

DR. BOB DAVIDOV

Симулятор устройств ModBUS RTU-TCP/IP

Цель работы: Изучить структуру ModBUS RTU/TCP устройства на примере mod_RSsim симулятора.

Задача работы: Обеспечить взаимодействие среды МатЛАБ с симулятором устройств

Приборы и принадлежности: Персональный компьютер, симулятор modbus RTU/TCP устройства (mod_RSsim.exe), демонстрационный ModBUS OPC сервер компании ИнСАТ (MODBUS_OPC_SERVER_SETUP_DEMO32TAGS.exe), MatLAB.

ОБЩИЕ СВЕДЕНИЯ

НАЗНАЧЕНИЕ И ХАРАКТЕРИСТИКИ СИМУЛЯТОРА ПЛК MODBUS RTU, TCP/IP И ALLEN BRADLEY DF1

mod_RSsim.exe симулятор предназначен для тестирования modbus master устройств, его также можно использовать для отладки как виртуальный ПЛК (Программируемый Логический Контроллер) или группа ПЛК до 255 устройств.

Симулятор не поддерживает полную реализацией протоколов, он реализует только их общие функции.

Симулятор устройств имеет следующие характеристики.

- Симулятор реализует только общие функции протоколов MODBUS RTU, MODBUS TCP/IP, Allen-Bradley DF1. Эмулирует MODBUS MOSCAD RTU на MODBUS. Симулятор не поддерживает протокол MODBUS ASCII.
- Моделирует одновременно работу нескольких RTU устройств.
- Отображает все регистры, содержимое регистров может редактироваться;
- Обеспечивает обмен данными более чем по одному COM-порту. Для этого необходимо запустить две копии программы. Вторая программа по умолчанию будет использовать следующий свободный порт RS-232. Запуск из командной строки позволяет связать симулятор с конкретным портом.
- Реализована функция отключения станций (устройств) нажатием на кнопки станций (00 .. 254) в нижней части окна симулятора.
- Загружает из файла и сохраняет значения каждого регистра (файл хранится в каталоге симулятора), имеется автоматическое изменение значений регистров.
- Позволяет задавать длину кадра Modbus (PDU), для моделирования RTU с требуемыми размерами буфера.
- Отображает значения в различных числовых форматах.
- Задает время ответа симулятора для поддержания работы с медленным оборудованием.
- Моделируют некоторые типы ошибок соединений, которые трудно смоделировать в лабораторных условиях.
- Имеется обработчик соединений Master – Slave, который показывает кодовые последовательности запросов и ответов.

- Моделирует шум в каналах передачи;
- Проверяет CRC (контрольный код протокола RTU) всех сообщений.

УСТАНОВКА СИМУЛЯТОРОВ УСТРОЙСТВ

1. Откройте командную строку Windows выполнив последовательность Пуск > Все программы > Стандартные > Проводник.
2. Выполните следующие команды

```

C:\WINDOWS\system32\cmd.exe

C:\mopc_HELP_CFG\MOD_RSsim>REG ADD "HKEY_CURRENT_USER\Software\Adroit Technologies\Adroit\Simulators\ModbusRTUSim" /v RegisterSel /t REG_DWORD /d 3 /f
Операция успешно завершена

C:\mopc_HELP_CFG\MOD_RSsim>REG ADD "HKEY_CURRENT_USER\Software\Adroit Technologies\Adroit\Simulators\ModbusRTUSim" /v AnimationType /t REG_DWORD /d 3 /f
Операция успешно завершена

C:\mopc_HELP_CFG\MOD_RSsim>REG ADD "HKEY_CURRENT_USER\Software\Adroit Technologies\Adroit\Simulators\ModbusRTUSim" /v AnimationScript /t REG_SZ /d "C:\mopc_HELP_CFG\MOD_RSsim\SimulateSawTooth3.vbs" /f
Операция успешно завершена

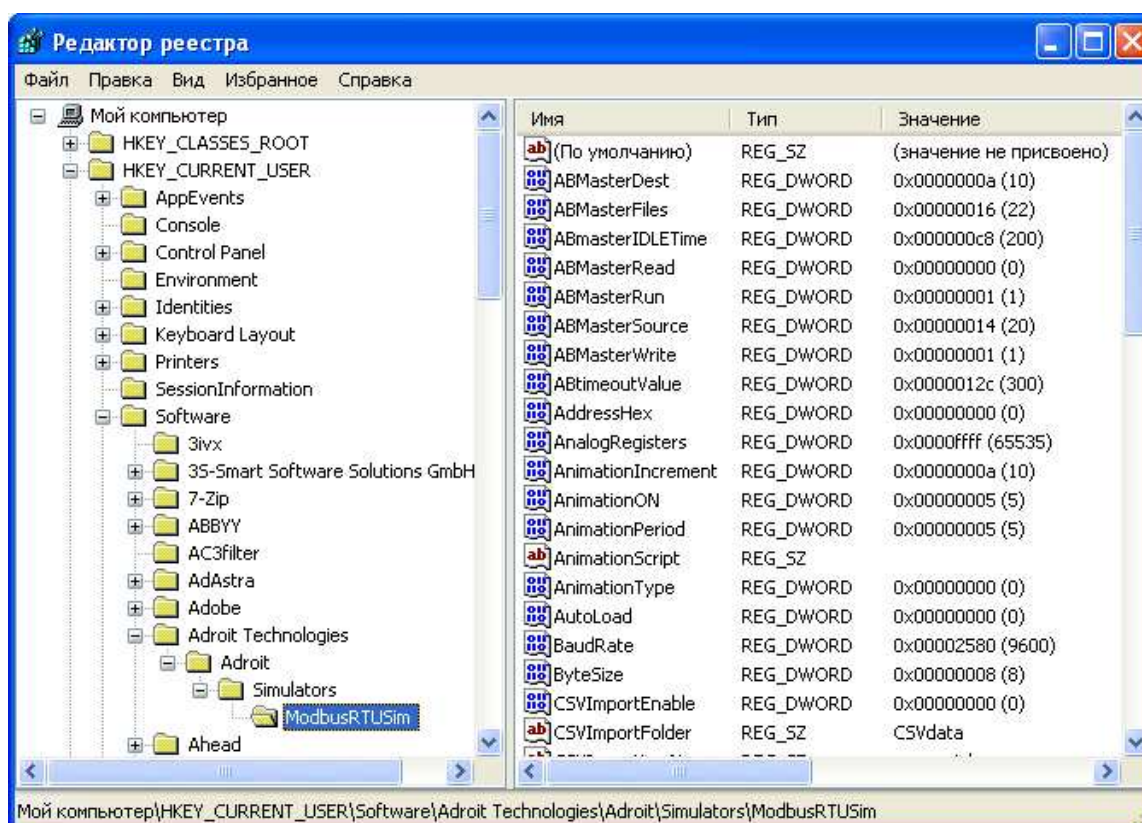
C:\mopc_HELP_CFG\MOD_RSsim>"D:\Program Files\EmbeddedIntelligence\Mod_RSsim\mod_RSsim.exe" modtcp:502
  
```

3. Запустится симулятор, настроенный на обмен по Modbus TCP/IP, с подключенным скриптом **SimulateSawTooth3.vbs**.

РЕГИСТРАЦИЯ Mod_RSsim

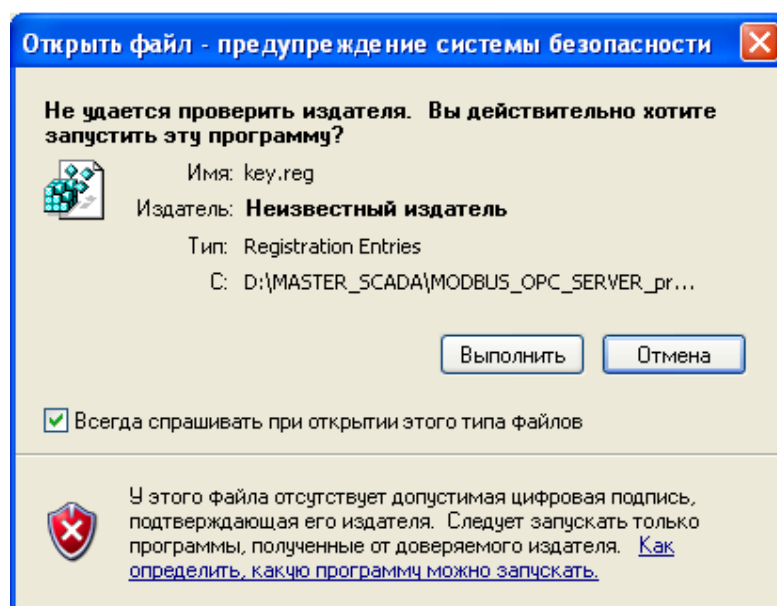
Симулятор работает и без регистрации, но ограниченное время. Для выполнения регистрации запустите файл **mod_RSsim.exe** из папки установки симулятора.

При первом запуске симулятора в реестре Windows создается следующий раздел для сохранения настроек продукта:

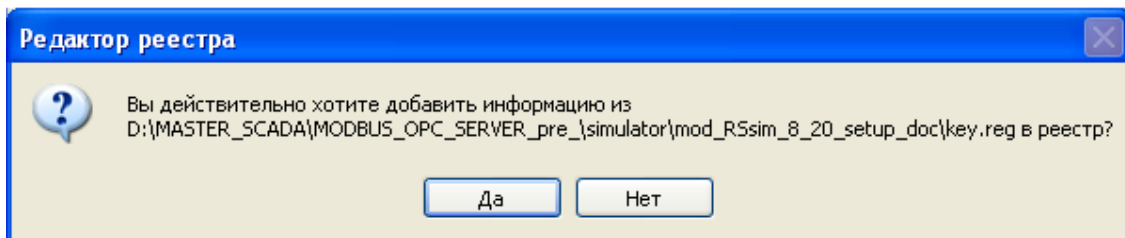


Чтобы правильно зарегистрировать симулятор, выполните следующие действия:

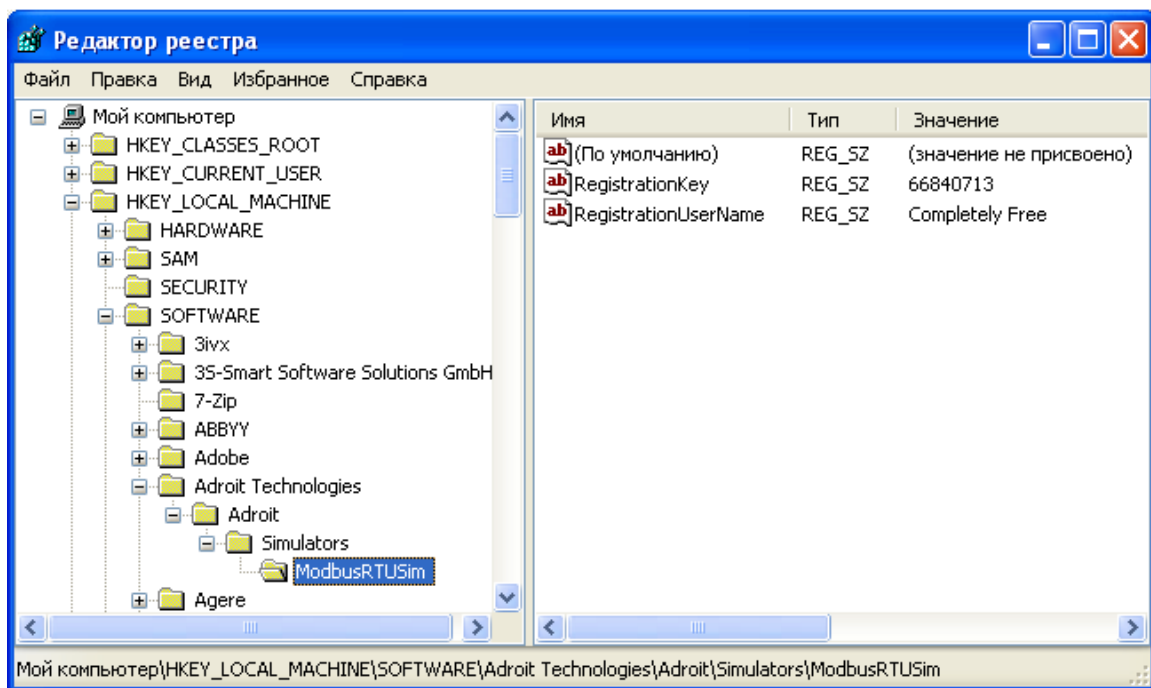
- запустите файл **key.reg** – откроется следующий диалог:



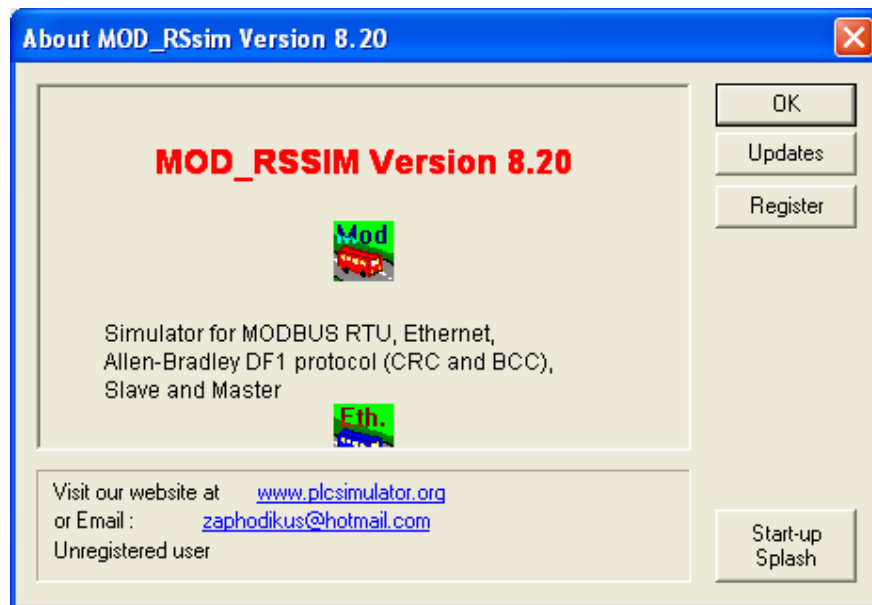
- Щелкните кнопку **Выполнить** – откроется следующий диалог:



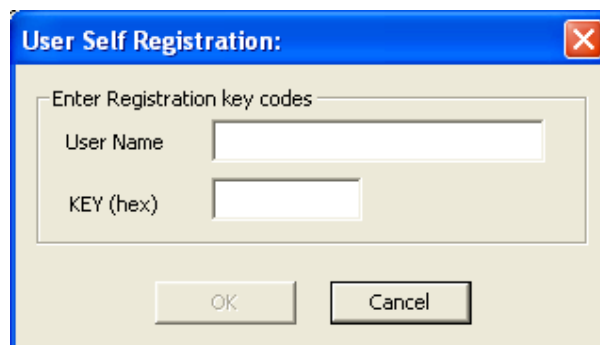
- Щелкните кнопку **Да** и информация из файла будет записана в реестр.
- Для завершения операций с файлом **key.reg** щелкните кнопку **ОК**. В результате в реестре создается следующий раздел регистрации продукта:



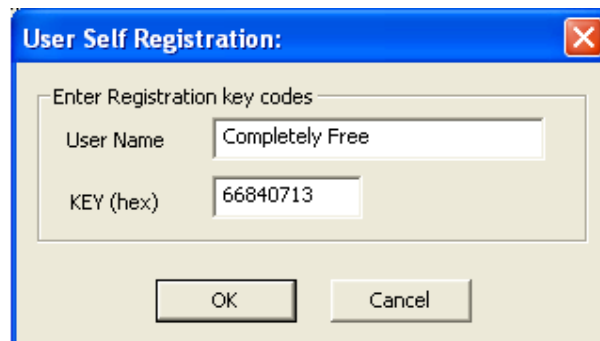
- щелкните кнопку  панели инструментов симулятора (или нажмите сочетание клавиш CTRL-B) – откроется диалог **О программе**:



- Щелкните кнопку **Register** – откроется диалог регистрации:



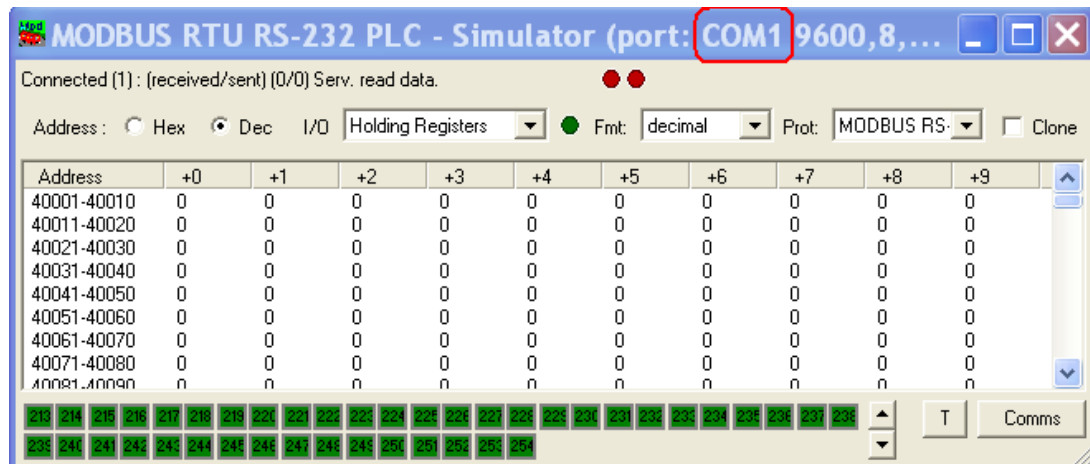
- Введите в поля **User Name** и **Key** (hex) значения, соответствующие значениям в реестре:



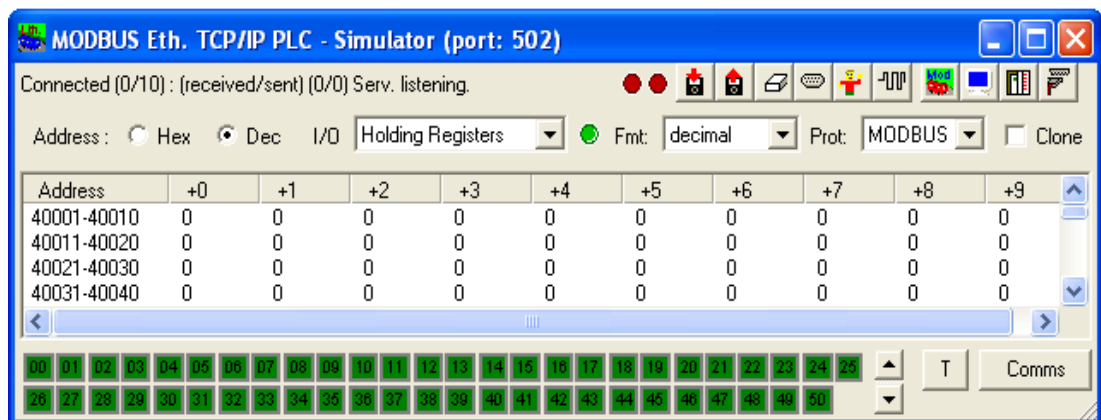
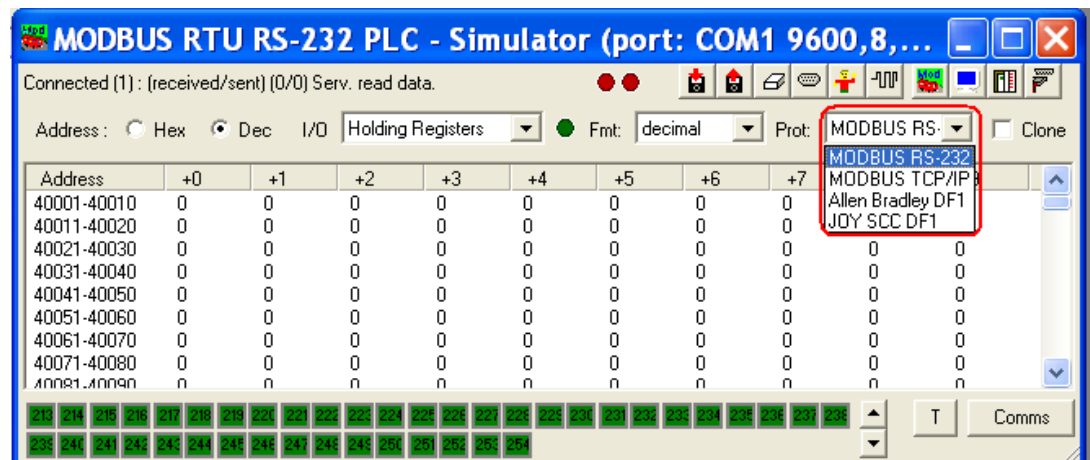
- Для завершения регистрации щелкните кнопку **OK** – при последующих запусках симулятора предупреждение о работе без ключа не будет появляться.

ОСОБЕННОСТИ СИМУЛЯТОРА УСТРОЙСТВ

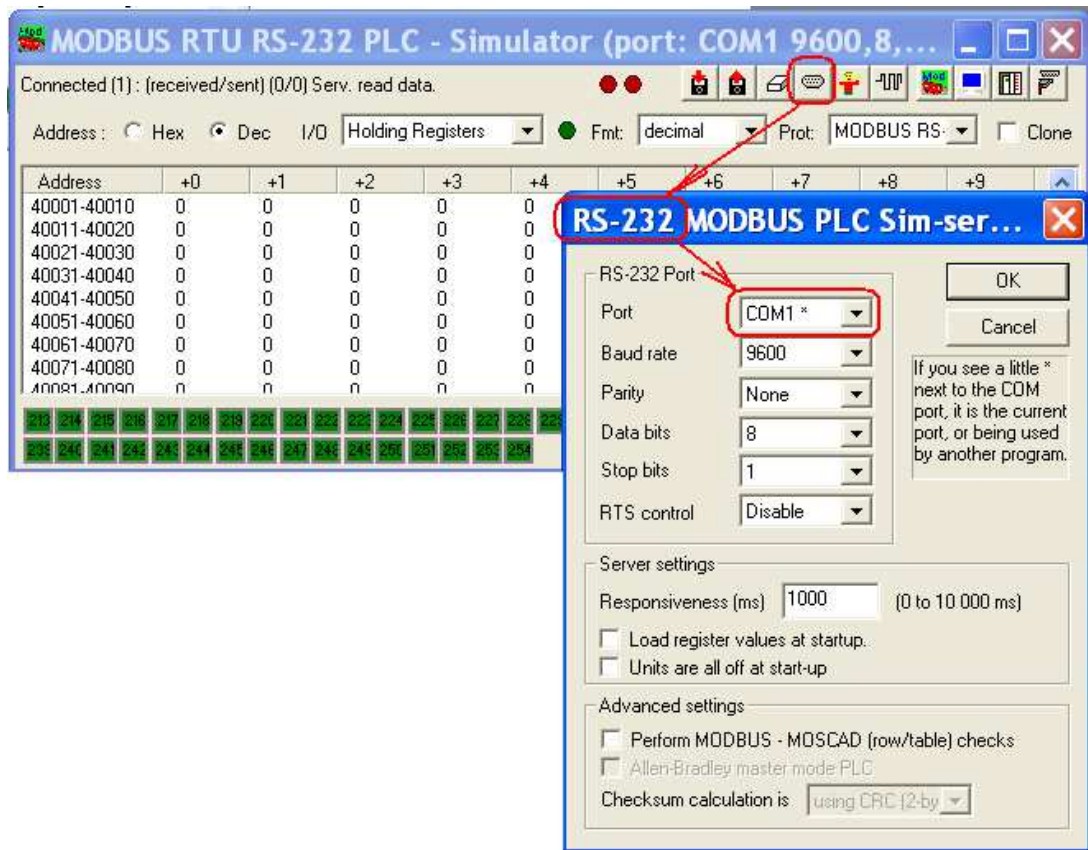
1. Запустите симулятор **mod_RSsim.exe** – он должен открыть порт (первый свободный порт для COM соединения или 502 порт для TCP/IP).



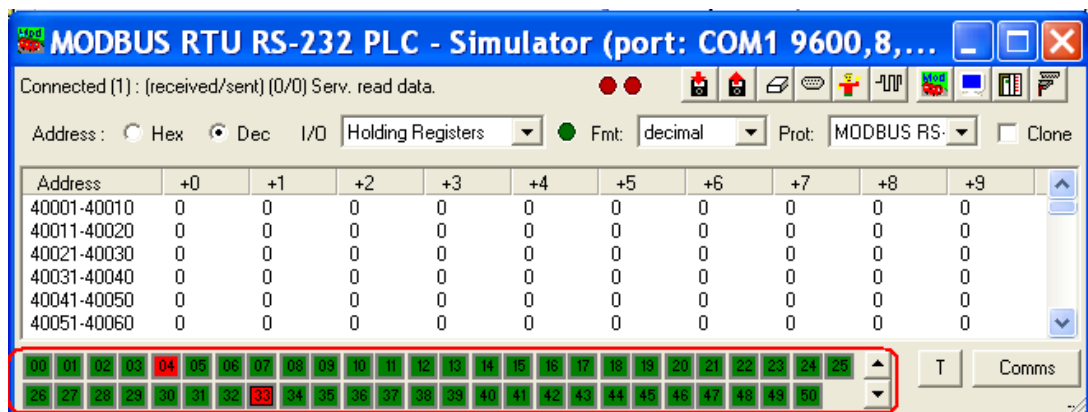
2. Выберите протокол.



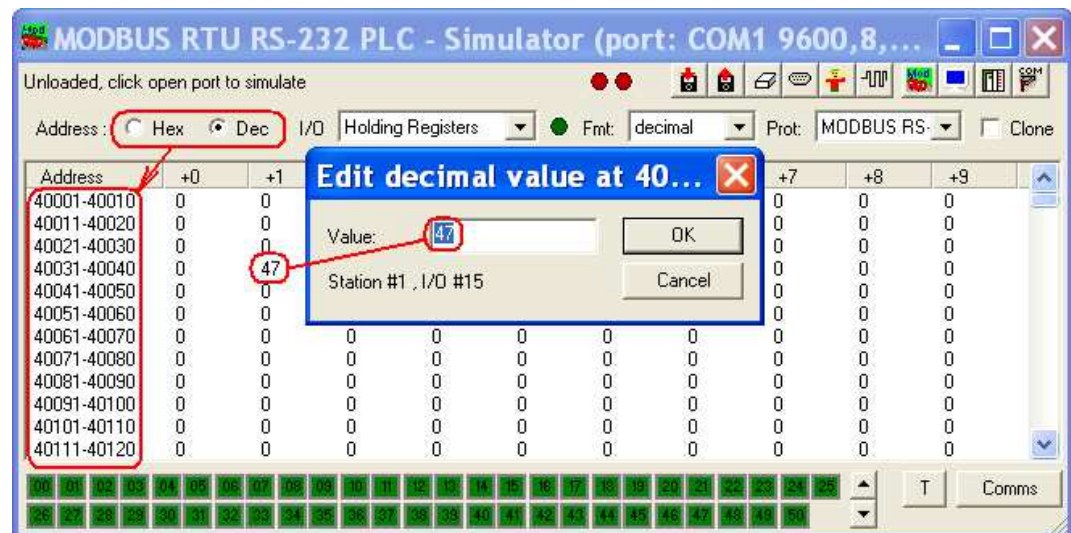
3. Щелкните кнопку Port Settings для перехода к редактированию параметров порта. Параметры сохраняются в реестре Windows.



4. Для отключения/подключения станций 0..254 используйте зеленые пронумерованные кнопки в нижней части окна. Кнопки работают и как индикаторы:
- зеленая кнопка показывает, что станция подключена,
 - красная - станция отключена,
 - когда контур кнопки меняет цвет – станция активна. Отключенные станции тоже могут проявлять активность, но симулятор в этом случае не отвечает.



- Для редактирования значений регистра дважды щелкните на нем. Двойной щелчок на цифровом вводе/выводе (0..1999) инвертирует его значение. Адрес регистра может отображаться в Hex/Dec формате.



- Кнопка Comms/Registers в правом нижнем углу симулятора переключает панели режима отладки и состояния связи.

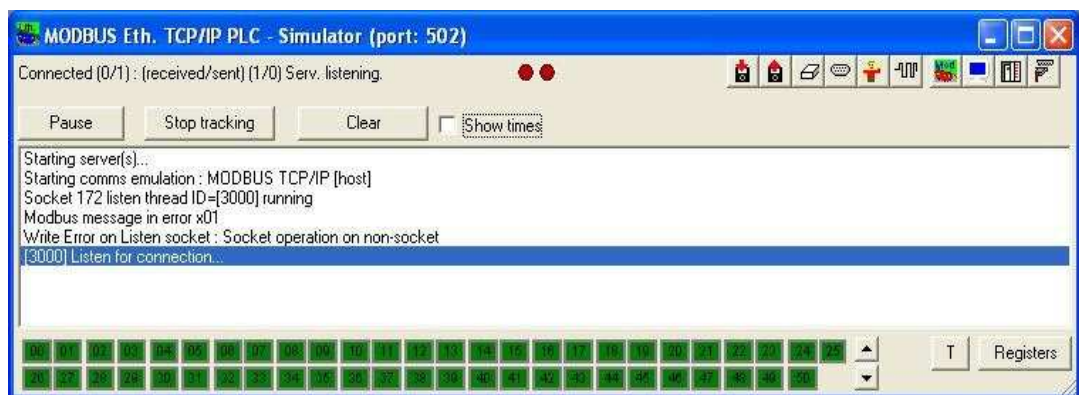
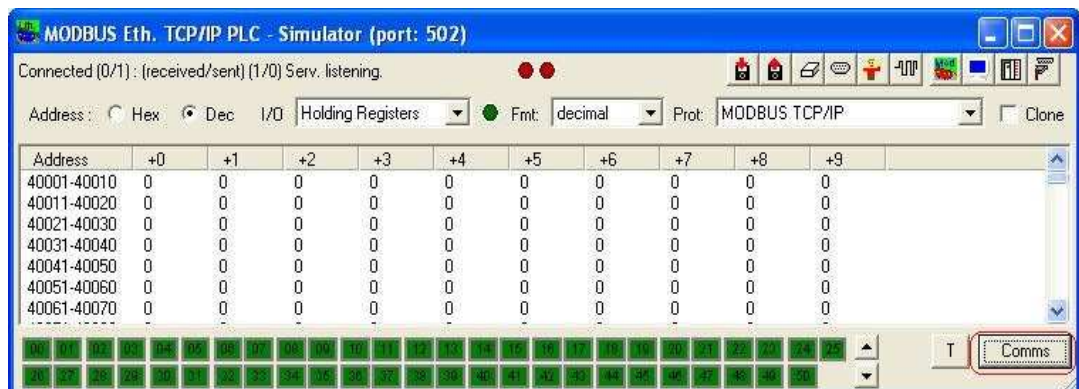


Рис. 1. В окне коммуникации имеются 4 кнопки Pause, Stop Tracking and Clear и опция отображения времени событий.

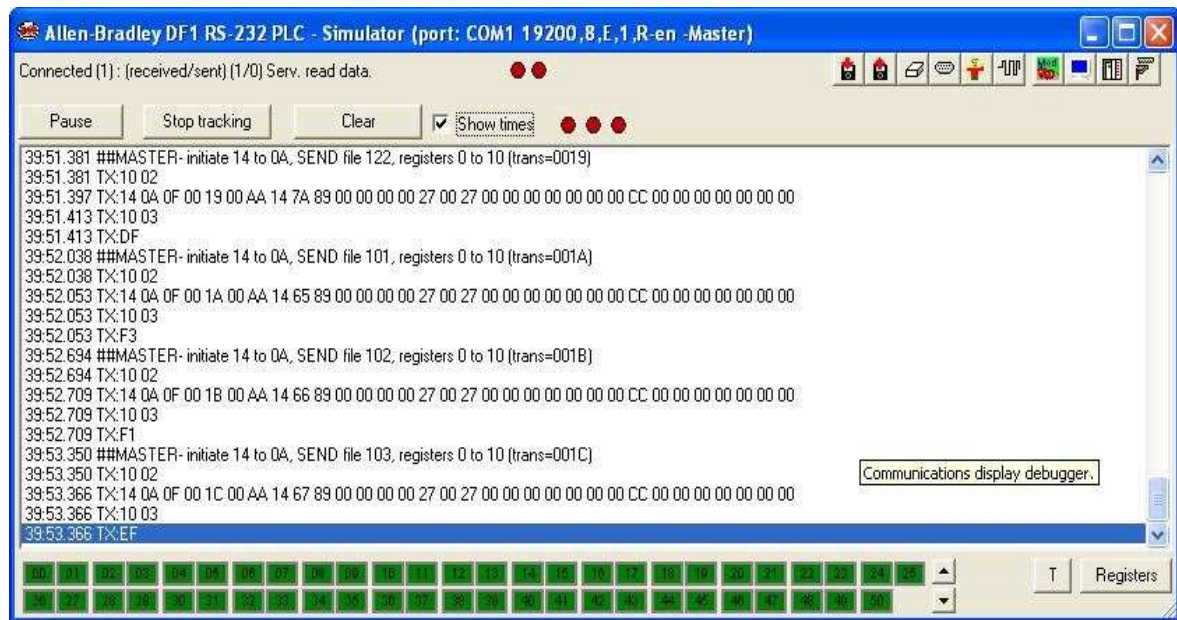
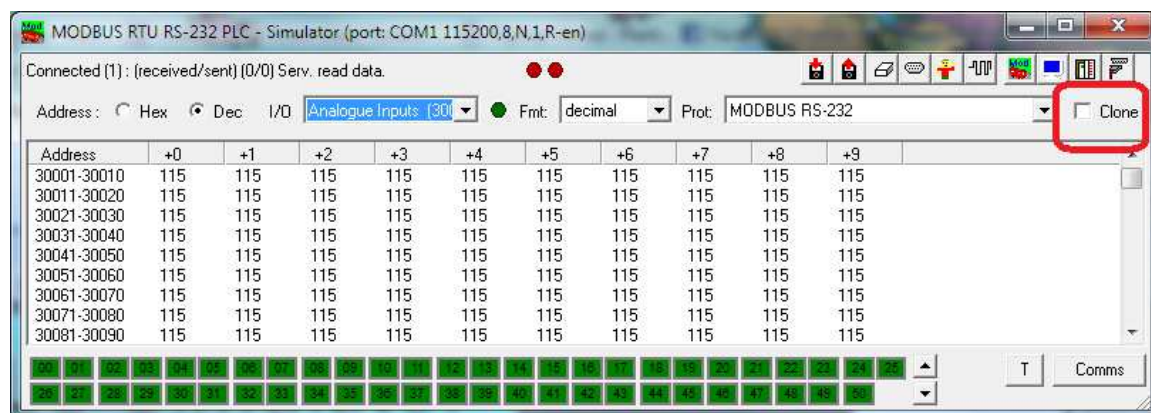


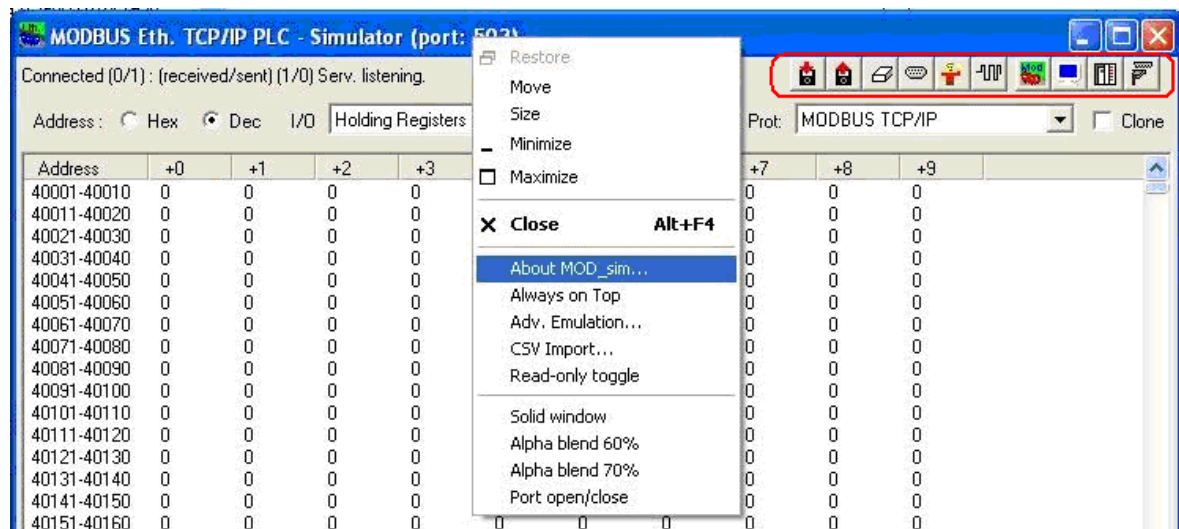
Рис. 2. Пример состояний приема передачи в режиме Allen-Bradley DF1bRS-232 PLC-Simulator (port: COM1 19200,8,E,1,R-en-Master).

7. Включение/выключение параметра **Clone** меняет порядок расположения регистров в строке.



КОМАНДЫ СИМУЛЯТОРА

Команды симулятора доступны через всплывающее меню, оно открывается щелчком ПКМ по заголовку панели симулятора и через клавиши верхней правой части панели симулятора.



Первая группа всплывающего окна относится к командам управления панелью симулятора:

- **About Mod_sim** – показывает версию симулятора и информацию разработчика
- **Always on top** – сохраняет окно симулятора на вершине других окон.
- **Advanced Emulation**  - устанавливает параметры эмуляции в соответствии с выбранным протоколом.

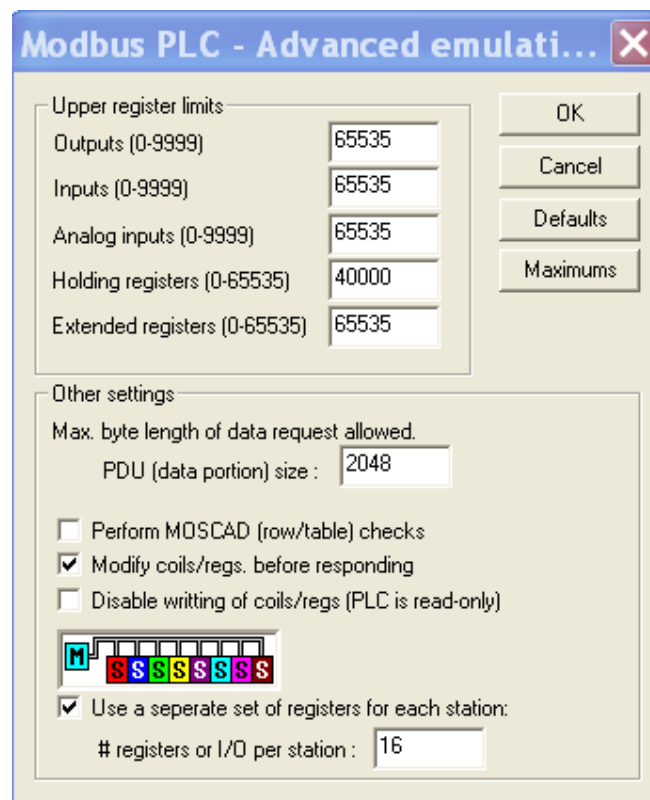


Рис. 3. Панель Advanced Emulation. **PDU (Protocol Data Unit)** определяет максимальный размер кадра сообщения. **Modify coils/regs** управляет последовательностью изменения

состояния перед ответом устройства. **Disable writting** устанавливает режим – только для чтения. **Use a separate set of registers** устанавливает заданное число регистров для каждого устройства. Например, на следующем рисунке показано разбиение зоны по 30 регистров для каждого устройства (30 рег. x 255 устройств)

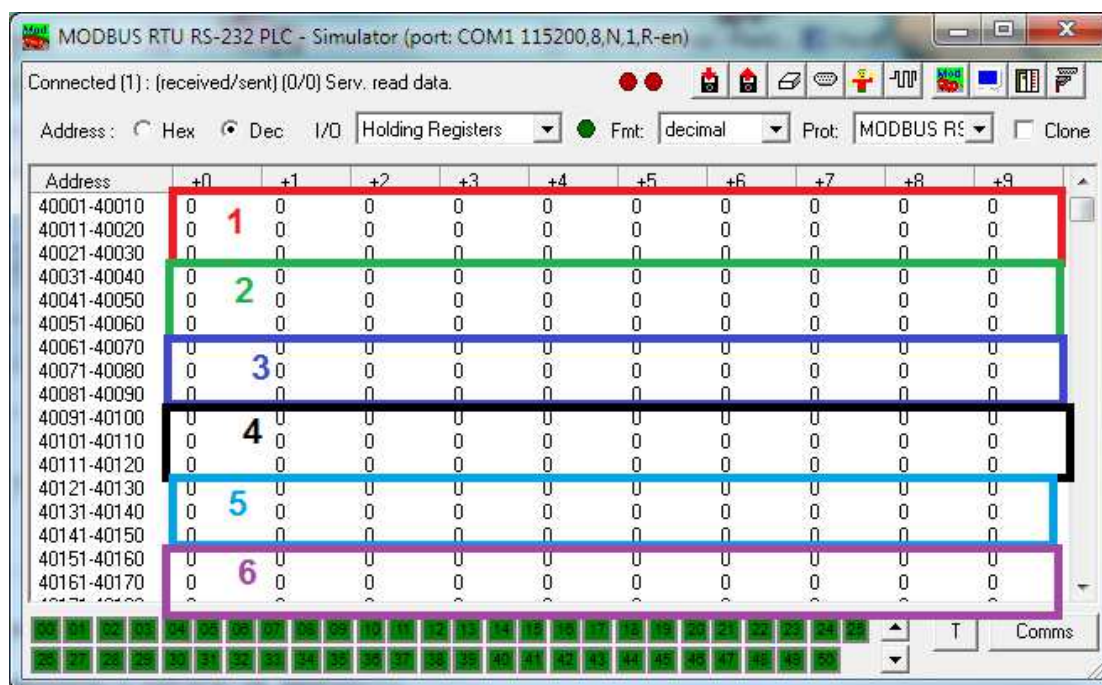


Рис. 4. Пример разбиения зоны по 30 регистров для каждого устройства (30 рег. * 255)

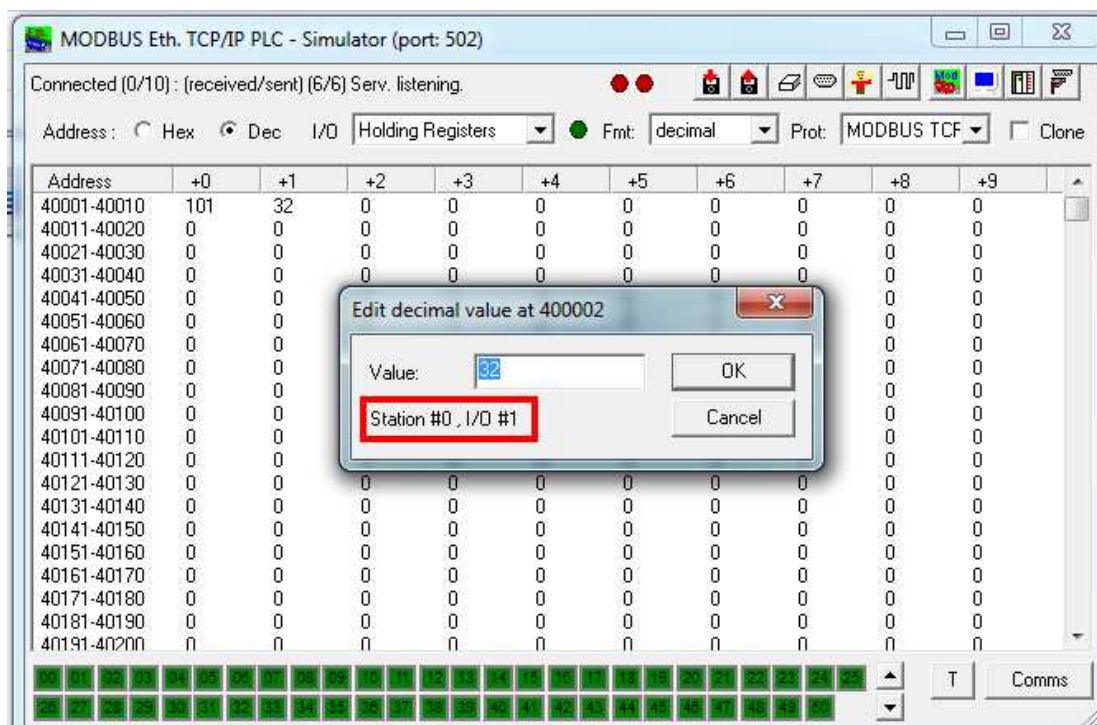
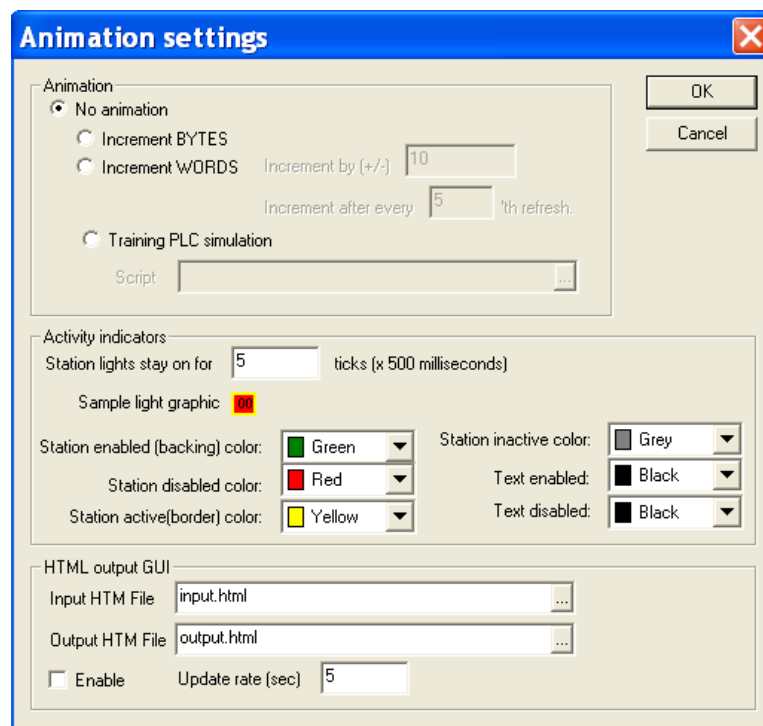


Рис. 5. Запись в Holding Register N1 станции N0, регистр работает в режиме запись/чтение (I/O).

Команды верхней правой части панели симулятора:

- **Load**  загрузка данных регистров которые были сохранены ранее.
- **Save**  сохраняет значения регистров в бинарном файле MODDATA.DAT (ABDATA.DAT в случае протокола DF1) в папке симулятора (не редактируйте эти файлы!). Эти файлы позволяют сохранить некоторые значения регистров и впоследствии восстановить их значения. Передача сообщений происходит медленнее и может приостанавливаться во время сохранения/загрузки данных, поскольку эти процессы взаимно блокируют друг друга.
- **Animation Settings**  задает цветовую палитру и настройки параметров симуляции.



- **Zero values**  обнуляет все значения, в т.ч. цифровые.
- **Responsiveness** - параметр порта задает задержку ответа после получения корректного запроса.
- **Toggle port open/close** – Устанавливает соединение с портом.

ГОРЯЧИЕ КЛАВИШИ

- <CTRL>-A = Запись значений регистров
- <CTRL>-B = О симуляторе
- <CTRL>-C = переключает панели Регистров - Коммуникации
- <CTRL>-E = Стирает содержимое всех регистров
- <CTRL>-J = вводит ошибочный символ в RS232 протокол
- <CTRL>-L = Загружает сохраненные значения регистров
- <CTRL>-M = Открывает панель ввода параметров протокола

- <CTRL>-N = Открывает панель задания шума для RS232 протокола
- <CTRL>-S = Открывает панель настройки параметров протокола
- <CTRL>-T = Изменяет “прозрачность” симулятора: прозрачный, полупрозрачный, не прозрачный.
- F1 = Help, справка

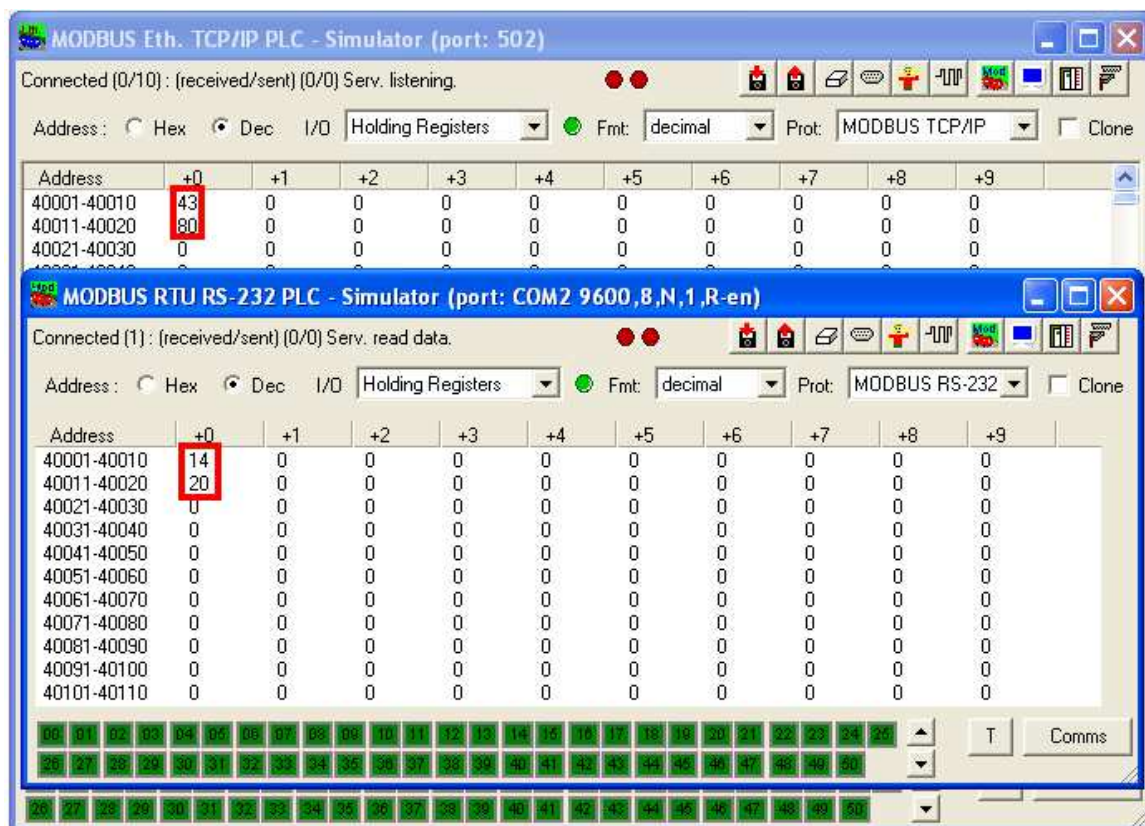


Рис. 6. Демонстрация одновременной работы двух симуляторов. Один работает по протоколу TCP/IP, другой поддерживает протокол RTU.

ВЗАИМОДЕЙСТВИЕ OPC СЕРВЕРА С СИМУЛЯТОРОМ УСТРОЙСТВ

Конфигурация `io_file.mbc` ModBUS OPC сервера компании ИнСАТ настроена на взаимодействие с симулятором устройств по протоколу TCP/IP, IP адрес: 127.0.0.1, IP порт: 502.

Конфигурация `io_file.mbc` демонстрирует использование функций библиотеки io LUA (см. раздел “помощь” OPC сервера: “5.7 Средства ввода-вывода”):

Скрипт тега `modbus1` обеспечивает архивирование данных, полученных от симулятора устройства TCP/IP, в произвольный файл;

Скрипт устройства `dTCP` реализует простейший дамп.

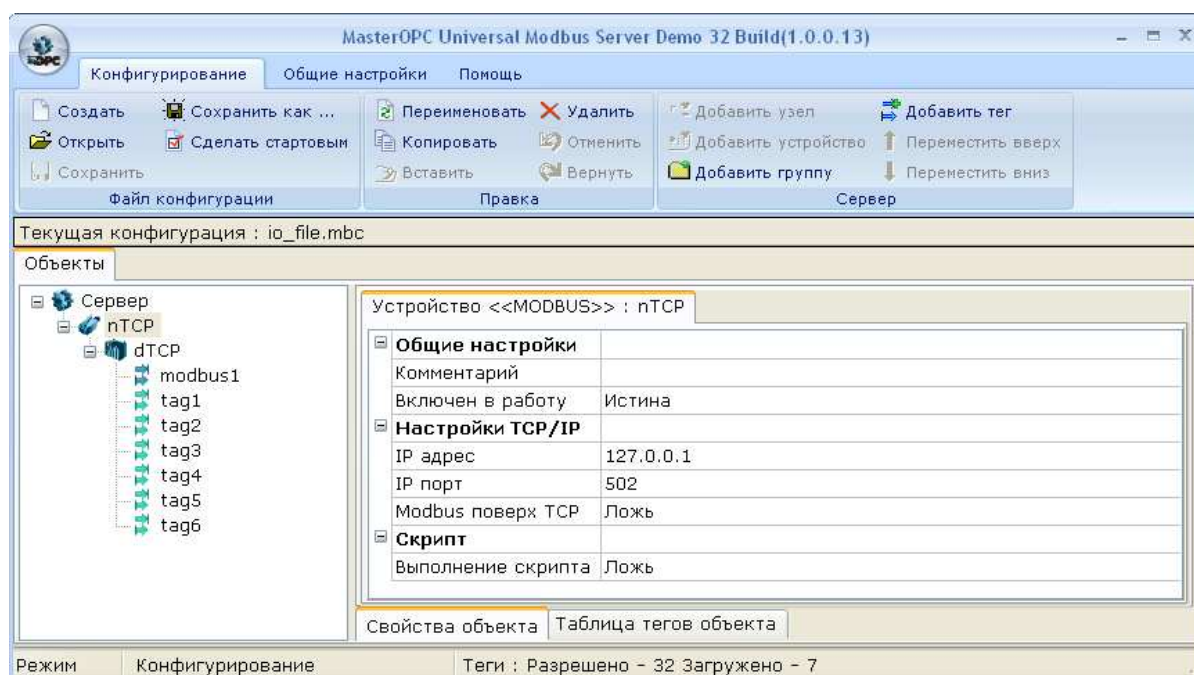


Рис. 7. Панель OPC сервера с *io_file.mbc* конфигурацией.

СОЗДАНИЕ КОМАНДНЫХ ФАЙЛОВ ЗАПУСКА СИМУЛЯТОРА УСТРОЙСТВ

Создание текстового командного файла на языке LUA для управления симулятором устройств описано в разделах “Помощь” ИнСАТ OPC сервера “Подготовка симуляторов устройств” и “Запуск симуляторов устройств”

В качестве эмулятора COM-портов используется *com0com*, дистрибутив этого эмулятора входит в состав пакета MasterOPC Universal Modbus Server. Обмен с симулятором Mod_RSSim может выполняться через виртуальные COM-порты. Перед выполнением примера необходимо установить симулятор Mod_RSSim и эмулятор *com0com* – см. разделы “Помощь” ИнСАТ OPC сервера: “Состав дистрибутива *com0com*”, “Установка *com0com* в Windows XP”, “Состав дистрибутива Mod_RSSim” и “Установка Mod_RSSim в Windows XP”.

MODBUS TCP

Для работы с симулятором в режиме TCP/IP, необходим ПК с установленным Ethernet адаптером или Microsoft loop-back адаптером.

Modbus Ethernet настройка включает:

- **Local IP** не устанавливается, отображает имя ПК или локальный IP адрес (пр. Di-fe36ebcc107)
- **Remote IP** не устанавливается, показывает имя компьютера или IP первого соединения с симулятором устройств.

- **# Server connections** – номер устройства подключенного к серверу (пр. 16).
- **Port (502)** – номер порта (пр. 502)
- **Station ID** – ID станции Modbus (не используется)
- **Responsiveness** – задержка ответа в мс (0 .. 10000). Используется для работы с медленными устройствами. Точность установки задержки – менее 20 сек.

Последовательность запросов и ответов modbus TCP начинается шестью байтами. Шестой байт содержит количество следующих байт в транзакции.

Байт 0:	Идентификатор транзакции – копируется сервером - обычно 0
байт 1:	Идентификатор транзакции – копируется сервером - обычно 0
Байт 2:	Протокол идентификатора = 0
байт 3:	Протокол идентификатора = 0
байт 4:	Длина поля (старший байт) = 0 (поскольку все сообщения меньше 256)
байт 5:	Длина поля (младший байт) = количество следующих байт
байт 6:	Идентификатор устройства (адрес устройства)
байт 7:	Код MODBUS функции
байт 8:	Необходимые данные

Так, например, операция чтения одного регистра содержащего 5 будет следующей.

запрос: 00 00 00 00 00 06 09 03 00 04 00 01,

ответ: 00 00 00 00 00 05 09 03 02 00 05

Следует отметить, что в режиме MODBUS. TCP / IP не требуется проверки "CRC-16" используемой в modbus RTU или проверки кодовой суммы "LRC" используемой в modbus ASCII поскольку механизм вычисления контрольная суммы для проверки точной доставки пакета используются внутри TCP / IP.

Последовательность действий клиента **MODBUS / TCP / IP**

- Создание TCP / IP объекта подключенного к порту 502 на требуемом сервере, используя команду connect ()
- Подготовка запроса MODBUS
- Отправка запроса MODBUS, в том числе 6-байт заголовка (prefix) MODBUS TCP / IP единой транзакцией командой send ()
- Ожидание ответа от того же TCP / IP соединения. Можно использовать фиксированную задержку командой select () для увеличения быстродействия в сравнении с использованием ожидания ответа TCP / IP.
- Чтение командой recv () первых 6 байт ответа которые содержат фактическую длину ответа
- Командой recv () читаются оставшиеся байты ответа.

ПРОТОКОЛ ALLEN BRADLEY

- Каждый из (0-250) PLC общается файлами по 255 слов.
- Этот протокол не поддерживает обращение к отдельным регистрам ввода вывода и другим. Принимаются только целые файлы.

Симулятор устройств управляется командами написанными на языке Microsoft VB:

- **GetRegisterValue (REG_TYPE as long , ADDRESS as long) As int**
возвращает содержимое modbus/AB регистра.
SetRegisterValue (REG_TYPE as long, ADDRESS as long, REGISTERVALUE as long) As none
устанавливает значение в указанном регистре.
- **DisableStation (STATIONID as int) none**
Исключает modbus slave модуль из сети устройств
- **EnableStation (STATIONID as int) None**
Подключает станцию к сети
- **DisplayAbout() None**
Показывает окно "About".
- **GetLastRunTime() As long**
Показывает время выполнения скрипта в мсек. Величина равна "-1" на первом шаге выполнения или если имеются сбои в работе скрипта.
- **StationEnabled (STATIONID as int) As long**
Возвращает состояние станции (0 – не работает, 1 -работает)
- **TransmitRegisters (STATIONSRC as int, STATIONDEST as int, REG_TYPE as long, ADDRESS as long, REGISTERS as long) As long**
Посылает значение на slave PLC (симуляция должна работать в режиме master)
Этот режим не поддерживает MODBUS.
AddDebugString (STATIONID as string) As None
Sends text to the communications debugger, the text will appear with a double hash ## in front of it.
- **TransactionBusy () As long**
Returns TRUE if this simulation device supports master-mode, and is busy with a transaction at this time,

где

REG_TYPE -тип регистра: 0 - ввод, 1 - вывод, 2 -аналоговый ввод, 3 - holding регистр. 4 дополнительный регистр. (Для AB, это файл #.)

ADDRESS ADDRESS - регистр или указатель ввода/вывода (Ноль всегда указывает на первый элемент). Например, 0 означает 40001 если тип регистра 3, а 63 является 64-м выходом coil если тип регистра 1

REGISTERVALUE - значение регистра в диапазоне (-32767 to +32767)

STATIONID - номер станции от 0 до 255; 0 - broadcast станция)

STATIONSRC - Allen-Bradley станция источника (кроме modbus протокола).

STATIONDEST – станция приемника

ПРИМЕР КОДА

Следующий код показывает как увеличить содержимое нескольких регистров а также как последовательно по одной подключить и отключить первые 12 станций. Скрипт запускается при каждой модификации анимации.

```
dim n
dim runtime
dim station
```

```
n=0
for n=0 to 240
  x = getregistervalue(3,n)
  SetRegisterValue 3, n, x+1
next
runtime = Getlastruntime
SetRegisterValue 3, 241, runtime

if (StationEnabled (station)) then
  Disablestation station
else
  enablestation station
end if

SetRegisterValue 3, 242, station

station = station + 1
if station > 12 then station = 0
```

КОМАНДЫ MODBUS ПРОТОКОЛА

Класс команд	Номер Modbus команды	Описание	Пример
0	3	Чтение нескольких регистров	Чтение регистра по ссылке 0 (40001 в Modicon 984) содержащего 1234 hex 03 00 00 00 01 => 03 02 12 34
	16	Запись нескольких регистров	Запись в регистр по ссылке 0 (40001 в Modicon 984) значения 1234 hex 10 00 00 00 01 02 12 34 => 10 00 00 00 01
1	1	Чтение катушек (coils)	Чтение катушки (coil) по ссылке 0 (00001 в Modicon 984) содержащей 1 01 00 00 00 01 => 01 01 01
	2	Чтение дискретных входов	Чтение дискретного входа по ссылке 0 (10001 в Modicon 984) содержащего 1 02 00 00 00 01 => 02 01 01
	4	Чтение входных регистров	Чтение одного входного регистра по ссылке 0 (30 001 в Modicon 984) содержащего 1234 hex 04 00 00 00 01 => 04 02 12 34
	5	Запись катушки (coil)	Запись в катушку по ссылке 0 (00001 в Modicon 984) значения 1 05 00 00 FF 00 => 05 00 00 FF 00
	6	Запись в отдельный регистр	Запись в регистр по ссылке 0 (40001 в Modicon 984) значения 1234 hex 06 00 00 12 34 => 06 00 00 12 34
	7	Чтение статуса исключений. Эта функция обычно имеет свое значение для каждого семейства устройств	Чтение статуса исключений содержащего 34 hex 07 => 07 34
2 Этот класс функций необходим для НМІ и для	15	Установить несколько катушек	Запись в 3 катушки (coils) по ссылке 0 (00001 в Modicon 984) значения 0,0,1 0F 00 00 00 03 01 04 => 0F 00 00 00 03
	20	Чтение расширенных данных Эта функция имеет возможность обрабатывать	Чтение расширенного (extended) регистра по ссылке 1:2 (групповой регистр 1 смещение 2 Modicon 984)

наблюдения
за
состоянием
устройств

- | | | |
|----|---|---|
| | несколько одновременных запросов, и может принимать 32 разрядные числа. В 584 и 984 ПЛК эта функция используется для приема значений типа 6 (расширенные групповые структуры). Эта функция подходит для обработки больших пространств регистров и данных, которые не имеют ссылок, таких как "unlocated" переменных. | содержащего 1234 hex
14 07 06 00 01 00 02 00 01 => 14 04 03 06 12 34 |
| 21 | Запись расширенных данных
Эта функция имеет возможность обрабатывать несколько одновременных запросов, и может принимать 32 разрядные числа. В 584 и 984 ПЛК она только принимает ссылки типа 6 (расширенный групповые регистры). Эта функция подходит для обработки больших пространств регистров и данных, которые не имеют ссылок, таких как "unlocated" переменных. | Запись в 1 расширенный регистр по ссылке 1:2 (группа 1 смещение 2 Modicon 984) значения 1234 hex
15 09 06 00 01 00 02 00 01 12 34 => 15 09 06 00 01 00 02 00 01 12 34 |
| 22 | Маска записи регистров | Изменить разряды 0-3 регистра по ссылке 0 (40001 в Modicon 984) на 4 hex, (AND с 000F, или с 0004)
16 00 00 00 0F 00 04 => 16 00 00 00 0F 00 04 |
| 23 | Чтение / Запись регистров
Эта функция позволяет обращаться к диапазону регистров за одну транзакцию. Это самый эффективный способ, с помощью MODBUS, выполнять регулярный обмен изображений, например, с модулем ввода / вывода. Таким образом, для повышения производительности может использовать команды 3, 16 и 23, совмещая быстрый регулярный обмен данными (ком. 23) с возможностью выполнять по требованию дополнительные обновления конкретных данных (ком. 3 и 16) | Запись 1 в регистр по ссылке 3 (40 004 в Modicon 984) значения 0123 hex и чтение 2-х регистров по ссылке 0 содержащих 0004 и 5678 hex
17 00 00 00 02 00 03 00 01 02 01 23 => 17 04 00 04 56 78 |
| 24 | Чтение FIFO очереди
Несколько специализированных функций, | Чтение содержимое блока FIFO, начиная со ссылки 0005 (40006 в Modicon 984), содержащего 2 слова: 1234 и 5678 |

предназначены, для передачи данных из таблиц, hex
 структурированных как FIFO (для использования с 18 00 05 => 18 00 06 00 02 12 34 56 78
 FIN и FOUT функциональными блоками на
 584/984 ПЛК). Полезно для логирования
 некоторых приложений

Эти	8	Диагностика
функции,	9	Программы (484 ПЛК)
хотя и	10	Опрос (poll)
упоминаются	11	Получение comx событий счетчиков
в	12	Получение comx событий логов
руководстве	13	Программы (584/984 ПЛК)
MODBUS	14	Опрос (584/984 ПЛК)
протокола,	17	Отчет ID устройства
не подходят	18	Программа (884/u84 ПЛК)
для	19	Сброс comx связи (884/u84 ПЛК)
обеспечения	40	Программа (ConCept)
совместимос	125	Замена прошивки
ти потому,	126	Программа (584/984 ПЛК)
что они	127	Сообщение местного адреса (MODBUS
слишком		
машинно-		
зависимы.		

КОДЫ ИСКЛЮЧЕНИЙ

Все **исключения** выделяются добавлением 0x80 в код запроса, и следующим единственным байтом с кодом причины исключения, например,

03 12 34 00 01 => 83 02

Запрос на чтение регистра с индексом 0x1234 дает ответ - исключение 2-го типа "неправильный адрес данных"

Номер исключения	Описание
01	НЕДОПУСТИМАЯ ФУНКЦИЯ Код функции, полученный в запросе не выполнен устройством. Причиной этого исключения может быть то, что код функции относится только к новым контроллерам, и не реализован в выбранном устройстве.
02	НЕДОПУСТИМЫЙ АДРЕС ДАННЫХ Адрес данных, указанный в запросе не допустим для устройства. Например, контроллер с 100 регистрами, на обращение к 4-м регистрам со смещением 96 даст правильный ответ тогда как обращение к 5-ти регистрам с тем же смещением выдаст исключение 02.
03	НЕДОПУСТИМЫЕ ДАННЫЕ Значения, содержащегося в поле данных запроса недопустимы для устройства. Это указывает на ошибки в структуре сложного запроса (например, длина неверна).
04	НЕДОПУСТИМАЯ ДЛИНА ОТВЕТА Указывает, что запрос требует ответ, размер которого превышает допустимый размер данных MODBUS. Исключение применяется только к функциям создания составного ответа, таким как 20 и 21.
05	ПОДТВЕРЖДЕНИЕ Зависит от используемых команд программирования
06	SLAVE УСТРОЙСТВО ЗАНЯТО Зависит от используемых команд программирования
07	ОТРИЦАНИЕ ПОДТВЕРЖДЕНИЯ Зависит от используемых команд программирования
08	ОШИБКА ПАРИТЕТА ПАМЯТИ Специальное применение в сочетании с функциями 20 и 21, чтобы показать, что расширенная область не прошла проверку целостности.
0A	НЕПРИЕМЛЕМЫЙ ПУТЬ ШЛЮЗА Применяется для MODBUS шлюзов. Указывает, что шлюз не смог выделить дополнительный путь Modbus для использования при обработки запроса. Обычно означает, что шлюз неправильный.
0B	ЦЕЛЕВОЕ УСТРОЙСТВО ШЛЮЗА НЕ ОТВЕЧАЕТ Применяется для MODBUS шлюзов. Указывает, что ответ не был получен от целевого устройства. Обычно означает, что устройство не присутствует в сети.

Команды МатЛАБ установки связи по протоколу modbus RTU.

Команда	Назначение
s = serial('COM1');	создание COM объекта
fopen(s); fclose(s)	подключение (отсоединение) объекта к серверу
fprintf(s,'RS232?')	запись и чтение данных
fwrite(s,[16 1 0 5 0 1 238 138])	
fread(s, 11)	
fscanf(s)	

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Задание 1. Обеспечение связи “МатЛАБ – Симулятор” по протоколу modbus RTU.

1. Соедините два компьютера нуль-модемным кабелем.
2. На одном компьютере запустите МатЛАБ. На другом – симулятор устройств mod_RSsim.exe.
3. Настройте СОМ порты компьютеров и симулятора, так чтобы их параметры (скорость передачи, количество бит данных и др.) не отличались.
4. Настройте симулятор на работу по протоколу modbus RTU через RS-232 интерфейс. Сделайте активным 16-е устройство (16).
5. В МатЛАБ считайте содержимое 5-го (+5 или +10 в “Clon” режиме – обратное расположение смещений адресов регистров) регистра (Адрес Analogue Inputs - 30006) 16-го устройства записью следующих десятичных кодов в СОМ порт [16 4 0 5 0 1 34 138],

>>fwrite(s,[16 1 0 5 0 1 238 138])
6. Примите отклик симулятора и сравните полученное значение регистра с данными, отображаемыми в окне симулятора.
7. Введите новое значение в 5-ый регистр симулятора. Считайте значение регистра из МатЛАБ.
8. Проверьте работоспособность следующих команд.

Регистр	Команда	Обращение к +5 регистру симулятора устройств
0..9999	1	Чтение Coil Outputs, [16 1 0 5 0 1 238 138]
10000..19999	2	Чтение дискретных входов (Digital Inputs) [16 2 0 5 0 1 170 138]
30000..39999	4	Чтение входных регистров (Analogue Inputs) [16 3 0 5 0 1 151 74]
40000..49999	3	Чтение (Holding Registers) [16 3 0 5 0 1 151 74]
	16	Запись (Holding Registers) [16 16 0 5 0 1 2 0 9 166 83]
60000..69999	20	Чтение (Extended Registers, General Reference) [16 20 0 5 0 1 227 73]
	21	Запись (Extended Registers, General Reference) [16 21 0 5 0 1 222 137]

Примечание: Последние два байта последовательности это коды контрольной суммы рассчитанной по следующему алгоритму:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% crc_calculator.m    v1.0a
% Matlab v7.0 (R14) SP 1
% Bob Davidov
% 25 February 2012
%
% CRC algorithm
% calculates Check sum of Modbus RTU sequence%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function output_hex_string = crc_calculator (Input_hex);
%Input_hex = F70302640008'; % <= 2 * 16 Char
F          = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
xor_constant = [1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1];
for i = 1 : length (Input_hex) / 2;
    A = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
    A_hex = Input_hex ((i-1)*2+1:i*2); % Two HEX bytes
    A_bin = dec2bin (hex2dec (A_hex));
    length_A_bin = length (A_bin);
    for j = 0 : length_A_bin - 1
        A (16 - j) = str2num(A_bin (length_A_bin - j));
    end
    F = xor (F,A);
    for ii = 1 : 8
        if F(16) ==1
            if xor_constant (1) == 0
                F_shift (1) = 0;
            else
                F_shift (1) = 1;
            end
            for j = 2 : 16;
                if xor_constant (j) == F (j-1);
                    F_shift (j) = 0;
                else
                    F_shift (j) = 1;
                end
            end
        else
            F_shift = circshift(F',1)';
        end
    end
    F = F_shift;
end
end
h = num2str(F);
h = h(1:3:length(h));
output_hex_string = num2str([dec2hex(bin2dec(h(9:12))) dec2hex(bin2dec(h(13:16)))
dec2hex(bin2dec(h(1:4))) dec2hex(bin2dec(h(5:8)))]);

% End of crc_calculator.m

```

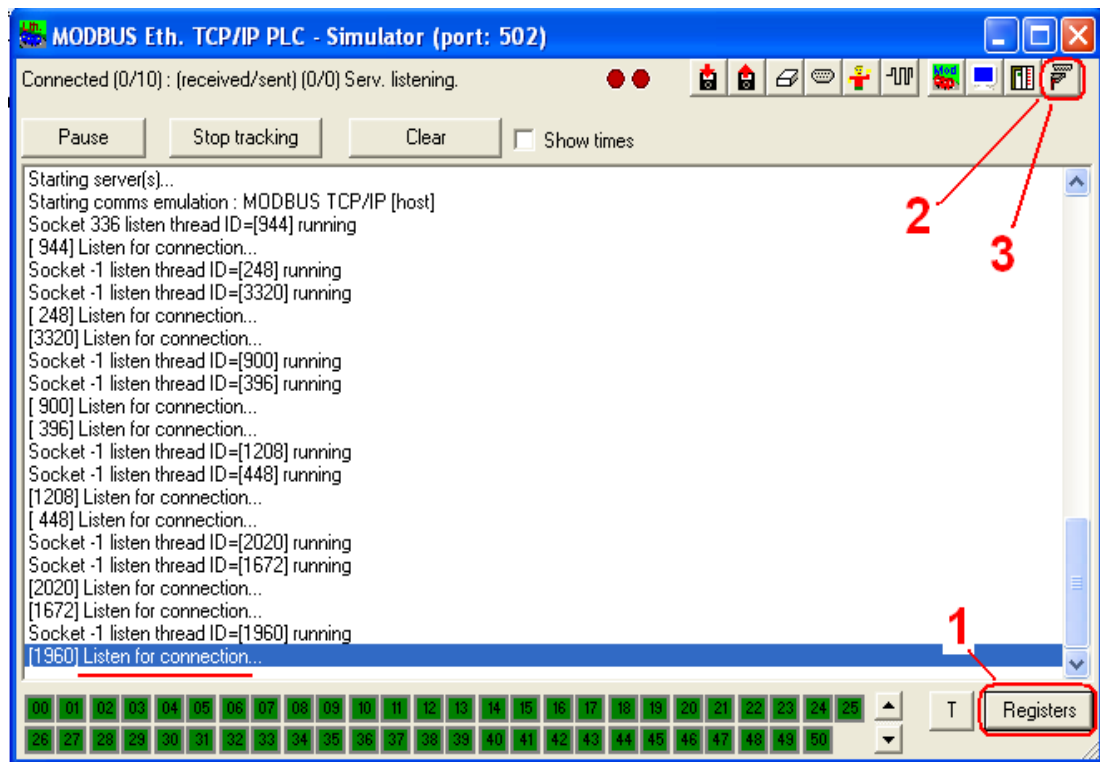
Задание 2. Запись данных пакетами.

1. Изменяя содержимое регистров проверьте работоспособность следующих команд обращения к группам регистров со смещением +5, +6 и + 7.

Регистр	Команда	Обращение к +5..+7 регистрам симулятора устройств
0..9999	1	Чтение Coil Outputs, [16 1 0 5 0 3 111 75]
10000..19999	2	Чтение дискретных входов (Digital Inputs) [16 2 0 5 0 3 43 75]
30000..39999	4	Чтение входных регистров (Analogue Inputs) [16 4 0 5 0 3 163 75]
40000..49999	3	Чтение (Holding Registers) [16 3 0 5 0 3 22 139], ответ Тх, например: [16 3 6 0 12 0 3 0 14 128 224] Рг.40006 = 12, Рг.40007 = 3, Рг.40008 = 14
	16	Запись (Holding Registers) [16 16 0 5 0 3 6 0 4 0 6 0 1 26 128], ответ: [16 16 0 5 0 3 147 72]
60000..69999	20	Чтение (Extended Registers, General Reference) [16 20 0 5 0 3 98 136], ответ Тх, например: [16 20 6 0 3 0 321 89 37 175] Рг.60006 = 3, Рг.60007 = 32, Рг.60008 = 345
	21	Запись (Extended Registers, General Reference) [16 21 0 5 0 3 6 0 4 0 6 0 1 22 140 8], ответ: [16 21 0 5 0 3 95 72]

Задание 3. Обеспечение связи “МатЛАБ – Симулятор” по протоколу TCP/IP.

1. Загрузите симулятор устройств: mod_RSsim.exe.
2. Загрузите МатЛАБ
3. Активируйте соединение с портом “502” симулятора устройств.
>>t = tcpip('localhost',502); или
>>t = tcpip('127.0.0.1',502);
4. Подготовьте команду чтения (Holding Registers)
>>req = [0 0 0 0 6 16 3 0 5 0 1],
где первые 6 байт – префикс TCP запроса; 6 – длина остатка сообщения [16 3 0 5 01] в байтах, в котором 16 – адрес устройства, 3 – код команды чтения Holding регистра; 0 5 – адрес регистра, 0 1 – количество считываемых регистров; кодовая сумма протоколов modbus RTU или ASCII в последовательность не входит поскольку TCP протокол сам контролирует доставку сообщений.
5. Переведите симулятор в режим отображения состояния связи (клавиша 1). Затем отключите и подключите канал клавишей 2 (3), так, чтобы симулятор находился в состоянии прослушивания канала: “Listen for connection”



Это необходимо сделать для того, чтобы убедиться, что устройство находится в режиме ожидания сообщений. После относительно непродолжительного времени устройство освобождает порт [216] Closing socket, idle for too long.

6. Соедините TCP объект с сервером

```
>> fopen(t)
```

Если устройство отключено от порта появляется сообщение:

```

??? Error using ==> icinterface.fopen at 82
Connection refused: connect

```

МатЛАБ не выводит сообщений если устройство подключено к 502 порту:

```

[ 576] Listen for connection...
[336][1464] Connection accepted.

```

7. Передайте запрос на чтение симулятору устройств

```
>> fwrite(t, req);
```

8. Проверьте получение ответа в симуляторе устройств и в МатЛАБ

```

[3680] Listen for connection...
[336][3316] Connection accepted.
TX:00 00 00 00 00 06 10 03 00 00 00 01
Read Register from 0 for 1 .
RX:00 00 00 00 00 05 10 03 02 00 00

```

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

```
>> t
```

TCP/IP Object : TCP/IP-localhost

Communication Settings

RemotePort: 502

RemoteHost: localhost

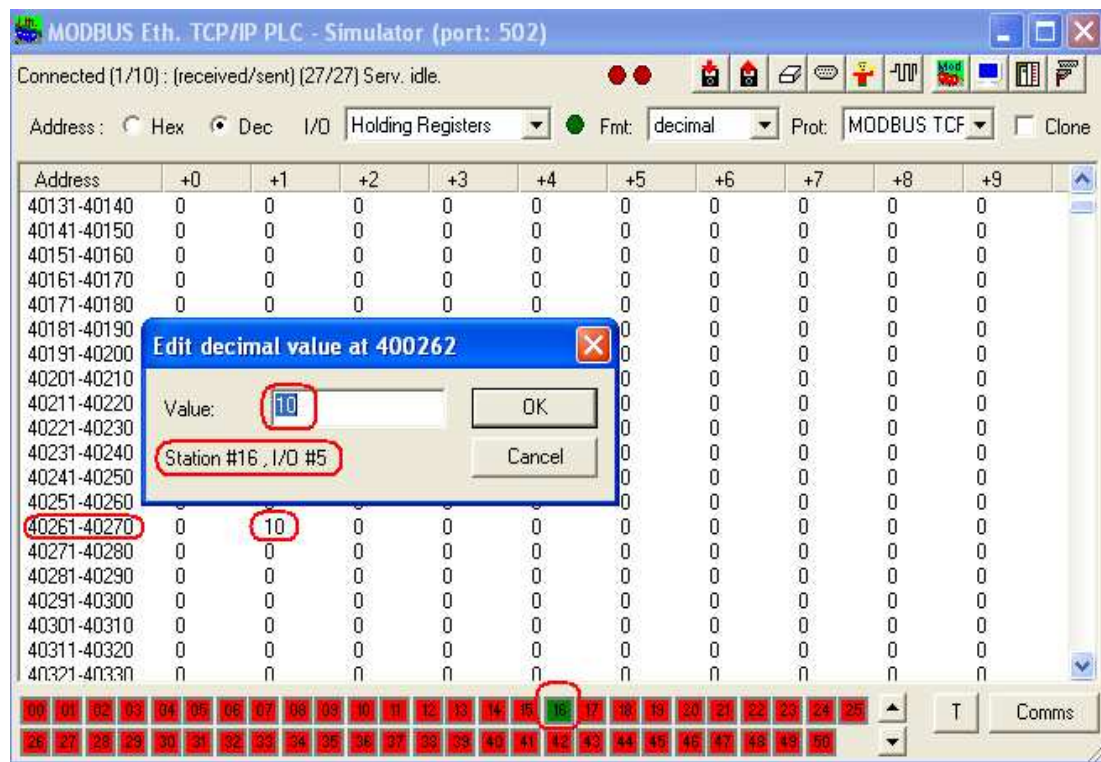
Terminator: 'LF'
 Communication State
 Status: open
 RecordStatus: off
 Read/Write State
 TransferStatus: idle
 BytesAvailable: 11
 ValuesReceived: 0
 ValuesSent: 12

9. Считайте полученные данные в Workspace МатЛАБ.

```
>> A = fread(t, 11);
```

```
0 0 0 0 0 5 16 3 2 0 0
```

10. Введите значение в #5 holding регистр 16-го устройства



11. Считайте значение регистра в Workspace МатЛАБ.

12. Запишите в симулятор устройств значение используя протокол Modbus TCP.

13. Убедитесь, что запись прошла успешно.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какое назначение симулятора устройств mod_RSsim?
2. Можно ли обеспечить взаимодействие Master программы и симулятора устройств по протоколу modbus RTU на одном компьютере?

3. Можно ли обеспечить работу нескольких симуляторов на одном компьютере, что для этого необходимо?
4. Чем отличаются протоколы Modbus RTU, ASCII, и TCP?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. MODBUS RTU, TCP/IP and Allen Bradley DF1 PLC Simulator (mod_RSsim.exe, симулятор доступен для свободной загрузки) <http://www.plcsimulator.org/>
2. Демо-версия OPC сервера компании InSAT <http://www.insat.ru/products/?category=792>
3. Modbus protocol. <http://www.rtaautomation.com/modbustcp/>