



Unimod Pro

ΠΡΟΤΟΚΟΛ ΜΟΔΒΟΥΣ

АО „ТРЭИ” постоянно совершенствует и развивает свою продукцию. В связи с этим информация, содержащаяся в данном документе, может изменяться без дополнительного предупреждения пользователей.

Все права на этот документ принадлежат АО „ТРЭИ”. Ни весь документ, ни какая-либо его часть не могут быть скопированы или воспроизведены без предварительного письменного разрешения АО „ТРЭИ”.

© 1990-2020 АО «ТРЭИ»

Россия,

440028, Пенза, ул. Титова, 1Г

Телефон (fax): +7 (8412) 55-58-90, 49-95-39

fax: +7 (8412) 49-85-13

e-mail: trei@trei-gmbh.ru

QNX® is a registered trademark of QNX Software Systems Ltd.

Windows® is a registered trademark of Microsoft Corporation.

DiskOnChip® and TrueFFS® are a registered trademark of M-systems Ltd.

iFIX® is a registered trademark of Intellution, Inc.

All other brand or product names are trademarks or registered trademarks of their respective holders

Оглавление

ВВЕДЕНИЕ.....	4
1. ПРОТОКОЛ MODBUS В СИСТЕМЕ UNIMOD PRO	5
2. ПРОТОКОЛ MODBUS-RTU MASTER	7
2.1 Общие сведения	7
2.2 Modbus-RTU Master на мастер-модуле	8
2.3 Modbus-RTU Master на интеллектуальном модуле.....	9
3. ПРОТОКОЛ MODBUS-RTU SLAVE.....	10
3.1 Общие сведения	10
3.2 Адресное пространство режима Slave.....	11
3.3 Modbus-RTU Slave на мастер-модуле	17
3.4 Modbus-RTU Slave на интеллектуальном модуле.....	18
4. ПРОТОКОЛ MODBUS ETHERNET	19
4.1 Общие сведения	19
4.2 Modbus ETHERNET на мастер-модуле	20
5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM	21
5.1 MB_PARAM на мастер-модуле	22
5.1.1 Мастер-модуль ECO - M1010E, M1011E, M1012E	22
5.1.2 Мастер-модуль M911E	22
5.1.3 Мастер-модуль M841E / M902E / M903E / M915E / M921E	22
5.1.4 Мастер-модуль M401E	25
5.1.5 Мастер-модуль M501E	27
5.2 MB_PARAM на интеллектуальном модуле (кроме M932C2).	31
5.3 MB_PARAM на интеллектуальном модуле M932C2	32

ВВЕДЕНИЕ

Протокол Modbus-RTU предназначен для организации связи в режимах Master и Slave между контроллерами фирмы **TREI** и другими устройствами, поддерживающими данный протокол. Среда **Unimod Pro** обладает необходимым набором функций и гибкой системой настроек для успешной реализации протокола Modbus.

Документ предназначен для разработчиков программного обеспечения, а также для проектировщиков систем контроля и управления.

При работе используйте документацию следующих фирм:

TREI

- «Unimod Pro. Руководство пользователя»
- «Unimod Pro. Руководство по программированию»

1. ПРОТОКОЛ MODBUS В СИСТЕМЕ UNIMOD PRO

1. ПРОТОКОЛ MODBUS В СИСТЕМЕ UNIMOD PRO

Гибкая система настроек протокола Modbus-RTU позволяет установить связь между интеллектуальным модулем и любым другим устройством, поддерживающим протокол Modbus.

Контроллеры взаимодействуют, используя метод "главный - подчиненный", в котором только одно устройство (master-устройство) может инициировать транзакции (называемые "запросами"). Другие устройства (slave-устройства) отвечают, обеспечивая главное устройство запрошенными данными, или выполняют действия, указанные в запросе. Типичные подчиненные устройства включают программируемые контроллеры.

Master-устройство может адресовать отдельные подчиненные устройства, или может инициализировать "широковещательное" сообщение для всех подчиненных. Slave-устройства возвращают сообщения (называемое "ответ") на запросы, которые адресованы отдельно к ним. На "широковещательные" запросы от главного устройства ответы не возвращаются.

Протокол Modbus устанавливает формат для запроса master-устройства, путем помещения в него адреса устройства (или широковещательного адреса), функционального кода, определяющего запрошенную операцию, данные подлежащие отправке и поле обнаружения ошибок.

Список поддерживаемых функциональных кодов Modbus в среде Unimod Pro представлен в таблице 1.

Таблица 1 – Поддерживаемые функциональные коды Modbus

Код	Выполняемая функция
01h	Чтение состояния ячейки (Read Coil Status)
02h	Чтение состояния входа (Read Contact Status)
03h	Чтение блокировочных регистров (Read Holding Registers)
04h	Чтение входных регистров (Read Input Registers)
05h	Управление одиночной ячейкой (Force Single Coil)
06h	Установка одиночного регистра (Preset Single Register)
08h	Диагностика связи (Loop-back Diagnostic)
0Fh	Управление многими ячейками (Force Multiple Coils)
10h	Установка многих регистров (Preset Multiple Registers)
17h	Чтение/установка многих регистров (Read/Write Multiple registers)

Сообщение ответа подчиненного устройства также конструируется, используя протокол Modbus. Оно содержит поля подтверждение принятой операции, любые данные, подлежащие возврату, и поле обнаружения ошибок. Если ошибка случилась во время получения сообщения, или если подчиненное устройство неспособно выполнить запрошенное действие, подчиненное устройство будет создавать сообщение об ошибке и посылать его в качестве ответа.

Поскольку интерфейсы связи Modbus Master и Modbus Slave используют разные физические линии, становится возможным параллельное использование режимов Master и Slave.

Подробное описание формата функций и спецификация протокола Modbus свободно доступна в глобальной сети Интернет по адресу <http://www.modbus.org/>.

В среде Unimod Pro реализован набор функциональных блоков, реализующих функции протокола Modbus-RTU, предназначенный для организации информационного обмена с подчиненными устройствами, поддерживающими данный протокол.

Для работы в режиме мастер используются следующие функциональные блоки: MB_R_V (чтение состояния 8-ми дискретных входов), MB_R_H (чтение целочисленного регистра), MB_R_I (чтение целочисленного входного регистра), MB_R_F (чтение пары регистров в формате с плавающей точкой), MB_R_R16 (чтение 16 целочисленных регистров), MB_R_F16 (чтение 16 пар регистров в формате с плавающей точкой), MB_W_C (запись в бинарную ячейку памяти), MB_W_H (запись целочисленного регистра), MB_W_F (запись пары регистров в формате с плавающей точкой), MB_DIAG (диагностика связи с подчиненным устройством), MB_STATE (получение ID подчиненного устройства). Функциональный блок MB_PARAM используется для установки параметров Modbus для всех режимов работы.

1. ПРОТОКОЛ MODBUS В СИСТЕМЕ UNIMOD PRO

Подробное описание функциональных блоков приведено в документе “Unimod Pro. Руководство по программированию”.

Протокол Modbus реализован на мастер-модулях и интеллектуальных модулях серий M800, M900, описанных в документе «Unimod Pro. Менеджер библиотек». В данном документе приведено описание особенностей реализации протокола Modbus на мастер-модулях и интеллектуальных модулей.

2. ПРОТОКОЛ MODBUS-RTU MASTER

2.1 Общие сведения

В режиме Master обмен с подчиненными устройствами осуществляется технологической программой модуля через функциональные блоки, предназначенные для работы с Modbus. Функциональные блоки реализуют следующие функции:

- Чтение одной или нескольких булевых переменных, 16-ти и 32-х разрядных целых и вещественных переменных.
- Запись булевой, целой или вещественной переменной в Slave-устройство;
- Установка параметров задачи связи Modbus, функции диагностики.

Включение (активация) и конфигурирование Master-интерфейса Modbus обычно осуществляется из технологической программы через функциональный блок MB_PARAM.

В режиме Modbus Master slave-устройства подключаются к модулю через последовательный канал RS-485, реализуемый юнитом UBUS (U485) (рисунок 1).

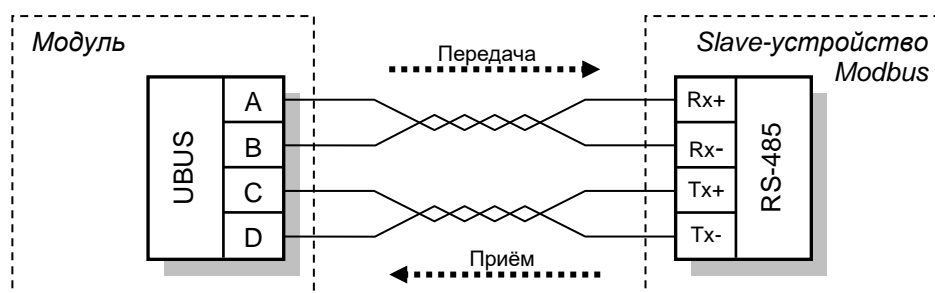


Рисунок 1

Реализация протокола Modbus Master для различных типов модулей имеет свои особенности.

2.2 Modbus-RTU Master на мастер-модуле

Для работы мастер-модуля **M911E** в качестве master-устройства на линии Modbus используется ICAT (встроенный коммуникационный адаптер). При этом должны быть выполнены следующие условия:

1. В ICAT загружена микропрограмма, реализующая протокол Modbus;
2. В конфигурации M911E установлена работа ICAT через юнит UBUS (U485).

Примечание: Описание устройств ICAT, ECAT, UBUS (U485) представлено в документе "Unimod Pro. Исполнительная система".

Для мастер-модулей **M841E/M902E/M921E/M915E/M903E** задействованы COM порты COM1..COM9 (количество COM портов, реально присутствующих на мастер-модуле, зависит от комплектации).

2. ПРОТОКОЛ MODBUS-RTU MASTER

2.3 Modbus-RTU Master на интеллектуальном модуле

Интеллектуальные модули не поддерживают функциональные блоки MB_R_R16 (чтение 16 целочисленных регистров), MB_R_F16 (чтение 16 пар регистров в формате с плавающей точкой).

3. ПРОТОКОЛ MODBUS-RTU SLAVE

3.1 Общие сведения

В режиме Slave модуль играет роль подчиненного устройства, предоставляя master-контроллеру возможность чтения и записи переменных приложения Unimod Pro посредством Modbus.

3.2 Адресное пространство режима Slave

Организация адресного пространства Modbus (таблицу 2) в режиме Slave основывается на индексированном доступе к массиву однотипных ячеек и состоит из 4-х независимых групп памяти, каждая из которых обладает определенными качествами.

Таблица 2 – Адресное пространство Modbus

Группа	Тип	Размер	Доступ	Описание
1	Дискретные вводы (Contacts)	1 бит	Только чтение	Данные каналов ввода
2	Дискретные выходы (Coils)	1 бит	Чтение и запись	Данные Управления
3	Входные регистры (Input Register)	16 бит	Только чтение	Данные каналов ввода
4	Выходные регистры (Holding Registers)	16 бит	Чтение и запись	Данные Управления

Диапазон адресов каждой из групп памяти ограничен 16-ти разрядным словом и на физическом уровне может принимать значения от 0 до 65535. Таким образом, для каждой группы памяти протокол Modbus выделяет адресное пространство, состоящее из 65536 ячеек, а также предоставляет простые методы для их чтения и записи, которые позволяют разбить большое количество последовательно идущих ячеек на небольшие блоки, ограниченные исключительно форматом пакета Modbus и/или свойствами линии передачи.

Логическое адресное пространство мастер-контроллера Modbus как правило отличается от такого представления, но независимо от типа применяемого контроллера, в результате преобразований логического адреса будет получен диапазон физических адресов 0...65535.

Протокол Modbus различает 2 базовых типа данных: 1-битовый дискретный (булевский) и 16-ти разрядное целое со знаком или без знака. Другие, более сложные структуры или форматы данных составляются путём группирования нескольких, как правило, следующих друг за другом переменных одного из базовых типов.

Обмен данными между интеллектуальным модулем и контроллером Modbus происходит через словарь обмена технологической программы. Более подробно словарь обмена описан в документе "Unimod Pro. Руководство пользователя". Все объявленные в словаре обмена булевские, целые и вещественные переменные становятся доступны для чтения и записи через функции протокола Modbus.

В реализации Modbus однотипные группы памяти совмещены, т.е. группы 1 и 2 относятся к булевским, а группы 3 и 4 - целым и вещественным переменным. Обмен переменными (чтение и запись) осуществляется через стандартные функции протокола Modbus.

Независимо для каждой из четырех групп памяти Modbus может быть задан базовый адрес блока переменных в словаре обмена технологической программы и количество переменных, доступное для чтения (и, в зависимости от атрибутов, записи) в данной группе памяти Modbus. Такое построение системы позволяет произвести разделение непрерывного адресного пространства словаря обмена на независимые блоки, обладающие разными атрибутами в соответствии со спецификой протокола Modbus.

При получении запроса, в зависимости от кода функции Modbus, интеллектуальный модуль производит преобразование адреса переменной Modbus в адрес словаря обмена. Для преобразования адреса используется таблица трансляции адресов Modbus. Эта таблица находится в ресурсах технологической программы и задается через функциональный блок MB_PARAM. Упрощенная схема преобразования адреса Modbus показана на рисунке 2.

Смещение и размер указываются для каждого типа памяти Modbus независимо. На практике это означает, что пространства адресов для Coils (витки), Contacts (контакты), Input Registers (входные регистры) и Holding Registers (выходные регистры) могут располагаться в любом порядке, пересекаться или относиться к одной и той же группе переменных словаря обмена технологической программы.

3. ПРОТОКОЛ MODBUS-RTU SLAVE

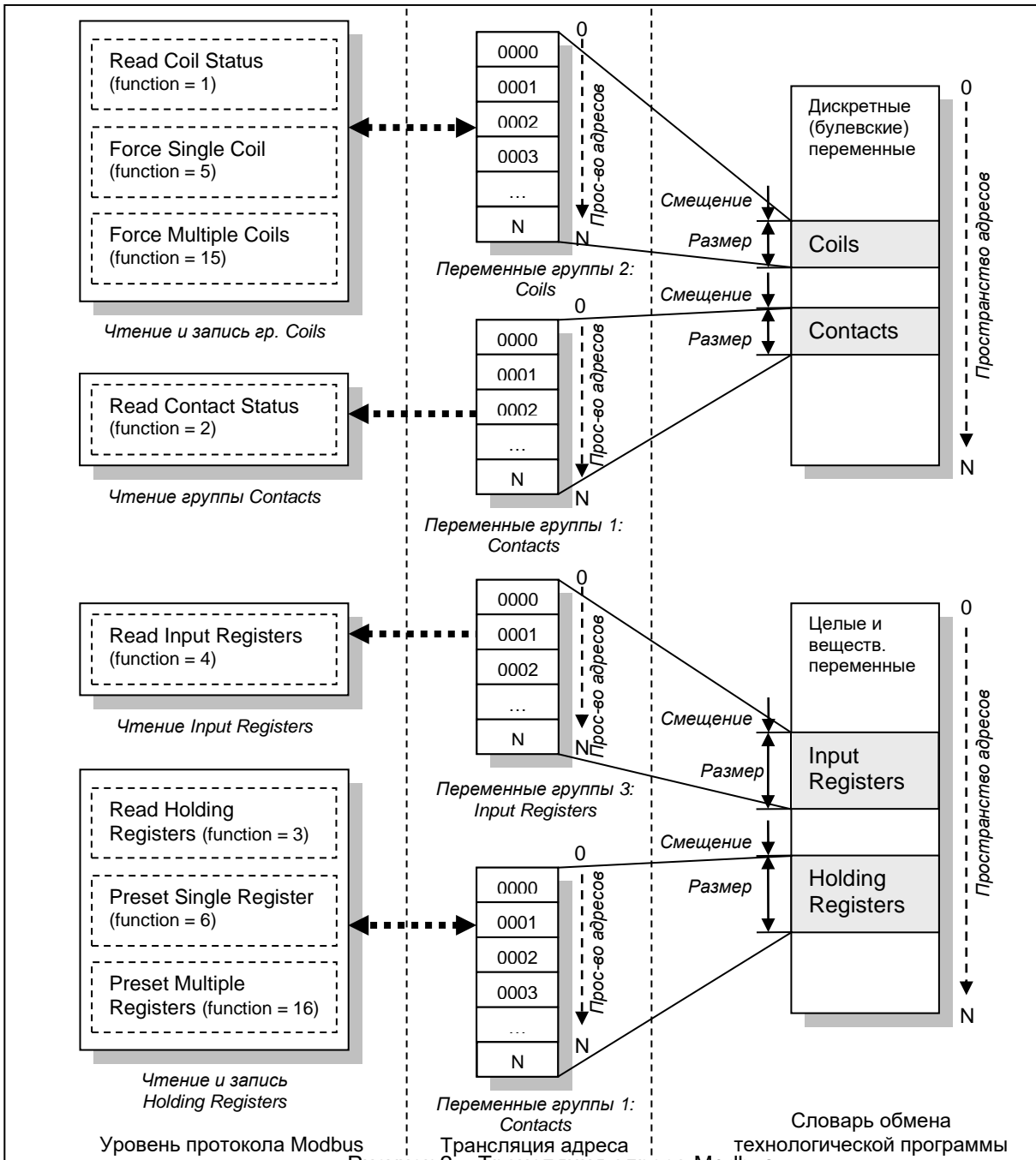


Рисунок 2 – Трансляция адреса Modbus

При использовании мастер-модуля в качестве slave-устройства переменные из словаря привязываются к адресному пространству Modbus с помощью инструмента “Карта адресов Modbus”, который вызывается из меню «Инструменты» программы Unimod Pro (рисунок 3). Подробная информация о программе Unimod Pro представлена в документе “Unimod Pro. Руководство пользователя”.

Таблица трансляции адресов Modbus на M911E содержит только смещение для каждого типа памяти Modbus. Единственное ограничение накладывается на размер – результирующий адрес не может превышать 65534 (0FFFFh).

3. ПРОТОКОЛ MODBUS-RTU SLAVE

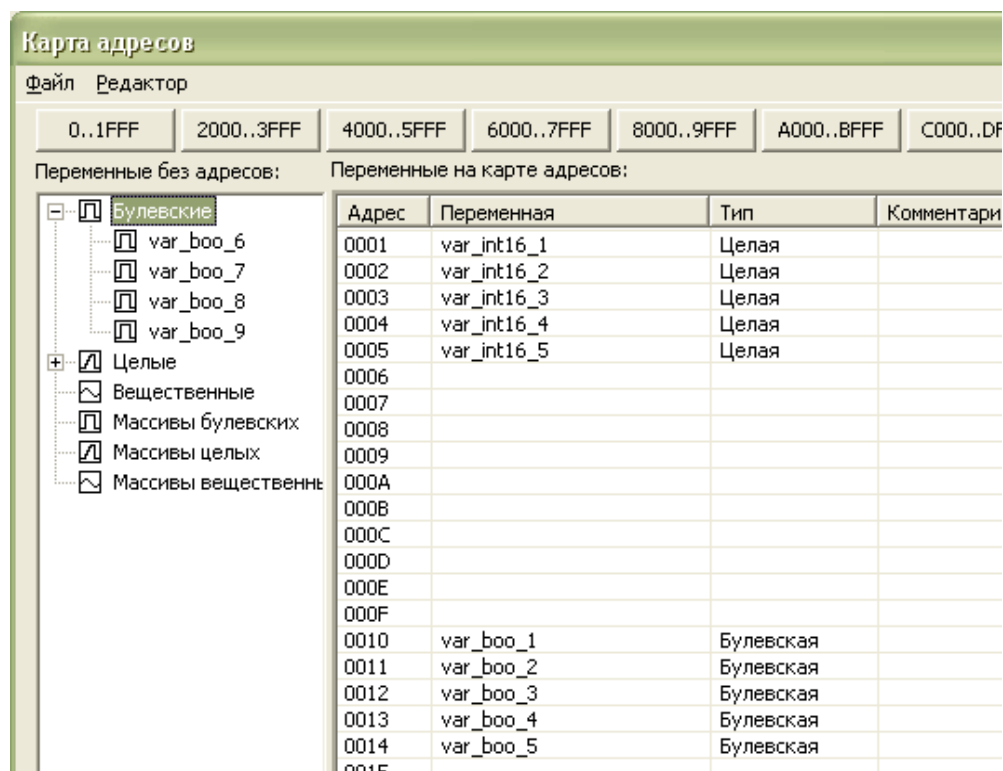


Рисунок 3

Для интеллектуальных модулей переменные читаются из словаря обмена. В этом случае адрес переменных в адресном пространстве Modbus зависит от их адреса в словаре обмена (рисунок 4).

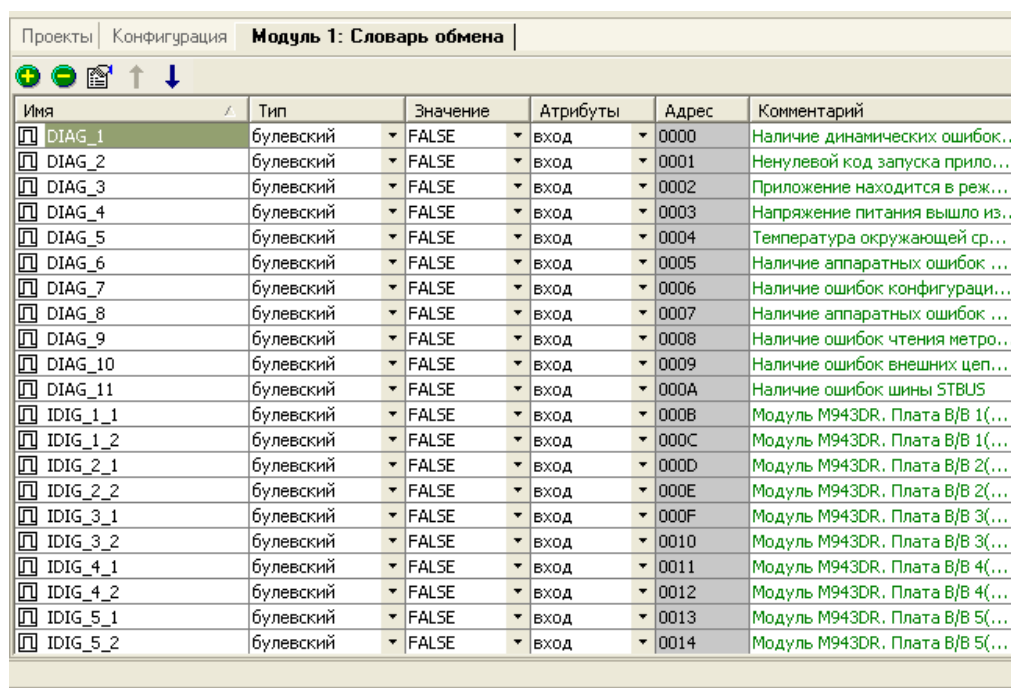


Рисунок 4

Протокол Modbus различает два базовых типа данных: 1-битовый дискретный (булевский) и 16-ти разрядное целое со знаком или без знака. Другие, более сложные структуры или форматы данных составляются путём группирования нескольких, как правило, следующих друг за другом переменных одного из базовых типов.

3. ПРОТОКОЛ MODBUS-RTU SLAVE

Каждой из четырех групп памяти Modbus должен быть присвоен адрес смещения. Фактический адрес, по которому Modbus контроллер должен читать/писать некоторую переменную, является суммой адреса смещения и адреса на «Карте адресов» к которому привязана эта переменная.

Адресация булевских переменных линейна. Каждой переменной соответствует один дискретный адрес в адресном пространстве Modbus (группы 1 и 2).

Целые и вещественные переменные занимают 32 разряда и поэтому не могут быть переданы по протоколу Modbus в явном виде.

Интеллектуальные модули реализуют два режима доступа к целым и вещественным переменным:

- **32-х разрядный:** Каждая переменная словаря обмена состоит из двух последовательно расположенных 16-ти разрядных регистров Modbus.
- **16-ти разрядный:** Каждой переменной соответствует один 16-ти разрядный регистр Modbus. При чтении текущее значение целой переменной ограничивается 16-ти разрядами, записанное значение расширяется до 32-х разрядов. Обмен вещественными переменными в этом режиме не поддерживается.

Для обеспечения максимальной совместимости с мастер-контроллером Modbus, в дополнение к выбранному режиму адресации, во время приема и перед передачей по последовательной линии, модуль может производить перестановку байт в границах одной переменной. Режим адресации и порядок замены байт устанавливается функциональным блоком MB_PARAM.

В 32-х разрядном режиме каждая целая или вещественная переменная разбивается на две независимые ячейки, расположенные под смежными адресами. Со стороны Modbus первая половина переменной доступна под четным адресом, вторая – под следующим за ним нечетным адресом. Следует отметить, что в зависимости от установленного режима работы 'половина' может соответствовать как старшим, так и младшим 16-ти разрядам переменной.

Таким образом, в физическом адресном пространстве Modbus 32-разрядные переменные располагаются под четными адресами, начиная с адреса ноль и заканчивая 65534 (0FFFEh).

32-х разрядный режим накладывает некоторые ограничения на способ обмена. Два регистра, из которых состоит переменная, должны обрабатываться как неделимое целое. Доступ к переменной должен происходить одним запросом.

В 16-ти разрядном режиме для передачи по Modbus 32-х разрядное целое значение преобразуется в его 16-ти разрядный эквивалент. При этом, значения переменных, выходящие за пределы 16-ти бит, ограничиваются в соответствии с текущей конфигурацией обмена. Варианты ограничений представлены в таблице 3.

Таблица 3 – Ограничение переменных при чтении

Значение в словаре обмена	Значение регистра Modbus	
	Знаковый	Без знаковый
-32769 (0FFFF7FFFh)	-32768 (8000h)	0 (0000h)
32768 (00008000h)	32767 (7FFFh)	32768 (8000h)
65536 (00010000h)	32767 (7FFFh)	65535 (0FFFFh)

Принятые для записи значения дополняются до 32-х бит. В таблице 4 показаны возможные варианты дополнений.

Таблица 4 – Дополнение переменных при записи

Значение, записанное в регистр Modbus	Значение, полученное в словаре обмена	
	Знаковый	Без знаковый

3. ПРОТОКОЛ MODBUS-RTU SLAVE

0FFFFh	-1	65535
8000h	-32768	32768
7FFFh	32767	32767

Для того чтобы упростить процесс настройки связи мастер-контроллера Modbus, на модуле может быть включен режим контроля доступа к памяти. В этом режиме исполнительная система модуля производит верификацию принятых команд и возвращает ответ исключение при обнаружении любой из следующих ситуаций:

- выход за адресное пространство словаря обмена, попытка чтения или записи несуществующей переменной;
- проба записи или чтения одной из частей 32-х разрядной переменной, деление переменной на два регистра.

В некоторых случаях контроль доступа может предотвратить появление ошибок в настройках карты адресов мастер-контроллера Modbus. При отключенном режиме контроля доступа модуль предоставляет полное адресное пространство для записи и чтения; запись несуществующих переменных отбрасывается, а чтение возвращает ноль.

Для **мастер-модулей M841E/M902E/M903E/M915E/M921E** режим доступа к переменным осуществляется в зависимости от их расположения на карте адресов Modbus. К переменной технологического приложения с Modbus-адресом m следует обращаться по адресу $(m-1)$. Сами адреса переменным должны назначаться следующим образом:

- для активации 16-ти разрядного режима адреса назначаются подряд:

Адрес	Переменная	Тип
0001	var_modbus1	Целая
0002	var_modbus2	Целая
0003	var_modbus3	Целая
0004	var_modbus4	Целая
0005	var_modbus5	Целая
0006	var_end	Целая
0007		
0008		
0009		
000A		

Рисунок 5

При этом для чтения и записи по Modbus будут доступны только младшие 2 байта переменных.

- для активации 32-х разрядного режима адреса назначаются “через один”:

Адрес	Переменная	Тип
0001	var_modbus1	Целая
0002		
0003	var_modbus2	Целая
0004		
0005	var_modbus3	Целая
0006		
0007	var_modbus4	Целая
0008		
0009	var_modbus5	Целая
000A		

Рисунок 6

При этом, например, чтение регистра по адресу 0 вернет значение младших 16 разрядов переменной var_modbus1, чтение по адресу 1 – старших 16 разрядов. Запись только старшей части 32-х разрядной переменной не допускается.

- для чтения/записи вещественных переменных двойной точности адреса назначаются “через три”:

3. ПРОТОКОЛ MODBUS-RTU SLAVE

Адрес	Переменная	Тип
0001	var_modbus1	Вещ. дв. точности
0002		
0003		
0004		
0005	var_modbus2	Вещ. дв. точности
0006		
0007		
0008		
0009	var_modbus3	Вещ. дв. точности
000A		
000B		
000C		
000D	var_modbus4	Вещ. дв. точности
000E		
000F		
0010		
0011	var_modbus5	Вещ. дв. точности
0012		

Рисунок 7

При этом, например, чтение регистра по адресу 0 вернет значение младших 16 разрядов переменной var_modbus1, чтение по адресу 1 – разрядов 16..31, 2 – разрядов 32..47, 3 – разрядов 48..63. Запись только одной из частей вещественной переменной двойной точности не допускается.

3.3 Modbus-RTU Slave на мастер-модуле

Для работы мастер-модуля **M911E** в качестве master-устройства на линии Modbus используется ECAT (сменный коммуникационный адаптер), в качестве которого установлен модуль-мезонин MACU. Модуль MACU реализует физический интерфейс RS485.

Для мастер-модулей **M841E/M902E/M921E/M915E/M903E** задействованы COM порты COM1..COM9 (количество COM портов, реально присутствующих на мастер-модуле, зависит от комплектации).

3. ПРОТОКОЛ MODBUS-RTU SLAVE

3.4 Modbus-RTU Slave на интеллектуальном модуле

Существует только одна линия связи интеллектуального модуля с мастер-устройством. На этой линии можно использовать либо протокол ST-BUS-M, либо протокол Modbus-RTU Slave. Таким образом, использование ST-BUS-M одновременно с Modbus-RTU Slave становится невозможным.

Активация Slave-интерфейса происходит либо через функциональный блок MB_PARAM, либо вручную установкой переключателя скорости RATE в положения '8'..'F'. Каждой позиции переключателя соответствует одна из скоростей передачи (таблица 5).

Таблица 5 – Скорости передачи Modbus

Положение	Скорость	Положение	Скорость
8	1200 бод	C	19200 бод
9	2400 бод	D	38400 бод
A	4800 бод	E	57600 бод
B	9600 бод	F	115200 бод

По умолчанию принимаются следующие установки Modbus Slave:

- адрес модуля в сети Modbus устанавливается переключателями ADRL и ADRH;
- формат слова: 8 бит, контроль четности отключен, 1 стоповый бит;
- 32-х разрядный знаковый режим обмена переменными для функций Modbus, относящихся к Input Registers и Holding Registers;
- мастер-контроллеру предоставляется полное адресное пространство словаря обмена модуля, контроль выхода за границы адресного пространства отключен.

После выполнения функционального блока MB_PARAM эти установки перекрываются значениями, заданными в MB_PARAM.

Режим работы интерфейса RS-485 интеллектуального модуля и подключение согласующих резисторов/терминаторов линии устанавливаются с помощью соответствующих джамперов и переключателей (таблицы 6 и 7).

Таблица 6 – Режим работы интерфейса RS-485

JP2	Режим работы
2-3, 4-5	Полный дуплекс
1-2, 3-4	Полудуплекс с дублированием
1-2, 3-4	Полудуплекс, пара 1 / пара 2*

Примечания:

1 Цифры соответствуют номерам контактов, на которые устанавливаются джамперы.

2 * неиспользуемая пара не подключается к разъему.

Таблица 7 – Подключение согласующих резисторов

Tr	Вкл	Выкл
Положение переключателя	ON (вверх)	OFF (вниз)

Примечание: Также можно ориентироваться по цветным меткам на корпусе, нанесенным со стороны включенного положения движка переключателя.

Примечание: Подробная информация о переключателях представлена в документе "Исполнительная система Unimod Pro. Руководство пользователя".

При необходимости настройки Slave-интерфейса могут быть изменены через функциональный блок MB_PARAM.

4. ПРОТОКОЛ MODBUS ETHERNET

4.1 Общие сведения

Протокол Modbus ETHERNET (МВЕ) предназначен для передачи обычных пакетов Modbus-RTU через TCP соединение. Другое часто используемое название этого протокола: Modbus на TCP/IP.

Устройство, поддерживающее МВЕ протокол, может работать в одном из двух режимов МВЕ–клиент или МВЕ–сервер. МВЕ–клиент всегда является инициатором начала обмена. Он устанавливает TCP соединение и посылает запросы серверу. Таким образом, режим МВЕ–клиент является аналогом режима Master в Modbus-RTU, а режим МВЕ–сервер - аналогом режима Slave. Соответственно, устройство, работающее в режиме МВЕ–сервер, должно иметь свой адрес (Modbus-адрес) по аналогии со slave-устройством.

Для установки соединения по протоколу МВЕ также задаются параметры, необходимые для инициализации TCP/IP соединения: IP-адрес и порт.

4.2 Modbus ETHERNET на мастер-модуле

Для использования MBE на мастер-модуле **M911E** в качестве ECAT должен быть установлен мезонин WIZNET. Работа по протоколу MBE в этом случае возможна только в режиме сервера.

Мастер-модуль M911E может работать только с одним клиентом по данному протоколу. Другой клиент может установить TCP соединение только когда предыдущий закончит свою работу. В случае прекращения запросов от клиента, установившего соединение, в течение некоторого времени – таймаута, мастер-модуль закрывает соединение и ожидает новых запросов на установку соединения.

Такие параметры, как таймаут ожидания запроса, номер порта и Modbus-адрес устанавливаются через функциональный блок MB_PARAM. IP-адрес задается в параметрах WIZNET, в конфигурации мастер-модуля с помощью программы UMDiag (программа UMDiag подробно описана в документе “Unimod Pro. Руководство пользователя”).

Для мастер-модулей **M841E/M902E/M903E/M915E/M921E**, а также для модуля **M932C2** доступна работа как в режиме master, так и в режиме slave. При этом доступны IP порты в диапазоне 502..65535. В режиме Modbus master такие параметры, как IP-адрес и порт подчиненного устройства, задаются через специальный ресурс (подробнее см. документ “ Unimod Pro. Руководство пользователя”, описание ФБ MB_PARAM).

5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM

Функциональный блок MB_PARAM, подробное описание которого приведено в документе "Unimod Pro. Руководство по программированию", служит для конфигурации связи по протоколу Modbus.

MB_PARAM имеет особенности реализации в соответствии с типом модуля.

5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM

5.1 MB_PARAM на мастер-модуле

5.1.1 Мастер-модуль ECO - M1010E, M1011E, M1012E

Параметр ID_DRV может принимать следующие значения:

- 0 интерфейс RS485 непосредственно на мастер-модуле.
- 1-32 интерфейс RS485 на удаленном модуле 1070S, где адрес 1070S по ITBUS равен ID_DRV.
- 33-38 MODBUS на TCP/IP, только на мастере M1011E, соответственно шесть сокетов.

При выполнении MB_PARAM с ID_DRV=0, необходимо предварительно в конфигурации мастера установить для интерфейса RS485 протокол MODBUS-MASTER.

В противном случае MB_PARAM вернет ошибку 138. Собственно, MB_PARAM с ID_DRV=0 можно не выполнять в приложении, если нет необходимости изменять настройки MODBUS заданные в конфигурации мастера.

5.1.2 Мастер-модуль M911E

Параметр ID_DRV может принимать следующие значения:

- 1 – встроенный коммуникационный адаптер ICAT в режиме Modbus-RTU Master
- 2 – сменный коммуникационный адаптер ECAT в режиме Modbus-RTU Slave
- 3 – сменный коммуникационный адаптер ECAT в режиме Modbus ETHERNET

В случае отсутствия заданного устройства на плате мастера или его неисправности, возвращается код ошибки 4 – “коммуникационный адаптер не отвечает”.

Параметр RATE определяет номер порта для TCP/IP соединения.

Таблица трансляции адресов Modbus-RTU Slave для модуля M911E состоит из 4-х конфигурационных параметров, объединенных в массив целых значений. Детальное описание отдельных полей этого массива приведено в таблице 8. Если размер ресурса не равен $4 * 4 = 16$ байта, формат таблицы считается некорректным, а функциональный блок возвращает код ошибки 140.

Таблица 8 – Таблица трансляции адресов Modbus модуля M911E

№ ячейки	Группа памяти Modbus	Описание
0	COILS	Начальный адрес
1	CONTACTS	Начальный адрес
2	INPUT REGISTERS	Начальный адрес
3	HOLDING REGISTERS	Начальный адрес

В режиме MBE мастер-модуля параметры PARITY, ORDER, PARAM, DELAY не используются. Параметр TOUT содержит время ожидания MBE сервером очередного запроса от клиента. При отсутствии запроса и истечении таймаута сервер разрывает TCP соединение и переводит сокет в состояние LISTEN (прослушивание).

5.1.3 Мастер-модуль M841E / M902E / M903E / M915E / M921E

Параметр ID_DRV может принимать следующие значения:

- 1..9 – номер COM порта (активация режима Modbus-RTU)
- 502..65535 – идентификатор связи в режиме Modbus-TCP Master
- 502..65535 – номер локального IP-порта в режиме Modbus-TCP Slave

Параметр ID_TAB задает идентификатор ресурса, который содержит:

- IP-адрес и порт подчиненного устройства (в режиме Modbus-TCP Master)
- Таблицу трансляции адресов Modbus (в режиме Slave)

5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM

Таблица трансляции адресов задается через специальный тип ресурса:

Имя	Значение	Комментарий
001. Таблица трансляции адресов Modbus	0	Смещение адресов Modbus для: Coils (Дискретные выходы)
	0	Contacts (Дискретные входы)
	0	Input Registers (Входные регистры)
	0	Holding Registers (Выходные регистры)

Рисунок 8 – Таблица трансляции адресов Modbus

На мастер-модуле в зависимости от параметров ID_DRV и SLAVE должна быть запущена соответствующая задача связи (*mb_rtu_mst*, *mb_rtu_slv*, *mb_tcp_mst*, *mb_tcp_slv*). Подробнее см. документ “Unimod Pro. Исполнительная система”, пункт “Система исполнения Unimod Pro”.

Таким образом, для инициализации Modbus необходимо выполнить следующее:

1. Для режима **Slave** переменным, к которым предполагается доступ по Modbus, задать адреса непосредственно в словаре технологического приложения или через меню “Инструменты”->”Карта адресов Modbus”:

Имя	Тип	Размер	Массив	Значение	Атрибуты	Чтение/запись	Хранить/резерв	Доступ	Ед.изм.	MODBUS
var_int16_1	целый			11	внутренняя		нет	полный		0001
var_int16_2	целый			12	внутренняя		нет	полный		0002
var_int16_3	целый			13	внутренняя		нет	полный		0003
var_int16_4	целый			14	внутренняя		нет	полный		0004
var_int16_5	целый			15	внутренняя		нет	полный		0005
var_boo_1	булевский			TRUE	внутренняя		нет	полный		0010
var_boo_2	булевский			FALSE	внутренняя		нет	полный		0011
var_boo_3	булевский			TRUE	внутренняя		нет	полный		0012
var_boo_4	булевский			FALSE	внутренняя		нет	полный		0013
var_boo_5	булевский			TRUE	внутренняя		нет	полный		0014
var_rea_1	вещественный			1.230000	внутренняя		нет	полный		0020
var_rea_2	вещественный			2.340000	внутренняя		нет	полный		0022
var_rea_3	вещественный			3.450000	внутренняя		нет	полный		0024
var_rea_4	вещественный			4.560000	внутренняя		нет	полный		0026
var_rea_5	вещественный			5.670000	внутренняя		нет	полный		0028
var_int32_1	целый			123456789	внутренняя		нет	полный		0030
var_int32_2	целый			234567891	внутренняя		нет	полный		0032
var_int32_3	целый			345678912	внутренняя		нет	полный		0034
var_int32_4	целый			456789123	внутренняя		нет	полный		0036
var_int32_5	целый			567891234	внутренняя		нет	полный		0038

Рисунок 9

Переменные без адресов:		Переменные на карте адресов:			
Имя	Тип	Адрес	Переменная	Тип	Комментарий
Булевские					
Целые		0001	var_int16_1	Целая	
Вещественные		0002	var_int16_2	Целая	
Массивы булевских		0003	var_int16_3	Целая	
Массивы целых		0004	var_int16_4	Целая	
Массивы вещественных		0005	var_int16_5	Целая	
		0006			
		0007			

Рисунок 10 – Карта адресов Modbus

Особенности задания адресов описаны в разделе “ПРОТОКОЛ MODBUS-RTU SLAVE”

2. Запустить задачу связи:

5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM

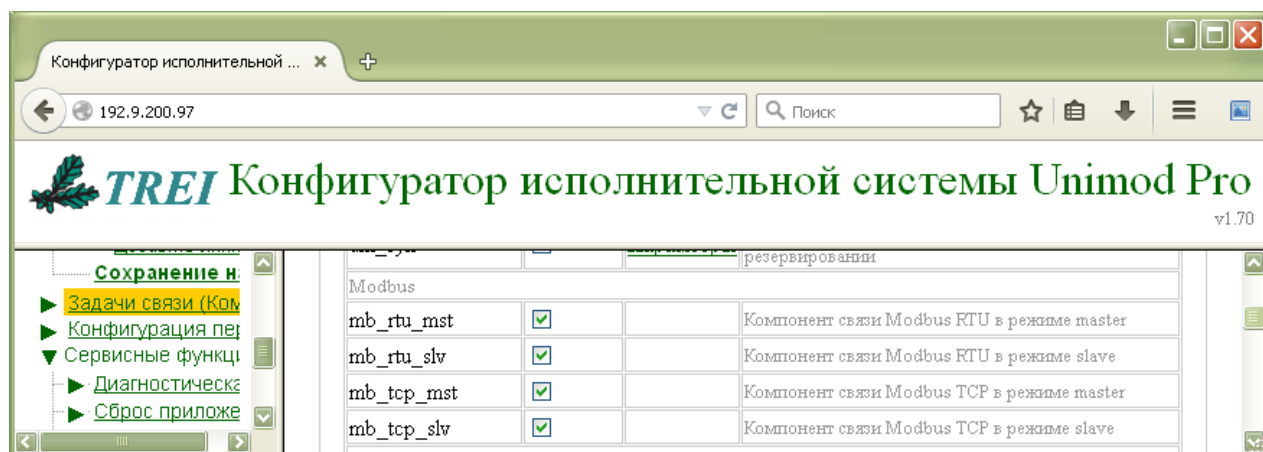


Рисунок 11 – Web-конфигуратор

3. Выполнить вызов функционального блока MB_PARAM с необходимыми параметрами (см. документ “Unimod Pro. Руководство по программированию”).
4. **Для режима Master** выполнить вызов блоков чтения/записи в соответствии с функциями контроллера.

Примечание. Если выполняется вызов функциональных блоков MB_R или MB_W, пункт 1 также необходимо выполнять.

5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM

5.1.4 Мастер-модуль M401E

Для мастер-модуля M401E блоки MB_PARAM и блоки чтения/записи (MB_*) не используются. Параметры физической линии и режим обмена устанавливаются на вкладке “Межконтроллерный обмен” (см. документ “TREI_UNIMOD_USER.pdf”, раздел “Межконтроллерный Обмен”).

Типовая последовательность действий для инициализации Modbus следующая:

1. Переменным, к которым предполагается доступ по Modbus, задать адреса непосредственно в словаре технологического приложения или через меню “Инструменты”->”Карта адресов Modbus”:

Имя	Тип	Размер	Массив	Значение	Атрибуты	Чтение/запись	Хранить/резерв	Доступ	Ед.изм.	MODBUS (10)
<input checked="" type="checkbox"/> var_int16_1	целый			11	внутренняя		нет	полный		1
<input checked="" type="checkbox"/> var_int16_2	целый			12	внутренняя		нет	полный		2
<input checked="" type="checkbox"/> var_int16_3	целый			13	внутренняя		нет	полный		3
<input checked="" type="checkbox"/> var_int16_4	целый			14	внутренняя		нет	полный		4
<input checked="" type="checkbox"/> var_int16_5	целый			15	внутренняя		нет	полный		5
<input type="checkbox"/> var_boo_1	булевский			TRUE	внутренняя		нет	полный		16
<input type="checkbox"/> var_boo_2	булевский			FALSE	внутренняя		нет	полный		17
<input type="checkbox"/> var_boo_3	булевский			TRUE	внутренняя		нет	полный		18
<input type="checkbox"/> var_boo_4	булевский			FALSE	внутренняя		нет	полный		19
<input type="checkbox"/> var_boo_5	булевский			TRUE	внутренняя		нет	полный		20
<input checked="" type="checkbox"/> var_rea_1	вещественный			1.230000	внутренняя		нет	полный		32
<input checked="" type="checkbox"/> var_rea_2	вещественный			2.340000	внутренняя		нет	полный		34
<input checked="" type="checkbox"/> var_rea_3	вещественный			3.450000	внутренняя		нет	полный		36
<input checked="" type="checkbox"/> var_rea_4	вещественный			4.560000	внутренняя		нет	полный		38
<input checked="" type="checkbox"/> var_rea_5	вещественный			5.670000	внутренняя		нет	полный		40
<input checked="" type="checkbox"/> var_int32_1	целый			123456789	внутренняя		нет	полный		48
<input checked="" type="checkbox"/> var_int32_2	целый			234567891	внутренняя		нет	полный		50
<input checked="" type="checkbox"/> var_int32_3	целый			345678912	внутренняя		нет	полный		52
<input checked="" type="checkbox"/> var_int32_4	целый			456789123	внутренняя		нет	полный		54
<input checked="" type="checkbox"/> var_int32_5	целый			567891234	внутренняя		нет	полный		56

Рисунок 12

Адрес	Переменная	Тип	Комментарий
0001	var_int16_1	Целая	
0002	var_int16_2	Целая	
0003	var_int16_3	Целая	
0004	var_int16_4	Целая	
0005	var_int16_5	Целая	
0006			
0007			

Рисунок 13 – Карта адресов Modbus

Особенности задания адресов описаны в разделе “ПРОТОКОЛ MODBUS-RTU SLAVE”

2. Добавить “задачу связи Modbus” на вкладке “Межконтроллерный обмен”:

5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM

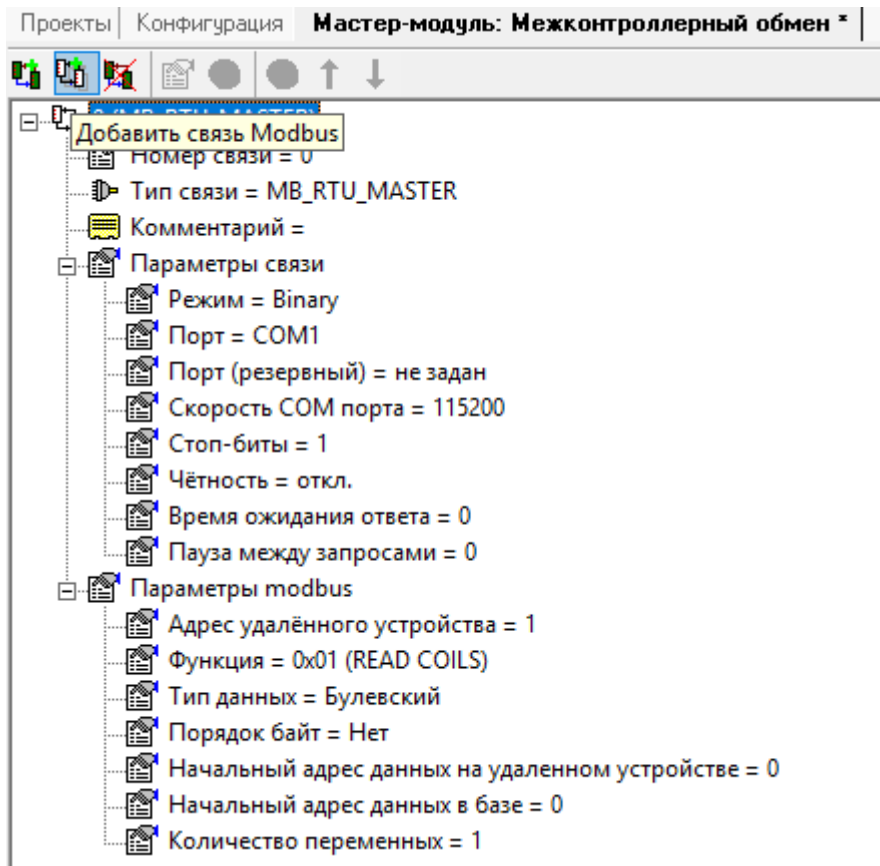


Рисунок 14 – Параметры задачи связи Modbus

3. Запустить соответствующую задачу связи на контроллере:

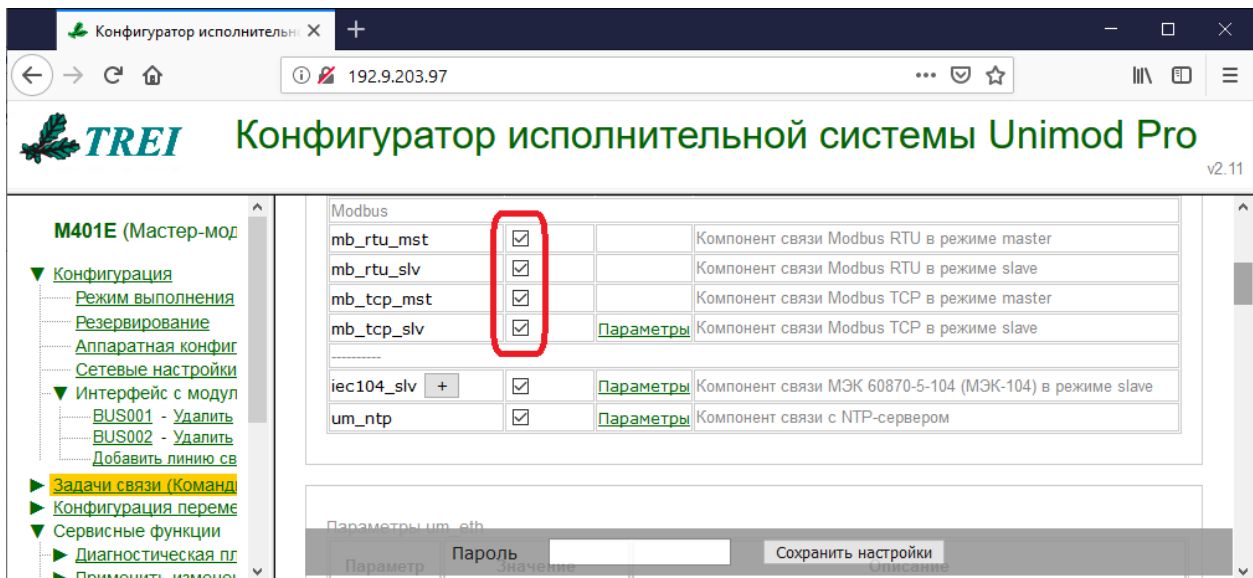


Рисунок 15 – Web-конфигуратор

4. Для режима Master контроль над выполнением запросов выполняется с помощью системного вызова system с кодами 36,37,38.

5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM

5.1.5 Мастер-модуль M501E

Для мастер-модуля M501E работа по Modbus может быть организована как через блоки с префиксом “MB_” (см. документ “TREI_UNIMOD_PROG.pdf”), так и через задачи связи на вкладке “Межконтроллерный обмен” (см. документ “TREI_UNIMOD_USER.pdf”, раздел “Межконтроллерный Обмен”).

5.1.5.1 Использование блоков с префиксом “MB_”

Описание блоков приведено в документе “TREI_UNIMOD_PROG.pdf”)

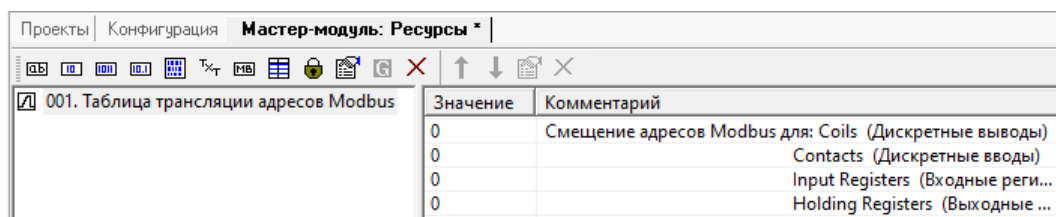
Параметр ID_DRV может принимать следующие значения:

- 1..9 – номер COM порта (активация режима Modbus-RTU)
- 502..65535 – идентификатор связи в режиме Modbus-TCP Master
- 502..65535 – номер локального IP-порта в режиме Modbus-TCP Slave

Параметр ID_TAB задает идентификатор ресурса, который содержит:

- IP-адрес и порт подчиненного устройства (в режиме Modbus-TCP Master)
- Таблицу трансляции адресов Modbus (в режиме Slave)

Таблица трансляции адресов задается через специальный тип ресурса:



Значение	Комментарий
0	Смещение адресов Modbus для: Coils (Дискретные выходы)
0	Contacts (Дискретные входы)
0	Input Registers (Входные реги...
0	Holding Registers (Выходные ...

Рисунок 16 – Таблица трансляции адресов Modbus

На мастер-модуле в зависимости от параметров ID_DRV и SLAVE должна быть запущена соответствующая задача связи (*mb_rtu_mst*, *mb_rtu_slv*, *mb_tcp_mst*, *mb_tcp_slv*). Подробнее см. документ “Unimod Pro. Исполнительная система”, пункт “Система исполнения Unimod Pro”.

Таким образом, для инициализации Modbus необходимо выполнить следующее:

1. Для режима **Slave** (и для режима **Master** при использовании блоков **MB_R/MB_W**) переменным, к которым предполагается доступ по Modbus, задать адреса непосредственно в словаре технологического приложения или через меню “Инструменты”->”Карта адресов Modbus”:

5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM

Проекты | Конфигурация | Мастер-модуль: Глобальный словарь

Имя	Тип	Размер	Массив	Значение	Атрибуты	Чтение/запись	Хранить/резерв	Доступ	Ед.изм.	MODBUS (10)
<input checked="" type="checkbox"/> var_int16_1	целый			11	внутренняя		нет	полный	1	
<input checked="" type="checkbox"/> var_int16_2	целый			12	внутренняя		нет	полный	2	
<input checked="" type="checkbox"/> var_int16_3	целый			13	внутренняя		нет	полный	3	
<input checked="" type="checkbox"/> var_int16_4	целый			14	внутренняя		нет	полный	4	
<input checked="" type="checkbox"/> var_int16_5	целый			15	внутренняя		нет	полный	5	
<input type="checkbox"/> var_boo_1	булевский			TRUE	внутренняя		нет	полный	16	
<input type="checkbox"/> var_boo_2	булевский			FALSE	внутренняя		нет	полный	17	
<input type="checkbox"/> var_boo_3	булевский			TRUE	внутренняя		нет	полный	18	
<input type="checkbox"/> var_boo_4	булевский			FALSE	внутренняя		нет	полный	19	
<input type="checkbox"/> var_boo_5	булевский			TRUE	внутренняя		нет	полный	20	
<input type="checkbox"/> var_rea_1	вещественный			1.230000	внутренняя		нет	полный	32	
<input type="checkbox"/> var_rea_2	вещественный			2.340000	внутренняя		нет	полный	34	
<input type="checkbox"/> var_rea_3	вещественный			3.450000	внутренняя		нет	полный	36	
<input type="checkbox"/> var_rea_4	вещественный			4.560000	внутренняя		нет	полный	38	
<input type="checkbox"/> var_rea_5	вещественный			5.670000	внутренняя		нет	полный	40	
<input checked="" type="checkbox"/> var_int32_1	целый			123456789	внутренняя		нет	полный	48	
<input checked="" type="checkbox"/> var_int32_2	целый			234567891	внутренняя		нет	полный	50	
<input checked="" type="checkbox"/> var_int32_3	целый			345678912	внутренняя		нет	полный	52	
<input checked="" type="checkbox"/> var_int32_4	целый			456789123	внутренняя		нет	полный	54	
<input checked="" type="checkbox"/> var_int32_5	целый			567891234	внутренняя		нет	полный	56	

Рисунок 17

Карта адресов

Файл Редактор

0..1FFF 2000..3FFF 4000..5FFF 6000..7FFF 8000..9FFF A000..BFFF C000..DFFF

Переменные без адресов: Булевские, Целые, Вещественные, Массивы булевских, Массивы целых, Массивы вещественных, Блоки данных

Переменные на карте адресов:

Адрес	Переменная	Тип	Комментарий
0001	var_int16_1	Целая	
0002	var_int16_2	Целая	
0003	var_int16_3	Целая	
0004	var_int16_4	Целая	
0005	var_int16_5	Целая	
0006			
0007			

Рисунок 18 – Карта адресов Modbus

Особенности задания адресов описаны в разделе “ПРОТОКОЛ MODBUS-RTU SLAVE”

2. Запустить задачу связи:

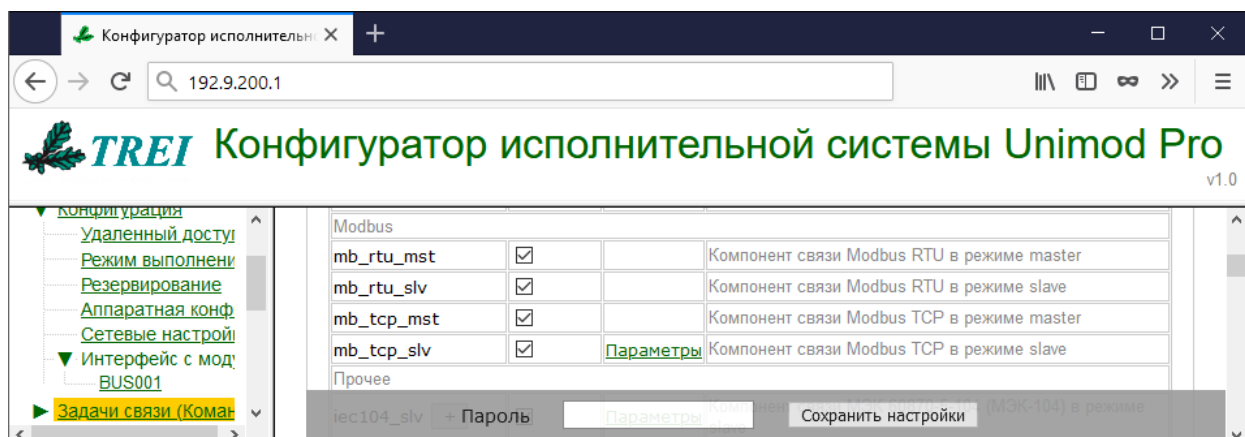


Рисунок 19 – Web-конфигуратор

3. Выполнить вызов функционального блока MB_PARAM с необходимыми параметрами (см. документ “Unimod Pro. Руководство по программированию”).
4. Для режима Master выполнить вызов блоков чтения/записи в соответствии с функциями контроллера.

5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM

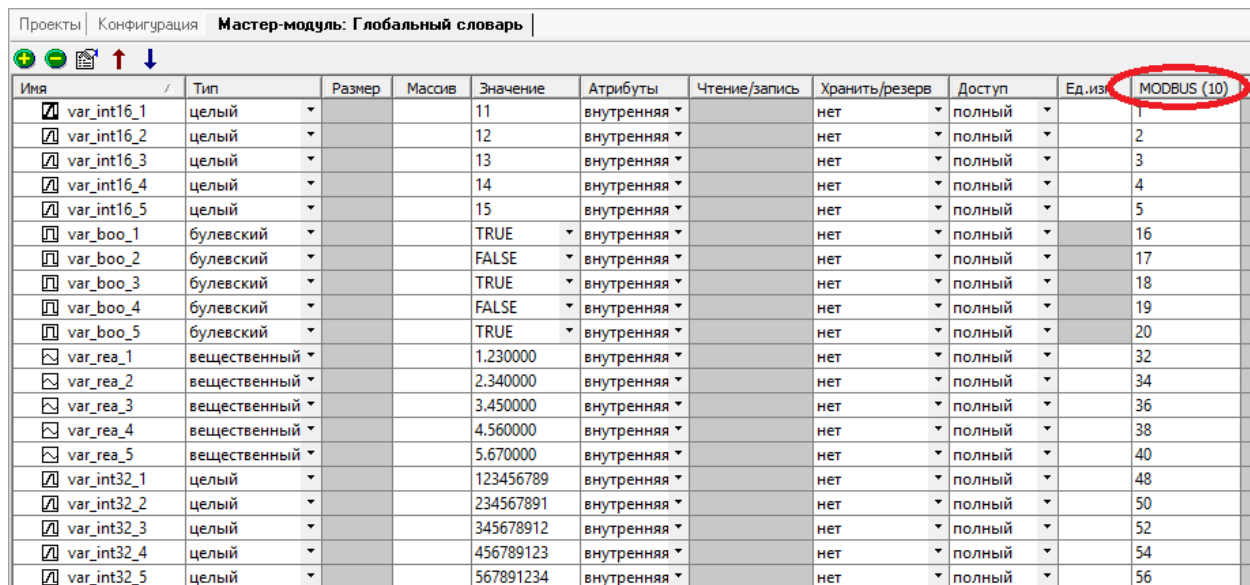
Примечание. Если выполняется вызов функциональных блоков MB_R или MB_W, пункт 1 также необходимо выполнять.

5.1.5.2 Использование задач связи на вкладке “Межконтроллерный обмен”

Описание вкладки приведено в документе “TREI_UNIMOD_USER.pdf”, раздел “Межконтроллерный Обмен”.

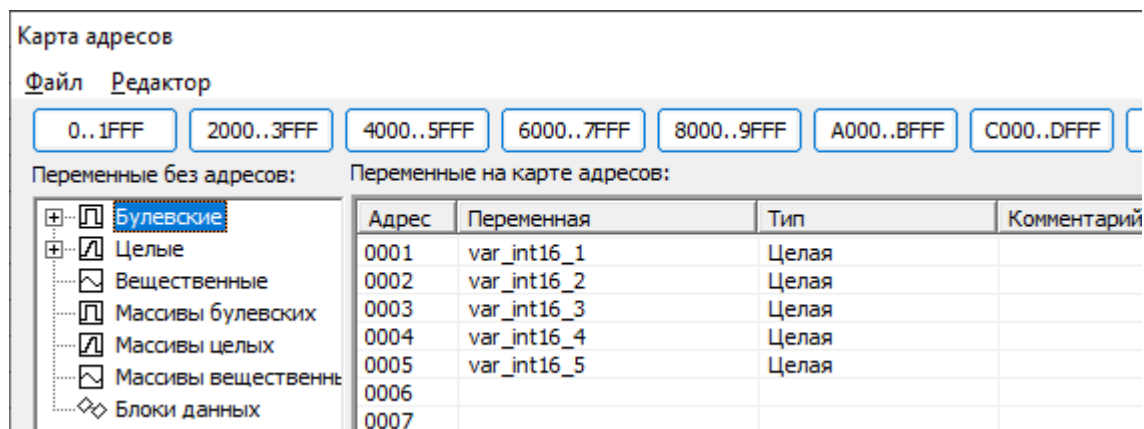
Типовая последовательность действий для инициализации Modbus следующая:

1. Переменным, к которым предполагается доступ по Modbus, задать адреса непосредственно в словаре технологического приложения или через меню “Инструменты”->”Карта адресов Modbus”:



Имя	Тип	Размер	Массив	Значение	Атрибуты	Чтение/запись	Хранить/резерв	Доступ	Ед.изм.	MODBUS (10)
<input checked="" type="checkbox"/> var_int16_1	целый			11	внутренняя		нет	полный		1
<input checked="" type="checkbox"/> var_int16_2	целый			12	внутренняя		нет	полный		2
<input checked="" type="checkbox"/> var_int16_3	целый			13	внутренняя		нет	полный		3
<input checked="" type="checkbox"/> var_int16_4	целый			14	внутренняя		нет	полный		4
<input checked="" type="checkbox"/> var_int16_5	целый			15	внутренняя		нет	полный		5
<input type="checkbox"/> var_boo_1	булевский			TRUE	внутренняя		нет	полный		16
<input type="checkbox"/> var_boo_2	булевский			FALSE	внутренняя		нет	полный		17
<input type="checkbox"/> var_boo_3	булевский			TRUE	внутренняя		нет	полный		18
<input type="checkbox"/> var_boo_4	булевский			FALSE	внутренняя		нет	полный		19
<input type="checkbox"/> var_boo_5	булевский			TRUE	внутренняя		нет	полный		20
<input checked="" type="checkbox"/> var_rea_1	вещественный			1.230000	внутренняя		нет	полный		32
<input checked="" type="checkbox"/> var_rea_2	вещественный			2.340000	внутренняя		нет	полный		34
<input checked="" type="checkbox"/> var_rea_3	вещественный			3.450000	внутренняя		нет	полный		36
<input checked="" type="checkbox"/> var_rea_4	вещественный			4.560000	внутренняя		нет	полный		38
<input checked="" type="checkbox"/> var_rea_5	вещественный			5.670000	внутренняя		нет	полный		40
<input checked="" type="checkbox"/> var_int32_1	целый			123456789	внутренняя		нет	полный		48
<input checked="" type="checkbox"/> var_int32_2	целый			234567891	внутренняя		нет	полный		50
<input checked="" type="checkbox"/> var_int32_3	целый			345678912	внутренняя		нет	полный		52
<input checked="" type="checkbox"/> var_int32_4	целый			456789123	внутренняя		нет	полный		54
<input checked="" type="checkbox"/> var_int32_5	целый			567891234	внутренняя		нет	полный		56

Рисунок 20



Карта адресов

Файл Редактор

0..1FFF 2000..3FFF 4000..5FFF 6000..7FFF 8000..9FFF A000..BFFF C000..DFFF

Переменные без адресов: Переменные на карте адресов:

Адрес	Переменная	Тип	Комментарий
0001	var_int16_1	Целая	
0002	var_int16_2	Целая	
0003	var_int16_3	Целая	
0004	var_int16_4	Целая	
0005	var_int16_5	Целая	
0006			
0007			

Рисунок 21 – Карта адресов Modbus

Особенности задания адресов описаны в разделе “ПРОТОКОЛ MODBUS-RTU SLAVE”

2. Добавить “задачу связи Modbus” на вкладке “Межконтроллерный обмен”:

5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM

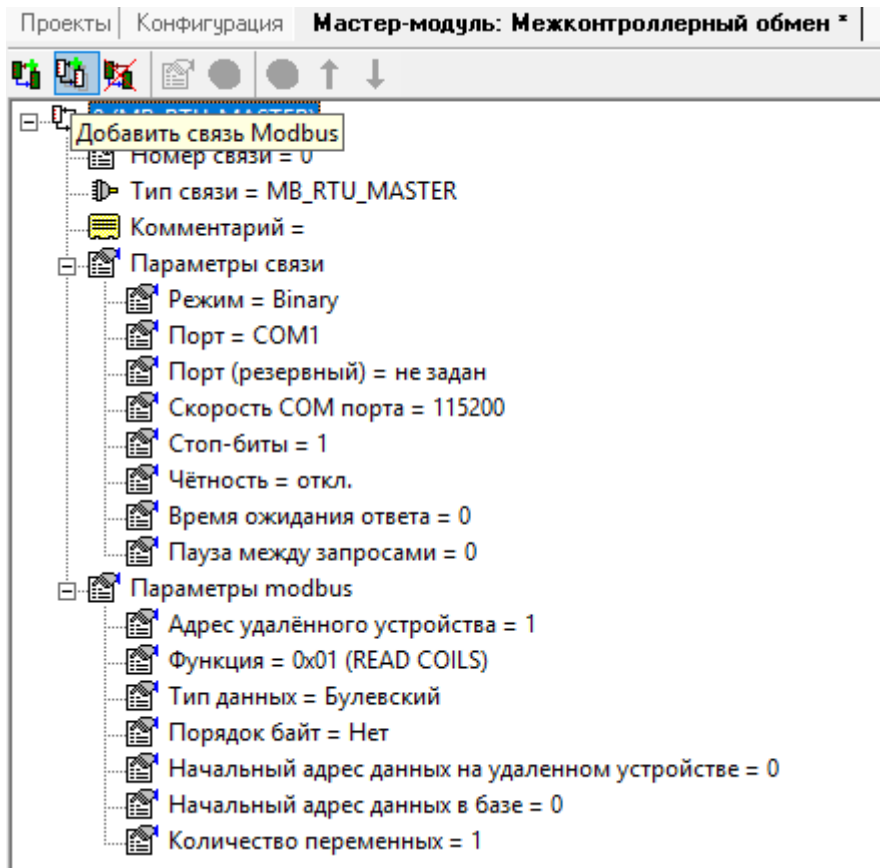


Рисунок 22 – Параметры задачи связи Modbus

3. Запустить соответствующую задачу связи на контроллере:

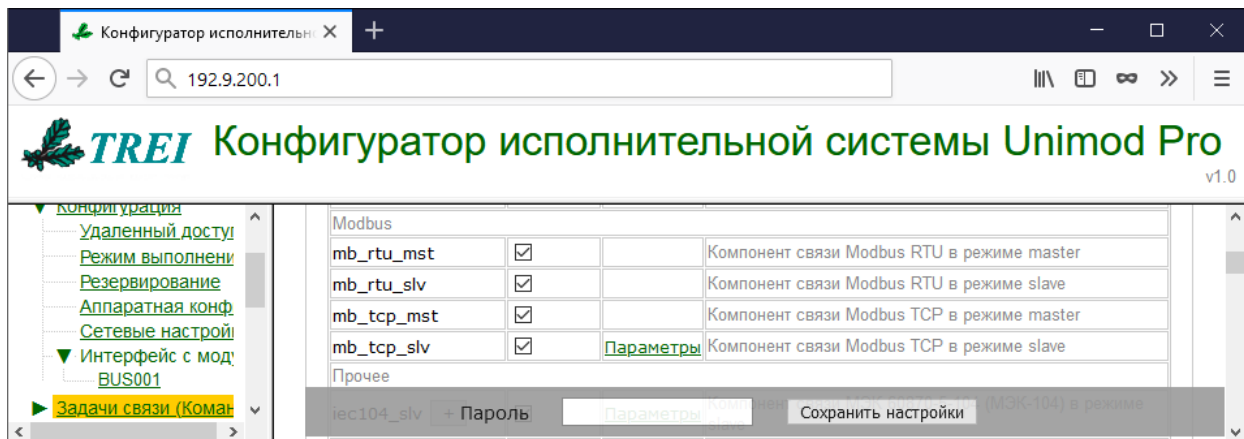


Рисунок 23 – Web-конфигуратор

4. Для режима **Master** контроль над выполнением запросов выполняется с помощью системного вызова system с кодами 36,37,38.

5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM

5.2 MB_PARAM на интеллектуальном модуле (кроме **M932C2**).

Интеллектуальный модуль серии M900 различает следующие идентификаторы устройств (параметр ID_DRV):

- 0 – Интерфейс Modbus-RTU Slave через основную последовательную линию вместо STBUS;
- 1 – Интерфейс Modbus-RTU Master через юнит STBU.

Таблица трансляции адресов Modbus-RTU Slave для интеллектуального модуля состоит из 8-ми конфигурационных параметров, объединенных в массив целых значений. Детальное описание отдельных полей этого массива приведено в таблице 9. Если размер ресурса не равен $4 * 8 = 32$ байта, формат таблицы считается некорректным, а функциональный блок возвращает код ошибки 140.

Таблица 9 – Таблица трансляции адресов Modbus модуля M900

№ ячейки	Группа памяти Modbus	Описание
0	COILS	Начальный адрес
1		Количество переменных
2	CONTACTS	Начальный адрес
3		Количество переменных
4	INPUT REGISTERS	Начальный адрес
5		Количество переменных
6	HOLDING REGISTERS	Начальный адрес
7		Количество переменных

При конфигурировании Master-интерфейса модуля M932C2 параметры TOUT и DELAY задают время ожидания ответа от Slave-устройства и задержку перед передачей и после получения ответа соответственно. Для Slave-устройства параметры TOUT и DELAY задают время обнаружение обрыва (окончания) пакета и задержку перед отправкой ответа.

5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ MB_PARAM

5.3 MB_PARAM на интеллектуальном модуле **M932C2**.

Интеллектуальный модуль M932C2 различает следующие идентификаторы устройств (параметр ID_DRV):

- 0 – Интерфейс Modbus-RTU Slave через юнит STBU;
- 1 – Интерфейс Modbus-RTU Master через юнит STBU;
- 2 – Интерфейс Modbus-RTU Slave через основную последовательную линию вместо STBUS;
- 3 – Интерфейс Modbus-RTU Master через основную последовательную линию вместо STBUS;
- 4 – Интерфейс Modbus-TCP Slave;
- 5 – Интерфейс Modbus-TCP Master.

Таблица трансляции адресов Modbus Slave для интеллектуального модуля состоит из 8-ми конфигурационных параметров, объединенных в массив целых значений. Детальное описание отдельных полей этого массива приведено в таблице 9. Если размер ресурса не равен $4 * 8 = 32$ байта, формат таблицы считается некорректным, а функциональный блок возвращает код ошибки 140.

Таблица 10 – Таблица трансляции адресов Modbus модуля M900

№ ячейки	Группа памяти Modbus	Описание
0	COILS	Начальный адрес
1		Количество переменных
2	CONTACTS	Начальный адрес
3		Количество переменных
4	INPUT REGISTERS	Начальный адрес
5		Количество переменных
6	HOLDING REGISTERS	Начальный адрес
7		Количество переменных

При конфигурировании Master-интерфейса модуля M900 параметры TOUT и DELAY задают время ожидания ответа от Slave-устройства и задержку перед передачей и после получения ответа соответственно. Для Slave-устройства параметры TOUT и DELAY задают время обнаружение обрыва (окончания) пакета и задержку перед отправкой ответа.

В режиме Modbus TCP Master параметры PARITY, ORDER, PARAM не используется. При установке параметров для Modbus, параметр RATE определяет номер порта для TCP/IP соединения. Также в режиме Modbus TCP Master функциональный блок необходимо вызывать в каждом цикле приложения, так как он осуществляет контроль TCP-соединения. В параметре SLAVE необходимо указать Modbus адрес устройства, к которому будут идти запросы.

Порт для режима Modbus TCP Slave задается в конфигурации модуля.