

Протокол передачи данных Modbus



Содержание:

Часть 1

1. Введение
2. Общая информация по Modbus
3. Пошаговая реализация Modbus RTU Slave (сервер)
4. Работа со сторонним ПО "Modbus Poll" (Master)

Часть 2

5. Пошаговая реализация Modbus RTU Master (клиент)
6. Работа со сторонним ПО "Эмулятор Modbus"
7. Дополнительная информация

ЧАСТЬ 1

1. Введение

Пошаговая инструкция по протоколу передачи данных Modbus подготовлена для технических специалистов, которые будут вводить в эксплуатацию и/или обслуживать устройства, которые обмениваются данными по протоколу Modbus.

1. В самом начале рассмотрена общая информация по протоколам Modbus RTU и Modbus TCP. Далее в разных частях дано пошаговое описание реализации протокола Modbus RTU Master (клиент, запрашивающий и/или записывающий данные) и Modbus RTU Slave (сервер, предоставляющий данные) на контроллерах Beckhoff.
2. Для Вашего удобства дано краткое описание работы со сторонним ПО для проверки корректности настройки протокола. По причине того, что это стороннее ПО, компания ООО "Бекхофф Автоматизация" не несет за них никакой ответственности.
3. В своей практической работе дополнительно Вам может понадобиться пошаговая инструкция "Первые шаги с СХ в TwinCAT". Она, как и данная инструкция, доступна для скачивания как отдельно, так и в составе общего практического пособия по программированию контроллеров Beckhoff.

2. Общая информация по Modbus

Что такое протокол передачи данных?

Протокол передачи данных – набор соглашений интерфейса логического уровня (SW), которые определяют обмен данными между различными программами. Эти соглашения задают единообразный способ передачи сообщений и обработки ошибок.

Стандартизированный протокол передачи данных также позволяет разрабатывать интерфейсы (HW) (уже на физическом уровне), не привязанные к конкретной аппаратной платформе и производителю (например, USB, Bluetooth).

	MODBUS RTU	MODBUS TCP
Физический интерфейс (аппаратная часть)	RS232 RS422/RS485	Ethernet

Как Modbus реализуется на BECKHOFF?

Общее описание на сайте www.beckhoff.com/modbus

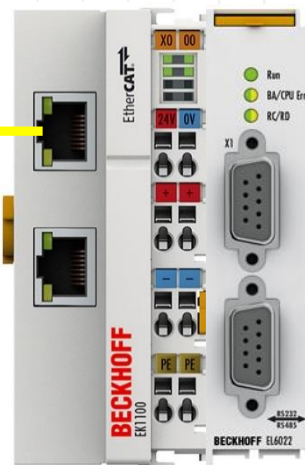
Реализация	Варианты	Modbus RTU	Modbus TCP
Аппаратная реализация интерфейса (HW)	Уже есть на "борту"	BC8150, BC7300, BC8050 BK7350, BK7300	CX80xx (кроме CX8091)
	Отдельным модулем или опцией	Реализация интерф. RSxxx: K-Bus:KL6001/6031/6021/6041 E-Bus:EL6001/6002/6021/6022 Оptionальный интерфейс для CX9020/CX50x0/CX20x0	Ethernet адаптер в ПЛК, модули I/O EL6601, EL6614 или опциональные интерфейсы для CX20x0
Программная реализация протокола (SW)	Уже есть на "борту"	У коптеров и контроллеров, у которых физический уровень Modbus RTU уже реализован на "борту"	У контроллеров серии CX80xx, у которых физический уровень Modbus TCP реализован на "борту"
	Через доп. библиотеку	В остальных случаях требуется библиотека TwinCAT Modbus RTU	В остальных случаях требуется ПО TwinCAT Modbus TCP Server

Реализация Modbus RTU в классе обучения

CP66xx



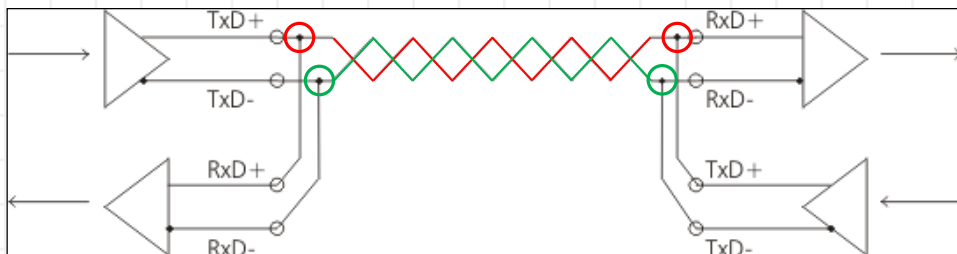
EL6022:



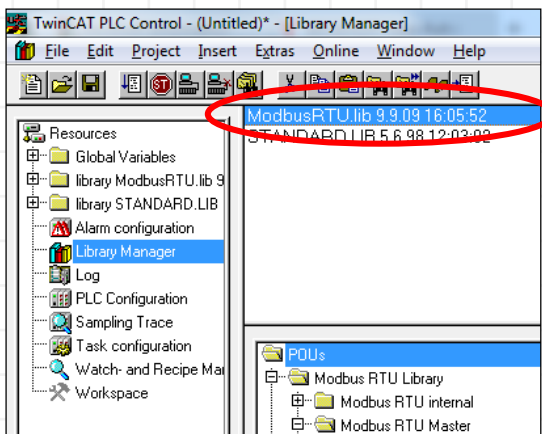
EtherCAT

RS422
RS485

Pin	RS485/RS422
1	-
2	Tx+
3	Rx+
4	-
5	GND
6	+5V
7	Tx-
8	Rx-
9	-



Больше информации по подключению



Библиотека:
ModbusRTU.lib –
Программная реализация
протокола Modbus RTU

Платный компонент.
Поставляется отдельно.

Modbus области

В процессе работы с документацией по Modbus часто встречается понятие "Modbus области" (Modbus-areas) – что это значит и как это применимо к различной памяти контроллеров?

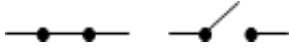

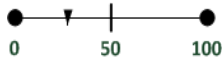

Modbus-areas	Data type	Access	I, Q, M	Example
Modbus области	Тип данных	Доступ		Пример
Digital inputs (Discrete Inputs)	1 Bit	Read only	I	
Digital outputs (Coils)	1 Bit	Read / write	Q	
Input registers	16 Bit	Read only	I	
Output registers	16 Bit	Read / write	Q, M	

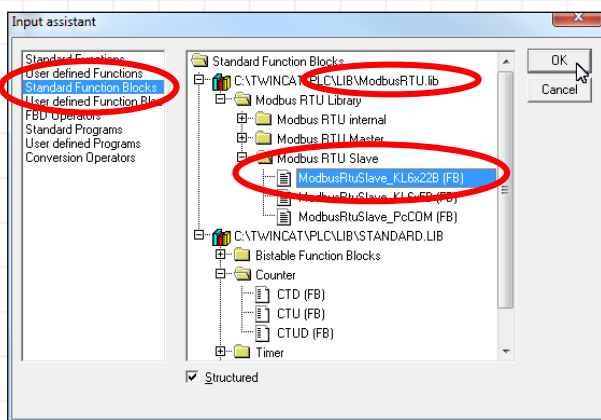
Таблица соответствия Modbus TCP функций Beckhoff

* - 1 регистр = 16 бит

Modbus TCP function Функция Modbus TCP	Function code Код функции (по стандарту Modbus)	PLC block Функциональный блок библиотеки
Read Coils	1 → Q (1 бит x N)	FB_MBReadCoils
Read Discrete Inputs	2 → I (1 бит x N)	FB_MBReadInputs
Read Registers*	3 → Q, M (16 бит x N)	FB_MBReadRegs
Read Input Registers*	4 → I (16 бит x N)	FB_MBReadInputRegs
Write Single Coil	5 → Q (1 бит)	FB_MBWriteSingleCoil
Write Single Register*	6 → Q, M (16 бит)	FB_MBWriteSingleReg
Write Multiple Coils	15 → Q (1 бит x N)	FB_MBWriteCoils
Write Multiple Registers*	16 → Q, M (16 бит x N)	FB_MBWriteRegs
Read/Write Multiple Registers* (не для RTU)	23 → Q, M (16 бит x N)	FB_MBReadWriteRegs
Diagnostic	8	FB_MBDiagnose

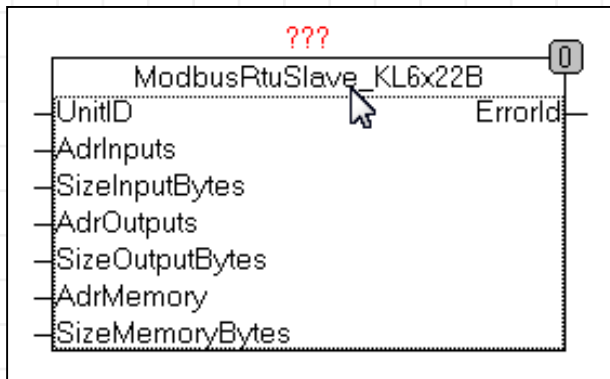
3. Пошаговая реализация Modbus RTU Slave (сервер)

1. Запустите TwinCAT PLC Control ¹⁾
2. Создайте новый пустой проект ¹⁾
3. Подключите к проекту библиотеку ¹⁾ *ModbusRTU.lib*
4. Выполните сохранение проекта ¹⁾
5. Путь к проекту не должен содержать кириллицу ¹⁾
6. Имя проекта не должно содержать кириллицу ¹⁾
7. В проект добавьте новую программу POU ¹⁾
8. Задайте имя, например *PRG_ModbusSlave*, тип программы Programm и язык программирования CFC ¹⁾
9. В программе *MAIN* напишите вызов только что созданной программы ¹⁾
10. Откройте созданную программу и добавьте пустой Box ¹⁾
11. Встаньте курсором в Box и нажмите F2 ¹⁾
12. Из раздела стандартных функциональных блоков (FB) ¹⁾ выберите *ModbusRtuSlave_KL6x22B* из библиотеки *ModbusRTU.lib*



¹⁾ Подробное описание см. в руководстве "TwinCAT 2: Первые шаги программирования контроллера CX"

13. Появится следующий функциональный блок



14. В области объявления переменных должны быть созданы следующие переменные

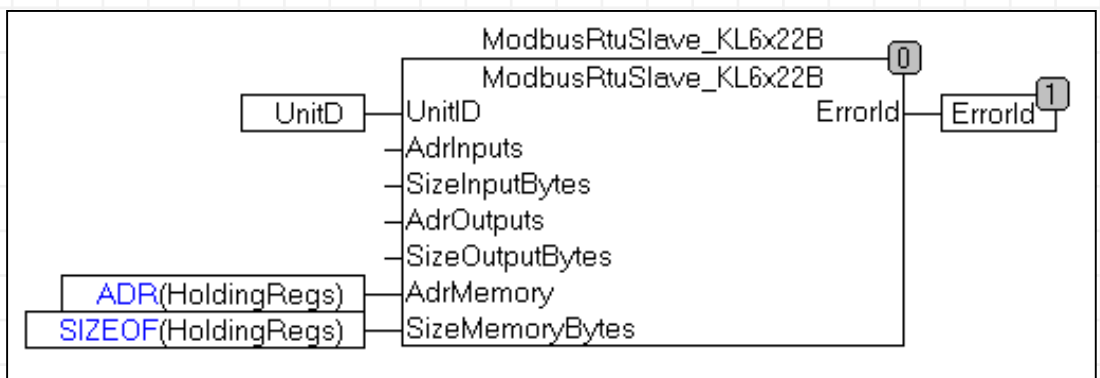
VAR

```

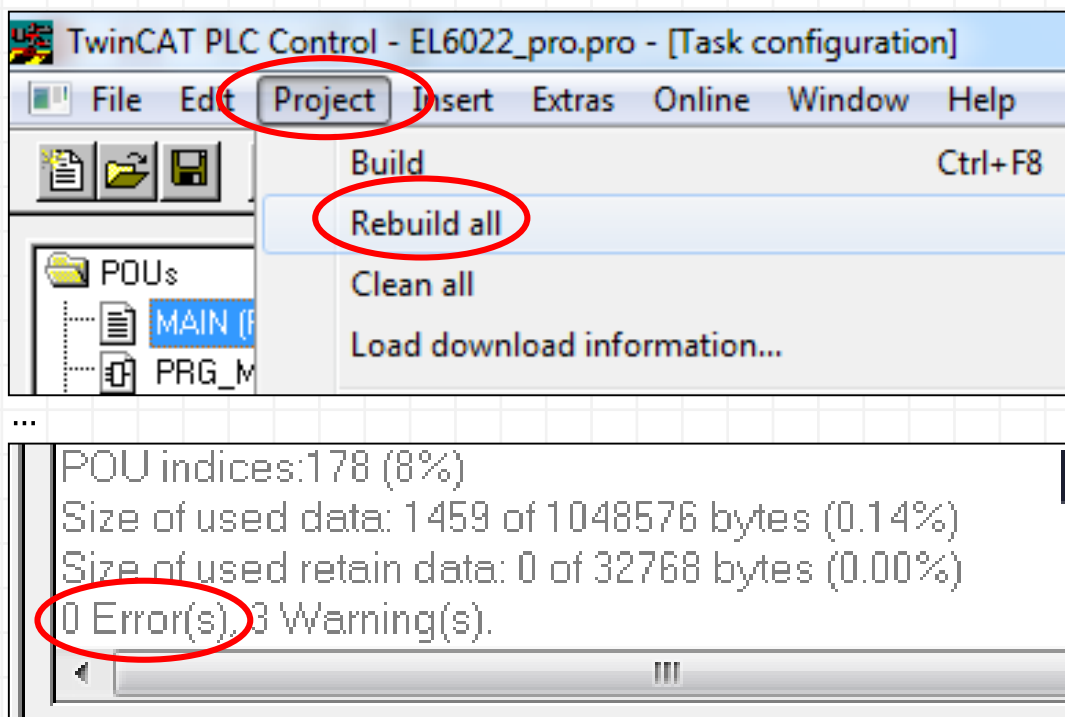
ModbusRtuSlave_KL6x22B: ModbusRtuSlave_KL6x22B;
UnitD :                      UINT:=1;
(*Номер Slave устройства*)
ErrorId :                    Modbus_ERRORS;
(*Переменная для хранения ошибок Modbus*)
HoldingRegs :                ARRAY [0..7] OF WORD;
(*Массив переменных для Modbus memory. Для примера используется 8 слов *)
    
```

END_VAR

15. Прилинкуйте переменные как показано ниже



16. Скомпилируйте проект ¹⁾



Ошибок быть не должно

17. Запустите TwinCAT System Manager ¹⁾

18. Создайте новый пустой проект ¹⁾

19. Выполните сохранение проекта ¹⁾

20. Путь к проекту не должен содержать кириллицу ¹⁾

21. Имя проекта не должно содержать кириллицу ¹⁾

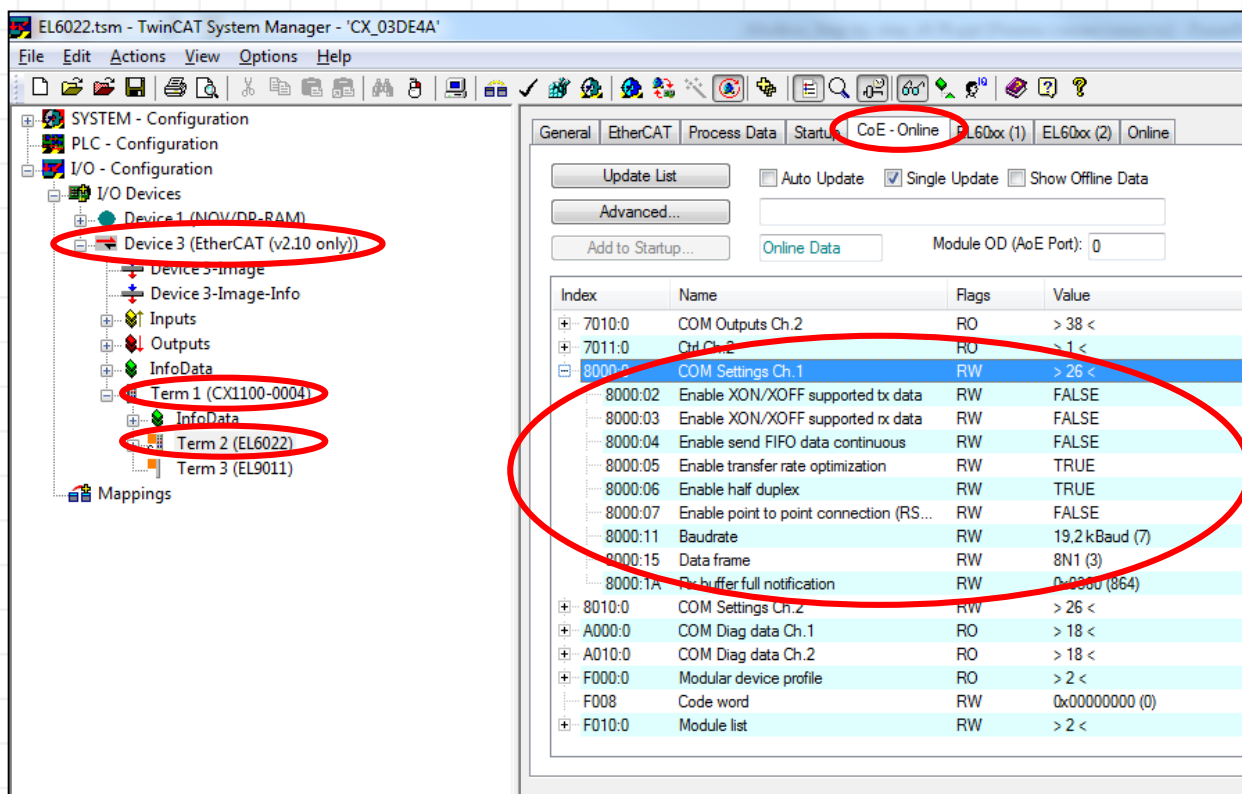
22. Выполните поиск устройства ¹⁾

23. Добавьте устройство в роутер ПК ¹⁾

24. Выберите добавленный контроллер как целевое устройство ¹⁾

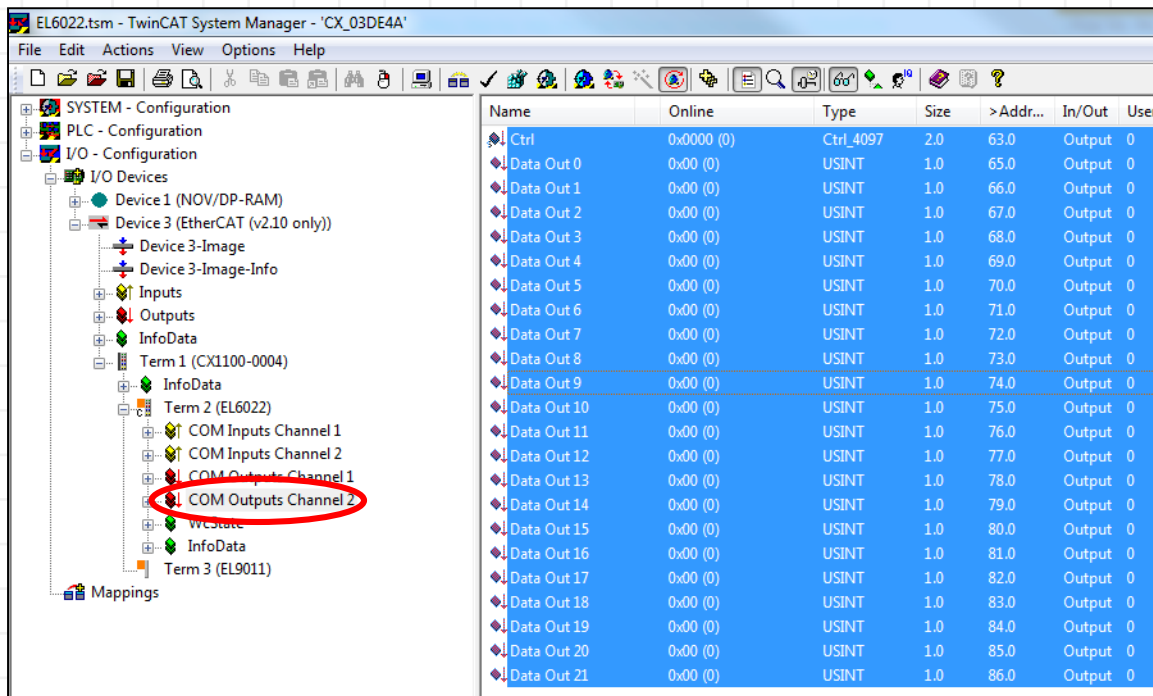
¹⁾ Подробное описание см. в руководстве "TwinCAT 2: Первые шаги программирования контроллера CX"

25. Убедитесь, что Вы выбрали правильное целевое устройство ¹⁾
26. Убедитесь, что у Вас есть связь с выбранным целевым устройством ¹⁾
27. Переведите целевое устройство в режим Config Mode ¹⁾
28. Выполните сканирование модулей ввода/вывода ¹⁾
29. Выберите вкладку PLC-Configuration ¹⁾
30. Найдите файл *.tpy, который создается автоматически после компиляции проекта ПЛК без ошибок ¹⁾
31. Убедитесь, что во вкладке PLC-Configuration появился раздел одноименный с именем файла *.tpy ¹⁾
32. Проверьте настройки модуля EL6022 для соответствующего номера COM порта

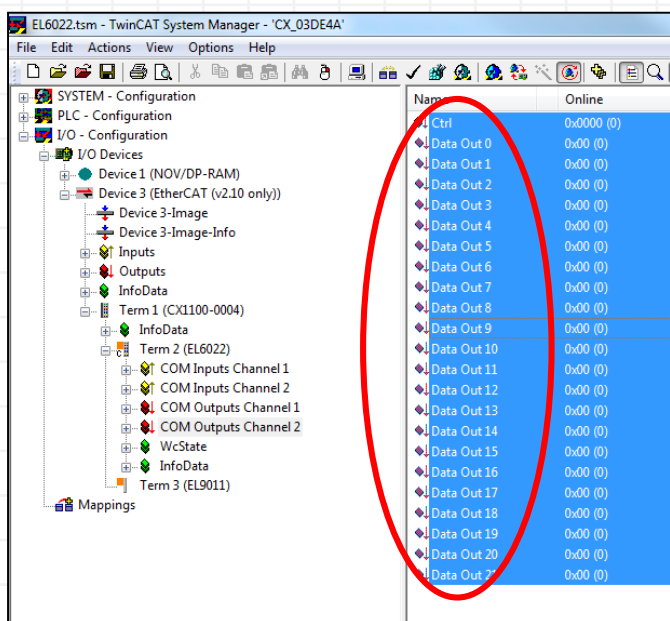


¹⁾ Подробное описание см. в руководстве "TwinCAT 2: Первые шаги программирования контроллера CX"

33. Раскройте модуль EL6022 и встаньте курсором на COM Outputs соответствующего канала. Здесь в примере выбран COM Outputs Channel 2



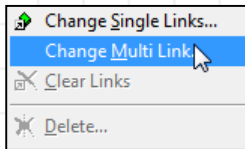
34. Выделите все переменные



35. Нажмите правую кнопку мыши

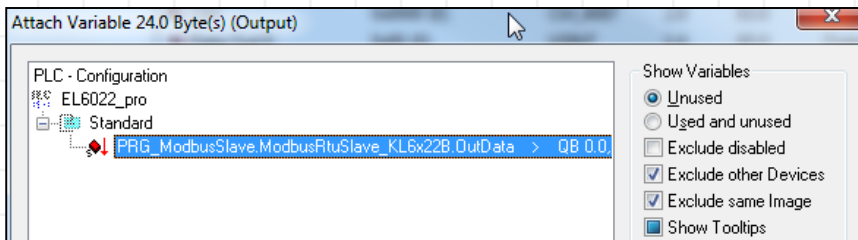


36. Выберите Change Multi Link

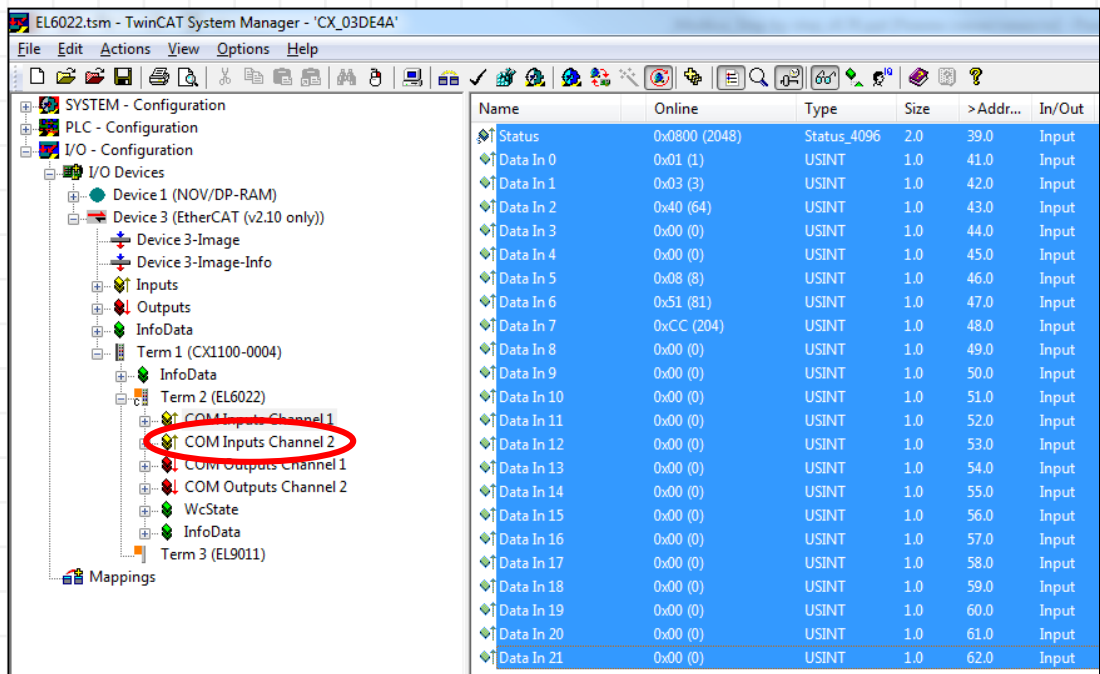


37. Прилинкуйте к

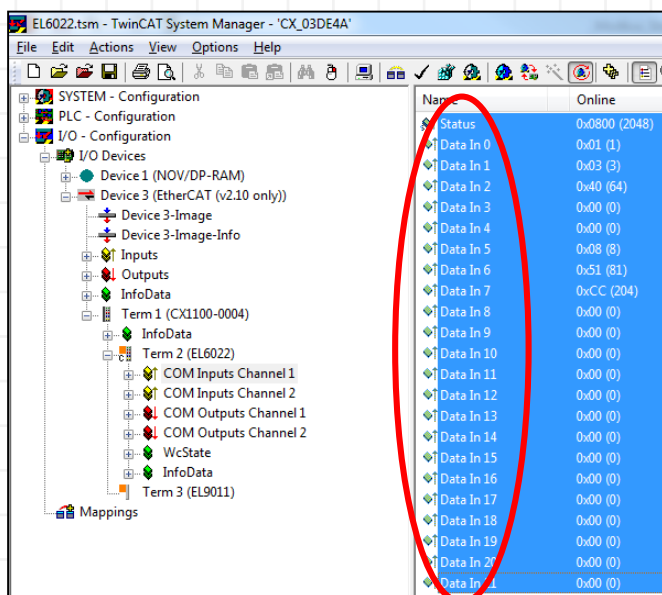
PRG_ModbusSlave.ModbusRtuSlave_KL6x22B.OutData



38. Теперь в этом же раскрывшемся модуле EL6022 вставьте курсором на COM Inputs соответствующего канала. Здесь в примере выбран COM Inputs Channel 2



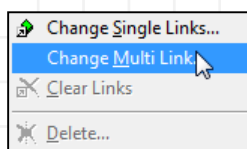
39. Выделите все переменные



40. Нажмите правую кнопку мыши

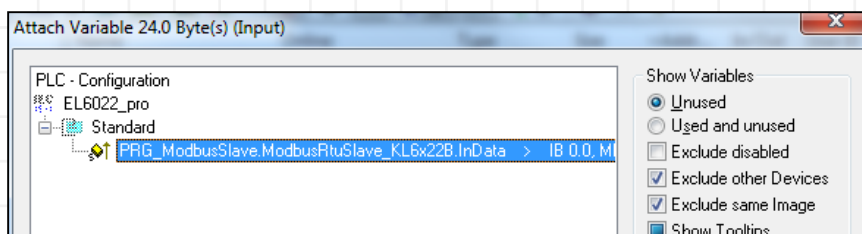


41. Выберите Change Multi Link



42. Прилинкуйте к

PRG_ModbusSlave.ModbusRtuSlave_KL6x22B.InData



43. Активируйте конфигурацию ¹⁾

¹⁾ Подробное описание см. в руководстве "TwinCAT 2: Первые шаги программирования контроллера CX"

44. Переведите целевое устройство в режим RUN ¹⁾
45. Проверьте, что осталась связь с целевым устройством и его состояние Config Mode изменилось на Run Mode ¹⁾
46. Вернитесь в TwinCAT PLC Control ¹⁾
47. Выберите Choose Run-Time System ¹⁾
48. Выберите контроллер и Порт ¹⁾
49. Выполните подключение к ПЛК и загрузите программу ¹⁾
50. Запустите выполнение программы ¹⁾
51. Конфигурирование и программирование Modbus RTU Slave окончено

В электронную версию инструкции вложены результирующие файлы проектов TwinCAT PLC Control и TwinCAT System Manager, на примере которых рассматривалась данная инструкция.



EL6022_tsm.tsm



EL6022_pro.pro

Результирующие файлы в архиве доступны для скачивания с сайта по этому QR-коду



Руководство "TwinCAT 2: Первые шаги программирования контроллера CX" доступно для скачивания с сайта по этому QR-коду



¹⁾ Подробное описание см. в руководстве "TwinCAT 2: Первые шаги программирования контроллера CX"

4. Работа со сторонним ПО "Modbus Poll" (Master)

Modbus Poll – это стороннее программное обеспечение, рекомендуемое инженерами ООО "Бекхофф Автоматизация", работа с которым коротко описана здесь исключительно для ознакомления. Для своей работы Вы в праве выбирать любой другой удобный для себя продукт. Полное описание Вы можете найти на сайте. Часть текста приведена ниже.

http://www.modbustools.com/modbus_poll.asp



ПО "Modbus Poll" является мастером в сети Modbus, поддерживает различные функции для считывания и записи данных.

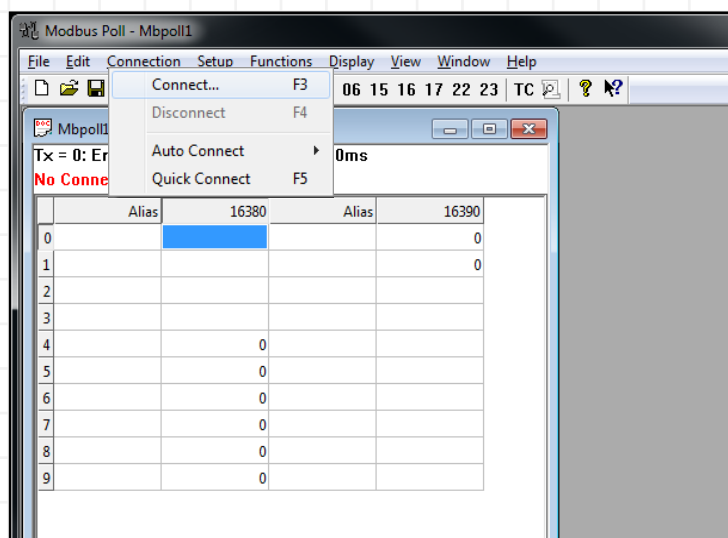
Поддерживаемые протоколы

Modbus RTU, Modbus ASCII, Modbus TCP/IP, Modbus RTU Over TCP/IP, Modbus ASCII Over TCP/IP
Modbus UDP/IP, Modbus RTU Over UDP/IP, Modbus ASCII Over UDP/IP

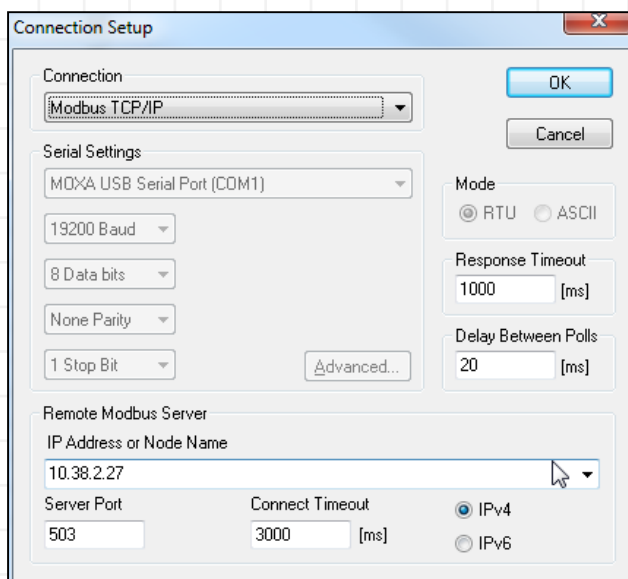
Чтение/запись до 125 регистров (Registers)

Чтение/запись до 2000 входов/выходов (Inputs/Coils)

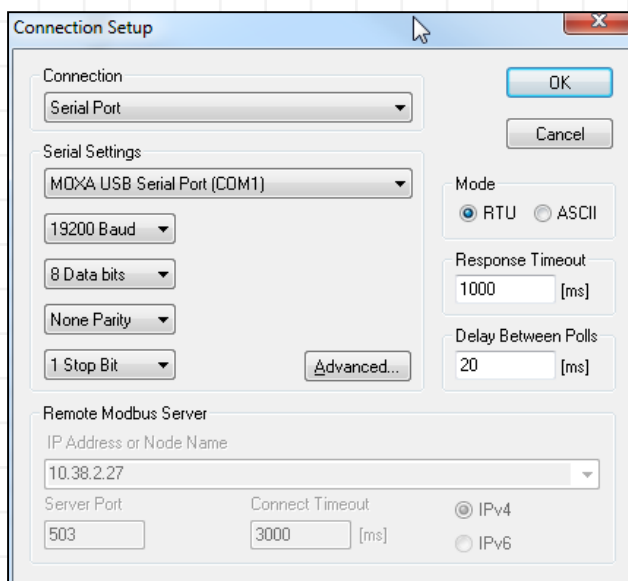
1. Скачайте, установите и запустите программу. В падающем меню выберите *Connection* → *Connect...*



2. Появившееся окно настройки порта

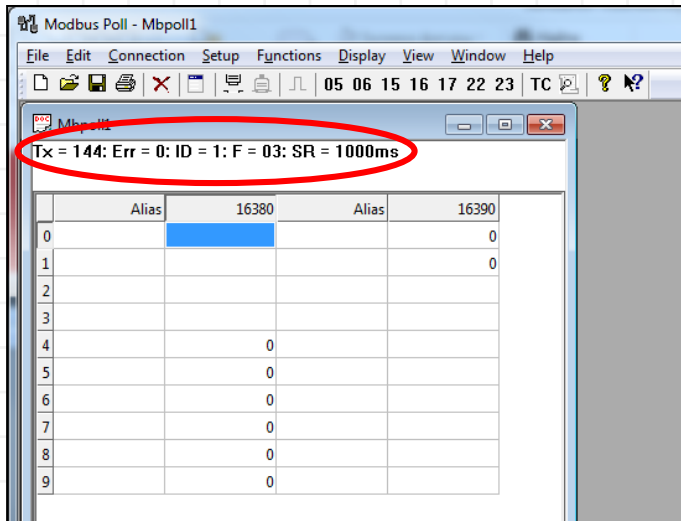


3. В *Connection* выберите *Serial Port*. Выполните настройки порта в соответствии с модулем EL6022 (или иным, что используете)

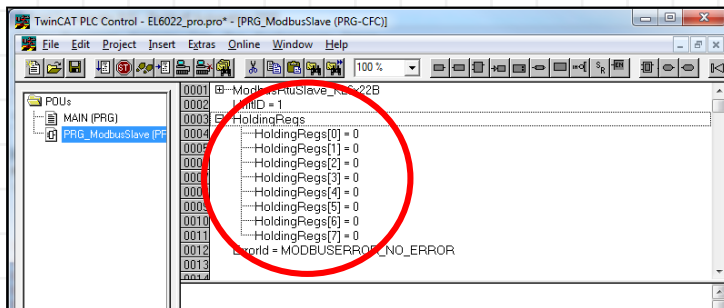


Проверка связи

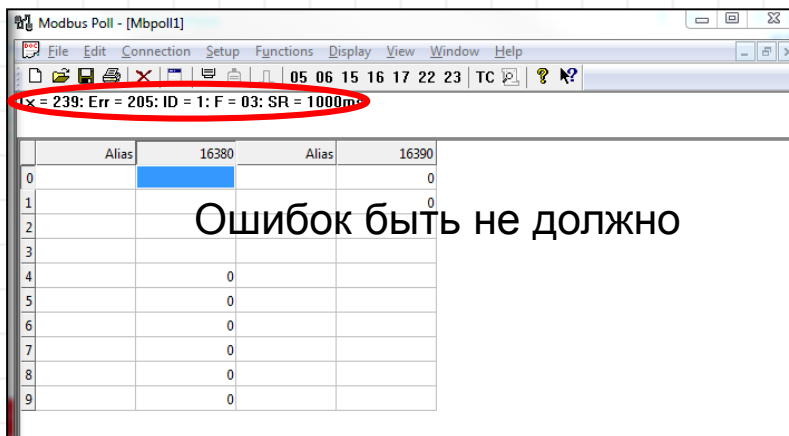
4. Результат подключения – ошибок связи быть не должно



5. TwinCAT PLC Control



6. Modbus Poll



Типы памяти ПЛК, их адресация и используемые функции

7. Используемые функции, тип адресации протокола и адреса можно найти на сайте infosys.beckhoff.com



Выбор Base 0 в Modbus Poll

Definition of the Modbus memory data as variable without flag address:

```
VAR
Memory : ARRAY[0..255] OF WORD;
END_VAR
```

Access to the *Memory* array via a Modbus master is possible with the following functions:

- 3 : Read Holding Registers
- 6 : Preset Single Register
- 16 : Preset Multiple Registers

Addressing

The *Memory* array is addressed with a 16#4000 offset, i.e. address 16#4000 in the output data array.

Examples:

PLC variable	Access type	Address in the Modbus telegram	Address in the end device (device-dependant)
Memory[0]	Word	16#4000	44001
Memory[1]	Word	16#4001	44002

8. Выбор функции

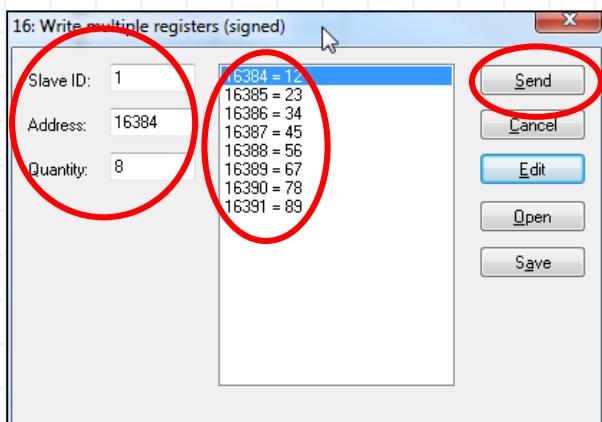
Modbus Poll - [Mbpoll1]

File Edit Connection Setup **Functions** Display View Window Help

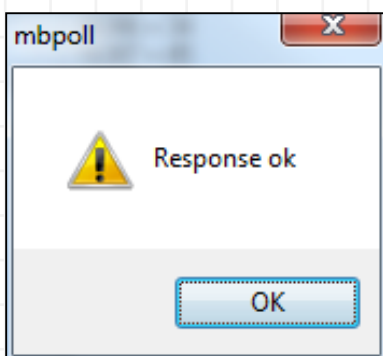
- 05: Write Single Coil... Alt+F5
- 06: Write Single Register... Alt+F6
- 15: Write Coils... Alt+F7
- 16: Write Registers... Alt+F8**
- 17: Report Slave ID...
- 22: Mask Write Register...
- 23: Read/Write Registers...
- Test Center... Alt+T

Запись данных в ПЛК - Function Code 16

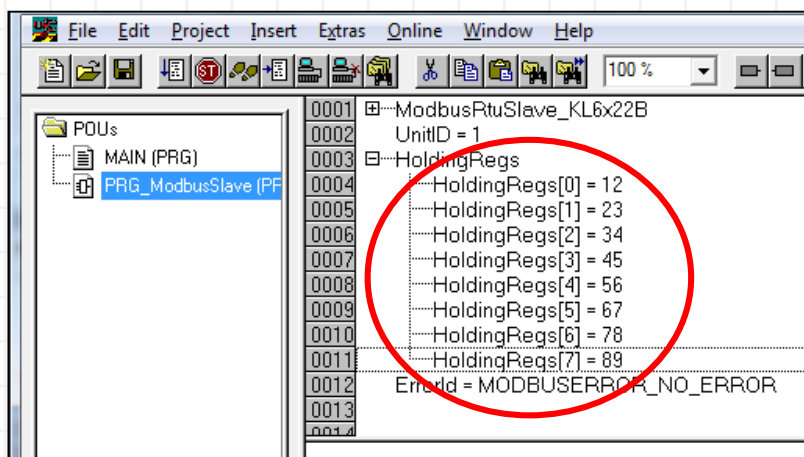
9. Указание ID номера устройства, начальный адрес, количество записываемых регистров и их значения



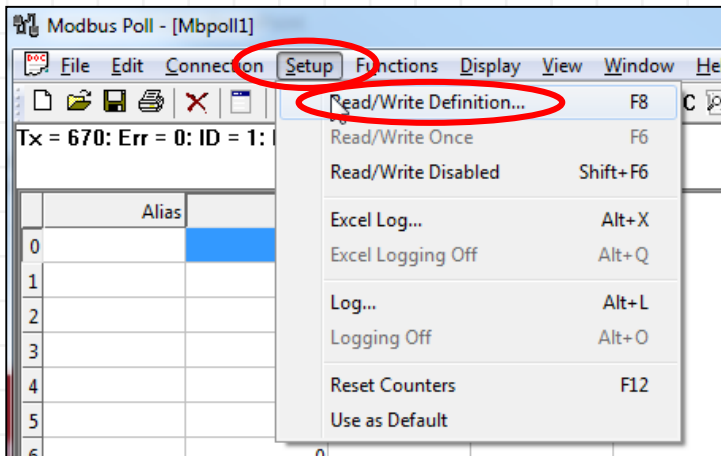
10. Успешная передача данных



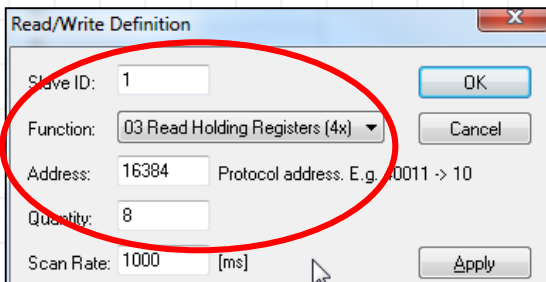
11. Проверка значений переменных в ПЛК



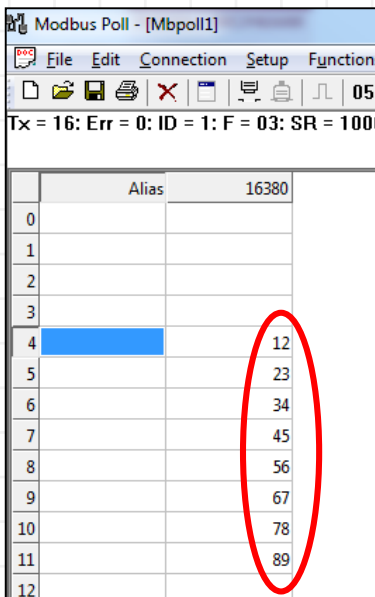
12. В падающем меню выберите: *Setup* → *Read/Write Definition...*



13. Выберите функцию *03 Read Holding Registers*. Введите адрес *16384*, количество считываемых регистров - *8*



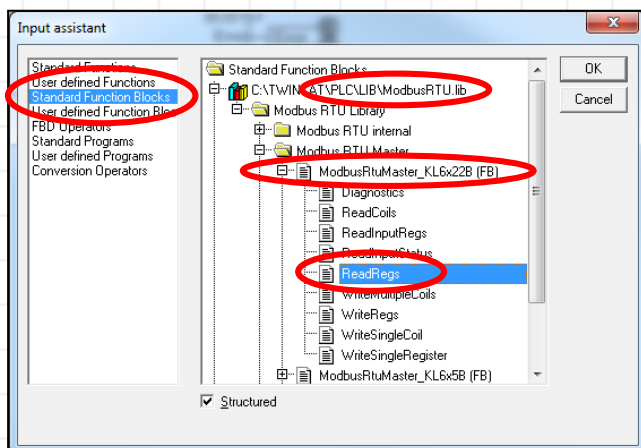
14. Результат считывания регистров



ЧАСТЬ 2

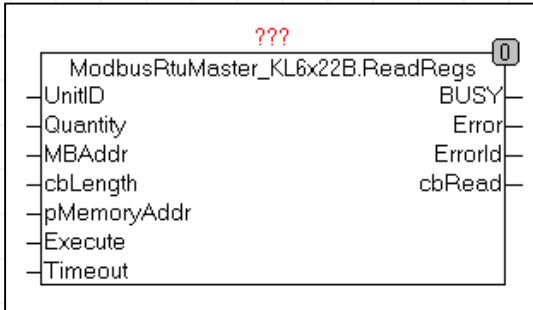
5. Пошаговая реализация Modbus RTU Master (клиент)

1. Запустите TwinCAT PLC Control ¹⁾
2. Создайте новый пустой проект ¹⁾
3. Подключите к проекту библиотеку ¹⁾ *ModbusRTU.lib*
4. Выполните сохранение проекта ¹⁾
5. Путь к проекту не должен содержать кириллицу ¹⁾
6. Имя проекта не должно содержать кириллицу ¹⁾
7. В проект добавьте новую программу POU ¹⁾
8. Задайте имя, например PRG_ModbusMaster, тип программы Programm и язык программирования CFC ¹⁾
9. В программе *MAIN* напишите вызов только что созданной программы ¹⁾
10. Откройте созданную программу и добавьте пустой Box ¹⁾
11. Встаньте курсором в Box и нажмите F2 ¹⁾
12. Из раздела стандартных функциональных блоков (FB) ¹⁾ выберите ReadRegs из *ModbusRtuMaster_KL6x22B* из библиотеки *ModbusRTU.lib*



¹⁾ Подробное описание см. в руководстве "TwinCAT 2: Первые шаги программирования контроллера CX"

13. Появится следующий функциональный блок



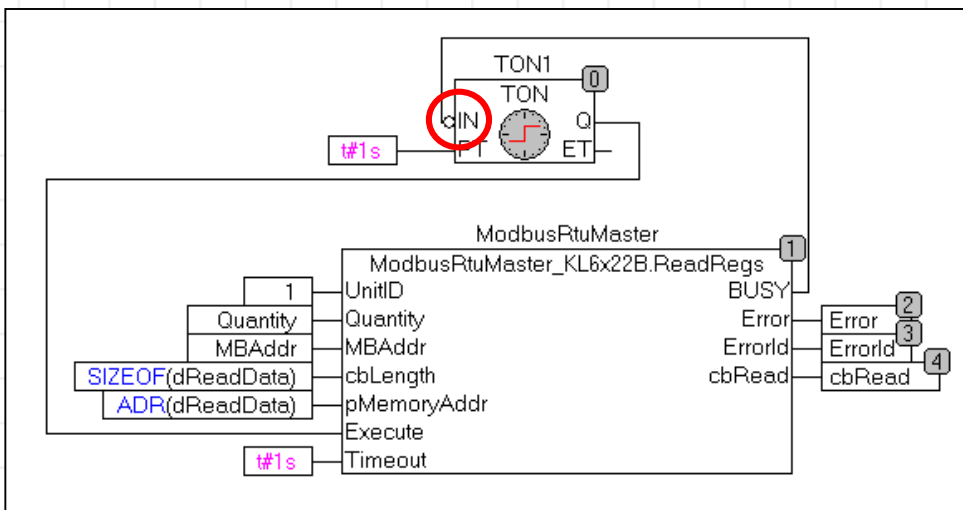
14. В области объявления переменных должны быть созданы следующие переменные

```
PROGRAM PRG_ModbusMaster
VAR
```

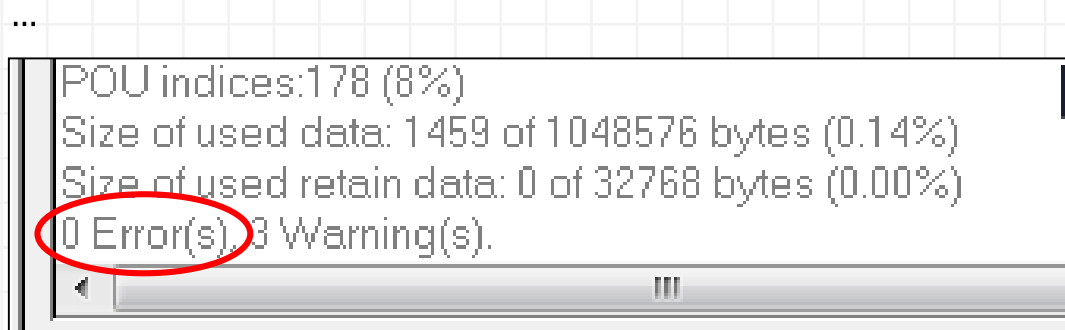
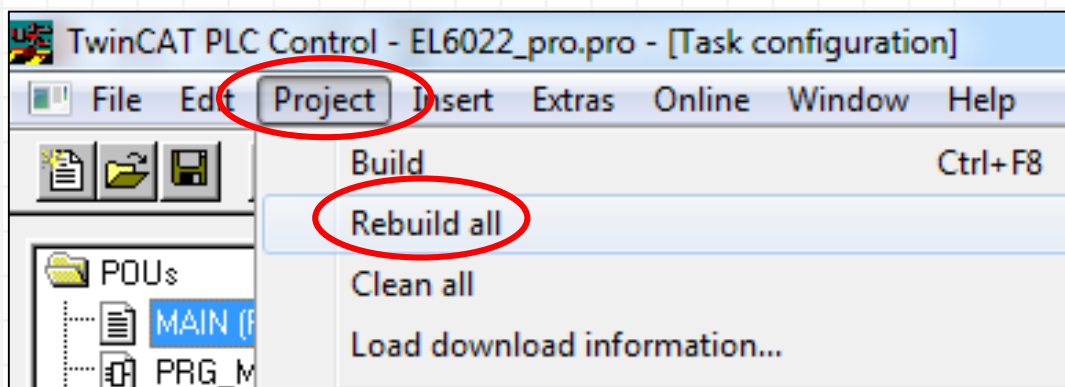
```
TON1: TON; (*Таймер задержки для периодического считывания данных*)
ModbusRtuMaster : ModbusRtuMaster_KL6x22B;
Quantity : WORD:=8; (*Количество считываемых слов*)
MAddr : WORD:=16#4000;
(*Адрес первого слова, с которого нужно считать данные = 16384 dec*)
dReadData : ARRAY [0..7] OF WORD;
(*Массив ПЛК, куда нужно сохранить считанный массив данных*)
Error : BOOL; (*Наличие/отсутствие ошибки*)
ErrorId : MODBUS_ERRORS; (*Текстовое описание ошибки*)
cbRead : UINT; (*Количество считанных байт*)
```

```
END_VAR
```

15. Реализуйте следующий алгоритм и прилинкуйте переменные как показано ниже



16. Скомпилируйте проект ¹⁾



Ошибок быть не должно

17. Запустите TwinCAT System Manager ¹⁾

18. Создайте новый пустой проект ¹⁾

19. Выполните сохранение проекта ¹⁾

20. Путь к проекту не должен содержать кириллицу ¹⁾

21. Имя проекта не должно содержать кириллицу ¹⁾

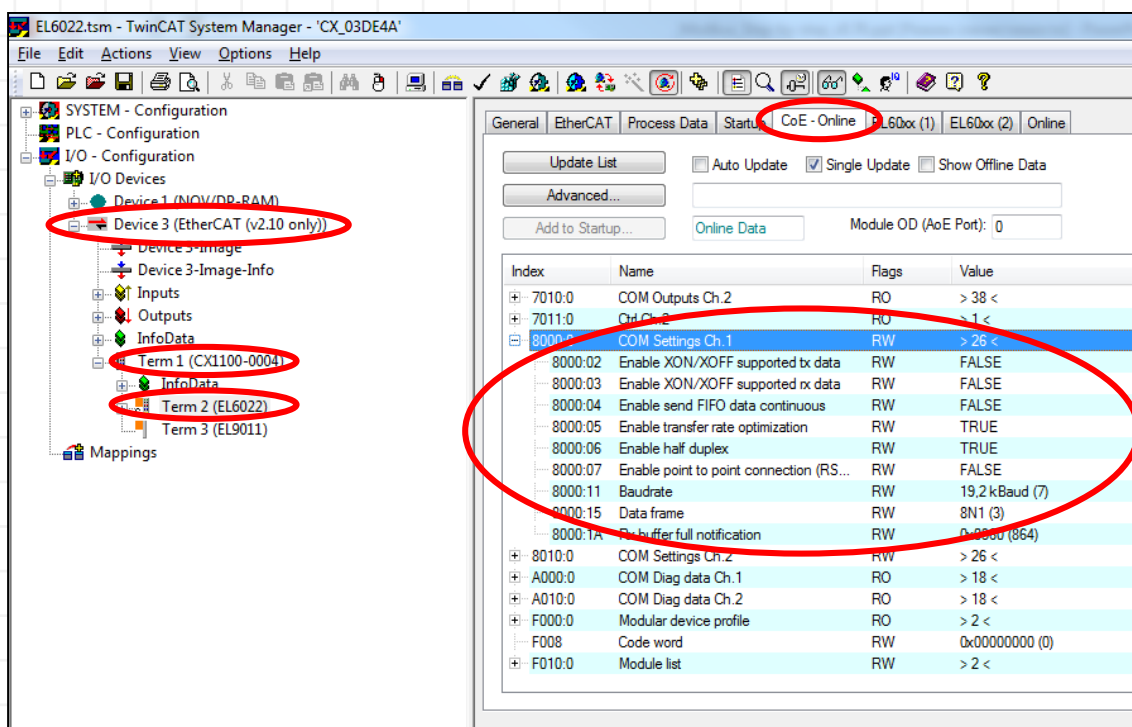
22. Выполните поиск устройства ¹⁾

23. Добавьте устройство в роутер ПК ¹⁾

24. Выберите добавленный контроллер как целевое устройство ¹⁾

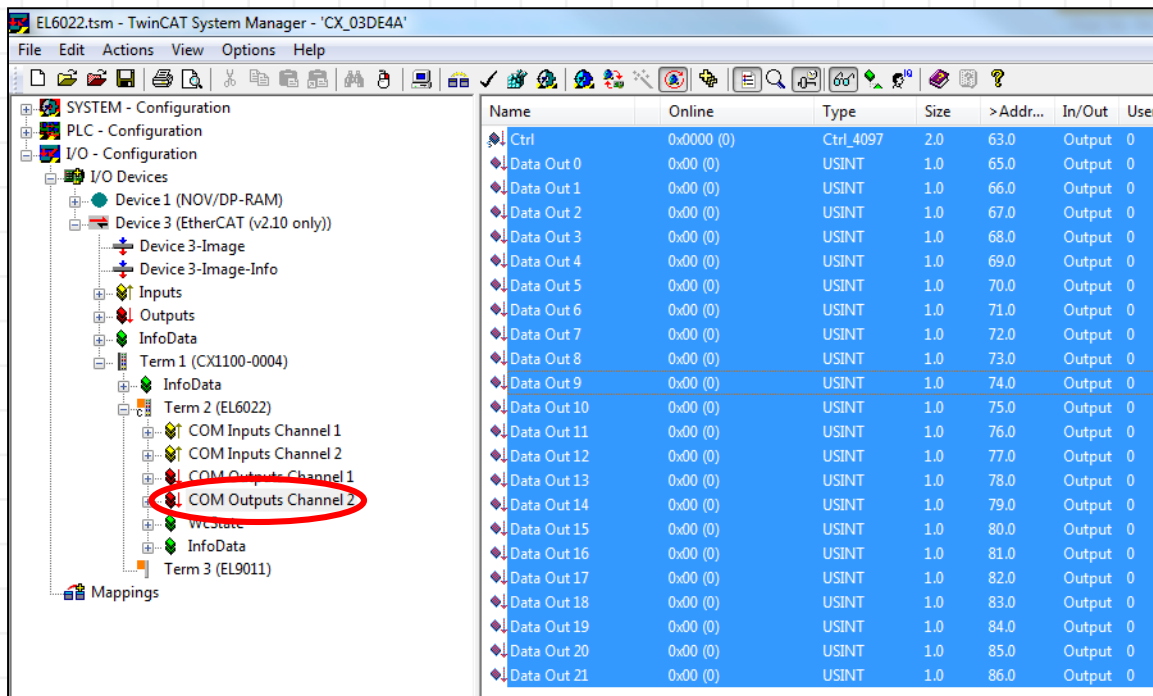
¹⁾ Подробное описание см. в руководстве "TwinCAT 2: Первые шаги программирования контроллера CX"

25. Убедитесь, что Вы выбрали правильное целевое устройство ¹⁾
26. Убедитесь, что у Вас есть связь с выбранным целевым устройством ¹⁾
27. Переведите целевое устройство в режим Config Mode ¹⁾
28. Выполните сканирование модулей ввода/вывода ¹⁾
29. Выберите вкладку PLC-Configuration ¹⁾
30. Найдите файл *.tpy, который создается автоматически после компиляции проекта ПЛК без ошибок ¹⁾
31. Убедитесь, что во вкладке PLC-Configuration появился раздел одноименный с именем файла *.tpy ¹⁾
32. Проверьте настройки модуля EL6022 для соответствующего номера COM порта

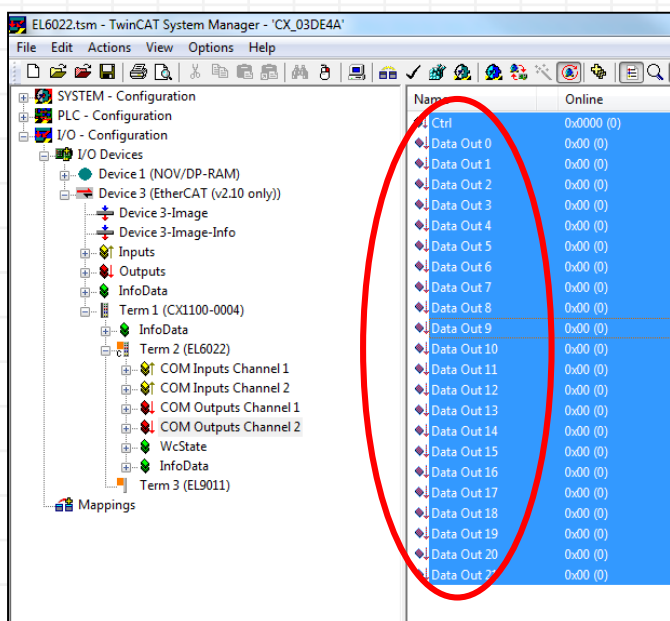


¹⁾ Подробное описание см. в руководстве "TwinCAT 2: Первые шаги программирования контроллера CX"

33. Раскройте модуль EL6022 и встаньте курсором на COM Outputs соответствующего канала. Здесь в примере выбран COM Outputs Channel 2



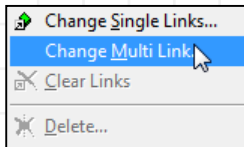
34. Выделите все переменные



35. Нажмите правую кнопку мыши

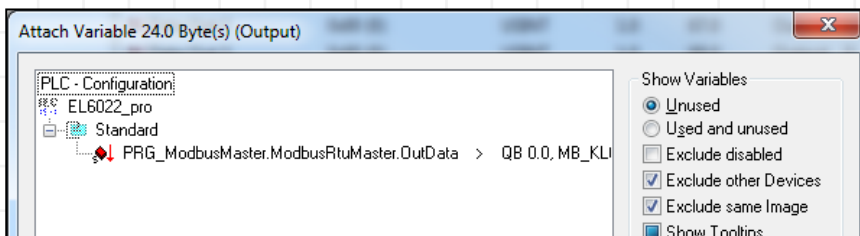


36. Выберите Change Multi Link

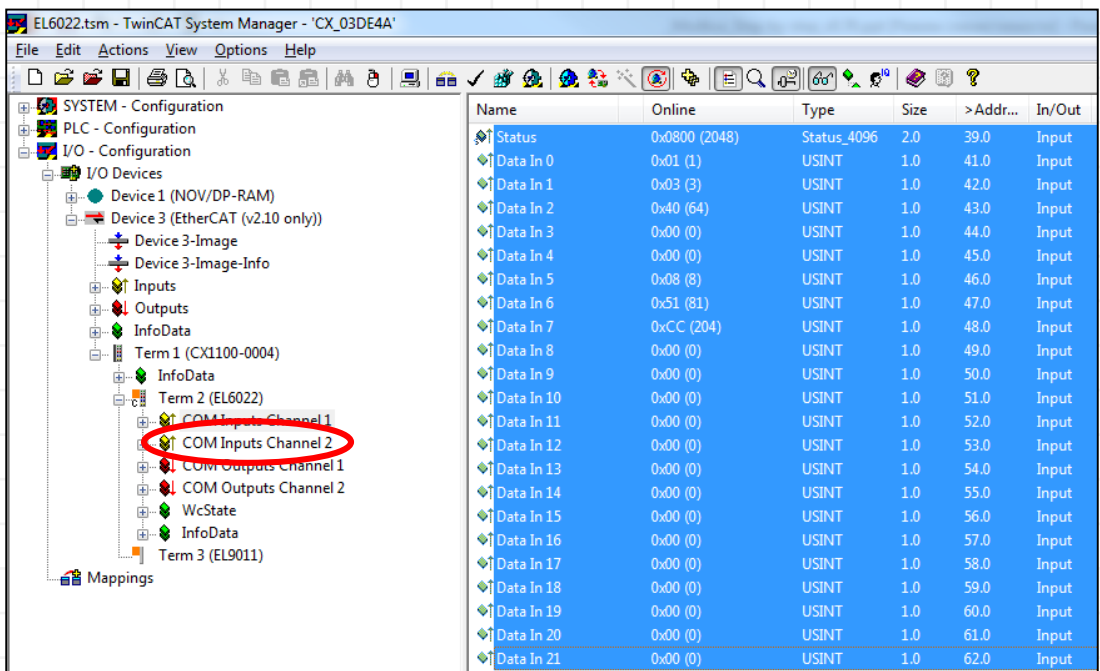


37. Прилинкуйте к

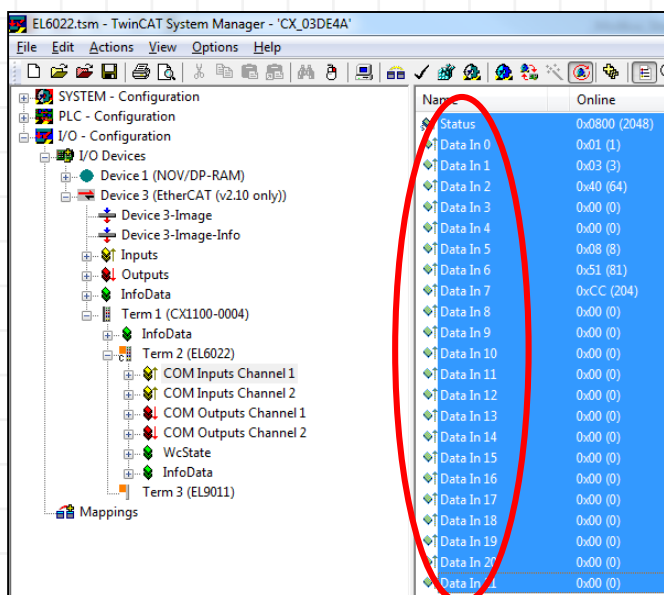
PRG_ModbusMaster.ModbusRtuMaster.OutData



38. Теперь в этом же раскрывшемся модуле EL6022 встаньте курсором на COM Inputs соответствующего канала. Здесь в примере выбран COM Inputs Channel 2



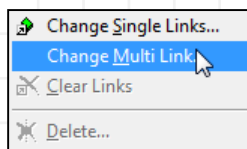
39. Выделите все переменные



40. Нажмите правую кнопку мыши

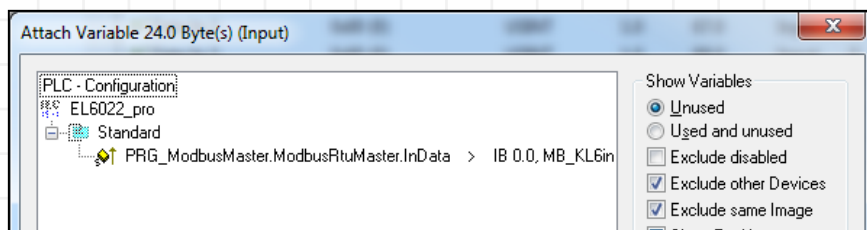


41. Выберите Change Multi Link



42. Прилинкуйте к

PRG_ModbusMaster.ModbusRtuMaster.InData



43. Активируйте конфигурацию ¹⁾

¹⁾ Подробное описание см. в руководстве "TwinCAT 2: Первые шаги программирования контроллера CX"

44. Переведите целевое устройство в режим RUN ¹⁾
45. Проверьте, что осталась связь с целевым устройством и его состояние Config Mode изменилось на Run Mode ¹⁾
46. Вернитесь в TwinCAT PLC Control ¹⁾
47. Выберите Choose Run-Time System ¹⁾
48. Выберите контроллер и Порт ¹⁾
49. Выполните подключение к ПЛК и загрузите программу ¹⁾
50. Запустите выполнение программы ¹⁾
51. Конфигурирование и программирование Modbus RTU Slave окончено

В электронную версию инструкции вложены результирующие файлы проектов TwinCAT PLC Control и TwinCAT System Manager, на примере которых рассматривалась данная инструкция.



EL6022_tsm.tsm



EL6022_pro.pro

Результирующие файлы в архиве
доступны для скачивания с сайта
по этому QR-коду



Руководство "TwinCAT 2: Первые шаги
программирования контроллера CX"
доступно для скачивания с сайта
по этому QR-коду



¹⁾ Подробное описание см. в руководстве "TwinCAT 2: Первые шаги программирования контроллера CX"

6. Работа со сторонним ПО "Эмулятор Modbus"

Эмулятор Modbus RTU/TCP устройств – это стороннее программное обеспечение, рекомендуемое инженерами ООО "Бекхофф Автоматизация", работа с которым коротко описана здесь исключительно для ознакомления. Для своей работы Вы в праве выбрать любой другой удобный для себя продукт. Полное описание Вы можете найти на сайте.

<http://ardsoft.ru/mEmulator.html>



Эмулятор Modbus устройств предназначен для создания виртуальной сети Modbus RTU или Modbus TCP устройств. Эмулятор полезен в первую очередь специалистам, разрабатывающим программное обеспечение опроса Modbus устройств, системным интеграторам, студентам для отладки Modbus клиентов, OPC/DDE серверов, тестирования оборудования, проверки SCADA проектов, создания стендов имитации технологического оборудования, систем обучения оперативного персонала, диагностики каналов связи и для многих других задач.

Ключевые особенности эмулятора:

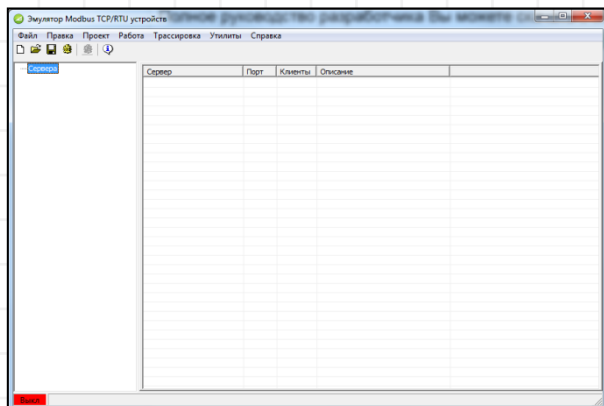
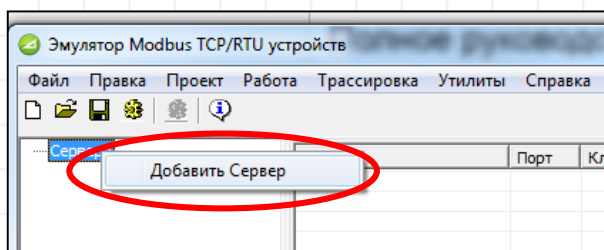
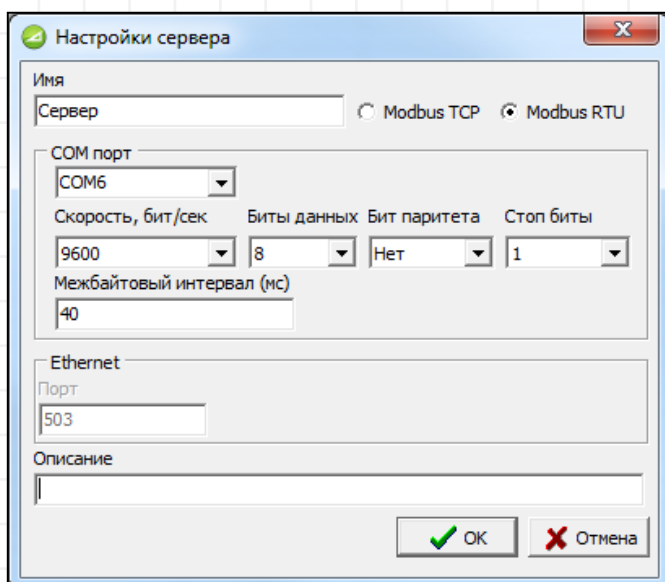
- работа по протоколам Modbus RTU и Modbus TCP;
- нет ограничений по конфигурированию;
- нет ограничений при подключении к OPC серверам;
- бесплатен для коммерческого использования;
- работа во всех версиях ОС Microsoft Windows, начиная с Windows 2000;
- работа в ОС Microsoft Windows 7 Embedded.

Цена: бесплатно.

Использование в коммерческих целях: можно все кроме продажи от своего имени.

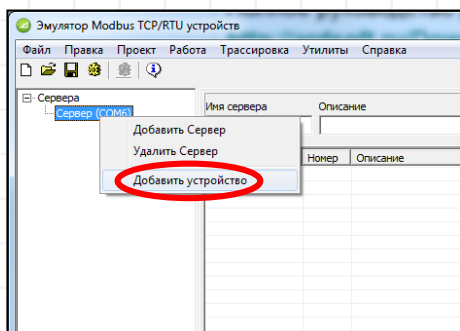
Запуск программы, добавление сервера и настройка COM порта

1. Скачайте ПО и запустите его

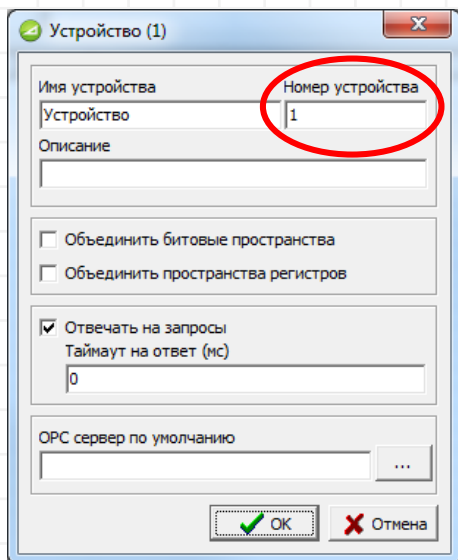
2. Кликните правой клавишей мыши на *Сервера*3. Кликните *Добавить Сервер* и выполните настройки COM порта. Параметры должны быть согласованы с настройками регистров модуля EL6022 (или иного модуля, который Вы используете по факту)

Добавление устройства

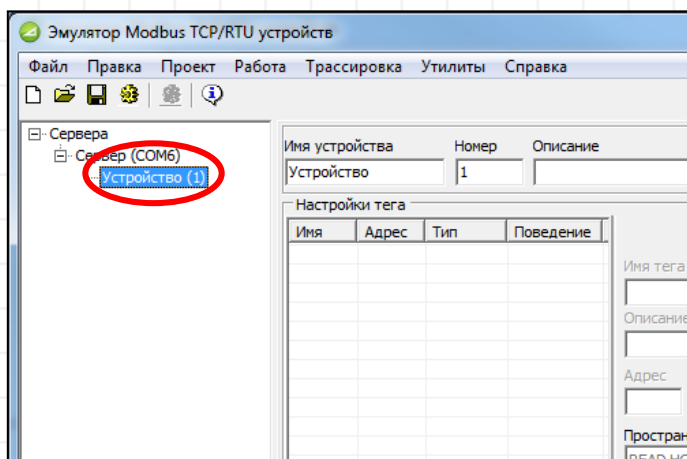
4. Кликните правой клавишей мыши на *Сервер (COM...)* и выберите *Добавить устройство*



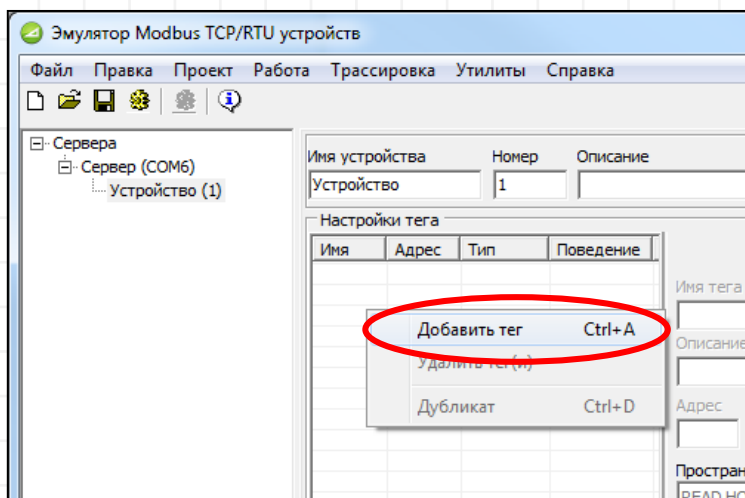
5. Задайте имя устройства, номер устройства (это UnitID в ПЛК) и прочие настройки



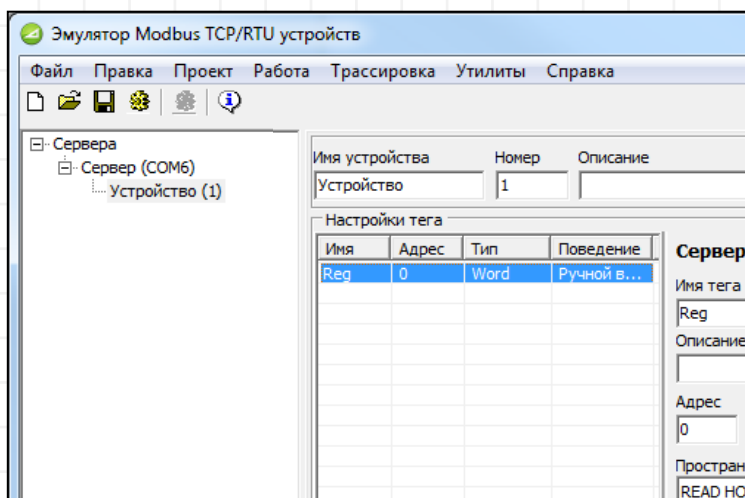
6. Дважды кликните по устройству



7. Кликните правой и выберите Добавить тег



8. Результат добавление тега



9. В следующих двух пунктах формируется список из 8 переменных и для каждой переменной задан:

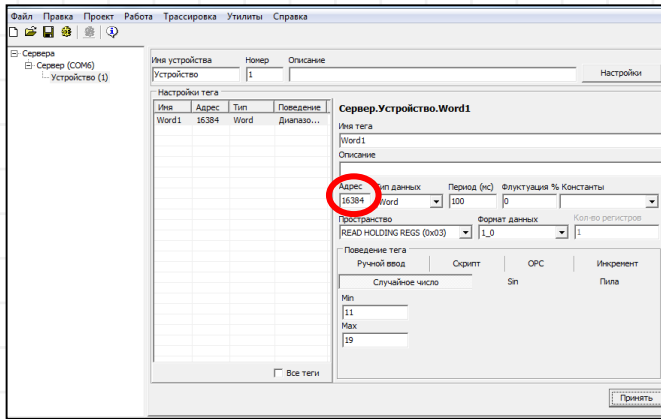
Адрес

Тип данных

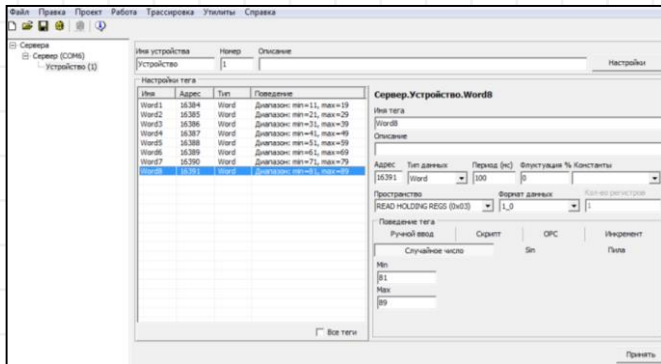
Пространство (0x03)

Поведение тега – случайное число в пределах устанавливаемых минимального (Min) и максимального (Max) значений

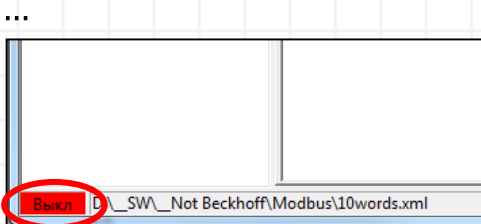
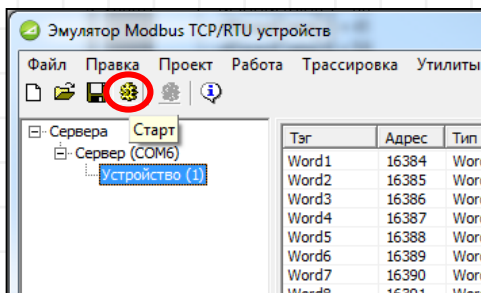
10. Настройка параметров добавленного тега



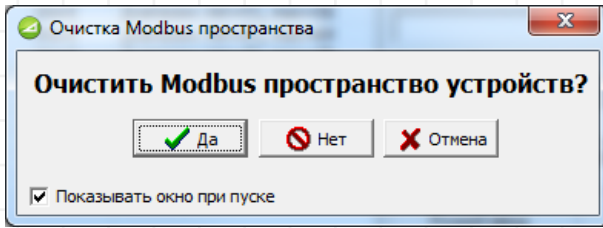
11. Список сформированных тегов



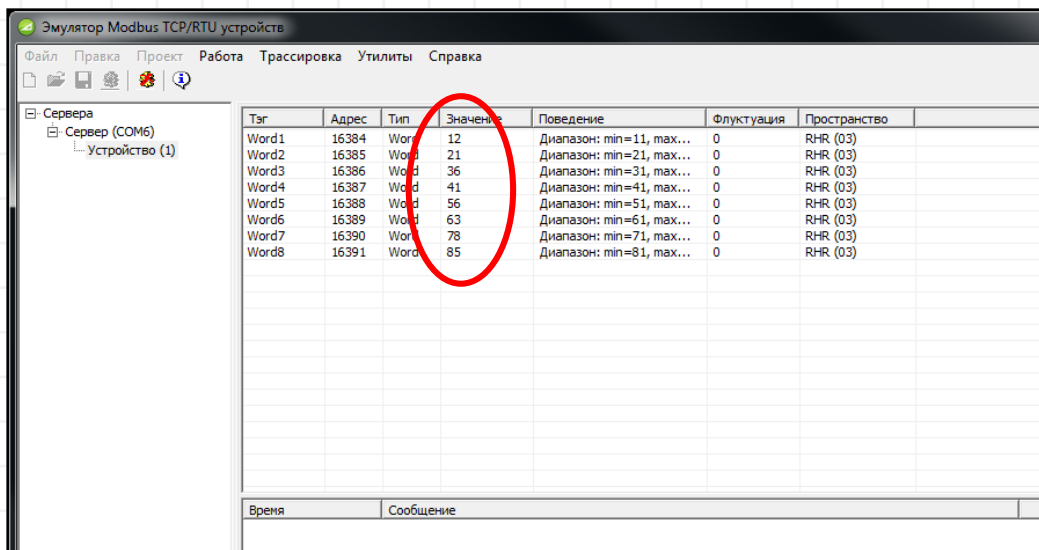
12. Запуск Сервера (эмулятора)



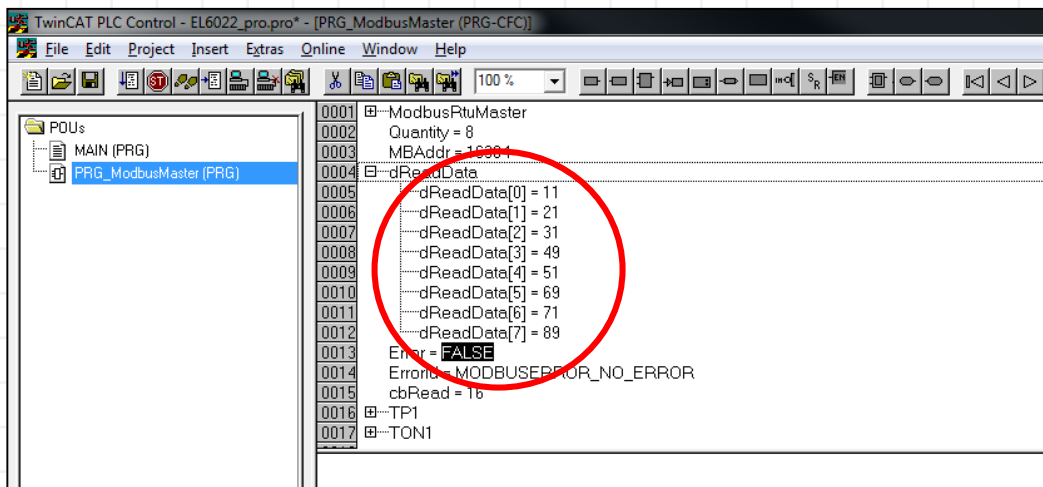
13. Очистка адресного пространства



14. Проверка изменения значений регистров в эмуляторе (сервер)



15. Проверка изменения значений регистров в ПЛК (клиент)



7. Дополнительная информация

В описании адресного пространства Modbus, например RTU, среди прочего, приведены таблицы адресов. Они разъясняют, что:

1. Для доступа к различной памяти и в зависимости от способа обращения (побитовое, порегистровое) используются различные номера функций
2. Существует понятие "смещение" адреса. Величина смещения может отличаться от указанных для тех контроллеров, у которых Modbus уже "на борту".
3. Система адресации переменных (Base 0, Base 1) разных SCADA для работы с одной и той же областью памяти ПЛК может отличаться.

Inputs в ПЛК (смещение 0)

2 : Read input status

4 : Read Input Registers

PLC variable	Access type	Address in the Modbus telegram, Base 0	Address in the end device (device-dependent), Base 1
Inputs[0]	Word	16#0 указание адреса	30001 (30000 + адрес hex + 1)
Inputs[1]	Word	16#1 напрямую	30002 (30000 + адрес hex + 1)
Inputs[0], Bit 0	Bit	16#0 как он есть	10001 (10000 + адрес hex + 1)
Inputs[1], Bit 14	Bit	16#1E в формате hex	1001F (10000 + адрес hex + 1)

Outputs в ПЛК (смещение 16#800)

1 : Read Coil Status
 3 : Read Holding Registers
 5 : Force Single Coil
 6 : Preset Single Register
 15 : Force Multiple Coils
 16 : Preset Multiple Registers

PLC variable	Access type	Address in the Modbus telegram, Base 0	Address in the end device (device-dependant), Base 1
Outputs[0]	Word	16#800 адрес	40801 (40000 + адрес ¹⁾ hex + 1)
Outputs[1]	Word	16#801 в формате hex	40802 (40000 + адрес ¹⁾ hex + 1)
Outputs[0], Bit 0	Bit	16#800 +	00801 (00000 + адрес ¹⁾ hex + 1)
Outputs[1], Bit 14	Bit	16#81E смещение	0081F (00000 + адрес ¹⁾ hex + 1)

Memory в ПЛК (смещение 16#4000)

3 : Read Holding Registers
 6 : Preset Single Register
 16 : Preset Multiple Registers

PLC variable	Access type	Address in the Modbus telegram, Base 0	Address in the end device (device-dependant), Base 1
Memory[0]	Word	16#4000 адрес	44001 (40000 + адрес ¹⁾ hex + 1)
Memory[1]	Word	16#4001 в формате hex	44002 (40000 + адрес ¹⁾ hex + 1)
Memory[10]	Word	16#400A +	4400B (40000 + адрес ¹⁾ hex + 1)
Memory[31]	Word	16#401F смещение	44020 (40000 + адрес ¹⁾ hex + 1)

¹⁾ - с учетом смещения

TS6250 TwinCAT Modbus TCP Server



ModbusTools



**TS6250-0030
TwinCAT Modbus TCP Server CE**



**Эмулятор Modbus RTU/TCP
Slave устройств**



**TS6255
TwinCAT PLC Lib: Modbus RTU**



Данная инструкция является частью общего документа "Практическое пособие по программированию контроллеров Beckhoff", электронную версию которого можно скачать с сайта по этому QR-коду

