase('Hello'); end;</pre> Результат: s=hello
function NameCase(s: String): String	приводит первый символ строки в верхнему регистру, остальные к нижнему	<pre>var s: string; begin s:=Nowercase('АНДрей'); end;</pre> Результат: s=Андрей
function Ord(ch: Char): Integer	возвращает цифровой код ASCII символа	<pre>var I: Integer; begin I:=Ord('A'); end;</pre> Результат: I=65
function Pos(substr, s: String): Integer	возвращает позицию подстроки substr в строке s, в случае отсутствия подстроки возвращает 0, функция чувствительна к регистру символов.	<pre>var I: Integer; J: Integer; begin I:=Pos('ell', Hello); J:=Pos('Ell',Hello); end;</pre> Результат: I=3, J=0
function SetLength(var S: Variant; L: Integer)	задает новую длину строки, применима только к локальным переменным.	
function Trim(s: String): String	удаляет из заданной строки все пробелы в начале и конце	<pre>var s: string; begin s:=Trim(' Hello world!'); end;</pre> Результат: s=Hello world!
function Uppercase(s: String): String	приводит все символы строки к верхнему регистру	<pre>var s: string; begin s:=Uppercase('Hello'); end;</pre>

		Результат: s=HELLO
--	--	--------------------

Функции для работы с датой и временем.

Функция	Описание	Пример
function Date: TDateTime	возвращает текущую дату	
function DayOfWeek(aDate: TDateTime): Integer	возвращает день недели заданной даты в диапазоне от 1 до 7, где 1 – воскресенье, 7 - суббота	
function DaysInMonth(nYear, nMonth: Integer): Integer	возвращает количество дней в месяце.	var I: Integer; begin I:=DaysOfMonth(2010,8); end; Результат: I=31
procedure DecodeDate(Date: TDateTime; var Year, Month, Day: Word)	декодирует дату и помещает результат в заданные переменные	var Year, Month, Day: Word; begin <i>//Текущая дата 20.08.2010</i> DecodeDate(Now,Year,Month,Day) end; Результат: Year=2010,Month=8,Day=20
procedure DecodeTime(Time: TDateTime; var Hour, Min, Sec, MSec: Word)	декодирует время и помещает результат в заданные переменные	var H, M, S, mS: Word; begin <i>//Текущая время 17:19:55.234</i> DecodeTime(Now,H,M,S,mS) end; Результат: H=17, M=19, S=55, mS=234
function EncodeDate(Year, Month, Day: Word): TDateTime	преобразует заданные год, месяц и день в формат TDateTime	
function EncodeTime(Hour, Min, Sec, MSec: Word): TDateTime	преобразует заданные час, минута, секунда, миллисекунда в формат TDateTime	
function IsLeapYear(Year: Word): Boolean	возвращает True если год високосный	
function Now: TDateTime	возвращает текущие дату и время в формате TDateTime	
function Time: TDateTime	возвращает текущее время в формате TDateTime	

Функции преобразования.

Функция	Описание	Пример
function DateTimeToStr(e: Extended): String	преобразует заданную время и дату в формате TDateTime в строку	<pre>var S: string; begin S := DateTimeToString(Now); end;</pre> Результат: S=23.08.2010 10:55:58
function DateToStr(e: Extended): String	преобразует заданную дату в формате TDateTime в строку	<pre>var S: string; begin S := DateToString(Now); end;</pre> Результат: S=23.08.2010
function FloatToStr(e: Extended): String	преобразует число с плавающей запятой в строку	
function IntToStr(i: Integer): String	преобразует целое число в строку	
function StrToDate(s: String): Extended	преобразует строку с датой в формат TDateTime	<pre>var T: TDateTime; begin T := StrToDate('20.08.2010'); end;</pre>
function StrToDateTime(s: String): Extended	преобразует строку с датой и временем в формат TDateTime	<pre>var T: TDateTime; begin T := StrToDateTime('20.08.2010 10:12:32'); end;</pre>
function StrToFloat(s: String): Extended	преобразует строку в число с плавающей запятой	<pre>var S: Single; begin S := StrToFloat('1.055'); end;</pre> Результат: S=1.055
function StrToInt(s: String): Integer	преобразует строку в целое число	<pre>var I: Integer; begin I := StrToInt('2'); end;</pre> Результат: I=2

function StrToTime(s: String): Extended	преобразует строку со временем в формат TDateTime	<pre> var T: TDateTime; begin T := StrToTime('10:12:32'); end; </pre>
function TimeToStr(e: Extended): String	преобразует заданное время в формате TDateTime в строку	<pre> var S: string; begin S := TimeToString(Now); end; </pre> <p>Результат: S=10:55:58</p>

Приложение №2

Соответствие типов данных тега к типам данных языков Pascal.

№	Тип тега	Pascal	Диапазон	Описание
1	Word	Word	$0 \div 65535$	16 разрядное целое число без знака
2	Float	Single	$1.5 \times 10^{-45} \div 3.4 \times 10^{38}$	32 разрядное число с плавающей запятой
3	ShortInt	SmallInt	$-32768 \div 32767$	16 разрядное целое число со знаком
4	Integer	Integer	$-2147483648 \div 2147483647$	32 разрядное целое число со знаком
5	DWord	Cardinal	$0 \div 4294967295$	32 разрядное целое число без знака
6	Int64	Int64	$-9223372036854775808 \div 9223372036854775807$	64 разрядное число со знаком
7	Bool	Boolean	True, False	Булева переменная, принимает только два значения
8	DateTime	TDateTime		64 разрядное число с плавающей запятой. Целая часть показывает количество дней прошедших с 30.12.1899г, дробная часть числа при умножении на 100 показывает сколько процентов времени от 24 часов истекло за текущие сутки.
9	Double	Double	$5.0 \times 10^{-324} \div 1.7 \times 10^{308}$	64 разрядное число с плавающей запятой
10	String	String		Строка переменной длины. Длина строки тега задается в регистрах, но все функции в скриптах оперируют размером в байтах.