

# Программное обеспечение SystemePLC Studio

для интеллектуальных реле SystemePLC SR  
и ПЛК SystemePLC S172

Быстрый старт. Часть 2.  
Сетевые коммуникации



## Содержание

Сетевые коммуникации.....	3
Протокол Modbus.....	3
Области данных Modbus.....	4
Поддержка кодов функций Modbus.....	5
Коммуникации Modbus RTU.....	6
Подключение ПЛК S172 по Modbus RTU.....	6
Конфигурация Modbus RTU Slave.....	7
Конфигурация Modbus RTU Master.....	10
Запись INT переменной в области 4х.....	13
Чтение INT переменной из области 3х.....	15
Чтение REAL переменной из области 4х.....	16
Запись BOOL переменной из области 0х.....	17
Чтение BOOL переменной из области 1х.....	19
Логика работы демонстрационных проектов Modbus RTU.....	21
Коммуникации Modbus TCP.....	24
Подключение ПЛК S172 по Modbus TCP.....	24
Настройка параметров IP.....	25
Конфигурация Modbus TCP Server.....	26
Конфигурация Modbus TCP Client.....	30
Запись INT переменной в области 4х.....	33
Чтение INT переменной из области 3х.....	35
Чтение REAL переменной из области 4х.....	36
Запись BOOL переменной из области 0х.....	38
Чтение BOOL переменной из области 1х.....	39
Логика работы демонстрационных проектов Modbus TCP.....	42
Используемые сокращения и термины.....	44

# Сетевые коммуникации

## Протокол Modbus

**Modbus** — это открытый коммуникационный протокол для организации связи между промышленными устройствами.

Работает по архитектуре **ведущий – ведомый** (клиент-сервер, master-slave).

Ведущее устройство (master или клиент) отправляет запрос на чтение или запись, ведомое устройство (slave или сервер), которому адресован запрос, проверяет корректность запроса и отвечает либо выполнением запроса (чтение/ запись значений), либо кодом ошибки.

### Поддержка Modbus в SystemePLC S172

SystemePLC S172 поддерживает следующие варианты работы с протоколом Modbus:

- Modbus RTU через два порта RS-485
- Modbus TCP через порт Ethernet RJ-45

### Режимы работы

Контроллер может функционировать в двух режимах:

- Master/Client (ведущий) — инициация обмена данными с другими ведомыми устройствами
- Slave/Server (ведомый) — ответ на запросы от ведущего

### Доступные интерфейсы:

- 2 порта RS-485 для Modbus RTU
- 1 порт Ethernet RJ-45 для Modbus TCP

### Важные характеристики:

- Все коммуникационные порты могут работать как в режиме Master, так и Slave
- Возможность одновременной работы всех трёх портов
- Поддержка стандартных функций Modbus для чтения/записи данных (см. раздел «Поддержка кодов функций Modbus»)

### Базовые функции обмена:

- Чтение состояния дискретных входов (DI)
- Запись состояния дискретных выходов (DO)
- Чтение значений аналоговых входов (AI)
- Запись значений аналоговых выходов (AO)

Для S172 существует ограничение на максимум **2000** модбас-переменных

## Рекомендации по применению

- Для связи с полевыми устройствами рекомендуется использовать Modbus RTU Master через RS-485
- Для интеграции с верхним уровнем АСУТП оптимально применять Modbus TCP Server
- При необходимости работы с несколькими сетями RS-485 можно использовать оба порта RS-485 одновременно

## Области данных Modbus

Модбас использует четыре области данных:

- **Discrete Inputs** — однобитовый тип, только для чтения (*область 1х*)
- **Coils** — однобитовый тип, для чтения и записи (*область 0х*)
- **Input Registers** — 16-битовый тип, только для чтения (*область 3х*)
- **Holding Registers** — 16-битовый тип, для чтения и записи (*область 4х*)

У устройства, к которому необходимо обращаться по сети, данные будут расположены в одной или нескольких указанных областях.

Информация про адресацию данных производитель указывает в документации («Список коммуникационных переменных», «Modbus-коммуникации», «Карта адресов» и т.п.)

Кроме знания адреса нужного параметра необходимо также выяснить, какие **коды функций Modbus (FC)** поддерживает устройство в качестве ведущего и какие функции Modbus поддерживает устройство в качестве ведомого. И ведущее, и ведомое устройства должны иметь поддержку нужного кода функций модбас

### **Обратите внимание!**

Если ведомое устройство поддерживает только определённый код функции Modbus, например, **FC\_06** в качестве ведомого, а ведущее устройство **не имеет поддержки** этой функции – связать эти устройства по Modbus **не получится**

## Поддержка кодов функций Modbus

ПЛК SystemePLC S172 поддерживает следующие функции (FC) Modbus в качестве ведущего устройства:

- FC\_01 (Read Coils)
- FC\_02 (Read Discrete inputs)
- FC\_03 (Read Holding Register)
- FC\_04 (Read Input Registers)
- FC\_05 (Write Single Coil)
- FC\_06 (Write Single Register)
- FC\_15 (Write Multiple Coils)
- FC\_16 (Write Multiple Register)

Код FC DEC	Код FC HEX	Описание	Разрядность
FC_01	0x01	Чтение дискретных выходов (Coils)	1 бит
FC_02	0x02	Чтение дискретных входов (Discrete Inputs)	1 бит
FC_05	0x05	Запись одного дискретного выхода	1 бит
FC_15	0x0F	Запись нескольких дискретных выходов	1 бит
FC_03	0x03	Чтение регистров хранения (Holding Registers)	16 бит
FC_04	0x04	Чтение входных регистров (Input Registers)	16 бит
FC_06	0x06	Запись одного регистра хранения	16 бит
FC_16	0x10	Запись нескольких регистров хранения	16 бит

### **Обратите внимание!**

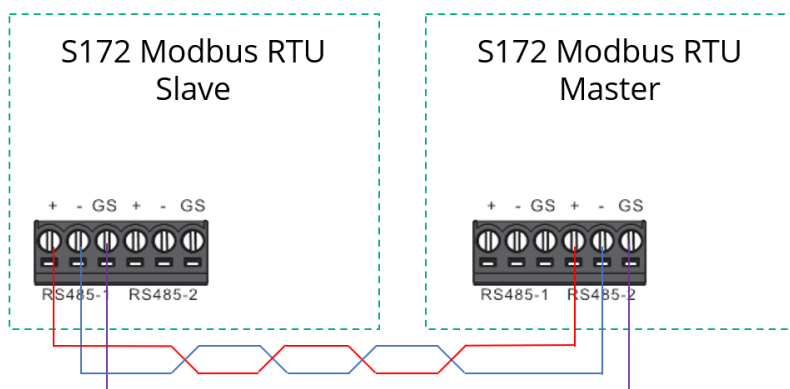
В документации может быть использована как **десятичная (DEC)**, так и **шестнадцатеричная (HEX)** системы счисления для указания адресов – будьте внимательны!

## Коммуникации Modbus RTU

ПЛК SystemePLC S172 поддерживает режим Modbus RTU по обоим портам RS485-1 и RS485-2. По любому порту контроллер может быть ведущим (Master) или ведомым (Slave).

Для демонстрации настройки Modbus RTU будет показано создание проектов для двух ПЛК SystemePLC S172, в котором один ПЛК будет сконфигурирован ведомым (Slave) для порта RS485-1, а второй ПЛК будет сконфигурирован ведущим (Master) для порта RS485-2

### Подключение ПЛК S172 по Modbus RTU



Для предлагаемого демонстрационного проекта нужно соединить контроллеры между собой в соответствии с приведённой схемой.

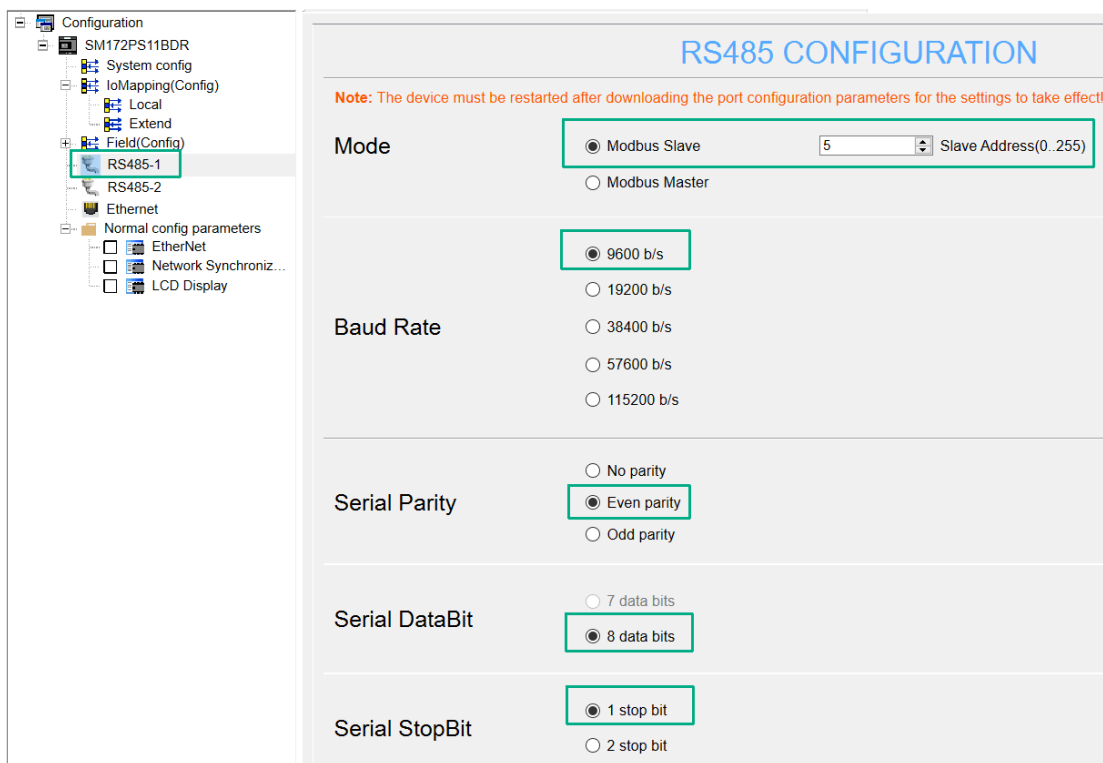
Подключение рекомендуется произвести симметричным кабелем для передачи данных по RS-485 с волновым сопротивлением 120 Ом (КИПЭВ, КИПвЭВ, КИС-П, ИТК RS-485 и аналогичными).

Монтаж связи проводить при отключённом питании

## Конфигурация Modbus RTU Slave

Создайте демонстрационный проект с именем **S172\_MDB\_Slave**.

1. Во вкладке Configuration дважды щёлкните по строчке RS485-1. В открывшемся окне введите параметры, указанные на рисунке:



Mode: Modbus Slave - режим Модбас слэйва (ведомого)

Slave Address: 5 - сетевой адрес: пятый

Baud Rate: 9600 b/s - скорость передачи: 9600 бит/с

Serial Parity: Even parity - проверка чётности: чётный

Serial DataBit: 8 data bit - количество бит: восемь

Serial StopBit: 1 stop bit - количество стоповых бит: один

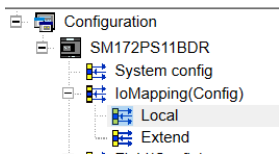
Параметры: скорость передачи, количество бит в посылке, количество стоповых бит для всех устройств данного сегмента Modbus RTU сети должны быть одинаковыми. Адреса ведомых устройств (Slave Address) не должны повторяться и должны находиться в диапазоне 1-247.

После изменения сетевых параметров необходимо **перезапустить ПЛК по питанию**

2. Переменные ПЛК, доступные для доступа в режиме Modbus RTU, могут располагаться в двух разделах:

- IoMapping (Config) – переменные ввода-вывода
- Field (Config) – глобальные сетевые переменные

Для нахождения модбас-адреса нужного входа/выхода нужно открыть в меню Configuration раздел IoMapping(Config) – Local (для локальных) и Extend (для модулей расширения) и посмотреть значение в столбце Modbus Address:



Name	Variable	Type	Option	fault Val	in Val	Max Value	Address	Modbus Address	Description
DI1		BOOL			0	1	%IX0.0	1x0001	DI1 digital input
DI2		BOOL			0	1	%IX0.1	1x0002	DI2 digital input
DI3		BOOL			0	1	%IX0.2	1x0003	DI3 digital input

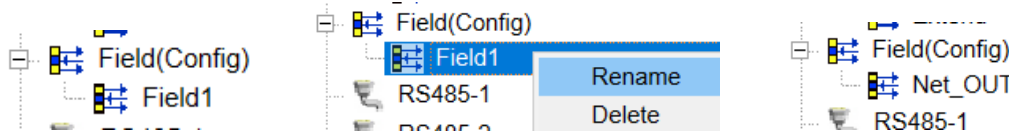
3. Для предоставления данных в режиме Modbus RTU Slave создайте группу глобальных сетевых переменных устройства (Field) и назовите её Net\_OUT.

Для этого:

а) в меню Configuration щёлкните правой кнопкой по Field (Config) и выберите Add Field



б) Переименуйте созданную группу. Щёлкните правой кнопкой по Field1 и выберите Rename, введите Net\_OUT:



4. Добавьте глобальные сетевые переменные слэйва

- Откройте группу переменных устройства Net\_OUT в меню Configuration – Field (Config)
- Нажмите кнопку +Add (Добавить)
- В поле Name введите имя переменной nVar1
- В поле Data Type введите тип INT
- В поле Register Type введите Holding

В меню Configuration – Field(Config) поле Register Type определяет тип доступа переменной в ведомом устройстве.

Если нужно только чтение переменной другим ведущим устройством, то следует выбрать Input.

Если нужны чтение/запись переменной другим ведущим устройством, то следует выбрать Holding:

Name	Data Type	Register Type	Init Value	Address	Modbus Address
0 nVar1	INT	Holding		%QW11.0	4x1001
1 nVar2	INT	Input		%IW11.0	3x1001
2 nVar3	REAL	Holding		%QD11.1	4x1003
3 nVar4	BOOL	Holding		%QX10.0	0x1001
4 nVar5	BOOL	Input		%IX10.0	1x1001

5. Добавьте переменные nVar2-nVar5 в соответствии с рисунком

В зависимости от типа переменной и типа регистра система автоматически выбирает требуемую Modbus-область и подставляет свободный адрес.

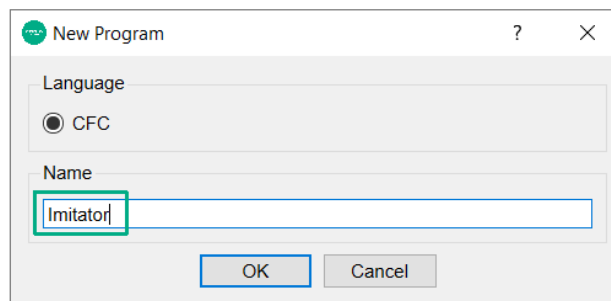
Именно эти адреса («карту адресов») нужно использовать для сетевого обмена с другим ведущим устройством – например, панелью оператора, SCADA или другим ПЛК.

6. Создайте программу Imitator для тестирования сетевого обмена

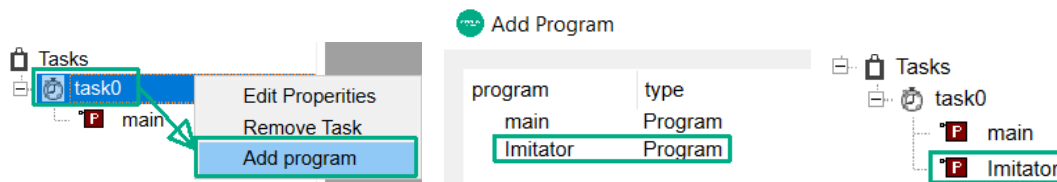
a) В меню Programming создайте новую программу, ПКМ по названию проекта New program



b) Введите имя Imitator и нажмите OK

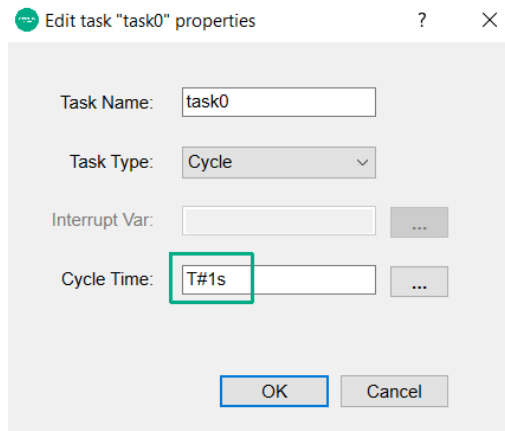


c) Добавьте программу Imitator в задачу task0:  
 Меню Programming – Tasks – ПКМ по task0 – Add Program

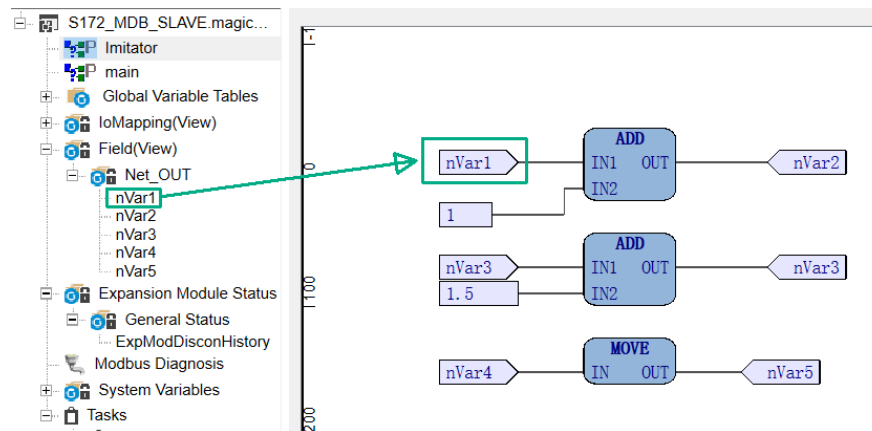


d) Измените период задачи task0 на T#1s:

Tasks – ПКМ по task0 – Edit properties – Cycle Time измените на T#1s



e) Добавьте код, изображённый на рисунке. Для добавления глобальных сетевых переменных перетащите их из группы Net\_OUT на рабочую область:



7. Скомпилируйте проект, исправьте ошибки при наличии, сохраните и загрузите в первый ПЛК.

## Конфигурация Modbus RTU Master

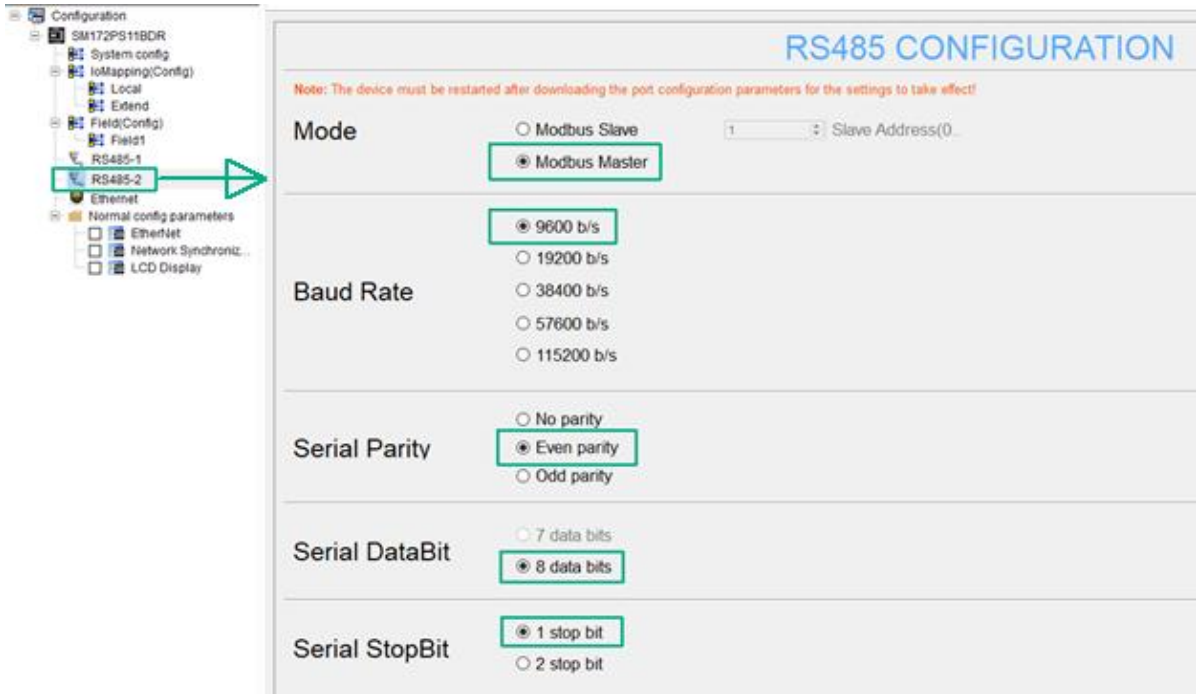
В этом режиме ПЛК сам инициирует запросы ведомым устройствам.

Создайте демонстрационный проект с именем **S172\_MDB\_Master**.

1. Настройте режим ведущего для порта RS485-2

a) Для этого щёлкните меню Configuration – RS485-2. Отметьте режим: Modbus Master.

b) Введите параметры, указанные на рисунке:



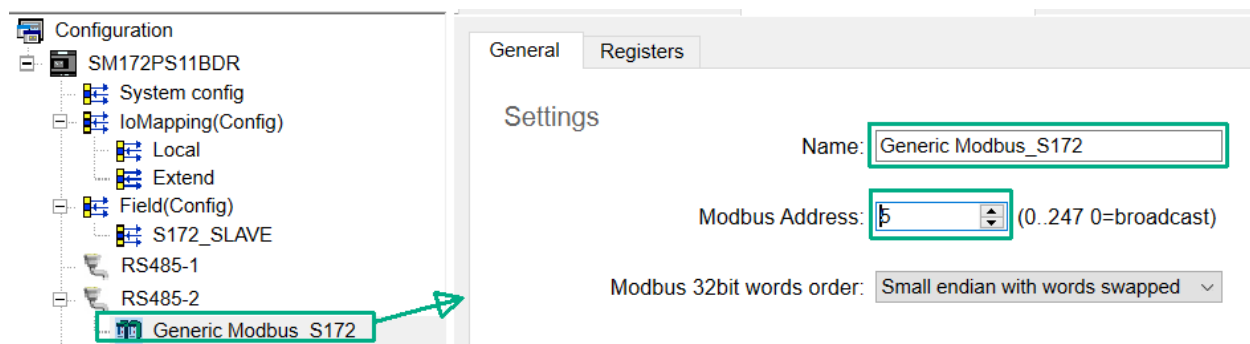
- Baud Rate: 9600 b/s - скорость передачи: 9600 бит/с
- Serial Parity: Even parity - проверка чётности: чётный
- Serial DataBit: 8 data bit - количество бит в посылке: восемь
- Serial StopBit: 1 stop bit - количество стоповых бит: один

После изменения сетевых параметров необходимо **перезапустить ПЛК по питанию**

## 2. Добавьте ведомое устройство

а) В меню Configuration – RS485-2 – ПКМ – Add Generic Modbus Device. Щёлкните по добавленному Generic Modbus Device и на вкладке General введите:

- Name: **Generic Modbus\_S172** - имя ведомого устройства
- Modbus Address: **5** - адрес ведомого устройства
- Modbus 32bit words order: - оставьте без изменений (Small endian with words swapped)



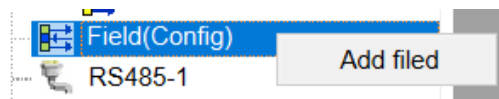
b) Определение последовательности байт в 32-битных переменных (Modbus 32-bit word order)

Возьмем для примера последовательность байт в 32-битной переменной перед обменом ABCD, где A — это старший байт, а D — это младший байт

Пункт	Порядок байт после обмена
Small endian with words swapped	CDAB
Small endian NOT words swapped	DCBA
Big endian with words swapped	BACD
Big endian NOT words swapped	ABCD

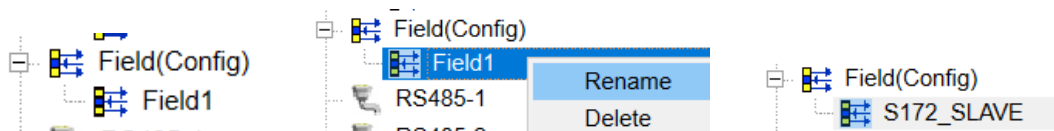
3. Для обмена с ведомым ПЛК S172 создайте группу переменных устройства (Field) и назовите её Slave\_S172.

a) В меню Configuration ПКМ по Field(Config) и выберите Add Field



b) Появившийся пункт Field1 переименуйте:

ПКМ по Field1 и выберите Rename, введите Slave\_S172:



4. Добавьте глобальные сетевые переменные для мастера

a) В меню Configuration – Field (Config) откройте группу переменных устройства Slave\_S172

b) Нажмите кнопку +Add (Добавить)

c) В поле Name введите имя переменной: nVar1

d) В поле Data Type введите тип: INT

e) В поле Register Type введите: Holding

f) В поле Init Value введите при необходимости значение, которое будет присвоено переменной при старте контроллера

g) В поле Retain отметьте при необходимости «галочку» (чекбокс) для сохранения значения переменной при отключении контроллера

h) Заполните далее таблицу в соответствии с рисунком

	Name	Data Type	Register Type	Init Value	Address	Modbus Address	Retain	Description
0	nVar1	INT	Holding		%QW11.1	4x1002	<input type="checkbox"/>	
1	nVar2	INT	Input		%IW11.0	3x1001	<input type="checkbox"/>	
2	nVar3	REAL	Input		%ID11.1	3x1003	<input type="checkbox"/>	
3	nVar4	BOOL	Holding		%QX10.0	0x1001	<input type="checkbox"/>	
4	nVar5	BOOL	Input		%IX10.0	1x1001	<input type="checkbox"/>	

## 5. Определение типа доступа к переменным из областей 0x и 4x

	Name	Data Type	Register Type
0	nVar1	INT	Holding
1	nVar2	INT	Input
2	nVar3	REAL	Holding
3	nVar4	BOOL	Holding
4	nVar5	BOOL	Input

*Переменные в ведомом ПЛК*

	Name	Data Type	Register Type
0	nVar1	INT	Holding
1	nVar2	INT	Input
2	nVar3	REAL	Input
3	nVar4	BOOL	Holding
4	nVar5	BOOL	Input

*Переменные в ведущем ПЛК*

Обратите внимание на переменную **nVar3** типа REAL, расположенную в области 4x в ведомом устройстве (Holding). Ведущее устройство может как читать, так и записывать эту переменную. Для того, чтобы **прочитать** nVar3 из ведомого устройства, тип доступа в ведущем устройстве нужно определить как **Input**, несмотря на то что в ведомом она определена как **Holding**. То же самое касается переменных типа BOOL, определённых в ведомом устройстве, как Holding

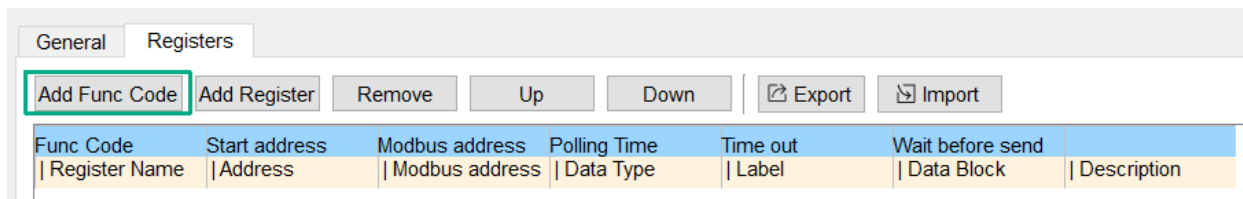
## 6. Конфигурация Modbus-запросов Modbus RTU Master

- Перейдите в меню Configuration – RS485-2 - Generic Modbus\_S172
- Откройте вкладку Registers добавленного устройства для конфигурации чтения/записи регистров ведомого устройства

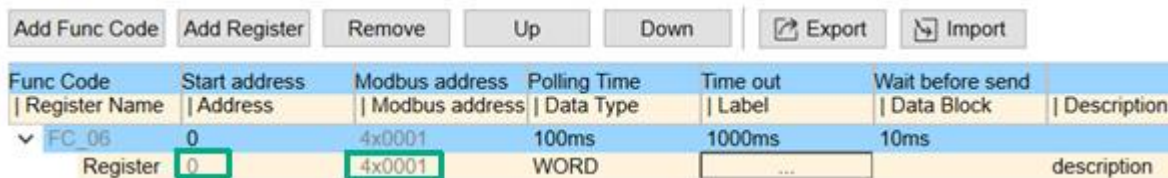
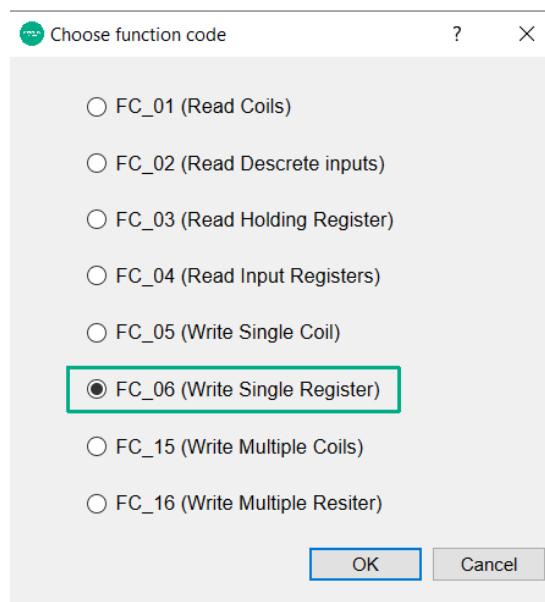
### Запись INT переменной в области 4x

nVar1	INT	Holding	4x1001	Запись
-------	-----	---------	--------	--------

- На вкладке Registers создайте новый запрос модбас, для чего щёлкните по кнопке Add Func Code:



b) Выберите FC\_06 и нажмите ОК



с) В строке кода функции модбас в ячейке Start Address нужно указать смещение начального адреса относительно первого регистра области. Если указан начальный адрес области 4x - 4x0001, то смещение (Start Address) относительно начального адреса будет:

$$4x0001 - 4x0001 = 0$$

Для записи адреса 4x1001 переменной nVar1 в ячейку Start Address строки кода функции модбас FC\_06 введите смещение от адреса 4x0001:

$$4x1001 - 4x0001 = 1000$$

Func Code	Start address	Modbus address
Register Name	Address	Modbus address
FC_06	1000	4x1001
Register	1000	4x1001

d) Далее настройте:

- Период опроса (Polling Time): 500ms
- Таймаут (Time out): 3000ms
- Паузу перед посылкой запроса (Wait before send): 10ms

Func Code	Start address	Modbus address	Polling Time	Time out	Wait before send
Register Name	Address	Modbus address	Data Type	Label	Data Block
FC_06	1000	4x1001	500ms	3000ms	10ms
Register	1000	4x1001	WORD	...	

- e) Щёлкните дважды по ячейке в столбце Register Name и переименуйте Register в nVar1
- f) В столбце Data Type выберите тип переменной – INT
- g) В следующем столбце щёлкните по ячейке «...» и в открывшемся окне выберите переменную для связывания nVar1 из группы переменных устройства Slave\_S172 и нажмите OK

Func Code	Start address	Modbus address	Polling Time	Time out	Wait before send	Description
Register Name	Address	Modbus address	Data Type	Label	Data Block	Description
FC_06	1000	4x1001	500ms	3000ms	10ms	
nVar1	1000	4x1001	INT	...		W1

Choose PLC variable

Filter:

holding: nVar1 (INT)

CLEAR

h) В столбце Description введите описание переменной: W1

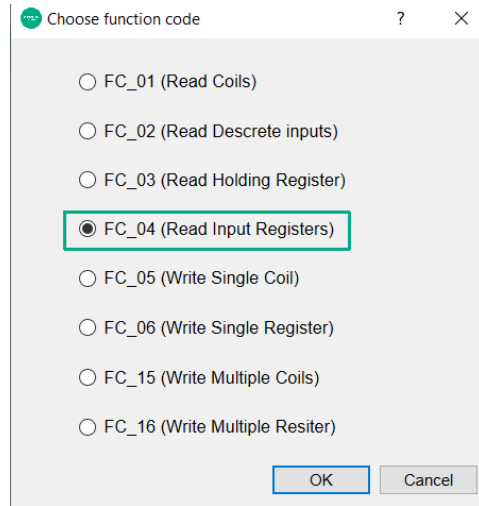
Результат должен выглядеть так:

Func Code	Start address	Modbus address	Polling Time	Time out	Wait before send	Description
Register Name	Address	Modbus address	Data Type	Label	Data Block	Description
FC_06	1000	4x1001	500ms	3000ms	10ms	
nVar1	1000	4x1001	INT	nVar1	%QW11.0	W1

### Чтение INT переменной из области 3x

nVar2	INT	Input	3x1001	Чтение
-------	-----	-------	--------	--------

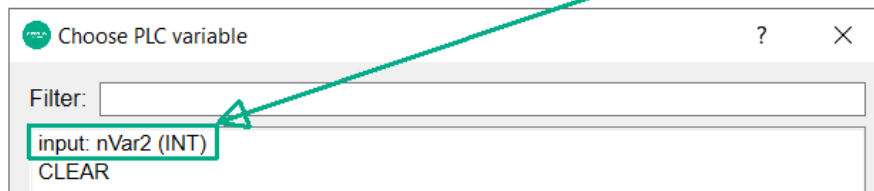
a) На вкладке Registers создайте новый запрос модбас, для чего щёлкните по кнопке Add Func Code



b) Выберите тип переменной – INT

c) Укажите смещение, период опроса и таймаут аналогично предыдущим настройкам

Func Code	Start address	Modbus address	Polling Time	Time out	Wait before send	
Register Name	Address	Modbus address	Data Type	Label	Data Block	Description
FC_06	1000	4x1001	500ms	3000ms	10ms	
nVar1	1000	4x1001	INT	nVar1	%QW11.1	W1
FC_04	1000	3x1001	500ms	3000ms	10ms	
Register	1000	3x1001	INT	...		description



d) Щёлкните дважды по ячейке в столбце Register Name и переименуйте Register в nVar2

e) В описании напишите R1

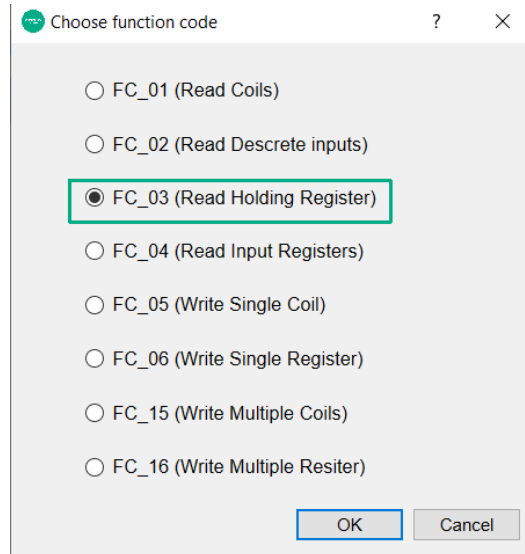
Результат должен выглядеть так:

FC_04	1000	3x1001	500ms	3000ms	10ms	
nVar2	1000	3x1001	INT	nVar2	%IW11.0	R1

## Чтение REAL переменной из области 4x

nVar3	REAL	Holding	4x1003	Чтение
-------	------	---------	--------	--------

a) На вкладке Registers создайте новый запрос модбас, для чего щёлкните по кнопке Add Func Code



b) Выберите тип переменной REAL

c) Укажите смещение (Start Address) Т.к. адрес переменной 4x1003, смещение будет 1002:

$$4x1003 - 4x0001 = 1002$$

FC_03	1002	4x1003	100ms
Register	1002	4x1003	REAL

d) Свяжите с регистром nVar3:

FC_03	1002	4x1003	500ms	3000ms	10ms
Register	1002	4x1003	REAL	...	

Choose PLC variable

Filter:

input: nVar3 (REAL)

CLEAR

e) Период опроса и таймаут аналогично предыдущим настройкам

f) Щёлкните дважды по ячейке в столбце Register Name и переименуйте Register в nVar3

g) В описании напишите R2

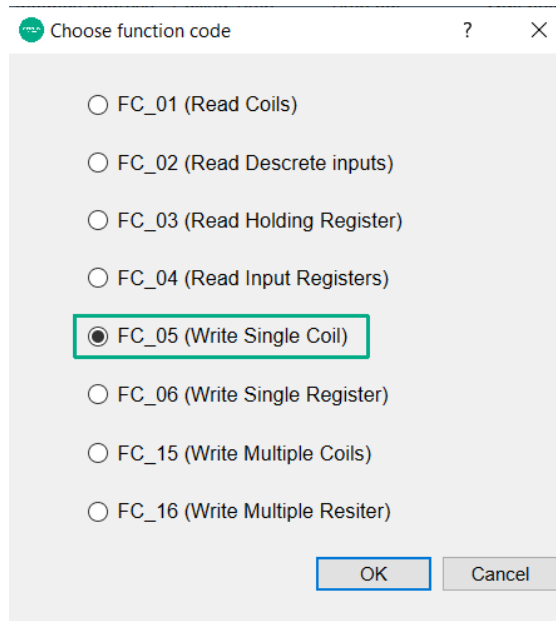
Результат должен выглядеть так:

FC_03	1002	4x1003	500ms	3000ms	10ms
nVar3	1002	4x1003	REAL	nVar3	%ID11.1 R2

Запись BOOL переменной из области 0x

nVar4	BOOL	Holding	0x1001	Запись
-------	------	---------	--------	--------

a) На вкладке Registers создайте новый запрос модбас, для чего щёлкните по кнопке Add Func Code



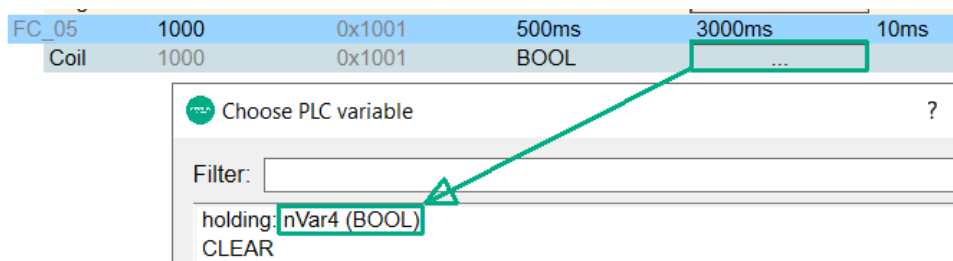
b) Выберите тип переменной – BOOL

c) Укажите смещение (Start Address) Т.к. адрес переменной 0x1001, смещение будет 1000:

$$0x1001 - 0x0001 = 1000$$

FC_05	1000	0x1001
Coil	1000	0x1001

d) Свяжите с регистром nVar4:



e) Щёлкните дважды по ячейке в столбце Register Name и переименуйте Register в nVar4

f) Период опроса и таймаут аналогично предыдущим настройкам

g) В описании напишите W2

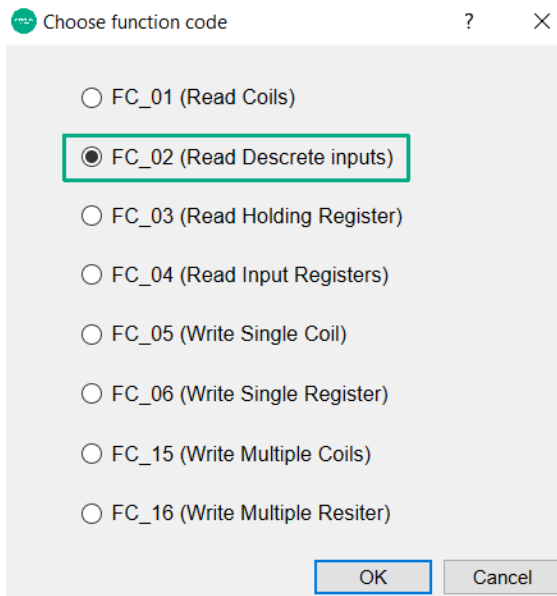
Результат должен выглядеть так:

FC_05	1000	0x1001	500ms	3000ms	10ms		
nVar4	1000	0x1001	BOOL	nVar4	%QX10.0	W2	

### Чтение BOOL переменной из области 1x

nVar5	BOOL	Input	1x1001	Чтение
-------	------	-------	--------	--------

a) На вкладке Registers создайте новый запрос модбас, для чего щёлкните по кнопке Add Func Code



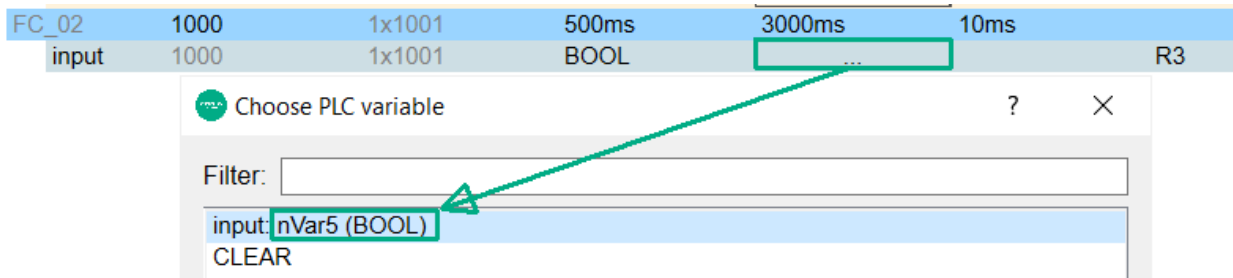
b) Выберите тип переменной – BOOL

c) Укажите смещение (Start Address) Т.к. адрес переменной 1x1001, смещение будет 1000:

$$1x1001 - 1x0001 = 1000$$

FC_02	1000	1x1001
input	1000	1x1001

d) Свяжите с регистром nVar5:



- e) Щёлкните дважды по ячейке в столбце Register Name и переименуйте Register в nVar5
- f) Период опроса и таймаут аналогично предыдущим настройкам
- g) В описании напишите R3

Результат должен выглядеть так:

FC_02	1000	1x1001	500ms	3000ms	10ms	
nVar5	1000	1x1001	BOOL	nVar5	%IX10.0	R3

Если при связывании запроса в окно выбора переменной **пустое** – значит, выбран неправильный тип данных (WORD вместо REAL, например) или неправильный тип доступа (Input для функции записи)

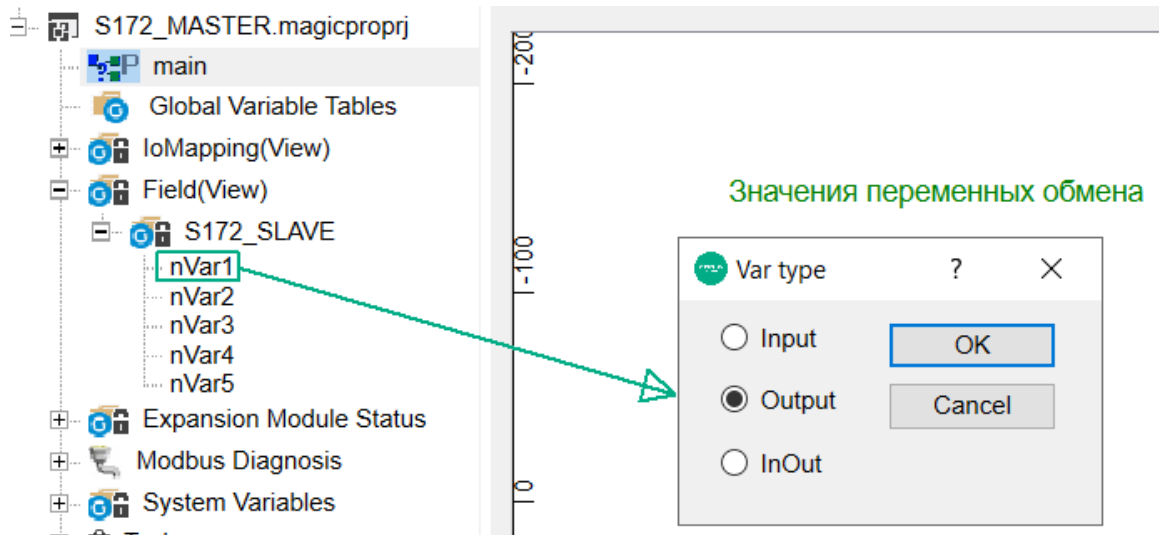
Итоговый результат должен выглядеть так:

Func Code	Start address	Modbus address	Polling Time	Time out	Wait before send	
Register Name	Address	Modbus address	Data Type	Label	Data Block	Description
FC_06	1000	4x1001	500ms	3000ms	10ms	
nVar1	1000	4x1001	INT	nVar1	%QW11.1	W1
FC_04	1000	3x1001	500ms	3000ms	10ms	
nVar2	1000	3x1001	INT	nVar2	%IW11.0	R1
FC_03	1002	4x1003	500ms	3000ms	10ms	
nVar3	1002	4x1003	REAL	nVar3	%ID11.1	R2
FC_05	1000	0x1001	500ms	3000ms	10ms	
nVar4	1000	0x1001	BOOL	nVar4	%QX10.0	W2
FC_02	1000	1x1001	500ms	3000ms	10ms	
nVar5	1000	1x1001	BOOL	nVar5	%IX10.0	R3

7. Если нужно групповым запросом прочитать/записать несколько переменных, адреса которых в слэиве расположены последовательно, выделите строку запроса, например, FC\_03 - Add register – Укажите тип – Свяжите с регистром глобальных сетевых переменных

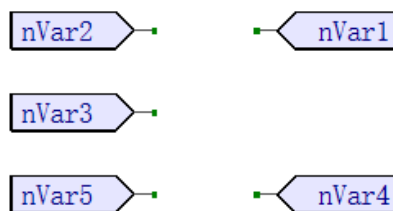
8. Добавьте в программу main глобальные сетевые переменные.

- a) На вкладке Programming откройте программу main и перетащите переменные из Field (View) – S172\_SLAVE на рабочую область программы
- b) В диалоговом окне для переменных nVar1 и nVar4 выберите Var type: Output
- c) В диалоговом окне для переменных nVar2, nVar3, nVar5 выберите Var type: Input



В результате должно получиться:

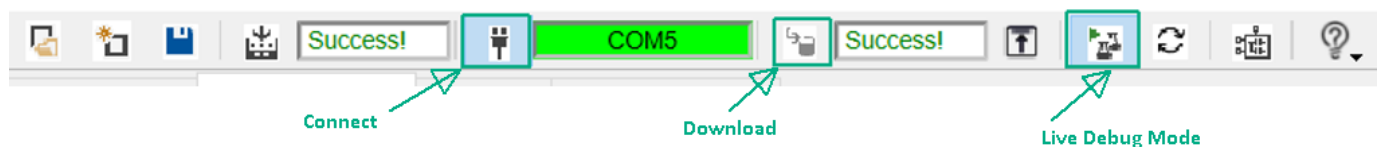
### Значения переменных обмена



9. Скомпилируйте проект, исправьте ошибки при наличии, сохраните и загрузите во второй ПЛК.

10. Соедините сетевым кабелем для RS485 порт RS485-1 контроллера с загруженной программой S172\_MDB\_SLAVE и порт RS485-2 контроллера с загруженной программой S172\_MDB\_MASTER, после чего подайте питание на контроллеры

11. Подключитесь к ПЛК-Master, откройте секцию main, включите режим Live Debug Mode и протестируйте работоспособность программ



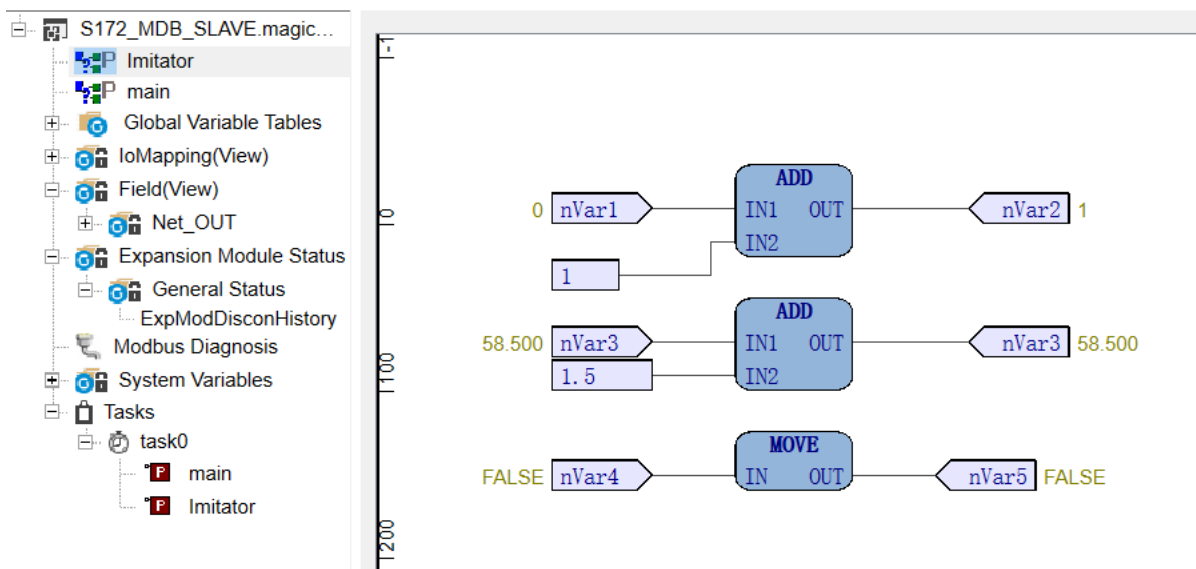
### Логика работы демонстрационных проектов Modbus RTU

Логика программ заключается в следующем:

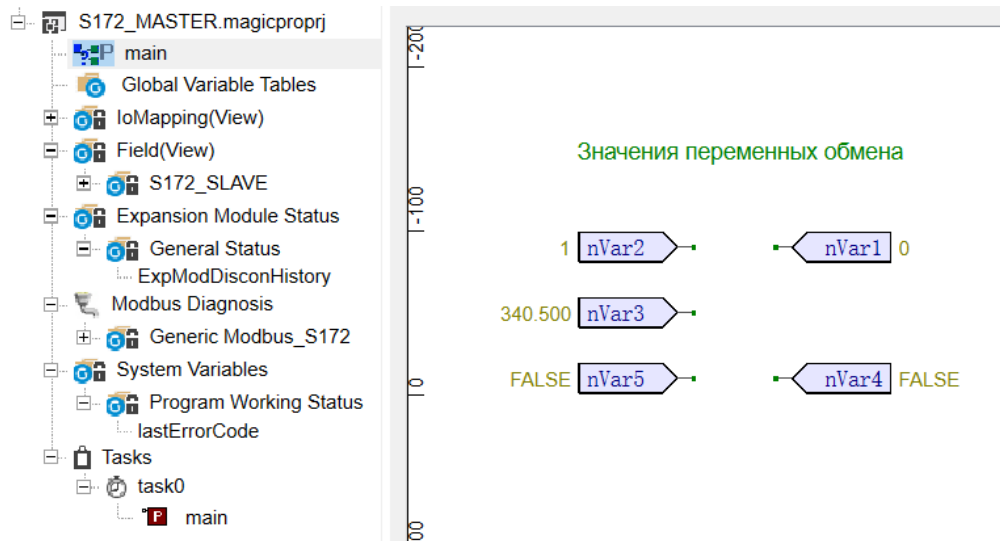
#### 1. Чтение REAL переменной

- а. Раз в секунду в слейве увеличивается значение переменной nVar3 на 1.5

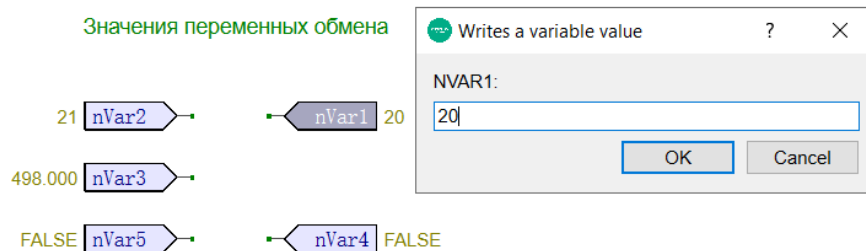
- b. Если связь установлена, и программа слэйва выполняется, то nVar3 в мастере будет каждую секунду увеличиваться на 1.5
2. **Чтение и запись INT переменных**
- a. Значение переменной nVar1 мастер записывает в слэйв
  - b. В слэйве к значению nVar1 прибавляется 1 и новое значение записывается в переменную nVar2
  - c. Мастер читает значение переменной nVar2
  - d. Если связь установлена, и программа слэйва выполняется, то после изменения nVar1 значение nVar2 будет равно nVar1+1
3. **Чтение и запись BOOL переменных**
- a. Значение переменной nVar4 мастер записывает в слэйв
  - b. В слэйве значение nVar4 копируется с помощью блока MOVE в переменную nVar5
  - c. Мастер читает значение переменной nVar5.
  - d. Если связь установлена, и программа слэйва выполняется, то после изменения nVar4 значение nVar5 будет таким же



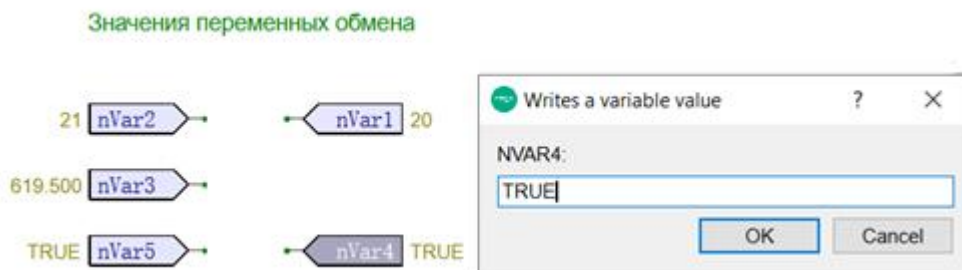
Программа Modbus Slave в режиме онлайн и Live Debug Mode



Программа Modbus Master в режиме онлайн и Live Debug Mode



Щёлкните дважды по переменной nVar1, введите новое значение, ОК



Щёлкните дважды по переменной nVar4, введите новое значение, ОК

## Коммуникации Modbus TCP

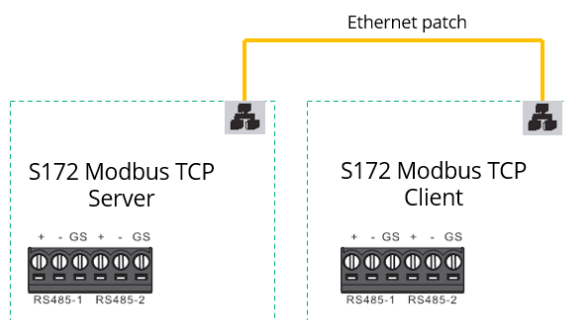
ПЛК SystemePLC S172 поддерживает режим Modbus TCP Server/ Modbus TCP Client через порт Ethernet.

Сконфигурированный Modbus TCP Server позволяет подключить до 8 устройств Modbus TCP Client, а сконфигурированный Modbus TCP Client — до 8 устройств Modbus TCP Server

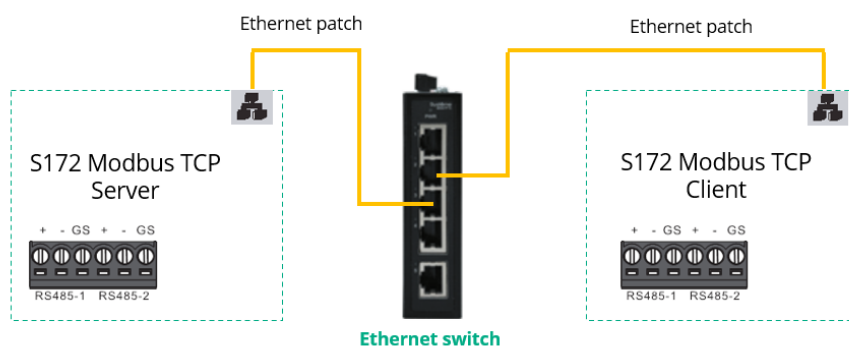
Для демонстрации настройки Modbus TCP будет показано создание проектов для двух ПЛК SystemePLC S172, в котором один ПЛК будет сконфигурирован ведомым (Server), а второй ПЛК будет сконфигурирован ведущим (Client)

### Подключение ПЛК S172 по Modbus TCP

Для предлагаемого демонстрационного проекта нужно соединить контроллеры между собой в соответствии с приведёнными схемами напрямую:



или через Ethernet-switch:



Подключение рекомендуется произвести Ethernet-кабелем для передачи данных категорий Cat5e и Cat6.

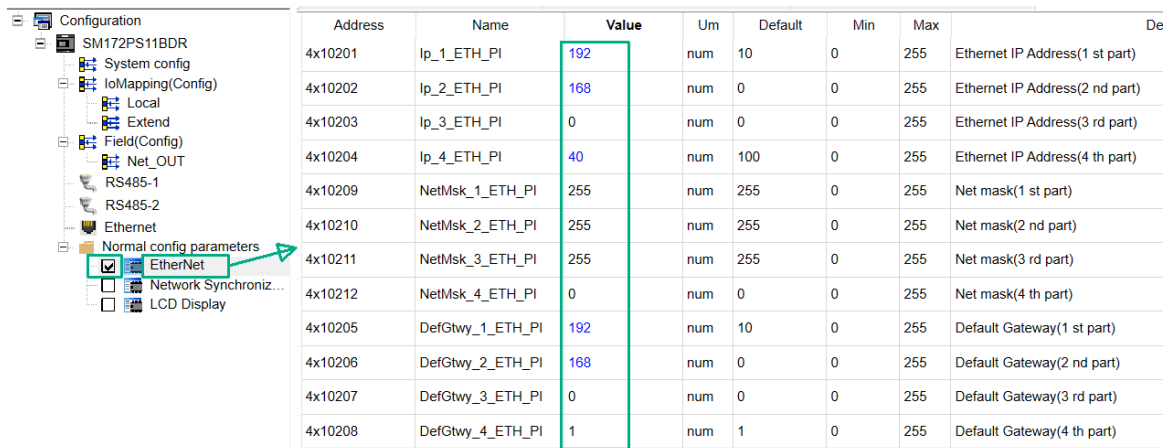
## Настройка параметров IP

Для подключения SystemePLC S172 по Modbus TCP может понадобиться изменить IP-адрес, маску подсети и адрес шлюза.

1. Для изменения IP-адреса через Systeme Studio:

a) Откройте меню Configuration – Normal Config Parameters – Ethernet

b) Установите «галочку» (чекбокс) напротив группы параметров Ethernet



Address	Name	Value	Um	Default	Min	Max	De
4x10201	Ip_1_ETH_PI	192	num	10	0	255	Ethernet IP Address(1 st part)
4x10202	Ip_2_ETH_PI	168	num	0	0	255	Ethernet IP Address(2 nd part)
4x10203	Ip_3_ETH_PI	0	num	0	0	255	Ethernet IP Address(3 rd part)
4x10204	Ip_4_ETH_PI	40	num	100	0	255	Ethernet IP Address(4 th part)
4x10209	NetMsk_1_ETH_PI	255	num	255	0	255	Net mask(1 st part)
4x10210	NetMsk_2_ETH_PI	255	num	255	0	255	Net mask(2 nd part)
4x10211	NetMsk_3_ETH_PI	255	num	255	0	255	Net mask(3 rd part)
4x10212	NetMsk_4_ETH_PI	0	num	0	0	255	Net mask(4 th part)
4x10205	DefGtwy_1_ETH_PI	192	num	10	0	255	Default Gateway(1 st part)
4x10206	DefGtwy_2_ETH_PI	168	num	0	0	255	Default Gateway(2 nd part)
4x10207	DefGtwy_3_ETH_PI	0	num	0	0	255	Default Gateway(3 rd part)
4x10208	DefGtwy_4_ETH_PI	1	num	1	0	255	Default Gateway(4 th part)

c) Установите для S172, который сконфигурирован как сервер (S172\_MDB\_TCP\_SERVER)

IP-адрес сервера (**Ip\_1\_ETH\_PI - Ip\_4\_ETH\_PI**): 192.168.0.40  
Маска подсети (**NetMsk\_1\_ETH\_PI - NetMsk\_4\_ETH\_PI**) 255.255.255.0  
Шлюз (**DefGtwy\_1\_ETH\_PI - DefGtwy\_4\_ETH\_PI**) 192.168.0.1

d) Скомпилируйте и сохраните проект



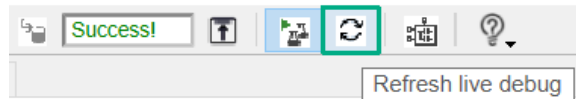
f) После подключения загрузите код в ПЛК, если не был загружен



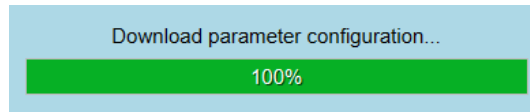
g) Включите Live Debug Mode



h) Включите обновление параметров



ПЛК обновит параметры



i) После обновления параметров нужно отключиться от ПЛК и перезапустить питание, если был подключён USB – временно отключить.

После перезапуска питания значения IP должны **обновиться**

j) Установите для S172, который сконфигурирован как клиент (S172\_MDB\_TCP\_CLIENT)

IP-адрес сервера ( <b>Ip_1_ETH_PI - Ip_4_ETH_PI</b> ):	192.168.0.41
Маска подсети ( <b>NetMsk_1_ETH_PI - NetMsk_4_ETH_PI</b> )	255.255.255.0
Шлюз ( <b>DefGtwy_1_ETH_PI - DefGtwy_4_ETH_PI</b> )	192.168.0.1

k) Скомпилируйте и сохраните проект, обновите значения IP как было описано выше

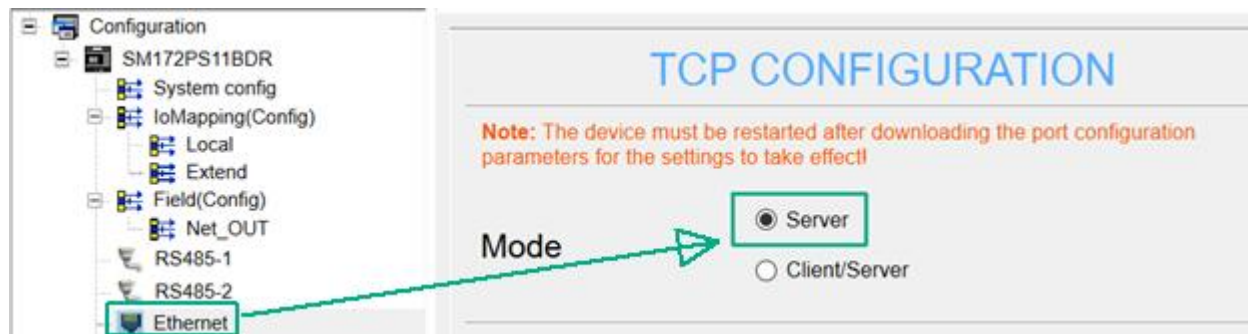
2. Параметры IP-адреса также можно изменить через **экранное меню** ПЛК.

- Для этого нужно на передней панели ПЛК нажать кнопку ОК
  - Кнопками ↑ и ↓ выбрать меню Network на главном экране, нажать кнопку ОК
  - В меню Network кнопками ↑ и ↓ выбрать нужный параметр, например, IP, нажать кнопку ОК для редактирования
  - Кнопками ← и → выбрать цифру для изменения
  - Кнопками ↑ и ↓ изменить значение, нажать кнопку ОК для подтверждения
- Переподключите Ethernet-патч к ПЛК

## Конфигурация Modbus TCP Server

Создайте демонстрационный проект с именем **S172\_MDB\_TCP\_SERVER**

1. Во вкладке Configuration щёлкните по строчке Ethernet. В открывшемся окне включите режим Server



2. После загрузки изменённой конфигурации порта ПЛК должен быть перезапущен для того, чтобы применить новые параметры

3. Переменные ПЛК, доступные для доступа в режиме Modbus Server, могут располагаться в двух разделах:

- IoMapping (Config) – переменные ввода-вывода
- Field (Config) – глобальные сетевые переменные

Для нахождения модбас-адреса нужного входа/выхода нужно открыть в меню Configuration раздел IoMapping(Config) – Local (для локальных) и Extend (для модулей расширения) и посмотреть значение в столбце Modbus Address:

Name	Variable	Type	Option	fault Val	in Val	Max Value	Address	Modbus Address	Description
DI1		BOOL			0	1	%IX0.0	1x0001	DI1 digital input
DI2		BOOL			0	1	%IX0.1	1x0002	DI2 digital input
DI3		BOOL			0	1	%IX0.2	1x0003	DI3 digital input

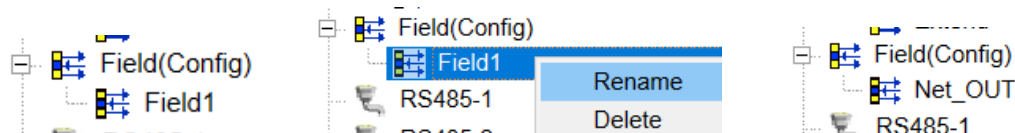
4. Для предоставления данных в режиме Modbus TCP Server создайте группу глобальных сетевых переменных устройства (Field) и назовите её Net\_OUT.

Для этого:

- а) в меню Configuration щёлкните правой кнопкой по Field (Config) и выберите Add Field



- б) Переименуйте созданную группу. Щёлкните правой кнопкой по Field1 и выберите Rename, введите Net\_OUT:



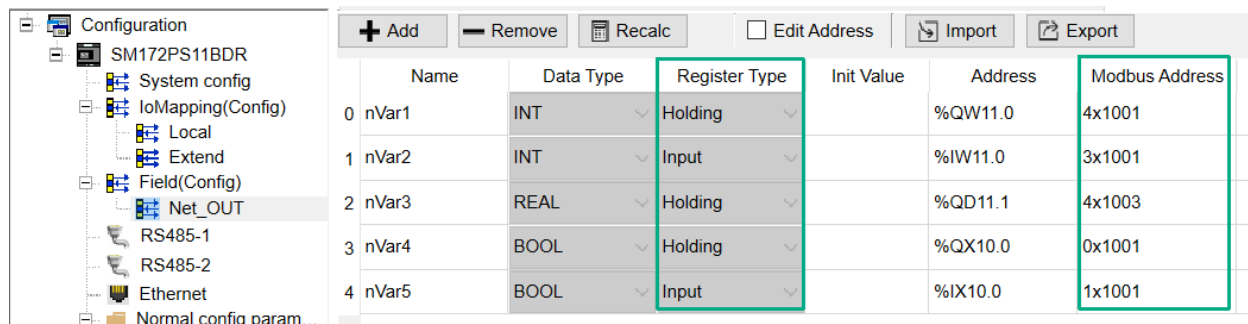
## 5. Добавьте глобальные сетевые переменные сервера:

- Откройте группу переменных устройства Net\_OUT в меню Configuration – Field (Config)
- Нажмите кнопку +Add (Добавить)
- В поле Name введите имя переменной nVar1
- В поле Data Type введите тип INT
- В поле Register Type введите Holding

В меню Configuration – Field(Config) поле Register Type определяет тип доступа переменной в ведомом устройстве.

Если нужно только чтение переменной другим ведущим устройством, то следует выбрать Input.

Если нужны чтение/запись переменной другим ведущим устройством, то следует выбрать Holding:



	Name	Data Type	Register Type	Init Value	Address	Modbus Address
0	nVar1	INT	Holding		%QW11.0	4x1001
1	nVar2	INT	Input		%IW11.0	3x1001
2	nVar3	REAL	Holding		%QD11.1	4x1003
3	nVar4	BOOL	Holding		%QX10.0	0x1001
4	nVar5	BOOL	Input		%IX10.0	1x1001

## 6. Добавьте переменные nVar2-nVar5 в соответствие с рисунком

В зависимости от типа переменной и типа регистра система автоматически выбирает требуемую Modbus-область и подставляет свободный адрес.

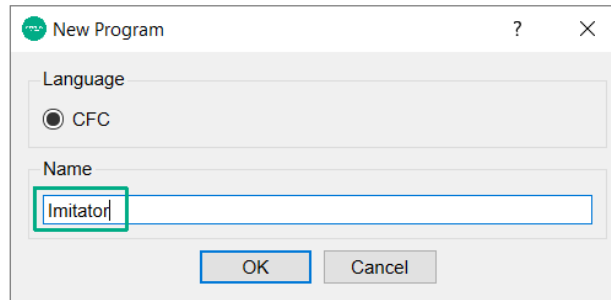
Именно эти адреса («карту адресов») нужно использовать для сетевого обмена с другим ведущим устройством – например, панелью оператора, SCADA или другим ПЛК.

## 7. Создайте программу Imitator для тестирования сетевого обмена

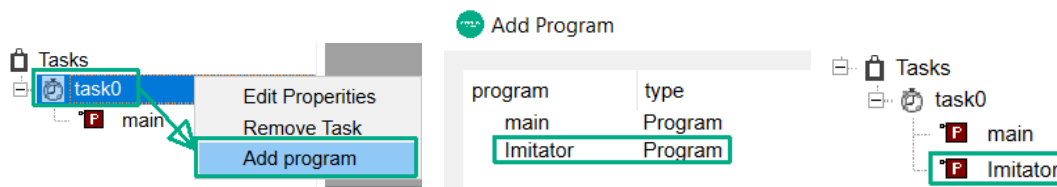
- В меню Programming создайте новую программу, ПКМ по названию проекта New program



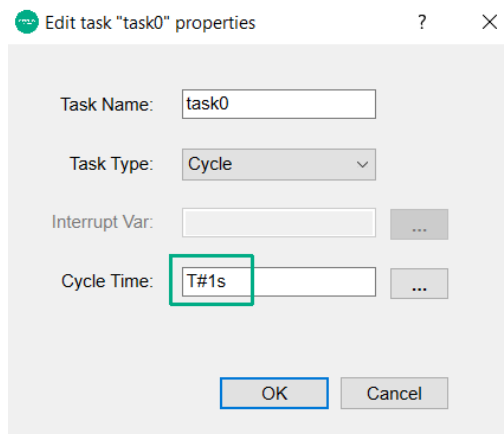
b) Введите имя Imitator и нажмите OK



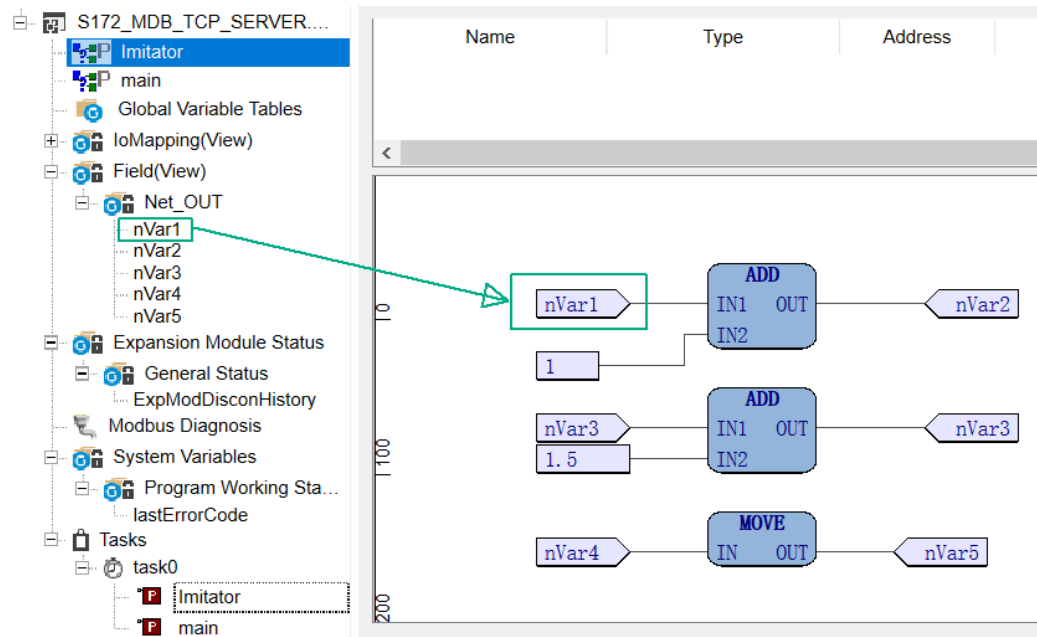
c) Добавьте программу Imitator в задачу task0:  
Меню Programming – Tasks – ПКМ по task0 – Add Program



d) Измените период задачи task0 на T#1s:  
Tasks – ПКМ по task0 – Edit properties – Cycle Time измените на T#1s



e) Добавьте код, изображённый на рисунке. Для добавления глобальных сетевых переменных перетащите их из группы Net\_OUT на рабочую область:



8. Скомпилируйте проект, исправьте ошибки при наличии, сохраните и загрузите в первый ПЛК.

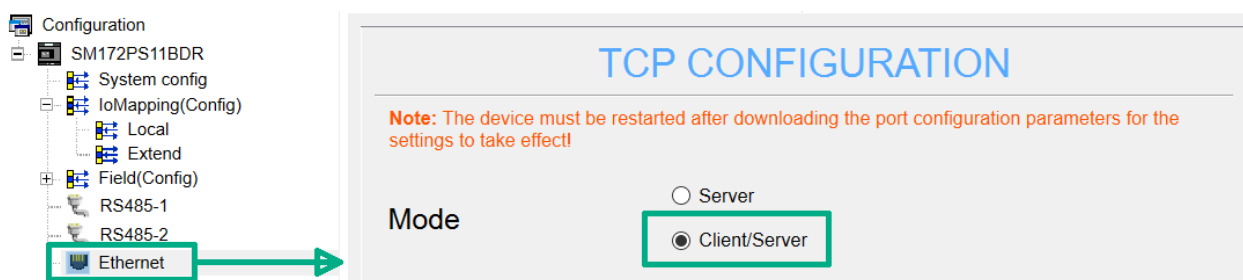
## Конфигурация Modbus TCP Client

В этом режиме ПЛК сам инициирует запросы ведомым устройствам, подключённым по Modbus TCP

Создайте демонстрационный проект с именем **S172\_MDB\_TCP\_CLIENT**

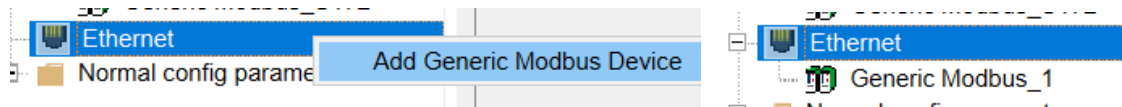
1. Включите режим Modbus TCP Client

а) Во вкладке Configuration щёлкните по строчке Ethernet. В открывшемся окне включите режим Client/Server

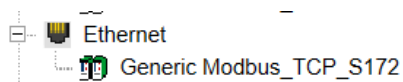


2. Добавьте ведомое устройство

а) В меню Configuration – ПКМ по Ethernet -- Add Generic Modbus Device. Добавилось устройство Generic\_Modbus\_1

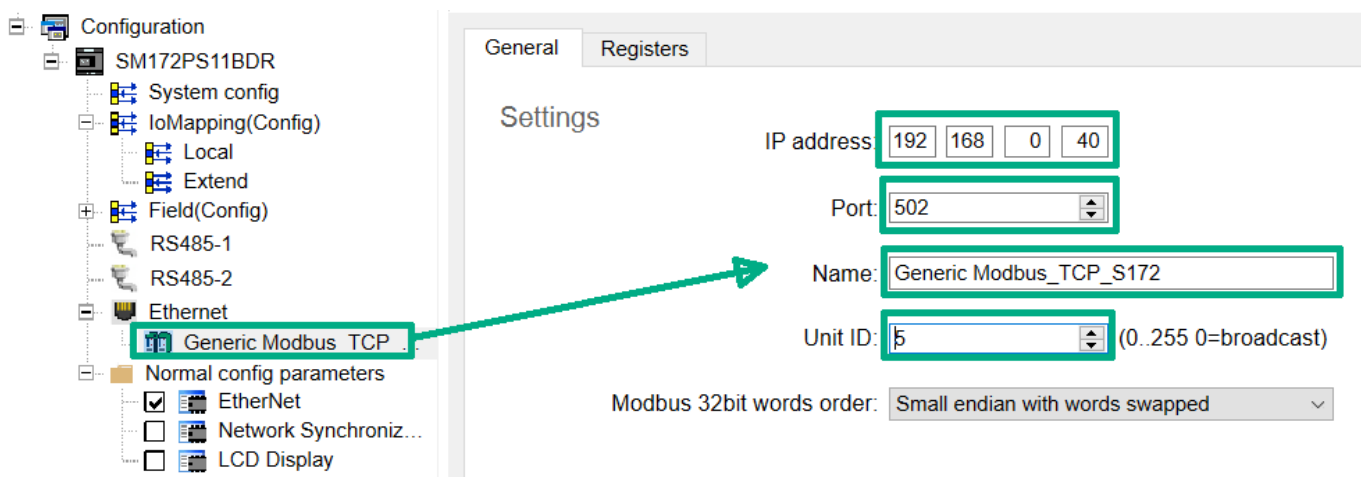


b) Переименуйте устройство в **Generic Modbus\_TCP\_S172**



c) Щёлкните по добавленному устройству и сконфигурируйте:

- IP address: 192.168.0.40                   - IP адрес ведомого устройства
- Port: 502                                   - Порт Modbus
- Name: Generic Modbus\_TCP\_S172       - Имя для идентификации ведомого
- Unit ID: 5                                 - Modbus адрес устройства
- Modbus 32bit words order             - Оставьте без изменений



### 3. Настройка параметров IP

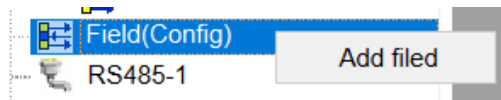
При необходимости настройте для ПЛК IP-адрес, маску подсети и шлюз. Процедура описана в разделе «Настройка параметров IP»

Установите для S172, который сконфигурирован как **клиент**:

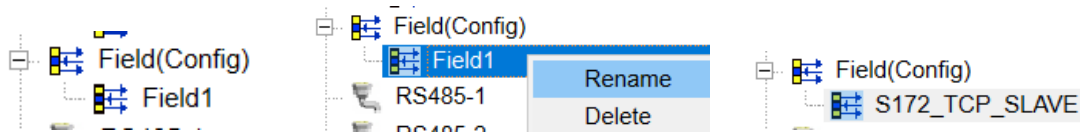
- IP-адрес сервера (**Ip\_1\_ETH\_PI - Ip\_4\_ETH\_PI**):                   192.168.0.41
- Маска подсети (**NetMsk\_1\_ETH\_PI - NetMsk\_4\_ETH\_PI**)           255.255.255.0
- Шлюз (**DefGtwy\_1\_ETH\_PI - DefGtwy\_4\_ETH\_PI**)                   192.168.0.1

4. Для обмена с ведомым ПЛК S172 создайте группу переменных устройства (Field) и назовите её S172\_TCP\_SLAVE

a) В меню Configuration ПКМ по Field(Config) и выберите Add Field



- b) Появившийся пункт Field1 переименуйте:  
ПКМ по Field1 и выберите Rename, введите S172\_TCP\_SLAVE:



5. Добавьте глобальные сетевые переменные для сервера

a) В меню Configuration – Field (Config) откройте группу переменных устройства S172\_TCP\_SLAVE

b) Нажмите кнопку +Add (Добавить)

c) В поле Name введите имя переменной: nVar1

d) В поле Data Type введите тип: INT

e) В поле Register Type введите: Holding

f) В поле Init Value введите при необходимости значение, которое будет присвоено переменной при старте контроллера

g) В поле Retain отметьте при необходимости «галочку» (чекбокс) для сохранения значения переменной при отключении контроллера

h) Заполните далее таблицу в соответствии с рисунком

		<b>+ Add</b>	<b>- Remove</b>	<b>Recalc</b>	<input type="checkbox"/> Edit Address	<b>Import</b>	<b>Export</b>		
	Name	Data Type	Register Type	Init Value	Address	Modbus Address	Retain	Description	
0	nVar1	INT	Holding		%QW11.1	4x1002	<input type="checkbox"/>		
1	nVar2	INT	Input		%IW11.0	3x1001	<input type="checkbox"/>		
2	nVar3	REAL	Input		%ID11.1	3x1003	<input type="checkbox"/>		
3	nVar4	BOOL	Holding		%QX10.0	0x1001	<input type="checkbox"/>		
4	nVar5	BOOL	Input		%IX10.0	1x1001	<input type="checkbox"/>		

## 6. Определение типа доступа к переменным из областей 0x и 4x

	Name	Data Type	Register Type
0	nVar1	INT	Holding
1	nVar2	INT	Input
2	nVar3	REAL	Holding
3	nVar4	BOOL	Holding
4	nVar5	BOOL	Input

*Переменные в ведомом ПЛК*

	Name	Data Type	Register Type
0	nVar1	INT	Holding
1	nVar2	INT	Input
2	nVar3	REAL	Input
3	nVar4	BOOL	Holding
4	nVar5	BOOL	Input

*Переменные в ведущем ПЛК*

Обратите внимание на переменную **nVar3** типа REAL, расположенную в области 4x в ведомом устройстве (Holding). Ведущее устройство может как читать, так и записывать эту переменную. Для того, чтобы **прочитать** nVar3 из ведомого устройства, тип доступа в ведущем устройстве нужно определить как **Input**, несмотря на то что в ведомом она определена как **Holding**. То же самое касается переменных типа BOOL, определённых в ведомом устройстве, как Holding

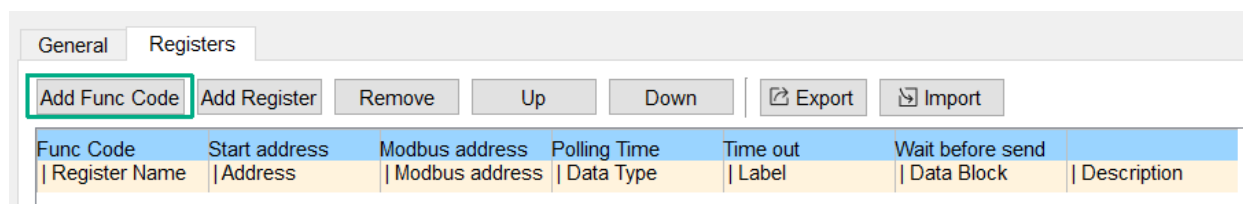
## 7. Конфигурация Modbus-запросов Modbus TCP Client

- а) Перейдите в меню Configuration – Ethernet - Generic Modbus\_S172
- б) Откройте вкладку Registers добавленного устройства для конфигурации чтения/записи регистров ведомого устройства

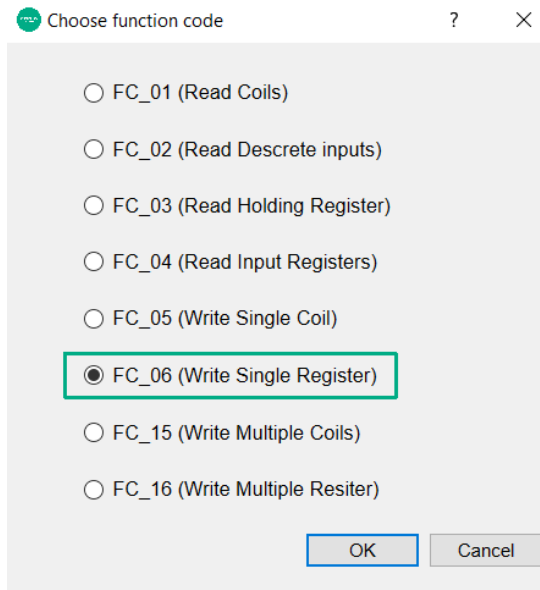
### Запись INT переменной в области 4x

nVar1	INT	Holding	4x1001	Запись
-------	-----	---------	--------	--------

- а) На вкладке Registers создайте новый запрос модбас, для чего щёлкните по кнопке Add Func Code:



- б) Выберите FC\_06 и нажмите ОК



Func Code	Start address	Modbus address	Polling Time	Time out	Wait before send	
Register Name	Address	Modbus address	Data Type	Label	Data Block	Description
FC_06	0	4x0001	100ms	1000ms	10ms	
Register	0	4x0001	WORD	...		description

с) В строке кода функции модбас в ячейке Start Address нужно указать смещение начального адреса относительно первого регистра области. Если указан начальный адрес области 4x - 4x0001, то смещение (Start Address) относительно начального адреса будет:

$$4x0001 - 4x0001 = 0$$

Для записи адреса 4x1001 переменной nVar1 в ячейку Start Address строки кода функции модбас FC\_06 введите смещение от адреса 4x0001:

$$4x1001 - 4x0001 = 1000$$

Func Code	Start address	Modbus address
Register Name	Address	Modbus address
FC_06	1000	4x1001
Register	1000	4x1001

d) Далее настройте:

- Период опроса (Polling Time): 500ms
- Таймаут (Time out): 3000ms

- Паузу перед посылкой запроса (Wait before send): 10ms

Func Code	Start address	Modbus address	Polling Time	Time out	Wait before send
Register Name	Address	Modbus address	Data Type	Label	Data Block
FC_06	1000	4x1001	500ms	3000ms	10ms
Register	1000	4x1001	WORD	...	

- Щёлкните дважды по ячейке в столбце Register Name и переименуйте Register в nVar1
- В столбце Data Type выберите тип переменной – INT
- В следующем столбце щёлкните по ячейке «...» и в открывшемся окне выберите переменную для связывания nVar1 из группы переменных устройства Slave\_S172 и нажмите ОК

Func Code	Start address	Modbus address	Polling Time	Time out	Wait before send	Description
Register Name	Address	Modbus address	Data Type	Label	Data Block	Description
FC_06	1000	4x1001	500ms	3000ms	10ms	
nVar1	1000	4x1001	INT	...		W1

Choose PLC variable ? X

Filter:

holding: nVar1 (INT)

CLEAR

- В столбце Description введите описание переменной: W1

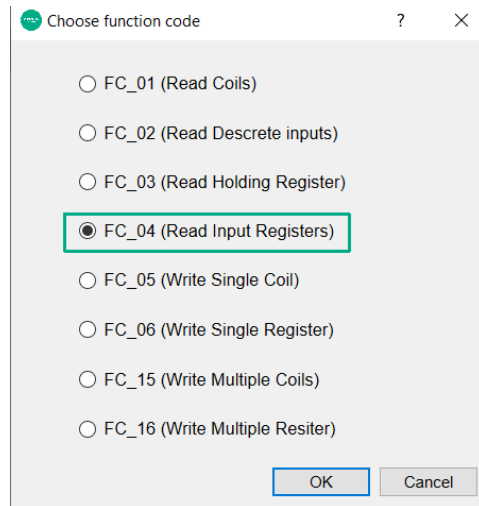
Результат должен выглядеть так:

Func Code	Start address	Modbus address	Polling Time	Time out	Wait before send	Description
Register Name	Address	Modbus address	Data Type	Label	Data Block	Description
FC_06	1000	4x1001	500ms	3000ms	10ms	
nVar1	1000	4x1001	INT	nVar1	%QW11.0	W1

### Чтение INT переменной из области 3x

nVar2	INT	Input	3x1001	Чтение
-------	-----	-------	--------	--------

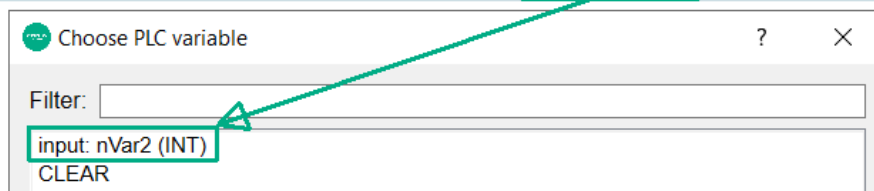
- На вкладке Registers создайте новый запрос модбас, для чего щёлкните по кнопке Add Func Code



b) Выберите тип переменной – INT

c) Укажите смещение, период опроса и таймаут аналогично предыдущим настройкам

Func Code	Start address	Modbus address	Polling Time	Time out	Wait before send	
Register Name	Address	Modbus address	Data Type	Label	Data Block	Description
FC_06	1000	4x1001	500ms	3000ms	10ms	
nVar1	1000	4x1001	INT	nVar1	%QW11.1	W1
FC_04	1000	3x1001	500ms	3000ms	10ms	
Register	1000	3x1001	INT	...		description



d) Щёлкните дважды по ячейке в столбце Register Name и переименуйте Register в nVar2

e) В описании напишите R1

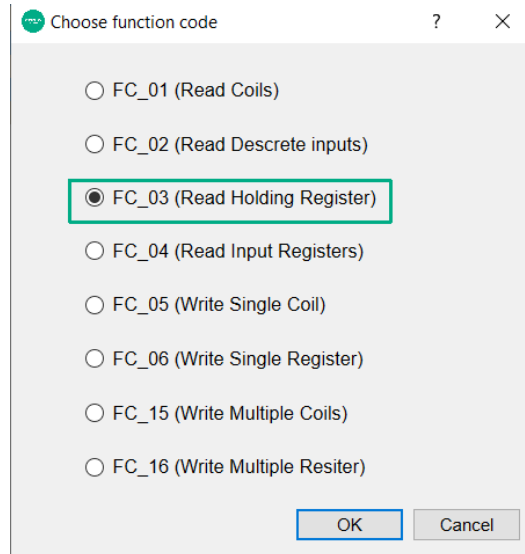
Результат должен выглядеть так:

FC_04	1000	3x1001	500ms	3000ms	10ms	
nVar2	1000	3x1001	INT	nVar2	%IW11.0	R1

## Чтение REAL переменной из области 4x

nVar3	REAL	Holding	4x1003	Чтение
-------	------	---------	--------	--------

a) На вкладке Registers создайте новый запрос модбас, для чего щёлкните по кнопке Add Func Code



b) Выберите тип переменной REAL

c) Укажите смещение (Start Address) Т.к. адрес переменной 4x1003, смещение будет 1002:

$$4x1003 - 4x0001 = 1002$$

FC_03	1002	4x1003	100ms
Register	1002	4x1003	REAL

d) Свяжите с регистром nVar3:

FC_03	1002	4x1003	500ms	3000ms	10ms
Register	1002	4x1003	REAL	...	

Choose PLC variable

Filter:

input: nVar3 (REAL)

CLEAR

e) Период опроса и таймаут аналогично предыдущим настройкам

f) Щёлкните дважды по ячейке в столбце Register Name и переименуйте Register в nVar3

g) В описании напишите R2

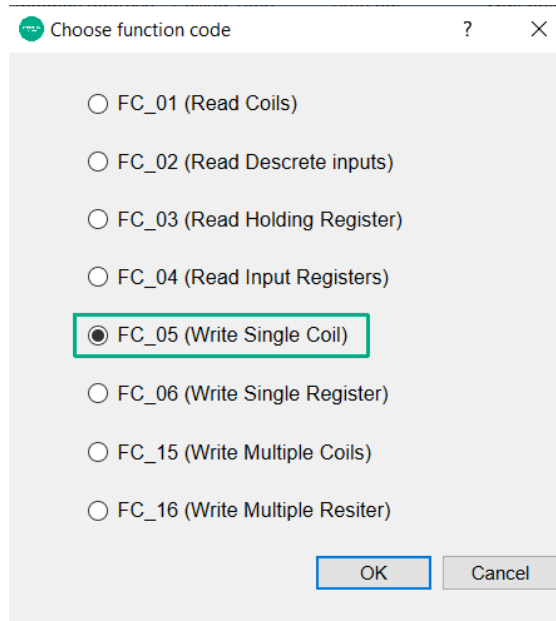
Результат должен выглядеть так:

FC_03	1002	4x1003	500ms	3000ms	10ms
nVar3	1002	4x1003	REAL	nVar3	%ID11.1 R2

## Запись BOOL переменной из области 0x

nVar4	BOOL	Holding	0x1001	Запись
-------	------	---------	--------	--------

a) На вкладке Registers создайте новый запрос модбас, для чего щёлкните по кнопке Add Func Code



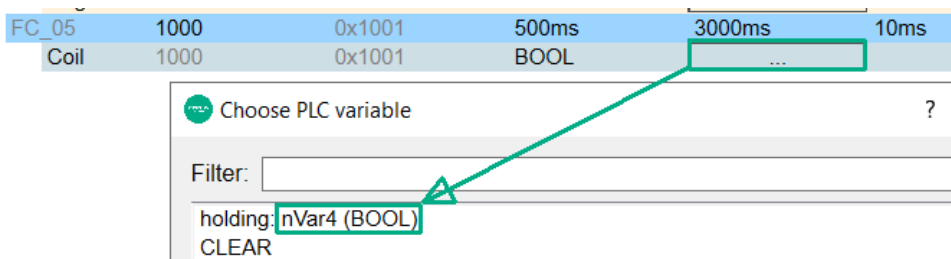
b) Выберите тип переменной – BOOL

c) Укажите смещение (Start Address) Т.к. адрес переменной 0x1001, смещение будет 1000:

$$0x1001 - 0x0001 = 1000$$

FC_05	1000	0x1001
Coil	1000	0x1001

d) Свяжите с регистром nVar4:



e) Щёлкните дважды по ячейке в столбце Register Name и переименуйте Register в nVar4

- f) Период опроса и таймаут аналогично предыдущим настройкам
- g) В описании напишите W2

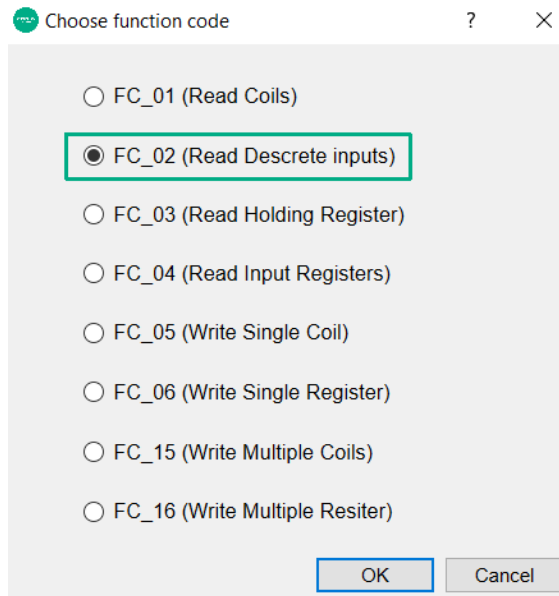
Результат должен выглядеть так:

FC_05	1000	0x1001	500ms	3000ms	10ms	
nVar4	1000	0x1001	BOOL	nVar4	%QX10.0	W2

### Чтение BOOL переменной из области 1x

nVar5	BOOL	Input	1x1001	Чтение
-------	------	-------	--------	--------

- a) На вкладке Registers создайте новый запрос модбас, для чего щёлкните по кнопке Add Func Code

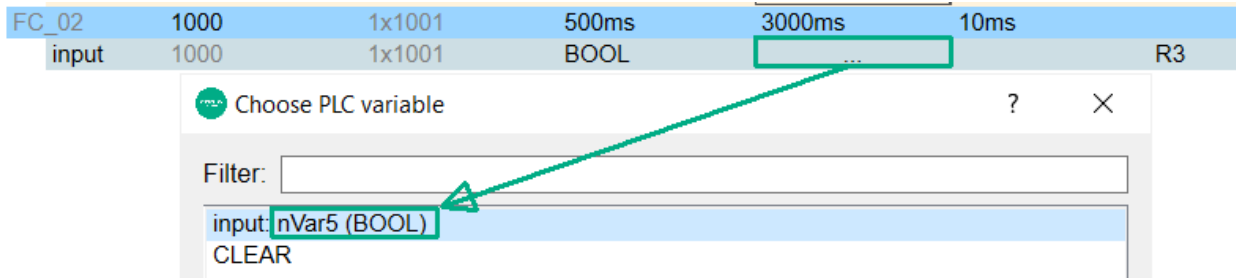


- b) Выберите тип переменной – BOOL
- c) Укажите смещение (Start Address) Т.к. адрес переменной 1x1001, смещение будет 1000:

$$1x1001 - 1x0001 = 1000$$

FC_02	1000	1x1001
input	1000	1x1001

- d) Свяжите с регистром nVar5:



- e) Щёлкните дважды по ячейке в столбце Register Name и переименуйте Input в nVar5
- f) Период опроса и таймаут аналогично предыдущим настройкам
- g) В описании напишите R3

Результат должен выглядеть так:

FC_02	1000	1x1001	500ms	3000ms	10ms		
nVar5	1000	1x1001	BOOL	nVar5	%IX10.0	R3	

Если при связывании запроса в окно выбора переменной **пустое** – значит, выбран неправильный тип данных (WORD вместо REAL, например) или неправильный тип доступа (Input для функции записи)

Итоговый результат должен выглядеть так:

Func Code	Start address	Modbus address	Polling Time	Time out	Wait before send		
Register Name	Address	Modbus address	Data Type	Label	Data Block	Description	
FC_06	1000	4x1001	500ms	3000ms	10ms		
nVar1	1000	4x1001	INT	nVar1	%QW11.1	W1	
FC_04	1000	3x1001	500ms	3000ms	10ms		
nVar2	1000	3x1001	INT	nVar2	%IW11.0	R1	
FC_03	1002	4x1003	500ms	3000ms	10ms		
nVar3	1002	4x1003	REAL	nVar3	%ID11.1	R2	
FC_05	1000	0x1001	500ms	3000ms	10ms		
nVar4	1000	0x1001	BOOL	nVar4	%QX10.0	W2	
FC_02	1000	1x1001	500ms	3000ms	10ms		
nVar5	1000	1x1001	BOOL	nVar5	%IX10.0	R3	

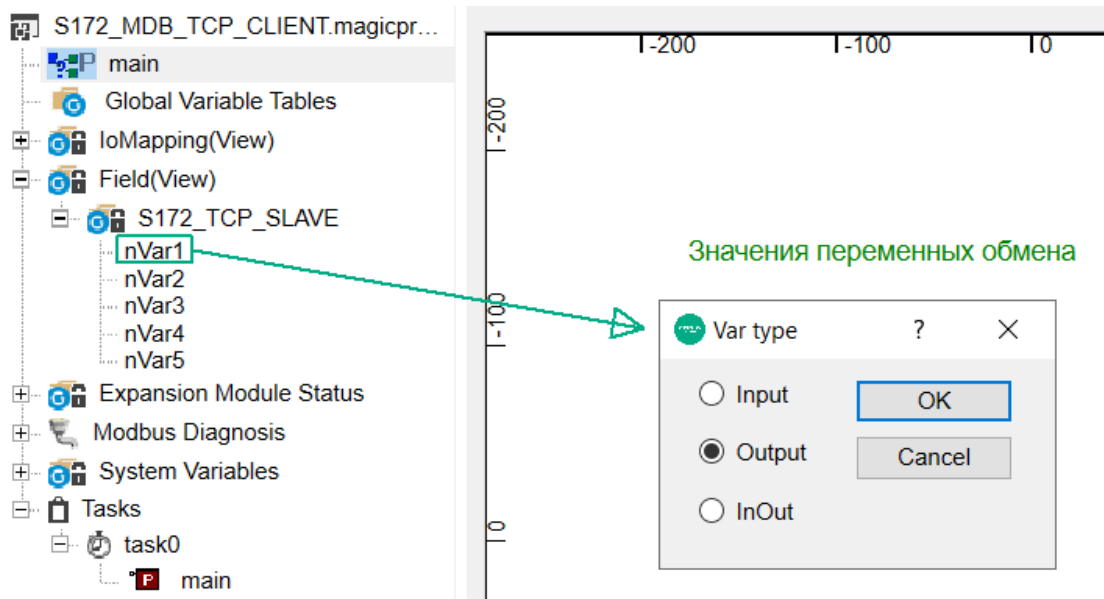
8. Если нужно групповым запросом прочитать/записать несколько переменных, адреса которых в слэиве расположены последовательно, выделите строку запроса, например, FC\_03 - Add register – Укажите тип – Свяжите с регистром глобальных сетевых переменных

9. Добавьте в программу main глобальные сетевые переменные.

a) На вкладке Programming откройте программу main и перетащите переменные из Field (View) – S172\_TCP\_SLAVE на рабочую область программы

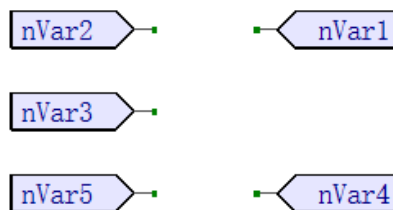
b) В диалоговом окне для переменных nVar1 и nVar4 выберите Var type: Output

c) В диалоговом окне для переменных nVar2, nVar3, nVar5 выберите Var type: Input

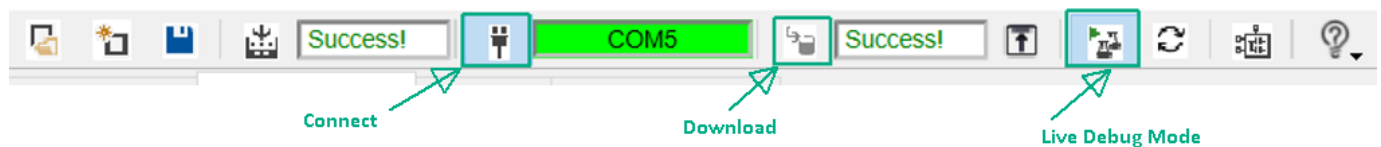


В результате должно получиться:

### Значения переменных обмена



10. Скомпилируйте проект, исправьте ошибки при наличии, сохраните и загрузите во второй ПЛК.
11. Соедините сетевым кабелем Ethernet контроллер с загруженной программой S172\_MDB\_TCP\_SERVER и контроллер с загруженной программой S172\_MDB\_TCP\_CLIENT, после чего подайте питание на контроллеры
12. Подключитесь к ПЛК-Client, откройте секцию main, включите режим Live Debug Mode и протестируйте работоспособность программ



## Логика работы демонстрационных проектов Modbus TCP

Логика программ заключается в следующем:

### 4. Чтение REAL переменной

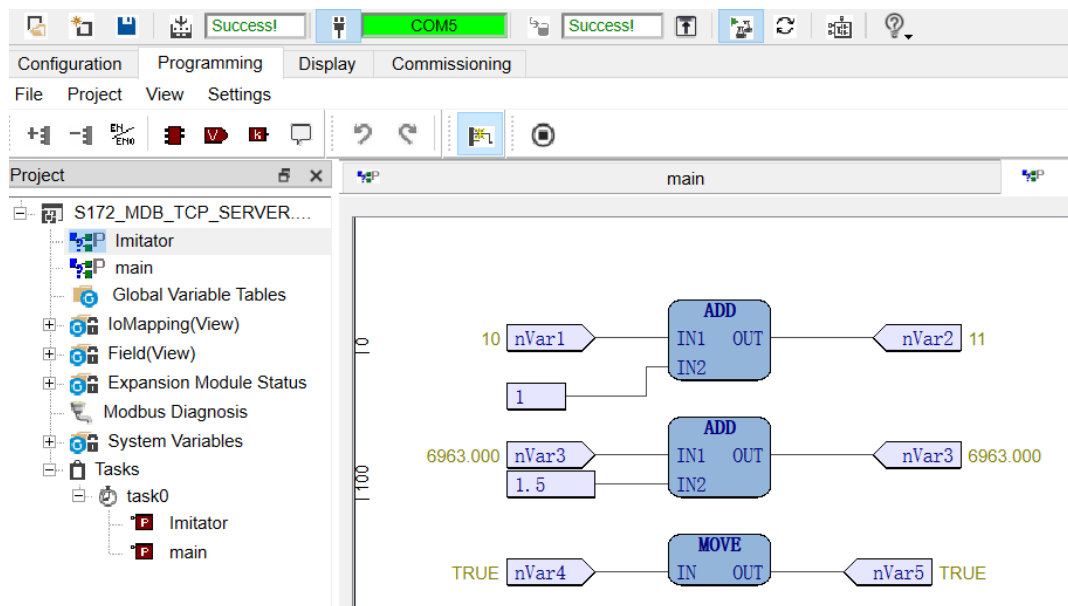
- Раз в секунду в сервере увеличивается значение переменной nVar3 на 1.5
- Если связь установлена, и программа сервера выполняется, то nVar3 в клиенте будет каждую секунду увеличиваться на 1.5

### 5. Чтение и запись INT переменных

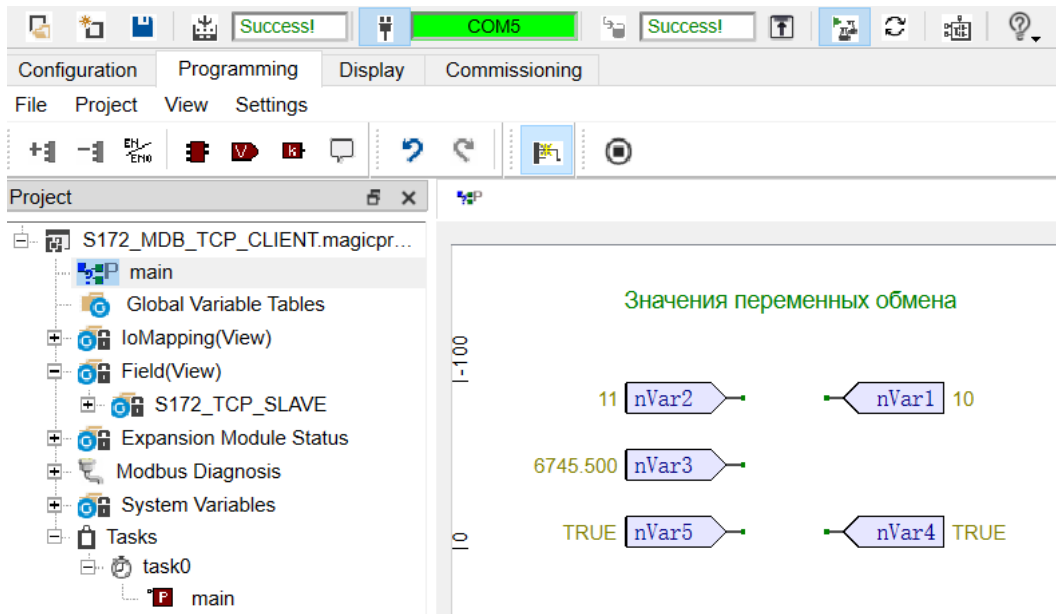
- Значение переменной nVar1 клиент записывает в сервер
- В сервере к значению nVar1 прибавляется 1 и новое значение записывается в переменную nVar2
- Клиент читает значение переменной nVar2
- Если связь установлена, и программа сервера выполняется, то после изменения nVar1 значение nVar2 будет равно  $nVar1 + 1$

### 6. Чтение и запись BOOL переменных

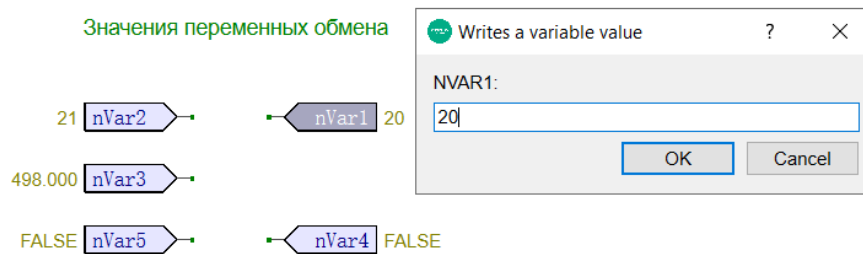
- Значение переменной nVar4 клиент записывает в сервер
- В сервере значение nVar4 копируется с помощью блока MOVE в переменную nVar5
- Клиент читает значение переменной nVar5.
- Если связь установлена, и программа сервера выполняется, то после изменения nVar4 значение nVar5 будет таким же



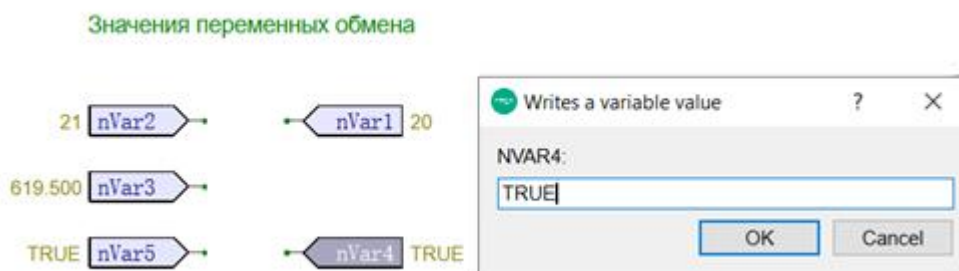
Программа Modbus TCP Server в режиме онлайн и Live Debug Mode



Программа Modbus TCP Client в режиме онлайн и Live Debug Mode



Щёлкните дважды по переменной nVar1, введите новое значение, ОК



Щёлкните дважды по переменной nVar4, введите новое значение, ОК

# Используемые сокращения и термины

## Сокращения

- PC – персональный компьютер (ПК)
- PLC – программируемый логический контроллер (ПЛК)
- LMB (Left Mouse Button) – левая кнопка мыши (ЛКМ)
- RMB (Right Mouse Button) – правая кнопка мыши (ПКМ)
- IO (Inputs/Outputs) – входы/выходы (в/в) ПЛК и модулей расширения
- POU (Program Operation Unit) – программа пользователя
- FB (Function Block) – функциональный блок
- Function Block Diagram (FBD) – язык функциональных блоков
- Continuous Function Chart (CFC) – язык непрерывных функциональных схем
- ВОС – внутренняя операционная система ПЛК [прошивка ПЛК]

## Термины частей интерфейса

- Toolbar – панель инструментов
- Tabs – вкладки меню: содержит основные разделы: Configuration (Конфигурация), Programming (Программирование), Display (Отображение), Commissioning (Ввод в эксплуатацию).
- Menu – пункты меню
- Project Tree – дерево проекта
- Library Tree – библиотека элементов
- Properties Window – свойства
- Output – поле вывода
- Status bar – строка состояния
- Programming Toolbar – панель инструментов программирования
- Variable Declaration Area – область объявления переменных
- Program editing area – область написания программного кода
- Display Toolbar – панель инструментов Дисплей
- Page editing area – область создания пользовательских экранов
- Run/Stop – режим работы ПЛК (пуск/стоп)
- Simulation Mode – оффлайн симулятор ПЛК
- Live Debug Mode – онлайн мониторинг ПЛК
- Watch Window – окно наблюдения за переменными
- ЛКМ – щелчок левой кнопкой мыши
- ПКМ – щелчок правой кнопкой мыши

## Социальные сети

 [systemelectric\\_official](https://t.me/systemelectric_official)

 [youtube.com/c/SystemeElectric](https://youtube.com/c/SystemeElectric)

 [vk.com/Systemelectric](https://vk.com/Systemelectric)

 Systeme Electric



[www.systeme.ru](http://www.systeme.ru)

## Наши бренды

**Systeme**  
electric

**Dēkraft**

 Механотроника

 **Systeme**  
soft