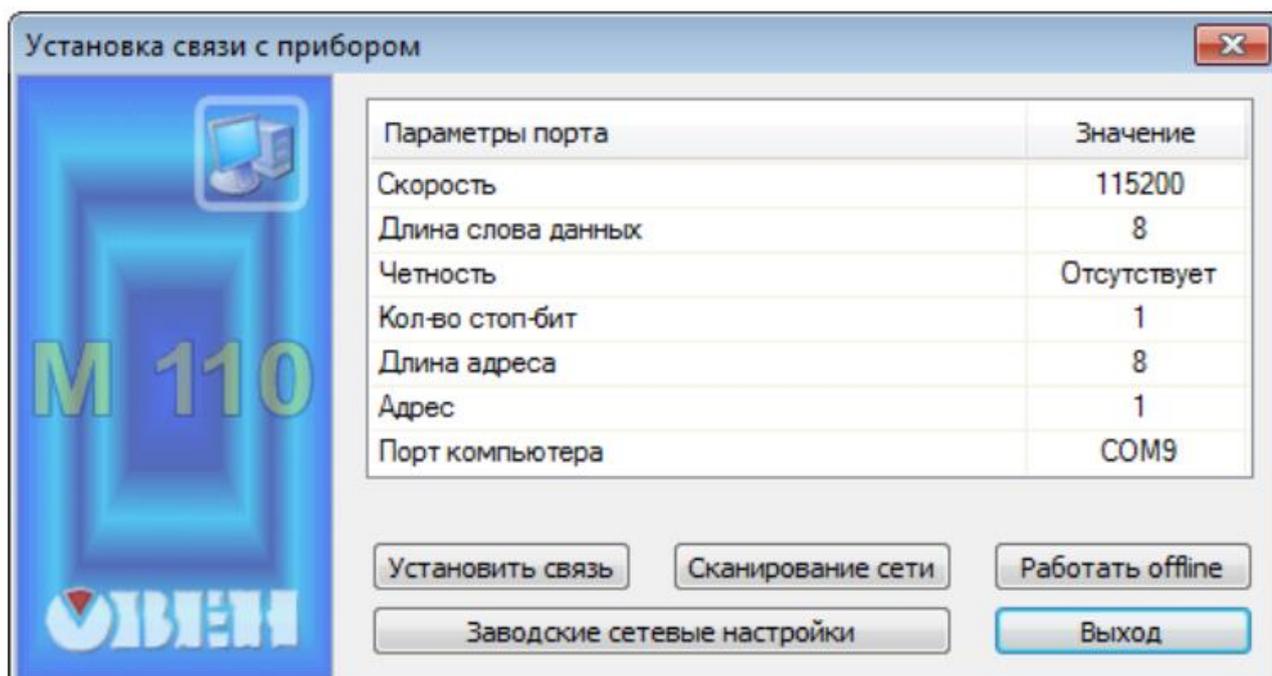


И. А. ЕЛИЗАРОВ, В. Н. НАЗАРОВ,  
А. А. ТРЕТЬЯКОВ, В. А. ПОГОНИН

# ПРОМЫШЛЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ



Тамбов  
Издательский центр ФГБОУ ВО «ТГТУ»  
2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Тамбовский государственный технический университет»

И. А. ЕЛИЗАРОВ, В. Н. НАЗАРОВ,  
А. А. ТРЕТЬЯКОВ, В. А. ПОГОНИН

# ПРОМЫШЛЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

Утверждено Ученым советом университета в качестве учебного пособия  
для студентов 4, 5 курсов, обучающихся по направлению подготовки  
27.03.04 «Управление в технических системах»,  
очной и заочной форм обучения

*Учебное электронное издание*



---

Тамбов  
Издательский центр ФГБОУ ВО «ТГТУ»  
2024

УДК 681.58(075.8)  
ББК Р973.202я73  
П78

Рецензенты:

Доктор технических наук, профессор, профессор кафедры теоретической  
и экспериментальной физики ФГБОУ ВО «ТГУ им. Г. Р. Державина»  
*И. И. Пасечников*

Кандидат технических наук, доцент, заведующий кафедрой  
«Системы автоматизированной поддержки принятия решений»  
ФГБОУ ВО «ТГТУ»  
*И. Л. Коробова*

**П78 Промышленные вычислительные сети** [Электронный ресурс] :  
учебное пособие / И. А. Елизаров, В. Н. Назаров, А. А. Третьяков,  
В. А. Погонин. – Тамбов : Издательский центр ФГБОУ ВО «ТГТУ», 2024. –  
1 электрон. опт. диск (CD-ROM). – Системные требования : ПК не ниже  
класса Pentium II ; CD-ROM-дисковод ; 4,0 Mb ; RAM ; Windows 95/98/  
XP ; мышь. – Загл. с экрана.  
ISBN 978-5-8265-2794-8

Приведены общие сведения о промышленных сетях, показано место промышлен-  
ных сетей в современной автоматизированной системе управления технологическими  
процессами и производством в целом, даны базовые понятия из теории построения  
цифровых сетей, описание распространенных сетевых интерфейсов.

Особое внимание уделено промышленным сетям, построенным с использованием  
протоколов MODBUS (ASCII, RTU, TCP). На примере выполнения лабораторных  
работ рассмотрено практическое применение этих протоколов при использовании  
с контроллерами ОВЕН ПЛК-100/110 и ПЛК-200/210, программируемых в среде  
CoDeSyS.

Предназначено для студентов 4, 5 курсов, обучающихся по направлению  
подготовки 27.03.04 «Управление в технических системах», очной и заочной форм  
обучения.

УДК 681.58(075.8)  
ББК Р973.202я73

*Все права на размножение и распространение в любой форме остаются за разработчиком.  
Нелегальное копирование и использование данного продукта запрещено.*

**ISBN 978-5-8265-2794-8**

© Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Тамбовский государственный технический  
университет» (ФГБОУ ВО «ТГТУ»), 2024

## ВВЕДЕНИЕ

---

Процесс автоматизации промышленных производств развивается все более ускоряющимися темпами: увеличивается количество «интеллектуальных» датчиков и исполнительных устройств, растет число вовлеченных в процессы контроля и управления технологическим процессом вычислительных систем на базе программируемых логических контроллеров и промышленных компьютеров. В этих условиях существенно возрастает роль данных, собираемых на всех уровнях АСУ ТП.

Требования, предъявляемые со стороны потребителей этой информации, все более ужесточаются в части объема, скорости и надежности получения данных, поэтому вопросы обеспечения коммуникаций становятся высокоприоритетными [1].

В течение многих лет автоматизированные системы управления строились по традиционной централизованной схеме, в которой имелось одно мощное вычислительное устройство и огромное количество кабелей, посредством которых осуществлялось подключение оконечных устройств (датчиков и исполнительных механизмов). Использование централизованной структуры объяснялось и относительно низким уровнем автоматизации производства и высокой ценой микропроцессорной управляющей техники.

В последнее время ситуация кардинально изменилась: стоимость меди, и, как следствие, кабельной продукции возрастает, а стоимость микропроцессорной техники стремительно снижается, повышается ее надежность и функциональность, микропроцессоры проникли уже на уровень датчиков и исполнительных устройств. За последние пару десятилетий появилось огромное количество недорогих программируемых логических контроллеров, интеллектуальных реле, модулей ввода-вывода с развитыми сетевыми возможностями и поддержкой различных промышленных сетей. Все это способствует интенсивному росту доли распределенных систем управления во всех отраслях промышленности.

# 1. ОБЩИЕ СВЕДЕНИЯ О ПРОМЫШЛЕННЫХ СЕТЯХ

## 1.1. МЕСТО ПРОМЫШЛЕННЫХ СЕТЕЙ В СОВРЕМЕННОЙ АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ УПРАВЛЕНИЯ

Современные автоматизированные системы управления промышленным предприятием имеют иерархическую структуру, которую принято представлять в виде пирамиды (рис. 1.1). Самый верхний уровень управления предприятием составляют системы планирования потребности в материалах MRP (Material Requirements Planning) и планирования и управления предприятием ERP (Enterprise Resource Planning), которые предназначены для решения задач, связанных с бизнес-процессами предприятия: планирование производства, закупки, сбыт, управление финансами, складским хозяйством и др.

Системы MES (Manufacturing Execution Systems) – производственные исполнительные системы, предназначены для решения оперативных задач управления производством.

Системы MES, MRP, ERP принято относить к автоматизированным системам управления производством (АСУ П).

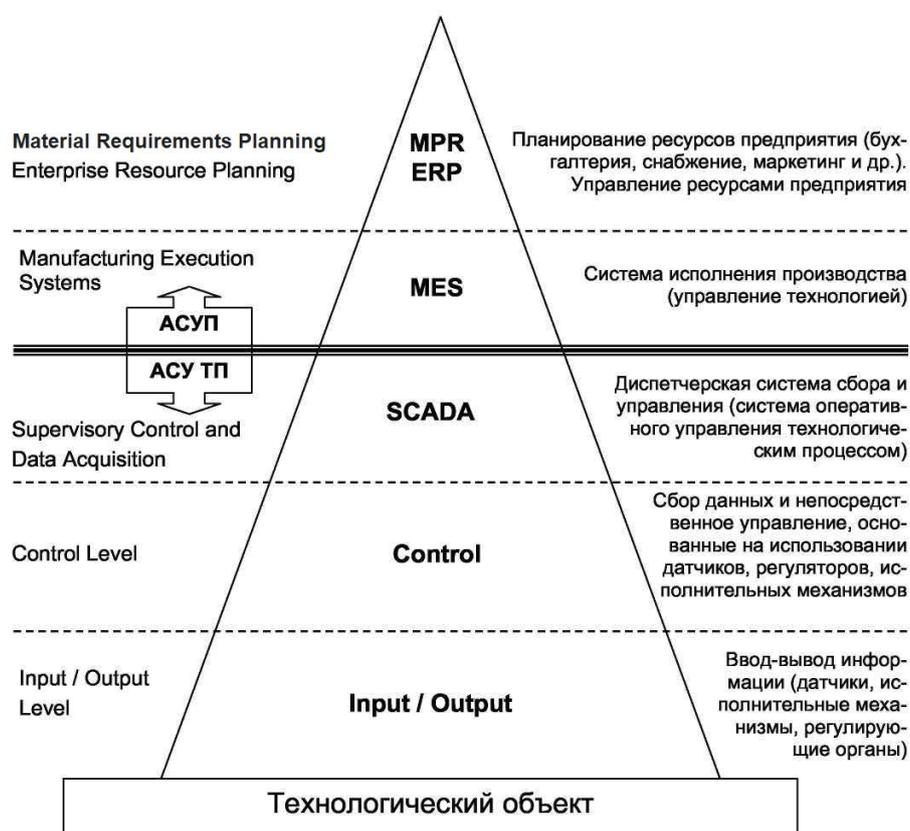


Рис. 1.1. Общая структура управления производством

Автоматизированные системы управления технологическими процессами (АСУ ТП) включают три нижних уровня пирамиды: уровень систем SCADA (Supervisory Control and Data Acquisition), контроллерный уровень (Control level) и уровень конечных устройств (Input/Output level). Информационный обмен между уровнями осуществляется в цифровом виде посредством промышленных вычислительных сетей.

На уровне управления предприятием располагаются компьютеры и файловые серверы, объединенные локальной сетью (LAN). Задача вычислительных систем на этом уровне – обеспечение визуального контроля основных параметров производства, построение отчетов, архивирование данных. Объемы передаваемых между узлами данных измеряются мегабайтами, а временные показатели обмена информацией с более низким уровнем не являются критичными [2].

На уровне управления технологическим процессом (уровень SCADA) осуществляется текущий контроль и управление либо в ручном режиме с операторских пультов, либо в автоматическом режиме по заложенному алгоритму. На этом уровне выполняется согласование параметров отдельных участков производства, отработка аварийных и предаварийных ситуаций, параметризация контроллеров нижнего уровня, загрузка технологических программ, дистанционное управление исполнительными механизмами. Информационный обмен с контроллерным уровнем осуществляется пакетами, которые содержат, как правило, несколько десятков байтов, а допустимые временные задержки могут составлять от 100 до 1000 миллисекунд в зависимости от режима работы.

На уровне управления устройствами располагаются «интеллектуальные» датчики и исполнительные механизмы, модули распределенного ввода/вывода, осуществляющие непосредственный сбор данных от датчиков и управление исполнительными устройствами. Размер данных, которыми контроллер обменивается с конечными устройствами, обычно составляет несколько байтов при скорости опроса устройств не более 10 мс.

В течение многих лет АСУ ТП строились по традиционной централизованной схеме, в которой имелось одно мощное вычислительное устройство и огромное количество кабелей для подключения конечных устройств (датчиков, исполнительных механизмов, устройств отображения информации). Такая структура диктовалась высокой ценой электронно-вычислительной техники [3].

На современном этапе, в условиях бурно растущего производства цифровой техники, стоимость вычислительных устройств зачастую оказывается ниже затрат на кабельно-проводниковую продукцию, и становится экономически целесообразной установка на площадке цеха или участка нескольких локальных контроллеров или интеллектуальных устройств связи с объектом, объединенных в единую сеть передачи данных. Таким образом, современные АСУ ТП являются распределенными системами управления, в которых обмен информацией между различными узлами осуществляется посредством промышленных сетей.

## 1.2. БАЗОВЫЕ ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Промышленная сеть – сеть передачи данных, связывающая различные «интеллектуальные» датчики, исполнительные механизмы, промышленные контроллеры, модули распределенного ввода-вывода и используемая в промышленной автоматизации [3].

В России промышленные сети появились, следуя тенденциям Запада, где, в соответствии со спецификацией IEC 61158, употребляется термин «Fieldbus», что дословно переводится как «полевая шина». Однако в русском языке в настоящее время принято использовать термин «промышленная сеть», что более точно отражает суть организации процесса передачи данных и не вводит никого в заблуждение, так как полевая шина фактически является частным случаем построения промышленной сети с использованием соответствующей топологии ее построения.

Полевая шина (Fieldbus) – это локальная сеть с линейной топологией, предназначенная для информационного обмена между территориально распределенными промышленными инструментальными устройствами: датчиками, преобразователями, средствами ручного ввода, исполнительными механизмами, средствами отображения информации и т.д.

То есть, строго говоря, термины «полевая шина» и «промышленная сеть» в общем случае синонимами не являются, но очень часто используются как таковые у различных авторов, особенно при рассмотрении сетей, построенных по шинной топологии.

Соединение промышленной сети с ее компонентами (устройствами, узлами сети) выполняется с помощью интерфейсов.

Интерфейс – совокупность возможностей, способов и методов взаимодействия двух и более систем, устройств или программ для обмена информацией между ними, определенная их характеристиками, характеристиками соединения, характеристиками сигналов обмена и т. п. Можно рассматривать интерфейс как логическую и (или) физическую границу между устройством и средой передачи информации. Для обмена информацией взаимодействующие устройства должны иметь одинаковый протокол обмена.

Протокол обмена – это набор правил, которые управляют обменом информацией. Протокол обмена определяет синтаксис и семантику сообщений, операции управления, синхронизацию и состояния при коммуникации. Он может быть реализован аппаратно-, программно- или программно-аппаратно.

Название сети обычно совпадает с названием протокола, что объясняется его определяющей ролью при создании сети.

Иногда сеть использует несколько протоколов, образующих стек протоколов – набор связанных коммуникационных протоколов, которые функционируют совместно и используют некоторые или все семь уровней модели взаимодействия открытых систем. Для большинства сетей стек протоколов реализован с помощью специализированных сетевых микросхем или встроен в универсальный микропроцессор.

### **1.3. ОБЩИЕ ТРЕБОВАНИЯ К ПРОМЫШЛЕННЫМ СЕТЯМ**

Промышленные сети логически мало чем отличаются от локальных вычислительных сетей, однако они должны отвечать специфическим требованиям [3, 4]:

- жесткая детерминированность (предсказуемость) поведения;
- обеспечение функций реального времени;
- работа на длинных линиях с использованием недорогих физических сред (например, витая пара);
- повышенная надежность физического и канального уровней передачи данных для работы в промышленной среде (например, при больших электромагнитных помехах);

– наличие специальных высоконадежных механических соединительных компонентов;

## 1.4. МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ OSI

Основной идеей, предложенной еще в 70-х годах прошлого века Международной организацией по стандартизации (ISO) модели OSI (Open System Interconnection Basic Reference Model – базовая эталонная модель взаимодействия открытых систем) является разбиение процесса информационного взаимодействия между системами на уровни с четко разграниченными функциями.

Модель позволяет разграничить, формализовать и стандартизировать функции, выполняемые различными аппаратными и программными средствами сетевой структуры. В соответствии с этой моделью все сетевые функции распределяются на семь уровней: физический, канальный, сетевой, транспортный, сеансовый, уровень представления и прикладной (табл. 1.1).

### 1.1. Уровни модели OSI

Номер уровня	Уровень (Layer)	Название протокола	Примеры	Название единицы обмена
7	Прикладной (Application)	Прикладной протокол	FTP, HTTP, SMTP	APDU, сообщение
6	Уровень представления (Presentation)	Протокол уровня представления	SSL	PPDU
5	Сеансовый (Session)	Сеансовый протокол		SPDU
4	Транспортный (Transport)	Транспортный протокол	TCP, UDP, SPX	TPDU
3	Сетевой (Network)	Сетевой	IP, IPX	Пакет
2	Канальный (Data Lin)	Протокол канального уровня		Кадр
1	Физический(Physical)	Протокол физического уровня		Бит

### *Прикладной уровень (Application Layer)*

При передаче функции прикладного уровня обеспечивают формирование пересылаемого сообщения, включающего получаемые из прикладной программы данные, информацию о том, куда эти данные должны передаваться, а также набор команд для принимающего устройства о том, что с этими данными он должен сделать. При приеме сообщения – наоборот, на прикладном уровне выполняются функции, в результате которой вся информация из сообщения передается в прикладную программу.

### *Уровень представления (Presentation Layer)*

Уровень представления предназначен для представления данных, подлежащих передаче между прикладными объектами в заранее оговоренную форму. Также на этом уровне могут быть использованы методы, связанные с преобразованием внешнего представления данных без изменения их содержания (шифрование или дешифрование, сжатие данных).

### *Сеансовый уровень (Session Layer)*

Сеансовый уровень предназначен для управления сеансами взаимодействия прикладных процессов пользователей. На этом уровне выполняется управление диалогами между взаимодействующими процессами и его синхронизация, т.е. регулируется какая из сторон осуществляет передачу, когда и как долго.

### *Транспортный уровень (Transport Layer)*

Транспортный уровень обеспечивает доставку сообщений с требуемым уровнем качества (надежности). Главная его задача – деление длинных сообщений, поступающих с верхних уровней на пакеты данных – при передаче информации, и формирование первоначальных сообщений из набора пакетов – при приеме информации.

### *Сетевой уровень (Network Layer)*

Сетевой уровень реализует функции буферизации и маршрутизации, т.е. прокладывает путь между отправителем информации и адресатом через всю сеть.

### *Канальный уровень (Data Link Layer)*

На этом уровне при побайтной передаче перед битами данных формируется старт-бит, после – бит паритета и стоп-бит. На этом уровне производится проверка получаемой информации и исправление ошибок, отключение канала при его неисправности. Но главное – здесь применяются правила совместного

использования физического уровня узлами сети, осуществляется управление доступом к передающей среде. Иными словами, решается вопрос: кто именно получает доступ к общему физическому каналу и ведет передачу.

#### *Физический уровень (Physical Layer)*

Физический уровень непосредственно связан с каналом передачи. Здесь формируются сигналы, с помощью которых представляются данные, устанавливается длительность каждого бита, реализуется конкретный способ перевода каждого бита в соответствующий электрический, оптический или радио-сигнал. Спецификации физического уровня определяют физические характеристики канала связи и параметры сигналов, например, вид кодировки, частота передачи, длина и тип линии, количество проводов, тип штекерного разъема и т.д. К числу наиболее распространенных спецификаций физического уровня относятся: EIA-RS-232C, EIA-RS-422, EIA-RS-485, IEEE 802.3 – Ethernet, IEEE 802.5 – Token Ring.

### **1.5. ТОПОЛОГИИ ПРОМЫШЛЕННЫХ СЕТЕЙ**

Сетевая топология описывает конфигурацию сети связи, физическое и логическое расположение узлов. Физическая топология описывает размещение узлов сети и физические соединения между ними. Логическая топология описывает пути, по которым проходят пакеты данных при их перемещении по сети. Логическая топология сети определяется протоколами, используемыми для передачи по ней данных.

В зависимости от расположения узлов в сети различают следующие физические топологии:

- точка-точка (PtP);
- линия (line);
- шина (bus);
- кольцо (ring);
- звезда (star);
- дерево (tree);
- ячеистая топология (mesh);
- полносвязная (fully connected).

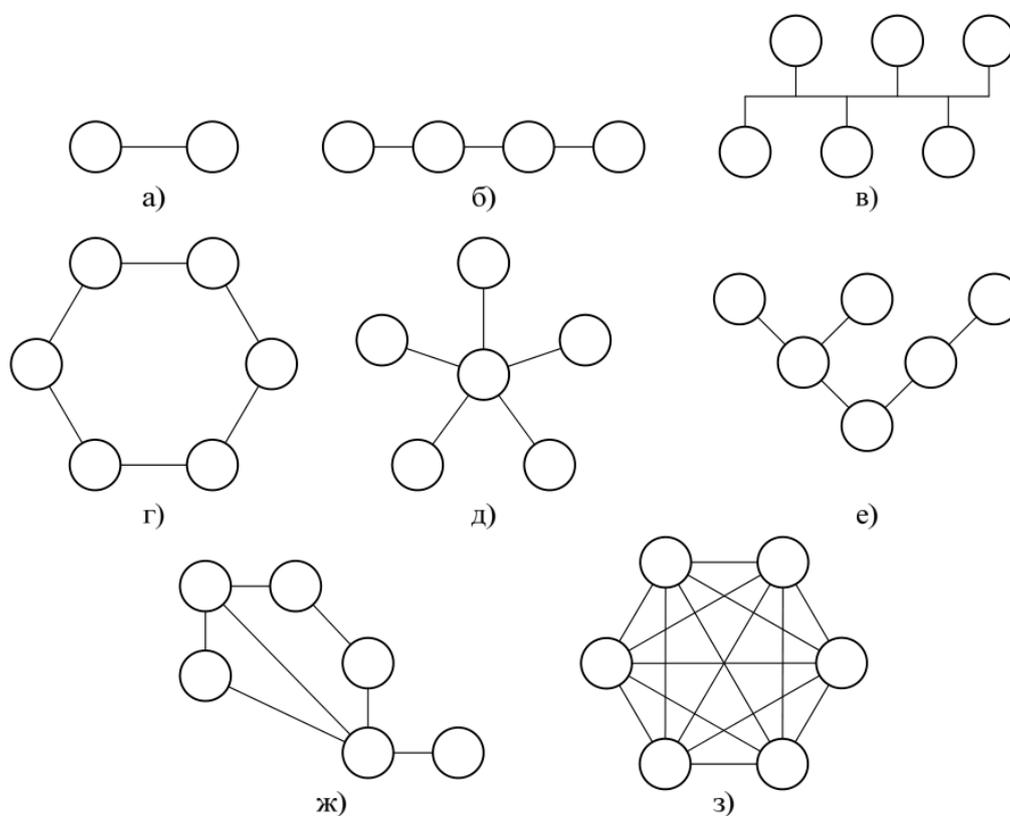
Топология «точка-точка» (Point-to-point, PtP) – это самый простой способ соединения двух узлов (рис. 1.2, а). Как следует из названия, это выделенный

постоянный канал связи между двумя узлами сети. Достоинство этого вида соединения – простота и дешевизна, недостаток – соединение только двух узлов.

В линейной топологии (рис. 1.2, б) узел передает сообщение другому узлу, а тот в свою очередь передает следующему узлу и так до тех пор, пока сообщение не придет на целевой узел. Отправленное с одного устройства сообщение может пройти через все устройства в сети. Данный тип построения сети используется при построении протяженных сетей. Повторители сигналов (репитеры) и шлюзы являются элементами такой сетевой топологии.

Одной из наиболее распространенных типов сетевой топологии в АСУ ТП является топология «шина» (рис. 1.2, в). В шинной структуре все устройства подсоединены к общей среде передачи данных или шине.

По концам шины устанавливаются согласующие сопротивления – терминаторы, предотвращающие отражение сигнала от конца шины. Основными преимуществами этой топологии являются простота, дешевизна и легкость переконфигурирования. Сеть на базе этой топологии не боится отключения или подключения устройств во время работы.



**Рис. 1.2. Топологии сетей:**

*a* – точка-точка; *б* – линия; *в* – шина; *г* – кольцо;  
*д* – звезда; *е* – дерево; *ж* – ячеистая; *з* – полносвязная

В кольцевой топологии (рис. 1.2, з) каждый узел в сети соединен с двумя другими узлами – от одного он только получает информацию, а другому только передает. Первый и последний узлы соединены друг с другом. Данные, передаваемые по сети, проходят через каждый из узлов кольца, пока не достигнут узла назначения. В кольцевой сети данные и сигналы, проходящие по сети, движутся в одном направлении.

Отказ в работе хотя бы одного узла приводит к нарушению работы кольца, а, следовательно, и к остановке всех передач. Чтобы этого избежать, необходимо включать в сеть автоматические переключатели, которые берут на себя инициативу, если данное устройство вышло из режима нормальной работы. То есть, они позволяют включать/выключать отдельные узлы без прерывания нормальной работы.

Существует также топология «двойное кольцо», которая отличается наличием двух соединений между каждым из узлов сети. Потоки данных в двух сформированных таким образом кольцах движутся в двух направлениях. Кольца независимы друг от друга, и потоки данных в них идут в противоположных направлениях. В случае отказа одного кольца другое служит его резервной копией для передачи данных.

В топологии «звезда» (рис. 1.2, д) каждый узел сети подключен к центральному узлу – концентратору. Данные, которые передаются между узлами сети, проходят через центральный узел. Между периферийными узлами и концентратором устанавливается отдельное физическое соединение. Отключение существующих или подключение новых узлов к концентратору не нарушает работу сети. К центральному узлу предъявляются повышенные требования по надежности, так как выход этого узла из строя приводит к обрушению всей сети.

Топологию «дерево» (рис. 1.2, е) можно рассматривать как объединение нескольких «звезд». Еще эту топологию в некоторых источниках называют как иерархическую. Древовидная топология масштабируема и проста в управлении. Недостатком этого типа топологии является то, что при отказе магистрали сеть

становится неработоспособной. В случае отказа корневого узла сети, отходящие от него, не могут взаимодействовать друг с другом.

Древовидная топология сегодня является наиболее популярной при построении локальных сетей.

В ячеистой (рис. 1.2, ж) топологии (mesh) каждый узел сети соединяется с некоторым количеством других узлов, при этом узлы также выполняют роль маршрутизаторов. Взаимосвязи между узлами создают избыточность и повышают ее отказоустойчивость. Даже если какой-либо из узлов выходит из строя или отключается, или происходит обрыв какого-либо соединения, сеть автоматически переконфигурируется и пакеты доходят до своих адресатов по альтернативным маршрутам. Ячеистая топология обычно используется в беспроводных сетях. В проводных сетях для организации сети с этой топологией требуется большое количество кабеля.

Полносвязная (Fully Connected) топология (рис. 1.2, з) – это такое размещение узлов, при котором каждый узел соединен с каждым.

## **1.6. ФИЗИЧЕСКИЕ ИНТЕРФЕЙСЫ**

### **Интерфейс RS-232C**

RS-232 – это название стандарта (RS – Recommended Standard – рекомендованный стандарт, 232 – его номер), описывающего интерфейс для соединения компьютера и устройства передачи данных.

Стандарт был разработан в 60-х годах XX века. В настоящее время действует редакция стандарта, принятая в 1991 году ассоциациями электронной и телекоммуникационной промышленности, под названием EIA/TIA-232. Однако, большинство людей по-прежнему использует название RS-232, которое закрепилось за этим интерфейсом.

Интерфейс RS-232 обеспечивает соединение двух устройств, одно из которых называется DTE (Data Terminal Equipment – Оконечное Оборудование Данных), второе – DCE (Data Communications Equipment – Оборудование Передачи Данных).

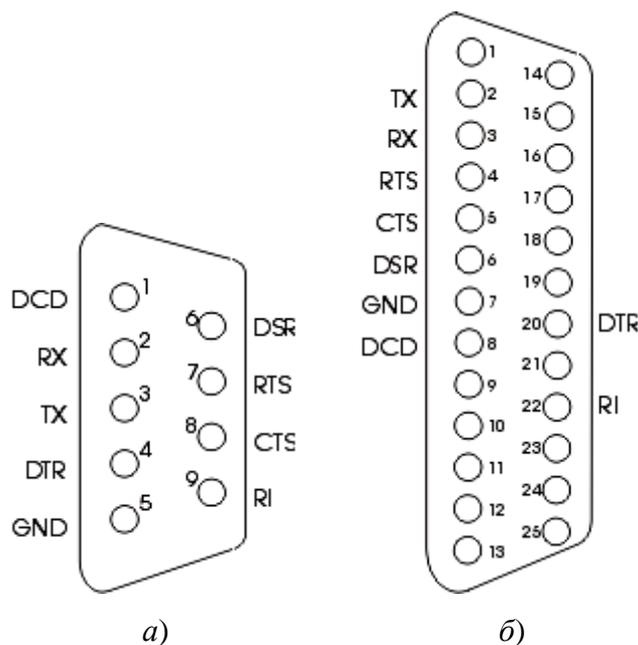
Как правило, DTE (ООД) – это компьютер, а DCE (ОПД) – это модем, хотя RS-232 использовался и для подключения к компьютеру периферийных устройств (мышь, принтер), и для соединения с другим компьютером или контроллером.

Изначально стандарт описывал применение 25-контактного соединителя типа DB25. DTE-устройство должно оснащаться вилкой (Male – «папа»), DCE-устройство – розеткой (Female – «мама»). Позднее, стали использовать усеченный вариант интерфейса и 9-контактные соединители DB9, наиболее распространенные в настоящее время (рис. 1.3).

Компьютер имеет 25-контактный (DB25P) или 9-контактный (DB9P) разъем для подключения RS-232C. Назначение контактов разъема приведено в табл. 1.2.

Назначение сигналов следующее:

- TxD – Последовательные данные – выход передатчика.
- RxD – Последовательные данные – вход приемника.
- RTS – Сигнал запроса передачи. Активен во все время передачи.
- CTS – Сигнал сброса (очистки) для передачи. Активен во все время передачи. Говорит о готовности приемника.
- DSR – Готовность данных. Используется для задания режима модема.



**Рис. 1.3. Разъем для подключения RS-232C:**

*а* – 9-контактный (DB9P); *б* – 25-контактный (DB25P)

## 1.2. Назначение контактов разъема DB9P

Наименование	Направление	Описание	Контакт (25-контактный разъем)	Контакт (9-контактный разъем)
DCD	IN	Carrier Detect (Определение несущей)	8	1
RXD	IN	Receive Data (Принимаемые данные)	3	2
TXD	OUT	Transmit Data (Передаваемые данные)	2	3
DTR	OUT	Data Terminal Ready (Готовность терминала)	20	4
GND	–	System Ground (Общий)	7	5
DSR	IN	Data Set Ready (Готовность данных)	6	6
RTS	OUT	Request to Send (Запрос на отправку)	4	7
CTS	IN	Clear to Send (Готовность приема)	5	8
RI	IN	Ring Indicator (Индикатор)	22	9

- GND – сигнальная земля, нулевой провод, общий провод.
- DCD – обнаружение несущих данных (детектирование принимаемого сигнала).
- DTR – готовность выходных данных.
- RI – индикатор вызова. Говорит о приеме модемом сигнала вызова по телефонной сети.
- PG – защитное заземление (экран кабеля, корпус устройства).

Для передачи данных предназначены цепи RXD и TXD. Остальные цепи предназначены для индикации состояния устройств (DTR, DSR), управления передачей (RTS, CTS) и индикации состояния линии (DCD, RI). Полный набор цепей используется только для подключения к компьютеру внешнего модема. В остальных случаях, например, при подключении к компьютеру промышленного контроллера, используется ограниченный набор цепей, зависящий от аппаратной и программной реализации стыка в контроллере.

Для соединения строго соответствующих стандарту устройств DTE и DCE нужен кабель «контакт в контакт». Для соединения двух DTE-устройств используют так называемые нуль-модемные кабели, в которых провода «перекрещиваются» в соответствии с назначением сигналов. На практике перед распайкой кабеля всегда следует разобраться с документацией на оба соединяемых устройства.

Для соединения многих устройств достаточно минимального набора цепей интерфейса RS-232: RXD, TXD и Signal Ground. На рисунке 1.4 показана распайка нуль-модемного кабеля, использующегося для соединения компьютера и многих контроллеров на соединителях DB9.

Стандарт определяет максимальную длину кабеля в 50 футов (примерно 15 метров) при скорости 9600 бит/с. На практике устойчивая работа может быть достигнута и при большей длине кабеля.

Рекомендуется использовать кабели на основе витой пары, где каждый из сигнальных проводов свит с общим проводом. Например, для этой цели хорошо подходит кабель для прокладки локальной сети Ethernet на неэкранированных витых парах (Unshielded Twisted Pair – UTP), а лучше – на экранированных – STP. Экран кабеля рекомендуется не объединять с сигнальным общим, а подключить к металлической оболочке разъема.

RS-232 является последовательным интерфейсом, поскольку поток данных передается по одному проводу бит за битом. В отсутствие передачи данных линия находится в состоянии логической единицы (–12 В). Скорость передачи данных стандартом не нормируется, но обычно выбирают из ряда 1200, 2400, 4800, 9600, 19 200, 38 400, 57 800, 115 200 бит в секунду. В основном используется асинхронный режим работы, при котором данные передаются фреймами.

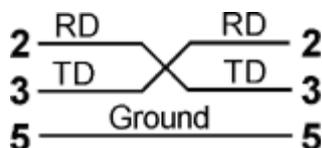


Рис. 1.4. Распайка нуль-модемного кабеля

Каждый фрейм состоит из стартового бита, битов данных, бита контроля четности (может отсутствовать), стопового бита. Биты байта данных передаются «хвостом вперед», начиная с младшего бита.

Для правильной стыковки приемопередатчики на обоих устройствах должны быть запрограммированы одинаковым образом, т.е. должны совпадать скорость, количество битов данных (7 или 8), тип контроля по четности, длина стопового бита (1 или 2).

В настоящее время из мира компьютерной техники интерфейс RS-232 практически ушел. Однако среди промышленных приборов и средств автоматизации в эксплуатации находится большое число устройств с интерфейсом RS-232. Для стыковки компьютеров с такими устройствами используют переходники USB-RS232. После подключения такого переходника и установки драйверов в компьютер появляется виртуальный COM-порт, через который можно общаться с устройством.

### **Интерфейсы RS-485/RS422**

Стандарты RS-485 и RS-422 были совместно разработаны двумя ассоциациями производителей: Ассоциацией электронной промышленности (EIA – Electronics Industries Association) и Ассоциацией промышленности средств связи (TIA – Telecommunications Industry Association). EIA/TIA официально заменил «RS» на «EIA/TIA» с целью облегчить идентификацию происхождения своих стандартов, однако многие инженеры продолжают использовать прежние обозначения. Интерфейс RS-485 является наиболее распространенным в промышленной автоматизации. Его используют промышленные сети Modbus, Profibus DP, ARCNET, BitBus, WorldFip, LON, Interbus и др. Связано это с тем, что по всем основным показателям данный интерфейс является наилучшим из всех возможных при современном уровне развития технологии. Основными его достоинствами являются:

- двусторонний обмен данными всего по одной витой паре проводов;
- работа с несколькими трансиверами, подключенными к одной и той же линии, т.е. возможность организации сети;
- большая длина линии связи;
- достаточно высокая скорость передачи.

Стандарты RS-485 и RS-422 имеют много общего (табл. 1.3). Основное отличие заключается в следующем: RS-485 является единственным стандартом EIA/TIA, допускающим множественные приемники и драйверы (передатчиков) на шине, EIA/TIA-422, наоборот, определяет единственный однонаправленный драйвер с множественными приемниками.

В основе построения интерфейса RS-485 лежит дифференциальный способ передачи сигнала, когда напряжение, соответствующее уровню логической единицы или нуля, отсчитывается не от «земли», а измеряется как разность потенциалов между двумя передающими линиями: Data+ и Data- (рис. 1.5). При этом напряжение каждой линии относительно «земли» может быть произвольным, но не должно выходить за диапазон  $-7...+12$  В [5].

### 1.3. Основные параметры интерфейсов RS-422/ RS-485

	RS-422	RS-485
Режим работы	Дифференциальный	Дифференциальный
Допустимое число Tx и Rx	1 Tx, 10 Rx	32 Tx, 32 Rx
Максимальная длина кабеля	1200 м	1200 м
Максимальная скорость передачи данных	10 Мбит/с	10 Мбит/с
Минимальный выходной диапазон драйвера	$\pm 2$ В	$\pm 1.5$ В
Максимальный выходной диапазон драйвера	$\pm 5$ В	$\pm 5$ В
Максимальный ток короткого замыкания драйвера	150 мА	250 мА
Сопротивление нагрузки Tx	100 Ом	54 Ом
Чувствительность по входу Rx	$\pm 200$ мВ	$\pm 200$ мВ
Максимальное входное сопротивление Rx	4 кОм	12 кОм
Диапазон напряжений входного сигнала Rx	$\pm 7$ В	от $-7$ В до $+12$ В
Уровень логической единицы Rx	$> 200$ мВ	$> 200$ мВ
Уровень логического нуля Rx	$< 200$ мВ	$< 200$ мВ

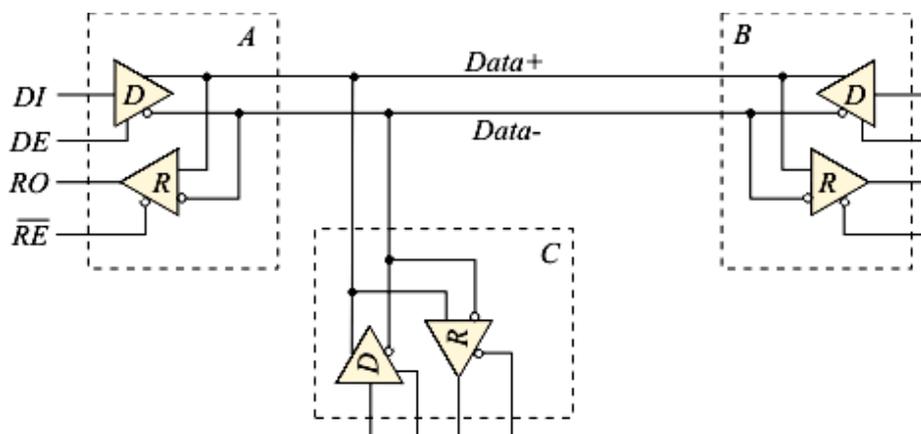


Рис. 1.5. Соединение трех устройств с интерфейсом RS-485 по двухпроводной схеме

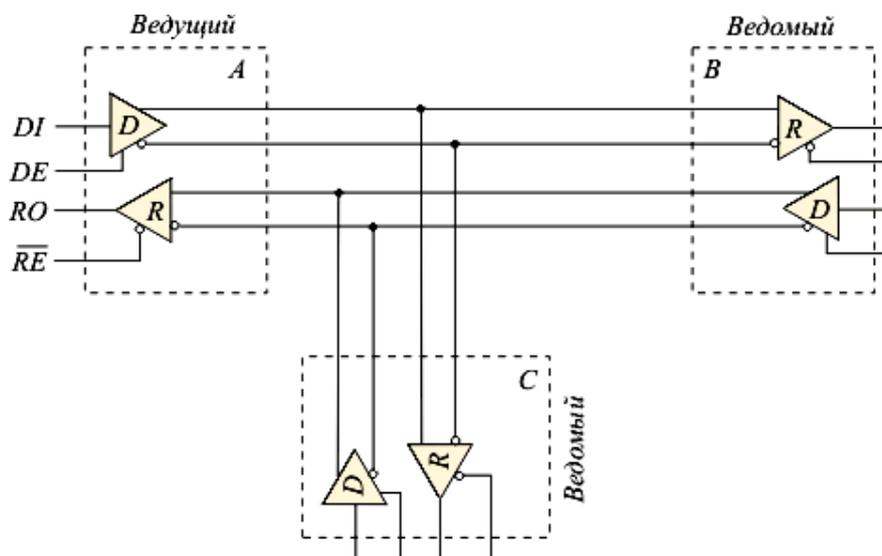
Приемники сигнала являются дифференциальными, т.е. воспринимают только разность между напряжениями на линии Data+ и Data-. При разности напряжений более 200 мВ, до +12 В считается, что на линии установлено значение логической единицы, при напряжении менее -200 мВ, до -7 В – логического нуля. Дифференциальное напряжение на выходе передатчика в соответствии со стандартом должно быть не менее 1,5 В, поэтому при пороге срабатывания приемника 200 мВ помеха (в том числе падение напряжения на омическом сопротивлении линии) может иметь размах 1,3 В над уровнем 200 мВ. Такой большой запас необходим для работы на длинных линиях с большим омическим сопротивлением. Фактически, именно этот запас по напряжению и определяет максимальную длину линии связи (1200 м) при низких скоростях передачи (менее 100 кбит/с).

Благодаря симметрии линий относительно «земли» в них наводятся помехи, близкие по форме и величине. В приемнике с дифференциальным входом сигнал выделяется путем вычитания напряжений на линиях, поэтому после вычитания напряжение помехи оказывается равным нулю. В реальных условиях, когда существует небольшая асимметрия линий и нагрузок, помеха подавляется не полностью, но ослабляется существенно.

Для минимизации чувствительности линии передачи к электромагнитной наводке используется витая пара проводов. Токи, наводимые в соседних витках вследствие явления электромагнитной индукции, по «правилу буравчика» оказываются направленными навстречу друг другу и взаимно компенсируются. Степень компенсации определяется качеством изготовления кабеля и количеством витков на единицу длины.

Интерфейс RS-485 имеет две версии: двухпроводную и четырехпроводную [5]. Двухпроводная используется для полудуплексной передачи (рис. 1.5), когда информация может передаваться в обоих направлениях, но в разное время. Для полnodуплексной (дуплексной) передачи используют четыре линии связи: по двум информация передается в одном направлении, по двум другим – в обратном (рис. 1.6).

Допустимая нагрузка драйвера RS-485/RS-422 количественно определяется в терминах единичной нагрузки, которая, в свою очередь, определяется как входной импеданс одного стандартного приемника RS-485 (12 кОм). Таким образом, стандартный драйвер RS-485 может управлять 32 единичными нагрузками (32 параллельных 12-килоомных нагрузки).



**Рис. 1.6. Четырехпроводное соединение устройств с интерфейсом RS-485**

Однако для некоторых приемников RS-485 входное сопротивление является более высоким – 48 кОм (1/4 единичной нагрузки) или даже 96 кОм (1/8 единичной нагрузки) – и, соответственно, к одной шине могут быть подключены сразу 128 или 256 таких приемников. Вы можете подключить любую комбинацию типов приемников, если их параллельный импеданс не превышает 32 единичных нагрузки (т.е. суммарное сопротивление не меньше 375 Ом).

При больших расстояниях между устройствами, связанными по витой паре и высоких скоростях передачи, начинают проявляться так называемые эффекты длинных линий. Причина этому – конечность скорости распространения электромагнитных волн в проводниках. Электрический сигнал имеет также свойство отражаться от открытых концов линии передачи и ее ответвлений. Грубая аналогия – желоб, наполненный водой. Волна, созданная в одном конце, идет по желобу и, отразившись от стенки в конце, идет обратно, отражается опять и так далее, пока не затухнет. Для коротких линий и малых скоростей передачи этот процесс происходит так быстро, что остается незамеченным. Однако, время реакции приемников – десятки/сотни нс. В таком масштабе времени несколько десятков метров электрический сигнал проходит отнюдь не мгновенно. И если расстояние достаточно большое, фронт сигнала, отразившись в конце линии и вернувшийся обратно, может исказить текущий или следующий сигнал. В таких случаях нужно каким-то образом подавлять эффект отражения.

Оказывается, что если на удаленном конце линии, между проводниками витой пары включить резистор с номиналом, равным волновому сопротивле-

нию линии, то электромагнитная волна, дошедшая до «тупика», поглощается на таком резисторе. Эти резисторы называются концевыми, согласующими или терминальными.

Для систем промышленной автоматики используются кабели с волновым сопротивлением от 100 до 150 Ом, однако кабели, спроектированные специально для интерфейса RS-485, имеют волновое сопротивление 120 Ом. На такое же сопротивление обычно рассчитаны микросхемы трансиверов интерфейса RS-485. Поэтому сопротивление терминального резистора выбирается равным 120 Ом, мощность – 0,25 Вт.

Для коротких линий (несколько десятков метров) и низких скоростей (меньше 38 400 бод) согласование может вообще не потребоваться.

Эффект отражения и необходимость правильного согласования накладывают ограничения на конфигурацию линии связи. Линия связи должна представлять собой один кабель витой пары. К этому кабелю присоединяются все приемники и передатчики. Расстояние от линии до микросхем интерфейса RS-485 должно быть как можно короче, так как длинные ответвления вносят рассогласование и вызывают отражения. В оба наиболее удаленных конца кабеля ( $Z_B = 120 \text{ Ом}$ ) включают согласующие резисторы по 120 Ом (рис. 1.7).

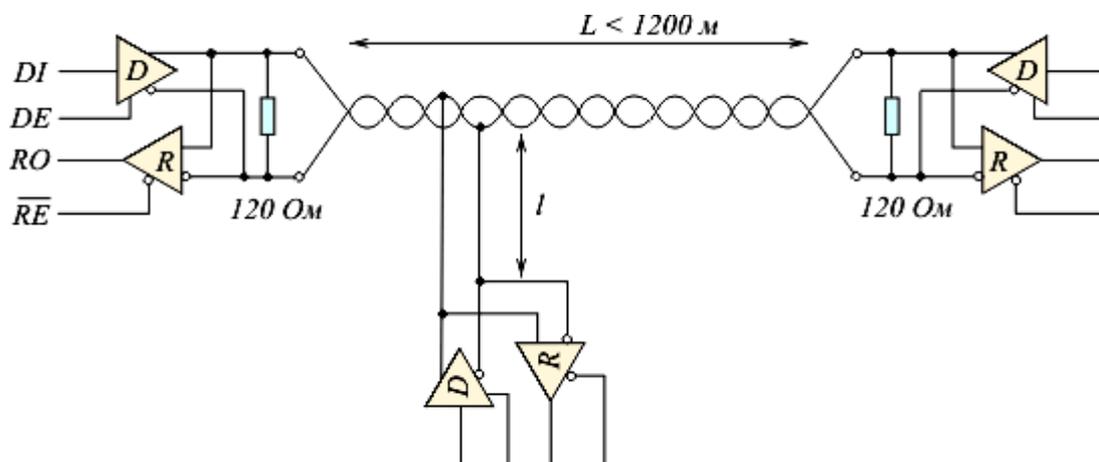


Рис. 1.7. Применение терминальных резисторов для согласования линии передачи

## 2. СЕТИ MODBUS

---

### 2.1. ОБЩИЕ СВЕДЕНИЯ

Протокол Modbus был первоначально разработан в 1979 году компанией Modicon (в настоящее время входит в состав Schneider Electric) для связи с программируемыми логическими контроллерами Modicon. В настоящее время развитием протокола Modbus занимается некоммерческая организация Modbus Organization ([www.modbus.org](http://www.modbus.org))

Несмотря на свой преклонный возраст протокол Modbus не потерял своей актуальности и сегодня. Это один из самых распространенных протоколов, использующихся для организации каналов связи между контроллерами, модулями ввода-вывода, интеллектуальными датчиками и исполнительными устройствами. Практически любой производитель программируемых логических контроллеров обеспечивает в своей продукции поддержку протокола Modbus.

Высокой популярности сетей Modbus способствует то, что для построения сети не требуется каких-либо специализированных интерфейсных контроллеров, а алгоритм взаимодействия узлов имеет простую программную реализацию. Тексты стандартов протокола Modbus являются полностью открытыми и доступны на сайте [www.modbus.org](http://www.modbus.org). Кроме того, Modbus имеет высокую достоверность передачи данных, связанную с применением надежного метода контроля ошибок.

В протоколе Modbus взаимодействие узлов сети осуществляется по принципу «Клиент – Сервер» и используется централизованный метод доступа к среде передачи данных Master-Slave («Ведущий – Ведомый») или иначе «Запрос/Ответ» (Request/Response). В соответствии с этим принципом в сети выделяется клиентское (ведущее, master) устройство (например, программируемый логический контроллер или НМИ), которое периодически отправляет

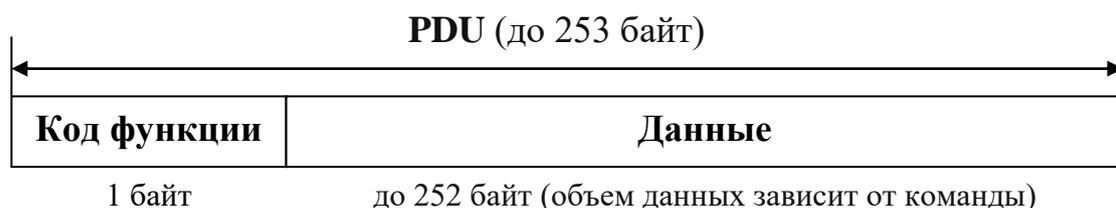
запросы на серверные (ведомые, slave) устройства с целью чтения или записи их параметров. В качестве ведомых (серверных, slave) устройств, обычно выступают модули распределенного ввода-вывода, микропроцессорные приборы, интеллектуальные датчики, частотные преобразователи, другие контроллеры.

По правилам протокола Modbus все запросы может инициировать только клиентское (master, ведущее) устройство. Серверные устройства (slave, ведомые) не могут самостоятельно выдавать сообщения, они только выдают ответы на запрос со стороны клиента (ведущего).

В документе MODBUS Protocol Specification (предыдущее название – Modbus Application Protocol) содержится спецификация прикладного уровня сетевой модели OSI: представлено описание структуры запросов и ответов. Их основа – элементарный пакет протокола, так называемый PDU (*Protocol Data Unit*). Структура PDU (рис. 2.1) не зависит от типа линии связи и содержит код функции (1 байт) и поле данных (0...252 байт).

Код функции показывает, что нужно сделать с полем данным. Код функции может принимать значения в диапазоне 1...127 (0x01...0x7F – в шестнадцатеричном формате). Диапазон значений 128...255 (0x80...0xFF) зарезервирован для кодов ошибок. Поле «Данные» не имеет строго определенной длины. В нем может содержаться информация о параметрах, используемых в запросах контроллера или ответах модуля.

Сетевое взаимодействие между Клиентом (Master) и Сервером (Slave) иллюстрируется следующим рисунком (рис. 2.2). Клиент инициирует запрос в серверное устройство, передавая в PDU код функции и данные. Если обработка запроса проходит без ошибок, то сервер возвращает пакет, содержащий исходный код функции и запрошенные данные.



**Рис. 2.1. Структура PDU**

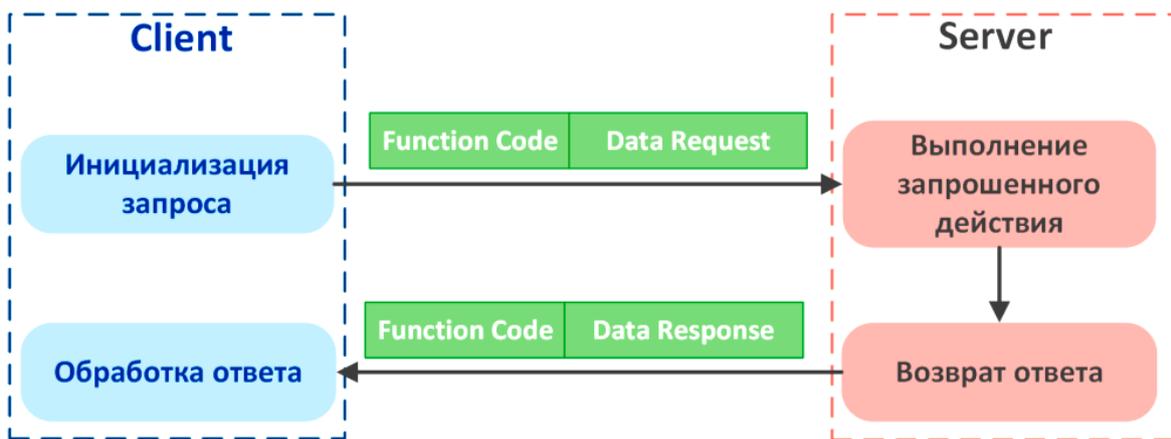


Рис. 2.2. Схема работы Modbus в случае отсутствия ошибок на серверном устройстве

При возникновении ошибки серверное (Slave) устройство возвращает в качестве данных код исключения (рис. 2.3), а вместо исходного кода функции его измененное значение (старший бит поля кода функции устанавливается в 1, что соответствует увеличению значения на 128). Например, если в исходном запросе код функции был равен 1, то в ответе, который содержит ошибку, код функции в PDU будет равен  $1 + 128 = 129$ .

Запрос Master-устройства представляет собой обращение к одной из областей памяти Slave-устройства с помощью определенной функции. Область памяти характеризуется типом хранящихся в ней значений (биты/регистры) и типом доступа (только чтение или чтение и запись) [6]. Спецификация Modbus определяет 4 области памяти (табл. 2.1).

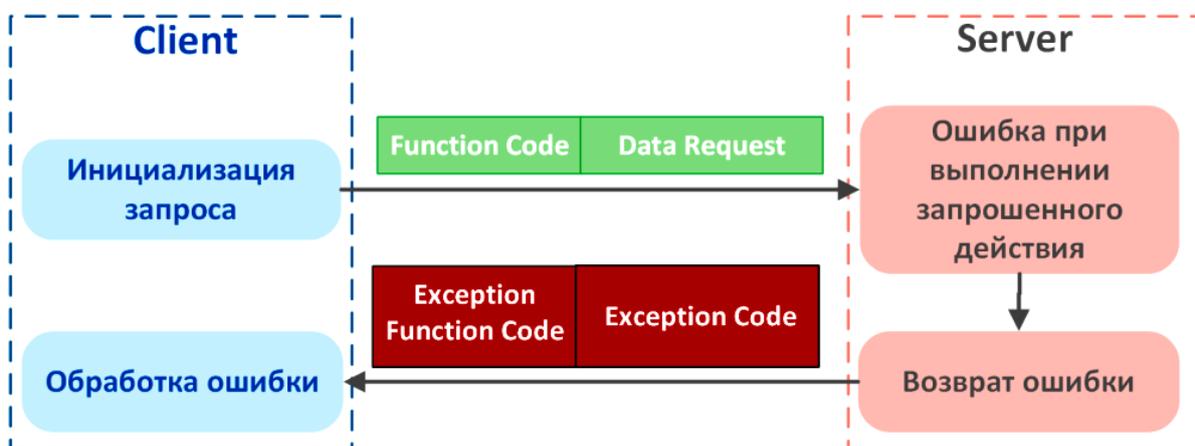


Рис. 2.3. Схема работы Modbus в случае возникновения ошибок на серверном устройстве

## 2.1. Области данных протокола Modbus

Область данных	Обозначение	Тип данных	Тип доступа
Coils (Катушки, регистры флагов) – DO	0x	BOOL	Чтение/запись
Discrete Inputs (Дискретные входы) – DI	1x	BOOL	Только чтение
Input Registers (Регистры ввода) – AI	3x	WORD	Только чтение
Holding Registers (Регистры хранения) – AO	4x	WORD	Чтение/запись

Каждая область памяти состоит из определенного (зависящего от конкретного устройства) количества ячеек. Каждая ячейка имеет уникальный адрес. Для конфигурируемых устройств (таких, как частотные преобразователи, цифровые измерительные приборы, модули распределенного ввода-вывода) производитель предоставляет карту регистров, в которой содержится информация об адресах и типах параметров устройства.

Для программируемых устройств пользователь формирует такую карту самостоятельно с помощью среды разработки.

В качестве примера в табл. 2.2 представлена карта адресов Modbus-регистров модуля дискретного ввода MB110-16ДН компании ОВЕН ([www.owen.ru](http://www.owen.ru)).

## 2.2. Карта адресов Modbus-регистров модуля MB110-16ДН

Параметр	Значение, [ед. изм.]	Тип	Адрес регистра	
			(Hex)	(Dec)
Максимальный сетевой тайм-аут	1...600 [с]	Uint16	0030	0048
Битовая маска значений входов	0...65535	Uint16	0033	0051
Значение счетчика входа № 1	0...65535 [имп]	Uint16	0040	0064
Значение счетчика входа № 2	0...65535 [имп]	Uint16	0041	0065
Значение счетчика входа № 3	0...65535 [имп]	Uint16	0042	0066
Значение счетчика входа № 4	0...65535 [имп]	Uint16	0043	0067
Значение счетчика входа № 5	0...65535 [имп]	Uint16	0044	0068
Значение счетчика входа № 6	0...65535 [имп]	Uint16	0045	0069
Значение счетчика входа № 7	0...65535 [имп]	Uint16	0046	0070
Значение счетчика входа № 8	0...65535 [имп]	Uint16	0047	0071
Значение счетчика входа № 9	0...65535 [имп]	Uint16	0048	0072
Значение счетчика входа № 10	0...65535 [имп]	Uint16	0049	0073
Значение счетчика входа № 11	0...65535 [имп]	Uint16	004A	0074
Значение счетчика входа № 12	0...65535 [имп]	Uint16	004B	0075
Значение счетчика входа № 13	0...65535 [имп]	Uint16	004C	0076
Значение счетчика входа № 14	0...65535 [имп]	Uint16	004D	0077
Значение счетчика входа № 15	0...65535 [имп]	Uint16	004E	0078
Значение счетчика входа № 16	0...65535 [имп]	Uint16	004F	0079

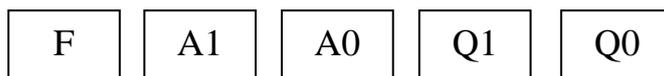
## 2.3. Часто используемые стандартные функции Modbus

Код функции	Имя функции	Выполняемая команда
1 (0x01)	Read Coil Status	Чтение значений из регистров флагов
2 (0x02)	Read Discrete Inputs	Чтение значений из дискретных входов
3 (0x03)	Read Holding Registers	Чтение значений из регистров хранения
4 (0x04)	Read Input Registers	Чтение значений из регистров ввода
5 (0x05)	Write Single Coil	Запись значения в один регистр флага
6 (0x06)	Write Single Register	Запись значения в один регистр хранения
15 (0x0F)	Write Multiple Coils	Запись значений в несколько регистров флагов
16 (0x10)	Write Multiple Registers	Запись значений в несколько регистров хранения

Для работы с modbus-регистрами модуль MB110-16ДН поддерживает следующие функции:

- Функция 16 (0x10) – запись в несколько регистров регистры;
- Функции 03 или 04 – чтение из нескольких регистров.

Так, например, кадр PDU в запросе на чтение счетчика № 15 с помощью функции 03 строится следующим образом:



где F – номер функции (0x03); A1 – старший байт адреса первого запрашиваемого регистра (0x00); A0 – младший байт адреса первого запрашиваемого регистра (0x4E); Q1 – старший байт количества запрашиваемых регистров (0x00); Q0 – младший байт количества запрашиваемых регистров (0x01);

То есть мы запрашиваем значение одного регистра, который имеет адрес 004E, при этом используем функцию номер 03. В PDU все значения записываются в шестнадцатеричном формате. В итоге кадр PDU этого запроса будет иметь вид:

**03 00 4E 00 01**

Все коды функций можно разделить на три группы:

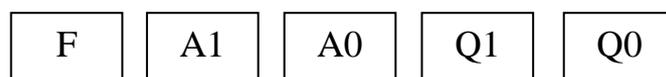
- Стандартные коды команд.
- Задаваемые пользователем (User-Defined Function Codes) – 65...72, 100...110. Эти коды не описаны в спецификации стандарта и могут использоваться в конкретных изделиях для собственных функций.
- Зарезервированные (Reserved). В эту группу входят коды 9, 10, 13, 14, 41, 42, 90, 91, 125, 126 и 127.

В таблице 2.3 представлены часто используемые стандартные функции Modbus.

## Функция 1 (0x01) – Read Coil Status

Функция 1 используется для чтения состояния дискретных выходов (coils) устройства. Количество считываемых значений подряд следующих дискретных выходов от 1 до 2000.

Структура PDU запроса имеет следующий вид:



где F – номер функции (0x01); A1, A0 – соответственно старший и младший байты адреса первого запрашиваемого дискретного выхода (coil). Адрес может лежать в диапазоне 0x0000...0xFFFF и занимает 2 байта; Q1, Q0 – соответственно старший и младший байты количества запрашиваемых дискретных выходов (coils). Количество запрашиваемых дискретных выходов от 1 до 2000 (0x0001...0x07D0).

В итоге PDU в запросе имеет размер 5 байт.

При удачном выполнении команды удаленное устройство выдает ответ, PDU которого имеет вид:



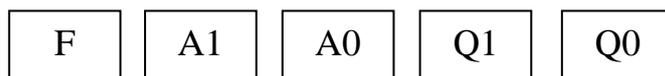
где F – номер функции (0x01); N – количество возвращаемых байтов, содержащих состояния дискретных выходов. Значение N лежит в диапазоне от 1 до 250 байт (0x01...0xFA); nnn – байты, содержащие состояния дискретных выходов (N байт).

Размер PDU в ответе зависит от количества запрашиваемых дискретных выходов и может лежать в диапазоне от 3 байт (при запросе состояний не более 8 дискретных выходов) до 250 байт (при запросе состояний 2000 дискретных выходов)

## Функция 2 (0x02) – Read Discrete Inputs

Функция 2 используется для чтения дискретных входов (Discrete Inputs) устройства. Количество считываемых значений подряд следующих дискретных входов от 1 до 2000.

Структура PDU запроса имеет следующий вид:



где F – номер функции (0x02); A1, A0 – соответственно старший и младший байты адреса первого запрашиваемого дискретного входа (coil). Адрес может лежать в диапазоне 0x0000...0xFFFF и занимает 2 байта; Q1, Q0 – соответственно старший и младший байты количества запрашиваемых дискретных выходов (Discrete Input). Количество запрашиваемых дискретных входов от 1 до 2000 (0x0001...0x07D0).

При удачном выполнении команды удаленное устройство выдает ответ, PDU которого имеет вид:



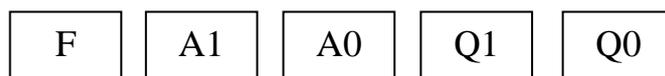
где F – номер функции (0x02); N – количество возвращаемых байтов, содержащих значения дискретных входов. Значение N лежит в диапазоне от 1 до 250 байт (0x01...0xFA); nnn – байты, содержащие состояния дискретных входов (N байт).

Размер PDU в ответе зависит от количества запрашиваемых дискретных входов устройства и может лежать в диапазоне от 3 байт (при запросе не более 8 дискретных входов) до 250 байт (при запросе 2000 дискретных входов).

### Функция 3 (0x03) – Read Holding Registers

Функция 3 используется для чтения нескольких непрерывно следующих регистров хранения (общего назначения) в удаленном устройстве. **Holding Register** в памяти устройства занимает 16 бит (2 байта).

Структура PDU запроса имеет следующий вид:



где F – номер функции (0x03); A1, A0 – соответственно старший и младший байты адреса первого запрашиваемого регистра. Адрес может лежать в диапазоне 0x0000...0xFFFF и занимает 2 байта; Q1, Q0 – соответственно старший



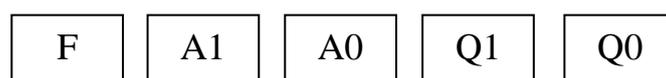
### Функция 4 (0x04) – Read Input Registers

Функция 4 используется для считывания от 1 до 125 последовательно расположенных входных регистров на удаленном устройстве. Структура PDU запроса и ответа при использовании функции 4 полностью **аналогична функции 3**. В PDU запроса указывается адрес начального регистра и количество регистров.

Каждый Input Register занимает в памяти устройства 2 байта (16-бит). В классическом представлении ModBus эти регистры предназначены для хранения значений сигнала на аналоговых входах устройства.

### Функция 5 (0x05) – Write Single Coil

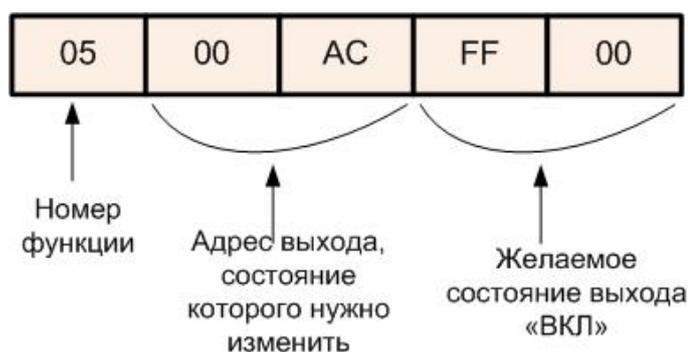
Функция 5 для изменения состояния одного из дискретных выходов (coil) удаленного устройства. Структура PDU запроса имеет следующий вид:



где F – номер функции (0x05); A1, A0 – соответственно старший и младший байты адреса изменяемого дискретного выхода (coil); Q1, Q0 – соответственно старший и младший байты устанавливаемого значения дискретного выхода. Значение 0xFF00 устанавливает адресуемый дискретный выход в состояние **ВКЛЮЧЕНО (ON)**, а значение 0x0000 – в состояние **ВЫКЛЮЧЕНО (OFF)**. Все остальные значения недопустимы и игнорируются удаленным устройством.

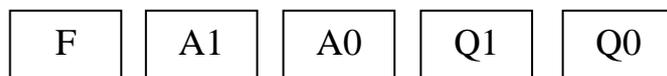
В случае успешного выполнения функции удаленное устройство отвечает сообщением с PDU являющимся копией PDU запроса.

Ниже приведен пример формирования PDU запроса на **ВКЛЮЧЕНИЕ** дискретного выхода с адресом 172 (0x00AC):



## Функция 6 (0x06) – Write Single Register

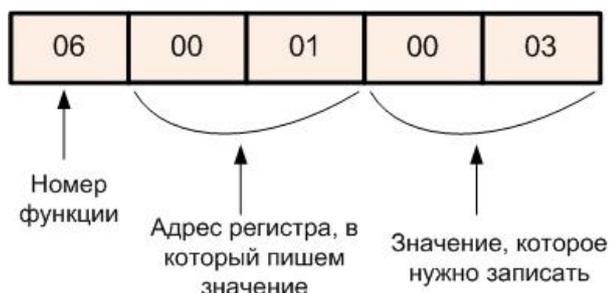
Функция 6 используется для записи значения в один из регистров хранения (holding registers) в удаленном устройстве. Структура PDU запроса имеет следующий вид:



где F – номер функции (0x06); A1, A0 – соответственно старший и младший байты адреса изменяемого регистра (Holding Register). Адрес может лежать в диапазоне 0x0000...0xFFFF и занимает 2 байта; Q1, Q0 – соответственно старший и младший байты устанавливаемого значения этого регистра. Значение может лежать в диапазоне 0x0000...0xFFFF и также занимает 2 байта.

В случае успешного выполнения функции удаленное устройство отвечает сообщением с PDU являющимся копией PDU запроса.

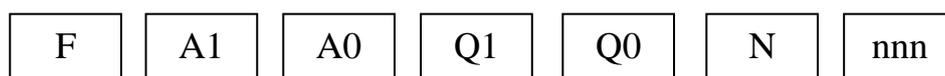
Ниже приведен пример PDU запроса записи в регистр с адресом 1 (0x0001), значения 3 (0x0003).



## Функция 15 (0x0F) – Write Multiple Coils

Функция 15 используется для одновременного изменения состояния нескольких подряд следующих дискретных выходов (coils) устройства. Количество изменяемых дискретных выходов может быть от 1 до 1968.

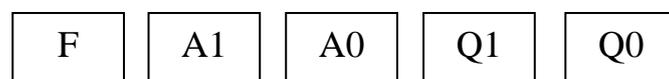
Структура PDU запроса имеет следующий вид:



где F – номер функции (0x0F); A1, A0 – соответственно старший и младший байты адреса первого запрашиваемого дискретного выхода (coil). Адрес может лежать в диапазоне 0x0000...0xFFFF и занимает 2 байта; Q1, Q0 – соответственно старший и младший байты количества  $N_{out}$  изменяемых непрерывно

следующих дискретных выходов (coils). Количество  $N_{out}$  может лежать в диапазоне от 1 до 1968 (0x0001...0x07B0). Это поле тоже занимает 2 байта; N – количество байт данных. Это поле считается как отношение  $N_{out}$  на 8 с округлением до целого в большую сторону; nnn – поле данных, которое занимает N байт. Каждый байт данных содержит состояние восьми дискретных выходов. Логическая «1» в разрядной позиции поля данных запрашивает включение соответствующего дискретного выхода, логический «0» – его выключение.

В случае успешного выполнения функции удаленное устройство отвечает сообщением с PDU следующего вида:



где F – номер функции (0x0F); A1, A0 – соответственно старший и младший байты адреса первого дискретного выхода; Q1, Q0 – соответственно старший и младший байты количества  $N_{out}$  записанных дискретных выходов (coils).

#### **Функция 16 (0x10) – Write Multiple Registers**

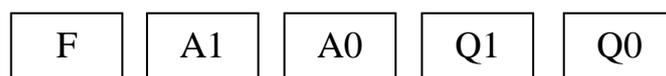
Функция 16 используется для записи значений в несколько (от 1 до 123) последовательно расположенных регистров хранения (holding registers).

Структура PDU запроса имеет следующий вид:



где F – номер функции (0x10); A1, A0 – соответственно старший и младший байты адреса первого изменяемого регистра хранения (holding registers). Адрес может лежать в диапазоне 0x0000...0xFFFF и занимает 2 байта; Q1, Q0 – соответственно старший и младший байты количества  $N_{рег}$  изменяемых непрерывно следующих регистров. Количество  $N_{рег}$  может лежать в диапазоне от 1 до 123 (0x0001...0x007B). Это поле тоже занимает 2 байта; nnn – поле данных, которое содержит значения, которые требуется записать. Так как один регистр занимает 2 байта, это поле занимает  $2 * N_{рег}$  байт.

При успешной выполнении функции удаленное устройство отвечает сообщением с PDU следующего вида:



где F – номер функции (0x10); A1, A0 – соответственно старший и младший байты адреса первого измененного регистра хранения; Q1, Q0 – соответственно старший и младший байты количества  $N_{\text{рег}}$  записанных регистров хранения (holding registers).

## 2.2. РАЗНОВИДНОСТИ ПРОТОКОЛА MODBUS

Modbus – это протокол прикладного (седьмого) уровня модели OSI (Open Systems Interconnection model). Он не зависит от нижележащих уровней и может использоваться совместно с другими протоколами, например Ethernet TCP/IP или UDP/IP, а в качестве физической среды для передачи сигналов применять последовательные интерфейсы RS-232, RS-422, RS-485, оптоволокно, радиоканалы и другое.

Для передачи пакета по физическим линиям связи PDU помещается в другой пакет, содержащий дополнительные поля. Этот пакет носит название ADU (Application Data Unit). Формат ADU зависит от типа линии связи. Существуют три варианта ADU: два для передачи данных через асинхронный последовательный интерфейс – Modbus ASCII Modbus RTU и один через TCP/IP сети – Modbus TCP.

Для всех разновидностей ADU общим является то, что помимо полезных данных (PDU), в них содержится адресная информация, позволяющая узлам сети определить, кто должен обрабатывать принятый пакет данных, а также некоторые виды ADU содержат дополнительные данные, позволяющие диагностировать сбои в процессе передачи данных [3].

В сетях Modbus каждое ведомое устройство должно иметь уникальный адрес в диапазоне от 1 до 247. Адрес 0 используется для организации широковещательной передачи данных. Адреса в диапазоне 248...255 – зарезервированы и не используются.

Ведомые устройства, получив сообщение по сети, должны реагировать на них только если удостоверятся в целостности ADU, и адрес, переданный в сообщении, совпадает с адресом самого устройства (или указан широко-вещательный адрес). На широковещательные команды ведомые устройства не выдают никакой ответ, а на индивидуальные запросы формируют ADU кадры-ответы, в поле адреса ведомого которых указывают свой индивидуальный адрес [3].

### 2.2.1. Modbus RTU

Протокол Modbus RTU (Remote Terminal Unit – удаленное терминальное устройство) является базовым вариантом протокола построения сетей Modbus по последовательным линиям связи. В качестве физического уровня сети чаще всего использует последовательный интерфейс RS-485, реже – RS-232 и RS-422. Структура передаваемого по сети кадра ADU для варианта Modbus RTU приведена на рис. 2.4: сначала передается адрес ведомого устройства (1 байт), затем – PDU, и завершает передачу контрольная сумма (2 байта), которая считается по алгоритму CRC-16.

Сообщения Modbus RTU передаются по сети в виде кадров, которые в явном виде не содержат символов начала и конца. Признаком начала (и окончания) ADU является интервал тишины (рис. 2.5) – время, по истечении которого принимающее устройство делает вывод об окончании процесса передачи данных, и может начать (если это разрешено) передавать свой кадр ADU [3].

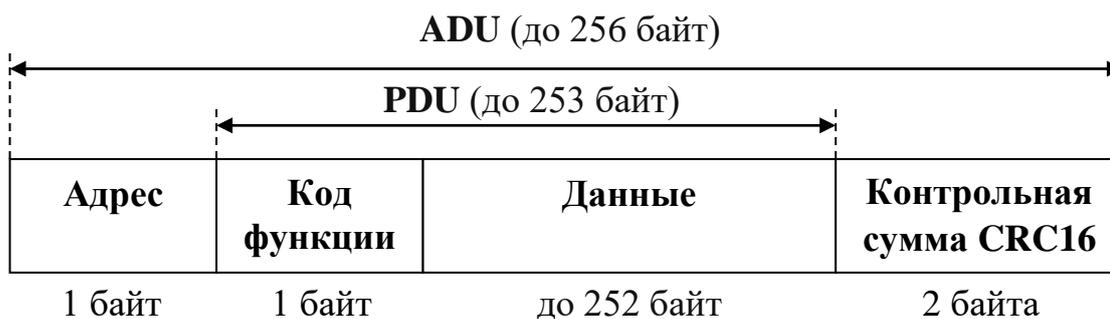


Рис. 2.4. Структура ADU для Modbus RTU



Рис. 2.5. Передача данных Modbus RTU

В соответствии со спецификацией Modbus RTU интервал тишины должен составлять не менее трех с половиной символов на выбранной скорости обмена данными. То есть, если в параметрах коммуникационного обмена установлена скорость 9600 бит/с и режим передачи данных 8e1, то интервал тишины

$$t_{3,5} = 3,5 \cdot \frac{1+8+1+1}{9600} \approx 4 \text{ мс.}$$

Вообще кадр ADU должен передаваться непрерывно, когда сразу по завершении передачи одного символа в линию связи передается другой, но Modbus RTU допускает задержку в передаче между символами на время, не превышающее времени передачи полутора символов. То есть для того же примера это время не должно превышать

$$t_{1,5} = 1,5 \cdot \frac{1+8+1+1}{9600} \approx 1,7 \text{ мс.}$$

Если при передаче кадра обнаруживается пауза более 1,5 символов, то считается, что кадр содержит ошибку и должен быть отклонен принимающей стороной. Если ADU начнется раньше, чем гарантированный интервал тишины в 3,5 символа, то оно будет восприниматься как часть предыдущего кадра, что в конечном итоге опять приведет к детектированию ошибки.

Эти интервалы связаны со скоростью передачи данных и должны строго соблюдаться при скоростях до 19 200 бит/с (включительно). При более высоких скоростях рекомендуется использовать фиксированные значения: минимальный интервал тишины – 1,75 мс, максимальный межсимвольный интервал – 0,75 мс.

Modbus RTU имеет два уровня контроля ошибок в сообщении: контроль бита паритета для каждого байта (опционально) и контроль целостности ADU при помощи 16-битной циклической контрольной суммы CRC с полиномом  $P(x) = x^{16} + x^{15} + x^2 + 1$ .

Контрольная сумма вычисляется передающим устройством и добавляется к сообщению. Принимающее устройство также вычисляет CRC в процессе приема и сравнивает вычисленную величину с полем контрольной суммы предыдущего сообщения. Если суммы не совпали, то имеет место ошибка, если же оба значения совпадают, то считается, что сообщение не содержит ошибок и получатель получил исходный кадр ADU.

### 2.2.2. Modbus ASCII

Протокол Modbus ASCII как и Modbus RTU для передачи данных использует последовательные интерфейсы RS-232/RS-485, но в отличие от Modbus RTU не является бинарным протоколом. Для кодирования сообщений он использует ASCII-символы.

Семибитовый код ASCII был разработан в начале 60-х годов прошлого века как универсальный код для отображения символов английского языка для телетайпов. ASCII является принятым в США стандартом для представления символов английского языка и управляющих символов, например, CR (возврат каретки) и LF (перевод строки). Наименования этих символов сохранились со времен электромеханических телетайпов.

В протоколе Modbus ASCII используется ограниченный набор ASCII-символов. Так как PDU в протоколе представляется в шестнадцатеричном формате, то для кодирования сообщения достаточно цифр от 0 до 9 и букв от A до F. Помимо этих 16 значений используются еще три: символ «двоеточие (:))» (0x3A), «перевод строки» (0x0A или <LF>), «возврат каретки» (0x0D или <CR>).



**Рис. 2.6. Структура ADU для Modbus ASCII**

Еще одним отличием Modbus ASCII от Modbus RTU является то, что каждый кадр ADU помечается разделителями, обозначающими начало и конец сообщения – сообщения начинаются с символа двоеточие «:» и заканчиваются последовательностью символов «возврат каретки», «перевод строки» (рис. 2.6). Интервалы тишины в данном случае не используются.

Приемник ведомого устройства в сети Modbus ASCII должен непрерывно отслеживать передачу символа «двоеточие». Как только он принят, продолжается прием адреса ведомого устройства, данных PDU и контрольной суммы до приема символов <CR><LF>.

Modbus ASCII так же имеет два уровня проверки достоверности передачи данных: контроль бита паритета для каждого символа (опционально) и контроль целостности ADU при помощи 8-битной контрольной суммы LRC (Longitudinal Redundancy Check), которая в составе ADU передается двумя символами.

LRC вычисляется передающим устройством и добавляется к концу сообщения. Принимающее устройство также вычисляет LRC в процессе приема и сравнивает вычисленную величину с полем контрольной суммы пришедшего сообщения. Если суммы не совпали – то имеет место ошибка.

В целом этот вариант протокола сейчас используется крайне редко – из-за сложностей кодирования и большого размера сообщений. Однако он может стать хорошей альтернативой Modbus RTU на линиях с сетевыми задержками и оборудованием с менее точными таймерами.

Наличие стартового и стопового символа в Modbus ASCII позволяет избежать проблем с определением конца фреймов – особенно если устройство не имеет точных аппаратных таймеров. Кроме того, это позволяет использовать в качестве среды передачи ненадежный канал связи – например, радиоканал, GPRS, спутниковую связь и т.д. – в которых во время передачи фрейма могут возникать задержки в несколько секунд.

### 2.2.3. Modbus TCP

Протокол Modbus TCP является одним из наиболее широко используемых и популярных протоколов в автоматизации промышленности. Он позволяет двум или более устройствам взаимодействовать друг с другом через сеть Ethernet. По сути Modbus TCP – это протокол Modbus RTU, работающий по сети Ethernet TCP / IP. Он также как Modbus RTU относится к бинарным протоколам и использует тот же формат PDU, что Modbus RTU.

Основное отличие Modbus TCP от Modbus RTU заключается в реализации нижних уровней сетевой модели взаимодействия открытых систем – OSI. В то время как Modbus RTU для передачи данных на физическом уровне в основном использует интерфейс RS-485, Modbus TCP использует сети Ethernet и протокол TCP/IP, состоящий из стека протоколов, включающего IP (Internet Protocol) и TCP (Transmission Control Protocol).

Кадр ADU для протокола Modbus TCP имеет вид (рис. 2.7):

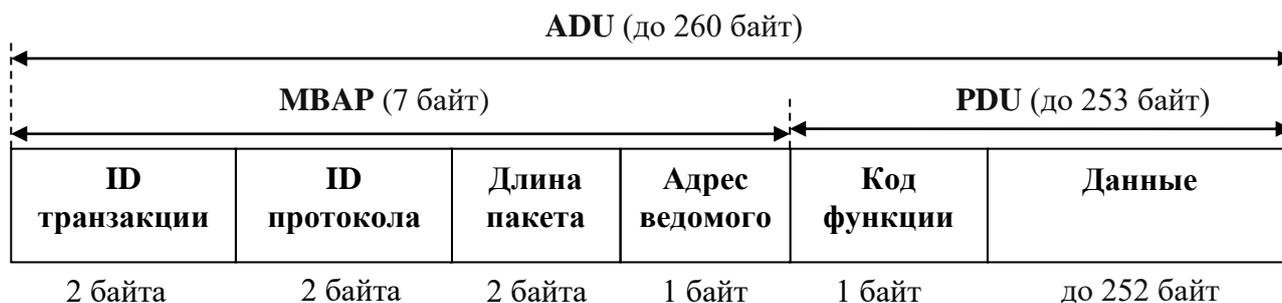


Рис. 2.7. Структура ADU для Modbus TCP

То есть передаче PDU предшествует передача заголовка протокола Modbus-приложения МВАР (Modbus Application Protocol Header), который включает:

**ID транзакции** (Transaction Identifier) – два байта, предназначены для идентификации запроса и ответа, обычно нули. Необходим для случаев, когда клиентское устройство отправляет несколько сообщений, не дожидаясь ответа на предыдущие, чтобы затем связать ответы с запросами.

**ID протокола** (Protocol Identifier) – два байта, используется преобразователями протоколов в смешанных сетях, обычно нули.

**Длина пакета** – два байта, длина следующей за этим полем части пакета: адреса сервера (ведомого устройства) и PDU.

**Адрес ведомого** – адрес подчиненного устройства, к которому адресован запрос. В отличие от Modbus RTU и ASCII, в Modbus TCP соединение устанавливается с конкретным устройством средствами TCP/IP. Поэтому адрес в пакете Modbus чаще всего игнорируется, а широковещательная рассылка сообщений не используется. Однако адрес может потребоваться, если соединение устанавливается со шлюзом, который, в свою очередь, связан с сетью Modbus RTU/ASCII.

В кадре ADU для Modbus TCP отсутствует поле для контроля ошибок в сообщении. Это связано с тем, что целостность данных обеспечивается за счет стека протокола TCP/IP.

Протокол TCP является одним из основных протоколов в сетях Ethernet и обеспечивает установление надежного соединения между двумя или более узлами сети. Перед передачей сообщений сначала устанавливается связь между определенными портами устройств (по умолчанию, в Modbus TCP используется **порт 502**). После передачи связь между портами закрывается. Данные, передаваемые по протоколу TCP, разделяются на пакеты, которые доставляются в порядке отправления и проверяются на целостность. Если какой-либо пакет данных потерян или поврежден, TCP повторно отправляет его, чтобы гарантировать надежную передачу данных.

Еще одно существенное различие – это архитектура сети устройств [8]. Архитектура Master-Slave в протоколе Modbus RTU предполагает наличие только одного ведущего и нескольких ведомых устройств. Мастер отправляет запросы к подчиненным устройствам для получения данных или выполнения определенных действий. А они, в свою очередь, могут отвечать на запросы мастера, предоставлять запрашиваемые данные или выполнять запрошенные действия.

С другой стороны, архитектура Server-Client в протоколе Modbus TCP допускает наличие нескольких серверов и нескольких клиентов. Клиенты отправляют запросы к серверам для получения данных или выполнения определенных действий. Серверы отвечают на запросы клиента, предоставляют необходимые данные или выполняют запрошенные действия.

Также стоит отметить, что протокол Modbus RTU в основном используется для связи с полевыми устройствами, такими как датчики и преобразователи частоты, в то время как протокол Modbus TCP более широко применяется в системах среднего и верхнего уровня.

## **2.3. ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ПРОТОКОЛОВ MODBUS**

### **Передача данных с плавающей точкой и 32-битных значений**

Изначально протокол Modbus RTU был разработан для устройств с 16-битной длиной регистра и он хорошо работает при передаче чисел типа INT или WORD. Для таких платформ диапазон значений, которое может хранить число с типом INT от –32 768 до 32 767, для чисел с типом WORD (или UINT) диапазон значений – от 0 до 65 535. Все эти числа занимают в памяти устройства 2 байта (16 бит). Однако в современных приложениях постоянно требуется передача чисел с плавающей точкой (чисел с типом FLOAT) или целых 32-битных чисел (DINT, UDINT, DWORD). Для хранения таких чисел требуется 4 байта. В памяти устройства эти числа занимают два соседних 16-битных регистра. Для передачи **одного** числа этих типов

данных возможно использовать функции 03 (Read Holding Registers), 04 (Read Input Registers), 16 (Write Holding Registers), при этом в запросе в качестве адреса указать адрес **первого** регистра, а количество читаемых (записываемых) регистров – 2.

### Порядок следования регистров/байт

Спецификация протокола Modbus, которая представлена в документе [7], определяет, что порядок байт при передаче «Big Endian», т.е. старшим байтом вперед. Так, например, при приеме/передаче 16-ти битного регистра, имеющего значение 0x1234, сначала передается старший байт 0x12, затем младший байт 0x34.

Как было описано выше, некоторые параметры занимают в памяти устройства 32 бит (4 байта), т.е. располагаются в двух регистрах. Порядок, в котором будут возвращены регистры при запросе такого параметра, спецификация Modbus не определяет. Поэтому этот порядок зависит от производителя конкретного устройства.

Большинство master-устройств предоставляют готовый функционал для изменения порядка регистров в принимаемых данных. Например, при программировании контроллеров Fastwel I/O в среде программирования CoDeSys v.2.3 при настройке устройства Serial Slave (рис. 2.8) имеется возможность перестановки выбора слов (регистров).

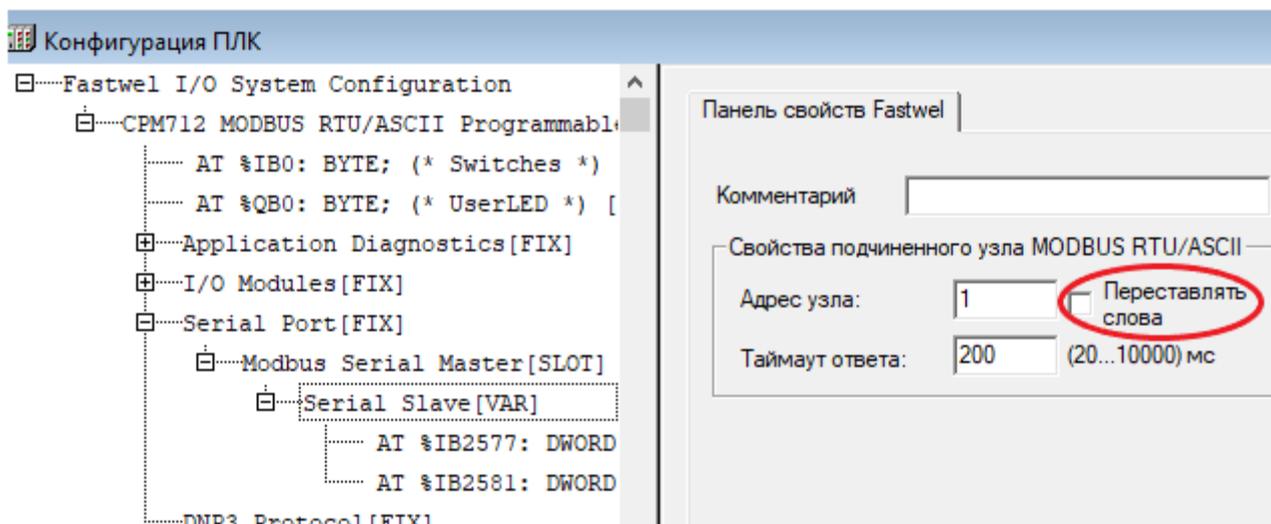
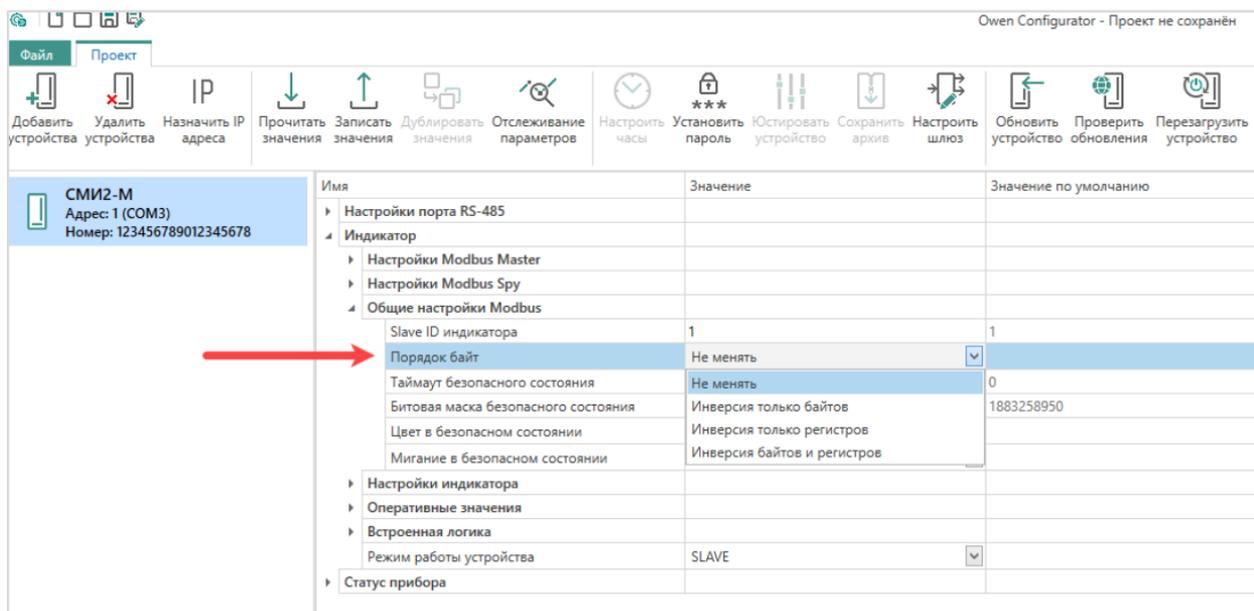


Рис. 2.8. Редактирование свойств Serial Slave для контроллера Fastwel I/O



**Рис. 2.9. Настройка порядка байт/регистров для Modbus в конфигураторе индикатора ОВЕН СМІ2-М**

Некоторые slave-устройства позволяют задать порядок регистров (а иногда даже байт – на тот случай, если порядок байт в master-устройстве отличается от slave-устройства), который будет использоваться в ответах, отправляемых ими на запрос master-устройства. На рисунке 2.9 приведен скриншот из конфигуратора индикатора СМІ2-М компании ОВЕН с соответствующей настройкой.

При передаче 32-битных параметров необходимо убедиться, что и в master-устройстве и slave-устройстве установлен одинаковый порядок следования слов (регистров) и байтов. При различном порядке следования слов (регистров) и байтов значения параметра будут некорректными. Например, число 123456.00, в slave-устройстве выглядит следующим образом (обозначим порядок следования байт – ABCD):

Десятичное представление	Шестнадцатеричное представление			
	A	B	C	D
<b>123456.00</b>	<b>00</b>	<b>20</b>	<b>F1</b>	<b>47</b>

При том же порядке следования байтов и слов на master-устройстве этот параметр отобразится корректно. При изменении порядка следования байтов (BADС – «замена байтов») на master-устройстве параметр отобразится следующим образом:

Шестнадцатеричное представление

Десятичное представление

<b>-9.85402689122801930E+029</b>
----------------------------------

**B A D C**

<b>20</b>	<b>00</b>	<b>41</b>	<b>F1</b>
-----------	-----------	-----------	-----------

Упорядочивание одних и тех же байтов в последовательности «CDAB», известное как «замена слов», приведет к результату, также сильно отличающемуся от исходного значения 123456.00:

Шестнадцатеричное представление

Десятичное представление

<b>+1.08658251394509650E-019</b>
----------------------------------

**C D A B**

<b>F1</b>	<b>47</b>	<b>00</b>	<b>20</b>
-----------	-----------	-----------	-----------

Изменение порядка следования байт и порядка следования слов также приводит к некорректному результату:

Шестнадцатеричное представление

Десятичное представление

<b>+1.257363081E-127</b>
--------------------------

**D C B A**

<b>47</b>	<b>F1</b>	<b>20</b>	<b>00</b>
-----------	-----------	-----------	-----------

# ЛАБОРАТОРНЫЕ РАБОТЫ

---

## *Лабораторная работа № 1*

### **КОНФИГУРИРОВАНИЕ МОДУЛЕЙ ВВОДА-ВЫВОДА ОВЕН МХ110 ДЛЯ РАБОТЫ В СЕТИ ПО ПРОТОКОЛУ MODBUS RTU**

*Цель работы:* Освоение программы «Конфигуратор Мх110» для модулей ввода-вывода компании ОВЕН и получение навыков организации работы модулей в сети с интерфейсом RS-485 по протоколу Modbus RTU.

*Задание на лабораторную работу.* Имеется следующий состав модулей ввода-вывода: МВ110-8АС – модуль 8-канального аналогового ввода; МУ110-6У – модуль 6-канального аналогового вывода; МВ110-16ДН – модуль 16-канального дискретного ввода; МУ110-16Р – модуль 16-канального дискретного вывода. Требуется для модулей задать сетевые параметры обмена:

- скорость обмена данными – 115 200 бит/с;
- формат передачи – 8N1 (8 – количество бит в посылке, N – бит контроля четности отсутствует, 1 – количество стоп-бит);
- протокол обмена по сети RS-485 – Modbus RTU;
- для каждого модуля установить индивидуальный адрес в сети RS-485: МВ110-8АС – адрес 1, МУ110-6У – адрес 2, МВ110-16ДН – адрес 3, МУ110-16Р – адрес 4.
- для модуля аналогового ввода МВ110-8АС дополнительно для всех входов задать: тип подключаемого датчика «0...10В»; нижняя и верхняя граница диапазона измерения – соответственно 0 и 10 000; смещение десятичной точки – 0.

## Краткие сведения о линейке модулей ввода-вывода ОВЕН Мх110

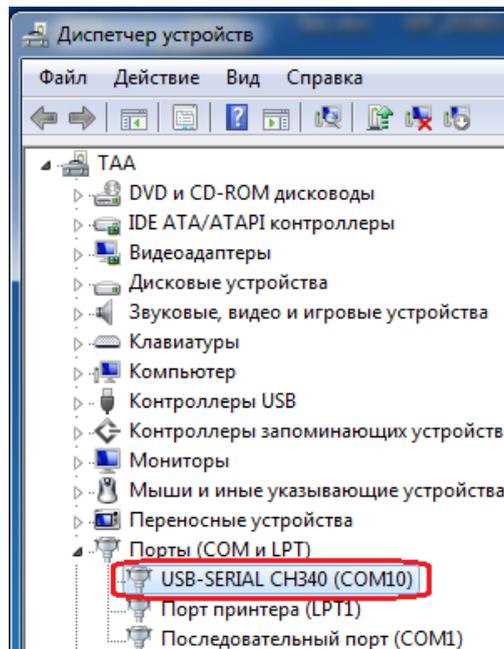
Модули ввода-вывода ОВЕН Мх110 широко применяются при автоматизации технологических процессов в различных отраслях промышленности. Линейка модулей ОВЕН Мх110 включает в себя модули аналогового и дискретного ввода-вывода, модули для измерения параметров электрической сети и модули для работы с сигналами с тензометрических датчиков. Сбор данных и управление с использованием этих модулей осуществляется по сети с интерфейсом RS-485, при этом модули ввода-вывода выступают как Slave-устройства. В качестве мастера в сети могут использоваться: программируемый логический контроллер (рис. 3.1), сенсорная панель оператора, компьютер с установленной SCADA-системой.

Конфигурирование модулей Мх110 осуществляется на компьютере через адаптер интерфейса RS-485/RS-232 или RS-485/USB (например, ОВЕН АС3-М или АС4, соответственно) с помощью программы «Конфигуратор М110», которую можно скачать с официального сайта компании ОВЕН из раздела «Поддержка/ Оборудование для автоматизации/Документация и ПО/Сервисное ПО». После скачивания программы необходимо запустить ее и следовать инструкциям, отображаемым в окнах в процессе установки. Для конфигурирования «новых» модулей ввода-вывода (с версией аппаратной платформы Н/В 2.0) используется программа «Owen Configurator», которую также можно найти на сайте ОВЕН в разделе «...Сервисное ПО».

При использовании для связи с модулями адаптера интерфейса (например, RS-485/USB АС4), предварительно на компьютер необходимо установить соответствующий драйвер адаптера, который также можно скачать с официального сайта ОВЕН.



Рис. 3.1. Пример использования модулей ввода-вывода Мх110



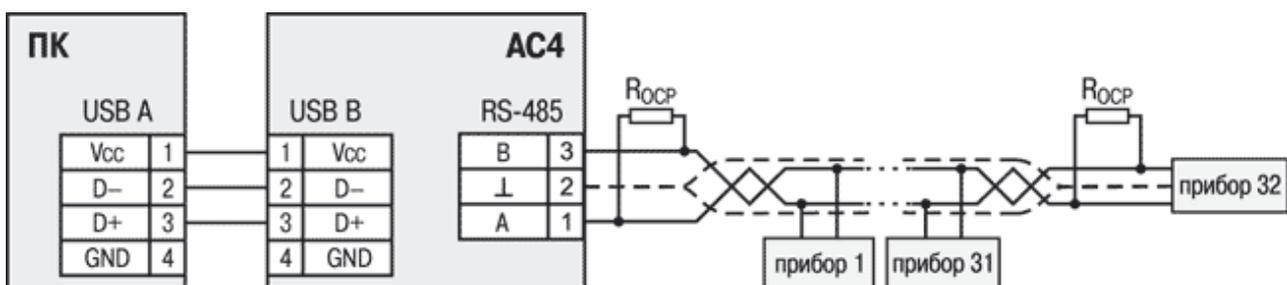
**Рис. 3.2. Виртуальный СОМ-порт**

После удачной установки драйвера в диспетчере устройств операционной системы будет отражаться виртуальный СОМ-порт (рис. 3.2).

### Порядок выполнения работы

1. *Подключение конфигурируемого модуля к компьютеру через адаптер USB/RS485.* Перед началом процедуры конфигурирования соответствующего модуля ввода-вывода его необходимо подключить к преобразователю интерфейсов АС4. Для этого следует:

- подсоединить USB-порт преобразователя интерфейсов к USB-порту персонального компьютера;
- подсоединить кабель интерфейса RS-485 к от преобразователя интерфейсов к модулю ввода-вывода, соблюдая полярность (рис. 3.3).



**Рис. 3.3. Схема подключения преобразователь интерфейсов АС4**

2. *Запуск конфигуратора и установка связи с модулем.* Для этого необходимо запустить программу «Конфигуратор M110». Установка связи происходит при значениях параметров порта, установленных при предыдущем запуске программы. Если связь с прибором не устанавливается, то открывается окно «Установка связи с прибором» (рис. 3.4). Для подключения к модулю следует указать его сетевые настройки. Если настройки неизвестны, то необходимо сбросить настройки на заводские (процесс сброса описан в руководстве по эксплуатации на модуль) и подключиться с помощью кнопки Заводские сетевые настройки. После успешной установки связи с модулем ввода-вывода открывается главное окно программы (рис. 3.5).

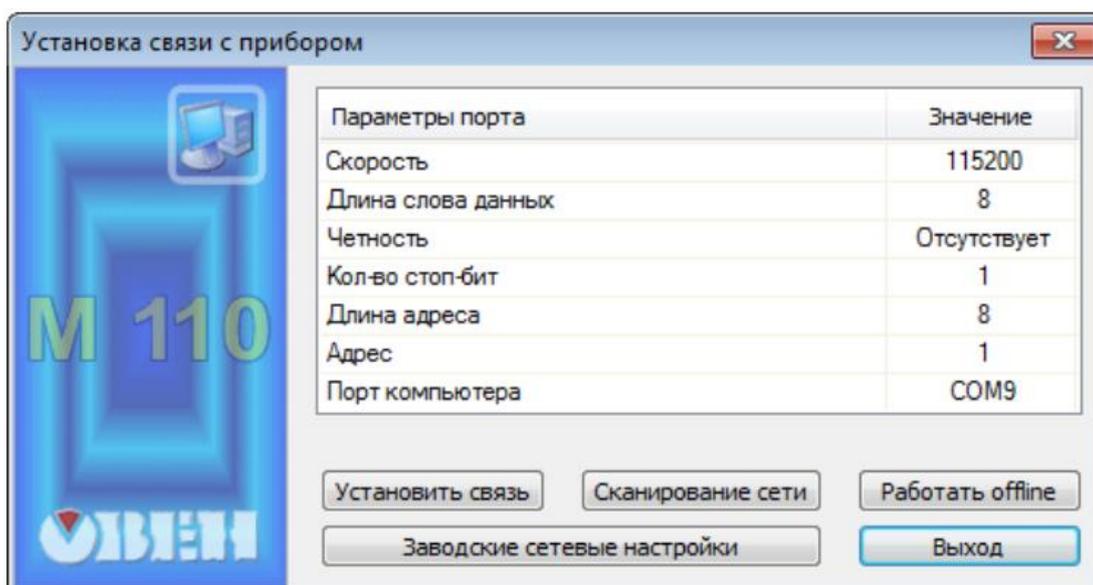


Рис. 3.4. Окно «Установка связи с прибором»

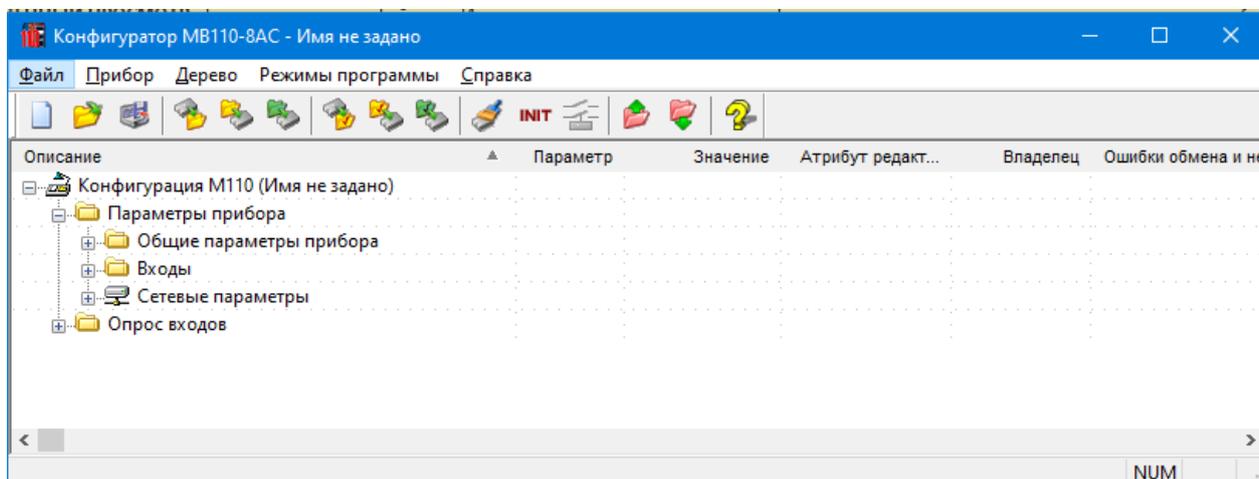


Рис. 3.5. Главное окно программы

3. Установка требуемых сетевых параметров модулей ввода-вывода. Для установки сетевых параметров модуля необходимо открыть группу «Сетевые параметры» (рис. 3.6) и задать/выбрать требуемые значения (в соответствии с заданием на лабораторную работу). Для примера на рис 3.6 приведена конфигурация сетевых параметров модуля аналогового ввода MB110-8AC. Некоторые модули ввода-вывода компании ОВЕН поддерживают автоматическое определение протокола обмена по RS-485. Модуль MB110-8AC относится к этой категории, поэтому среди перечня сетевых параметров отсутствует протокол обмена. Для некоторых модулей (например МУ110-6У) требуется явно задавать необходимый протокол обмена (рис. 3.7).

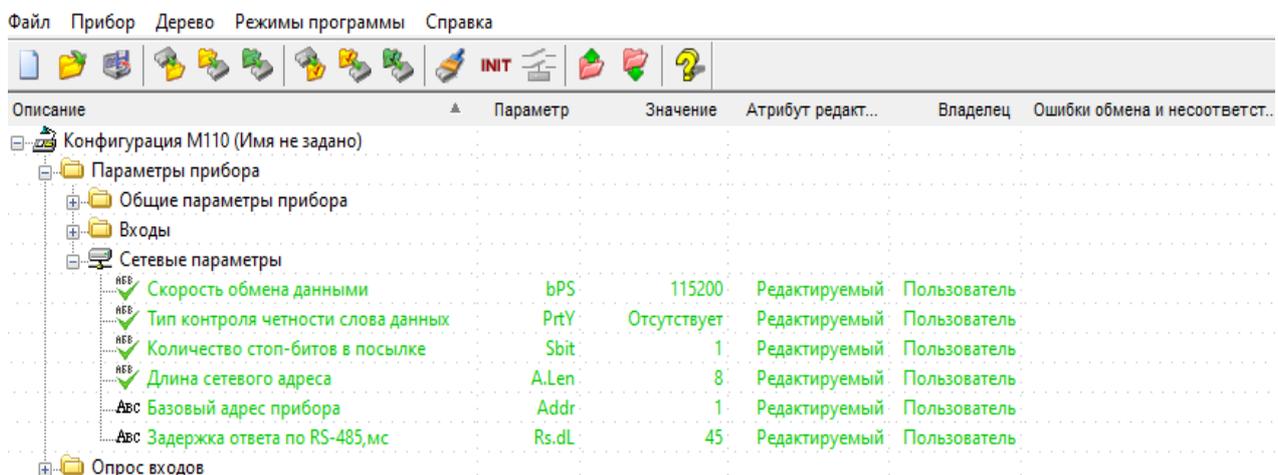


Рис. 3.6. Конфигурирование сетевых параметров модуля MB110-8AC

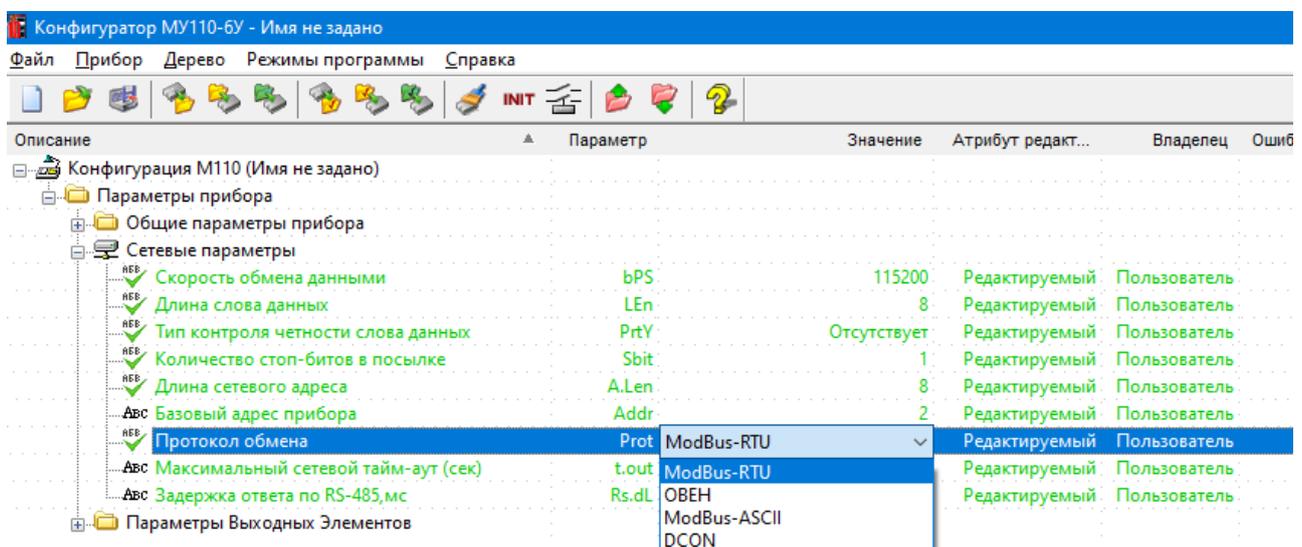


Рис. 3.7. Конфигурирование сетевых параметров модуля МУ110-6У

4. Конфигурирование входов для модуля MB110-8АС. Конфигурирование осуществляется в группе «Входы». Для каждого аналогового входа можно задать тип подключаемого датчика, постоянную времени для экспоненциального фильтра, нижнюю и верхнюю границы диапазона измерения активного датчика, смещение десятичной точки, ограничение скорости изменения измеряемого сигнала, тип выходного фильтра. В соответствии с заданием на лабораторную установить требуемые параметры для каждого из 8 аналоговых входов (рис. 3.8).

5. Сохранение измененных параметров в постоянной памяти модулей. Измененные значения параметров модулей отображаются зеленым шрифтом. Запись измененных значений параметров из программы в прибор осуществляется нажатием соответствующей кнопки панели инструментов (рис. 3.9). В случае записи измененных сетевых параметров появляется окно с предупреждением переключения прибора на новые сетевые параметры (рис. 3.10).

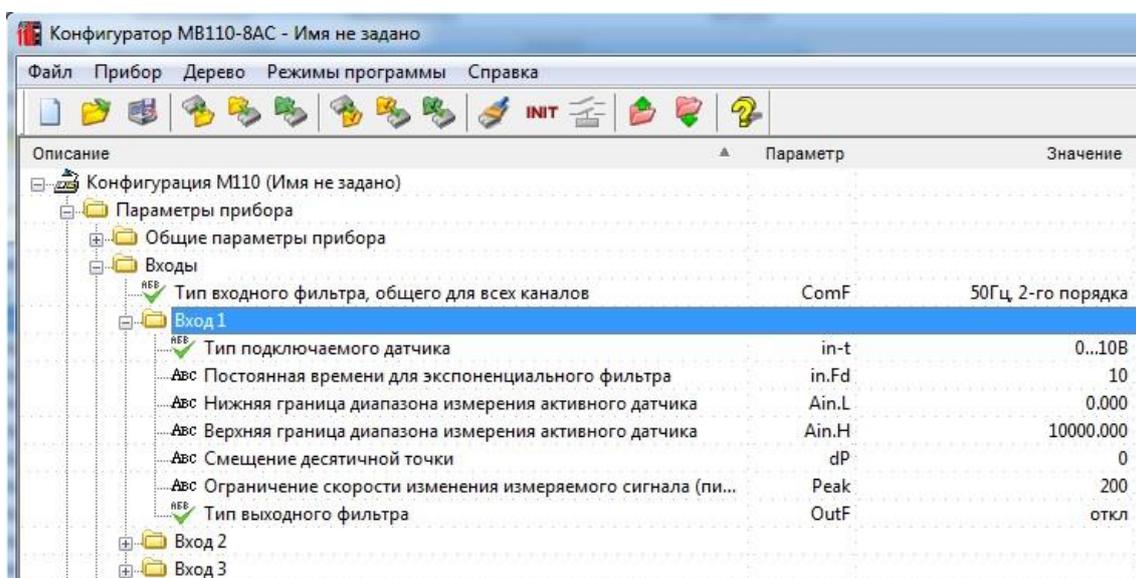


Рис. 3.8. Параметры входов модуля MB110-24/220.8АС

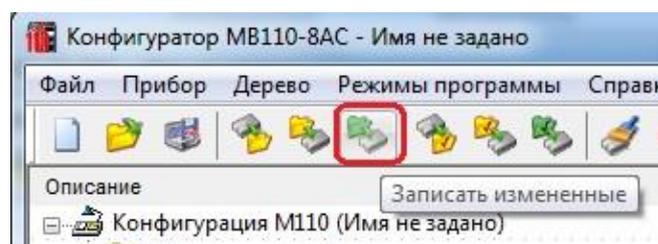


Рис. 3.9. Запись измененных значений параметров в прибор

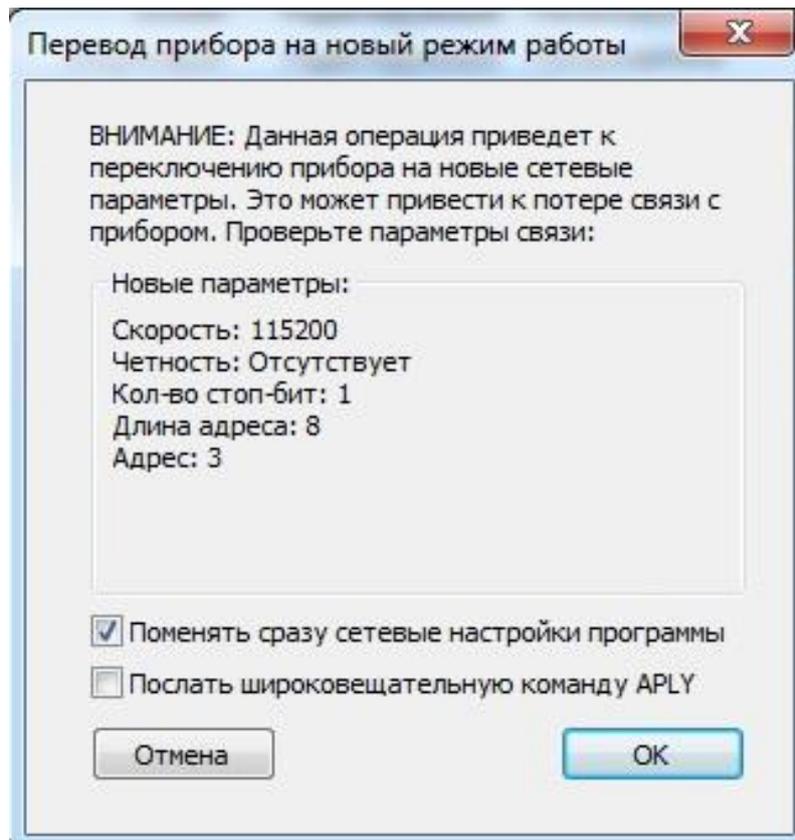


Рис. 3.10. Применение новых сетевых параметров

### Контрольные вопросы

1. Какие протоколы обмена поддерживают модули ввода-вывода ОВЕН Мх110?
2. Какие скорости обмена данными поддерживают модули ввода-вывода ОВЕН Мх110?
3. Как расшифровывается формат 8N1?
4. Какие значения адреса могут быть установлены модулям ввода-вывода при использовании протокола Modbus RTU?
5. Какая программа используется для конфигурирования модулей ввода-вывода ОВЕН Мх110? Где ее можно скачать?
6. Расскажите про процедуру установки сетевых параметров при конфигурировании модулей ввода-вывода ОВЕН Мх110?
7. Какие виды измеряемых сигналов поддерживает модуль аналогового ввода МВ110-8АС?

## **КОНФИГУРИРОВАНИЕ МОДУЛЕЙ ВВОДА-ВЫВОДА ОВЕН Mx210 ДЛЯ РАБОТЫ В СЕТИ ПО ПРОТОКОЛУ MODBUS TCP**

*Цель работы:* Освоение программы «Owen Configurator» для модулей ввода-вывода компании ОВЕН и получение навыков организации работы модулей в сети Ethernet по протоколу Modbus TCP.

*Задание на лабораторную работу.* Имеется следующий состав модулей ввода-вывода: MB210-101 – модуль 8-канального аналогового ввода; МУ210-502 – модуль 6-канального аналогового вывода; МК210-301 – модуль 6-канального дискретного ввода / 8-канального дискретного вывода. Требуется для модулей задать сетевые параметры обмена:

- для каждого модуля установить индивидуальный адрес в сети Ethernet: MB210-101 – адрес 192.168.0.1, МУ210-502 – адрес 192.168.0.2, МК210-301 – адрес 192.168.0.3;
- маска подсети – 255.255.255.0
- адрес шлюза – 192.168.0.254;
- для модуля аналогового ввода MB110-8АС дополнительно для всех входов задать: тип подключаемого датчика «–1...1 В»; нижняя и верхняя граница диапазона измерения – соответственно –100 и 100; положение десятичной точки – 1.

### **Краткие сведения о линейке модулей ввода-вывода ОВЕН Mx210**

Модули ввода-вывода ОВЕН Mx210 являются новой линейкой модулей ввода-вывода компании и ориентированы на работу в сети с интерфейсом Ethernet. Линейка модулей ОВЕН Mx210 включает в себя модули аналогового и дискретного ввода-вывода и модули для измерения параметров электрической сети. Основным протоколом для работы с модулями серии ОВЕН Mx210 является протокол Modbus TCP. Помимо него модули поддерживают протоколы MQTT, SNMP, NTP.

Сеть сбора данных и управления, базирующаяся на протоколе Modbus TCP, реализована по принципу Client-Server. При этом модули ввода-вывода выступают как серверные устройства, причем допускается до 4-х одновременных соединений. В качестве клиентов в сети могут использоваться: программируемые логический контроллеры (рис. 3.11), сенсорные панели оператора, компьютеры с установленной SCADA-системой.

Конфигурирование модулей Mx210 осуществляется по Ethernet или через USB (разъем типа microUSB) с помощью программы «Owen Configurator», которую можно скачать с официального сайта компании OWEN из раздела «Поддержка/Оборудование для автоматизации/Документация и ПО/Сервисное ПО». После скачивания программы необходимо запустить ее и следовать инструкциям, отображаемым в окнах в процессе установки.

При подключении модуля ввода-вывода через USB в системе создается виртуальный COM-порт (рис. 3.12), который становится доступен в программе «Owen Configurator». При этом для связи рекомендуется использовать специальный протокол «Owen Auto Detection Protocol».



Рис. 3.11. Пример использования модулей ввода-вывода Mx210

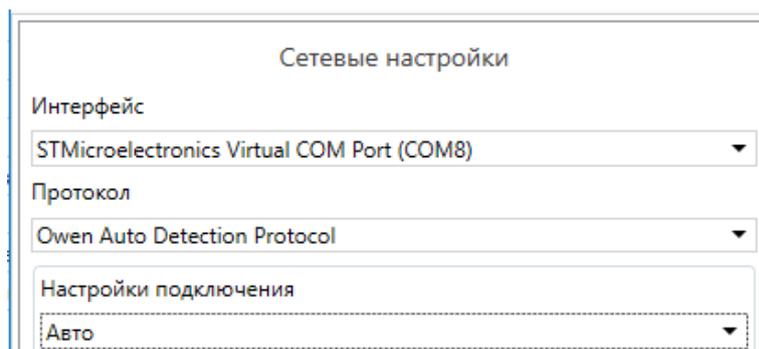


Рис. 3.12. Сетевые настройки для связи с модулем ввода-вывода через USB

## Порядок выполнения работы

1. Подключить к ПК конфигурируемое устройство с помощью одного из интерфейсов связи.
2. Установить на компьютер и запустить программу «Owen Configurator». После запуска «Owen Configurator» на мониторе ПК открывается Главное окно (рис. 3.13).

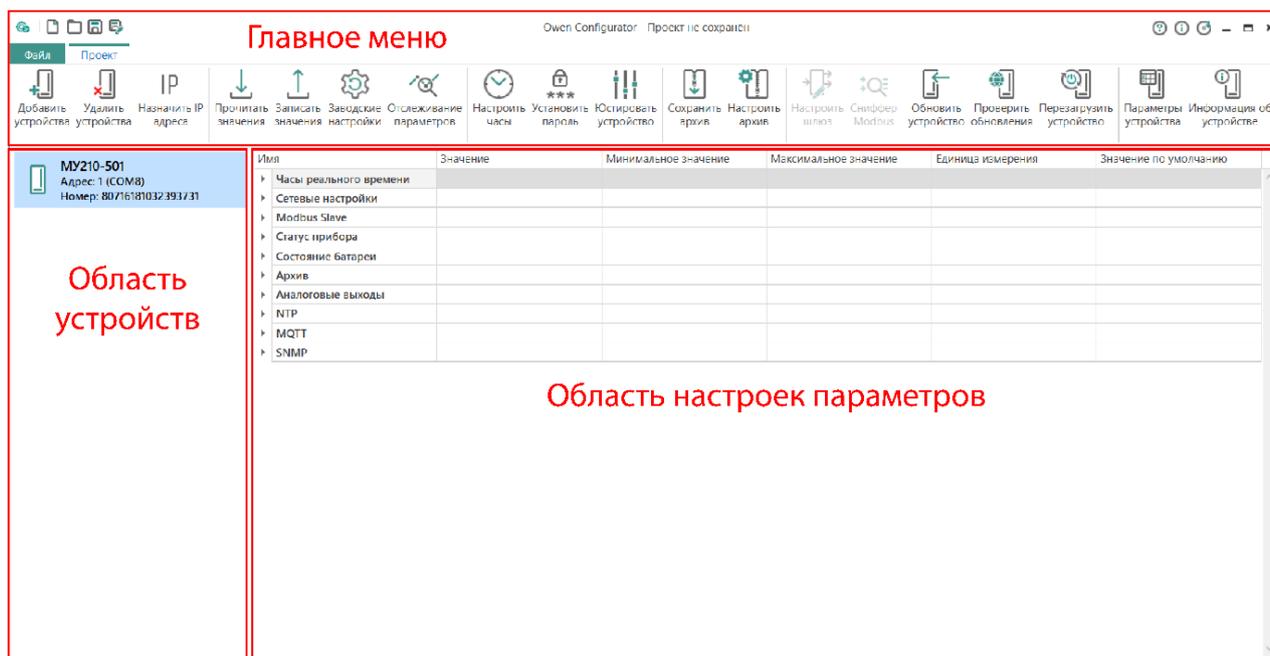
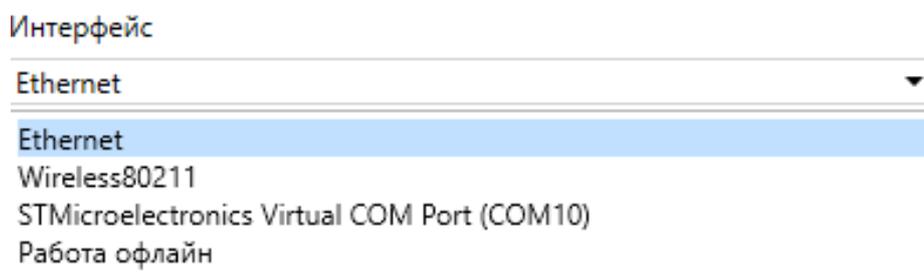
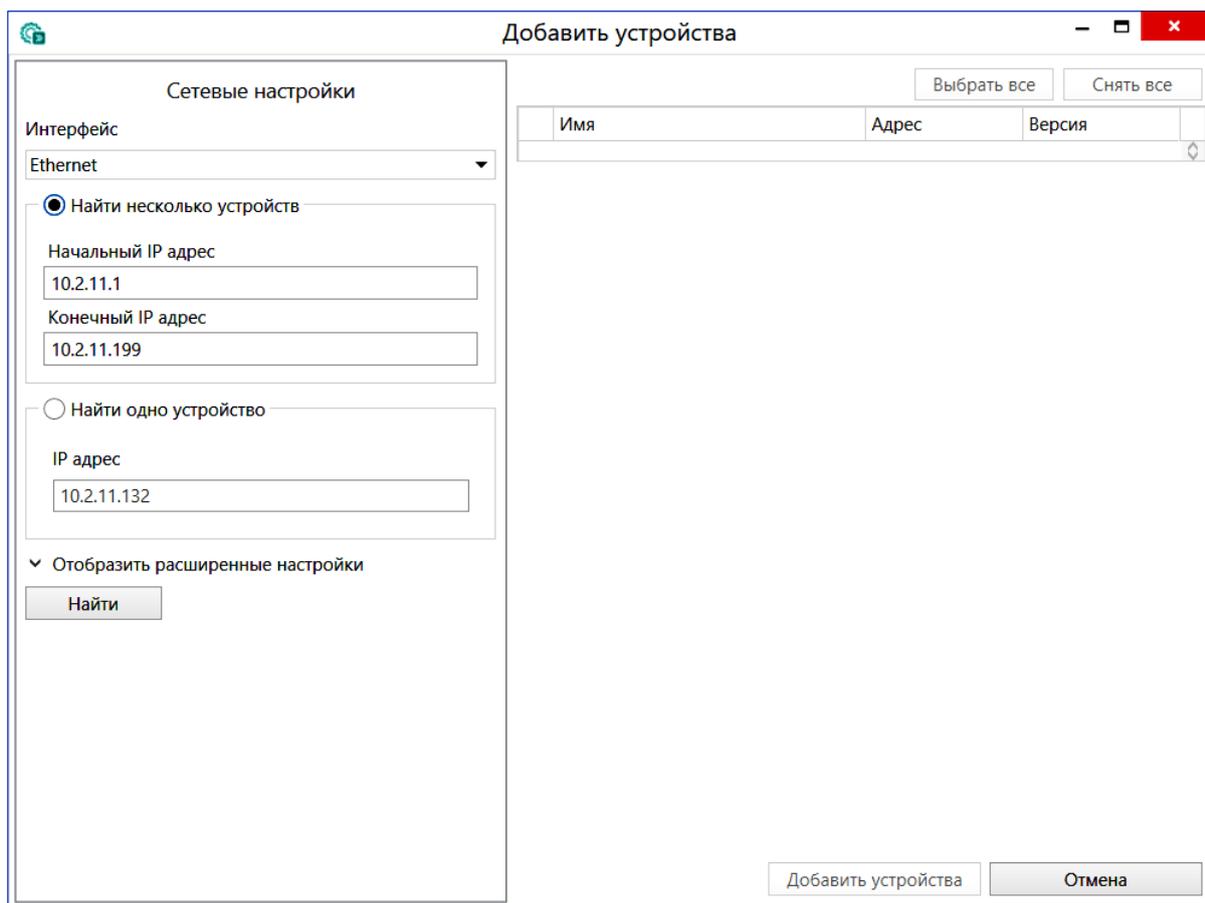


Рис. 3.13. Интерфейс программы Owen Configurator

3. Добавить устройство в проект. Для добавления устройства следует нажать кнопку  (Добавить устройства) в главном меню «Проект». Откроется окно выбора интерфейса подключения и поиска устройств (рис. 3.14). Для добавления устройства по интерфейсу Ethernet в выпадающем меню Интерфейс следует выбрать пункт Ethernet





**Рис. 3.14. Окно поиска и добавления устройств в проект**

Поиск устройств выполняется по IP-адресам. С помощью переключателя следует выбрать способ поиска:

- Найти несколько устройств – задается диапазон IP-адресов для сканирования;
- Найти одно устройство – задается один IP-адрес.

Для поиска группы устройств или устройства, адрес которого неизвестен, используется поиск в диапазоне адресов. В поля ввода следует задать начальный и конечный адреса. После нажатия кнопки Найти, будет просканирован каждый IP-адрес в указанном диапазоне. Результаты поиска для каждого IP-адреса будут отображены в поле справа.

\* Для вновь приобретенного модуля ввода-вывода IP-адрес 192.168.1.99

4. Настроить сетевые параметры устройств в соответствии с заданием на лабораторную работу. Для обмена данных модуля в сети Ethernet должны быть заданы параметры, приведенные ниже в таблице.



## **Контрольные вопросы**

1. Какие протоколы обмена поддерживают модули ввода-вывода ОВЕН Мх210?
2. Какие физические интерфейсы могут быть использованы для конфигурирования модулей ввода-вывода ОВЕН Мх210?
3. Какой физический интерфейс имеют сети с протоколом Modbus TCP?
4. Какая программа используется для конфигурирования модулей ввода-вывода ОВЕН Мх210? Где ее можно скачать?
5. Какими способами может быть установлены сетевые параметры при конфигурировании модулей ввода-вывода ОВЕН Мх210?
6. Расскажите про процедуры установки сетевых параметров при конфигурировании модулей.
7. Какие виды измеряемых сигналов поддерживает модуль аналогового ввода ОВЕН МВ210-101?

**ОРГАНИЗАЦИЯ СВЯЗИ КОНТРОЛЛЕРОВ ОВЕН ПЛК1XX  
С МОДУЛЯМИ ДИСКРЕТНОГО ВВОДА-ВЫВОДА МХ110  
ПО ПРОТОКОЛУ MODBUS RTU**

*Цель работы:* получение навыков организации обмена данными между контроллером и модулями дискретного ввода-вывода компании ОВЕН с использованием протокола Modbus RTU.

*Задание на лабораторную работу.* Имеется контроллер ПЛК-100 (сетевой адрес 10.0.6.10), к которому по сети RS-485 подключены модули ввода-вывода: модуль дискретного ввода МВ110-16ДН (адрес в сети RS-485 – 3) и модуль дискретного вывода МУ110-16Р (адрес 4). Модули ввода-вывода имеют следующие сетевые настройки: скорость обмена данными по сети – 115200 бит/с; формат передачи данных – 8N1 (8 – количество бит в посылке, N – бит контроля четности отсутствует, 1 – количество стоп-бит); протокол обмена по сети RS-485 – Modbus RTU.

В ходе выполнения работы необходимо:

1) получить информацию от модуля МВ110-16ДН (адрес 3) о состоянии двух дискретных входов  $x_1$  и  $x_2$ ;

2) осуществить вычисление следующих выражений:

$$y_1 = x_1 \cup x_2, \quad y_2 = x_1 \cap x_2;$$

3) вывести через модуль МУ110-16Р (адрес 4) полученные результаты выражений  $y_1$  и  $y_2$ .

**Краткие сведения по организации обмена**

**с модулями ввода/вывода ОВЕН по протоколу Modbus RTU**

Протокол Modbus является самым распространенным в мире. Различают три разновидности протокола Modbus: Modbus ASCII, Modbus RTU и Modbus TCP. Сеть, построенная на базе протокола Modbus RTU, работает по принципу Master/Slave, при этом в сети может быть только один Master и до 247 Slave-

узлов. Мастером в сети может быть контроллер, а Slave-узлами могут выступать модули ввода/вывода.

Для организации связи по протоколу Modbus RTU необходимо знать:

1) сетевые параметры, среди которых скорость обмена по сети, вид протокола. Все устройства в одной сети должны иметь одинаковые сетевые параметры;

2) сетевые адреса Slave-узлов, Master-узел сетевого адреса не имеет;

3) распределение Modbus-регистров для Slave-узлов. Эта информация обычно приводится в руководствах по эксплуатации на соответствующее устройство. В таблицах 3.1 и 3.2 приведены в качестве примера распределение Modbus-регистров модуля MB110-16ДН и некоторых параметров модуля МУ110-16Р;

4) поддерживаемые функции (команды) протокола Modbus для чтения и записи Modbus-регистров. Эта информация также приводится в руководствах по эксплуатации. Для модулей ввода/вывода ОВЕН запись в регистры осуществляется командой 16 (0x10), чтение – командами 03 или 04 (прибор поддерживает обе команды).

### 3.1. Карта Modbus-регистров модуля MB110-16ДН

Параметр	Значение, [ед. изм.]	Тип	Адрес регистра	
			(Hex)	(Dec)
Максимальный сетевой тайм-аут	1...600 [с]	Int16	0030	0048
Битовая маска значений входов	0...65535	Int16	0033	0051
Значение счетчика входа № 1	0...65535 [имп]	Int16	0040	0064
Значение счетчика входа № 2	0...65535 [имп]	Int16	0041	0065
Значение счетчика входа № 3	0...65535 [имп]	Int16	0042	0066
Значение счетчика входа № 4	0...65535 [имп]	Int16	0043	0067
Значение счетчика входа № 5	0...65535 [имп]	Int16	0044	0068
Значение счетчика входа № 6	0...65535 [имп]	Int16	0045	0069
Значение счетчика входа № 7	0...65535 [имп]	Int16	0046	0070
Значение счетчика входа № 8	0...65535 [имп]	Int16	0047	0071
Значение счетчика входа № 9	0...65535 [имп]	Int16	0048	0072
Значение счетчика входа № 10	0...65535 [имп]	Int16	0049	0073
Значение счетчика входа № 11	0...65535 [имп]	Int16	004A	0074
Значение счетчика входа № 12	0...65535 [имп]	Int16	004B	0075
Значение счетчика входа № 13	0...65535 [имп]	Int16	004C	0076
Значение счетчика входа № 14	0...65535 [имп]	Int16	004D	0077
Значение счетчика входа № 15	0...65535 [имп]	Int16	004E	0078
Значение счетчика входа № 16	0...65535 [имп]	Int16	004F	0079

### 3.2. Карта некоторых Modbus-регистров модуля МУ110-16Р

Параметр	Значение (ед. изм.)	Тип	Адрес регистра	
			(Hex)	(Dec)
Коэффициент заполнения ШИМ на выходе 1	0...1000 (0,1 %)	Uint16	0x0000	0000
Коэффициент заполнения ШИМ на выходе 2	0...1000 (0,1 %)	Uint16	0x0001	0001
Коэффициент заполнения ШИМ на выходе 3	0...1000 (0,1 %)	Uint16	0x0002	0002
Безопасное состояние выхода 1	0...1000 (0,1 %)	Uint16	0x0010	0016
Безопасное состояние выхода 2	0...1000 (0,1 %)	Uint16	0x0011	0017
Безопасное состояние выхода 3	0...1000 (0,1 %)	Uint16	0x0012	0018
Период ШИМ на выходе 1	1...900 с	Uint16	0x0020	0032
Период ШИМ на выходе 2	1...900 с	Uint16	0x0021	0033
Период ШИМ на выходе 3	1...900 с	Uint16	0x0022	0034
Битовая маска значений выходов	0...65535	Uint16	0x0032	0050

5) работу с дискретными входами и выходами модулей удобно (и эффективно) осуществлять через битовую маску значений входов и выходов.

#### Порядок выполнения работы

1. Запустить CoDeSys и выбрать целую платформу контроллера, на котором будет исполняться проект, например, PLC100.R-L. Предварительно target-файлы соответствующего контроллера должны быть установлены. Скачать таргет-файлы контроллеров ОВЕН ПЛК можно с официального сайта ОВЕН из раздела «Каталог продукции / Программное обеспечение, устройства связи / Среда программирования CODESYS».

2. При создании нового программного компонента (POU) выбрать пункт «Программа», в поле названия программы – оставить имя PLC\_PRG и выбрать язык «ST».

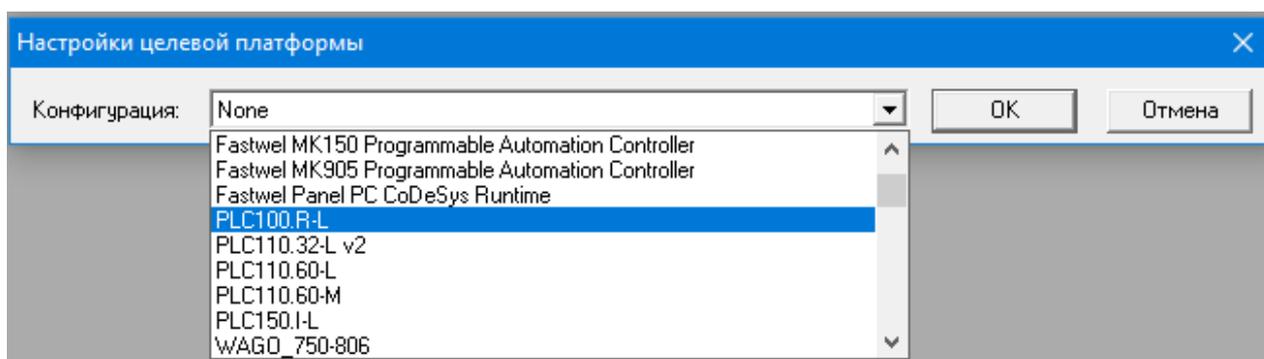


Рис. 3.16. Выбор целевой платформы

3. Перейти на вкладку «Ресурсы» и открыть окно «Конфигурация ПЛК».
4. Добавить поддержку портов интерфейса связи с модулями ввода/вывода – подэлемент Modbus (Master) (рис. 3.17).
5. Выбрать для связи с модулями ввода/вывода порт RS-485-1 (рис. 3.18).
6. Настроить параметры связи порта RS-485-1 на вкладке «Параметры модуля». При этом выбрать (рис. 3.19): скорость обмена (Communication Speed) 115200, проверка на четность (Parity) – NO PARITY CHECK; количество бит даны (Data Bits) – 8; количество стоп-бит (Stop Length) – One Stop Bit, тип интерфейса (Interface Type) – RS485; вид протокола (Frame Oriented) – RTU.

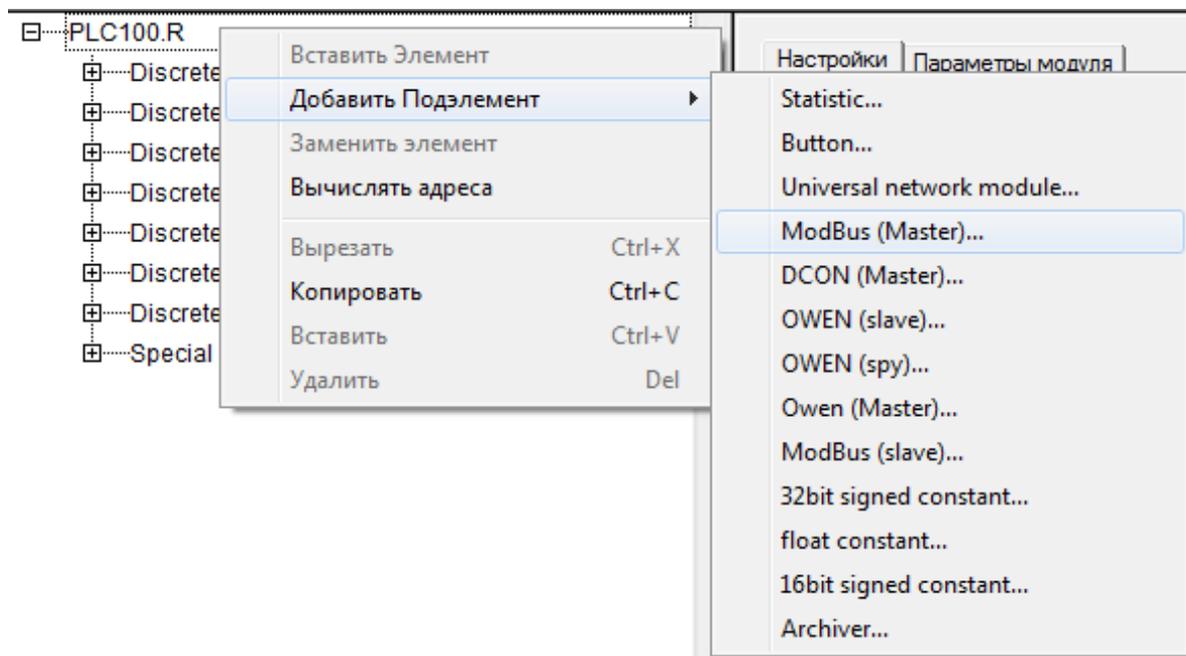


Рис. 3.17. Добавление подэлемента Modbus (Master)

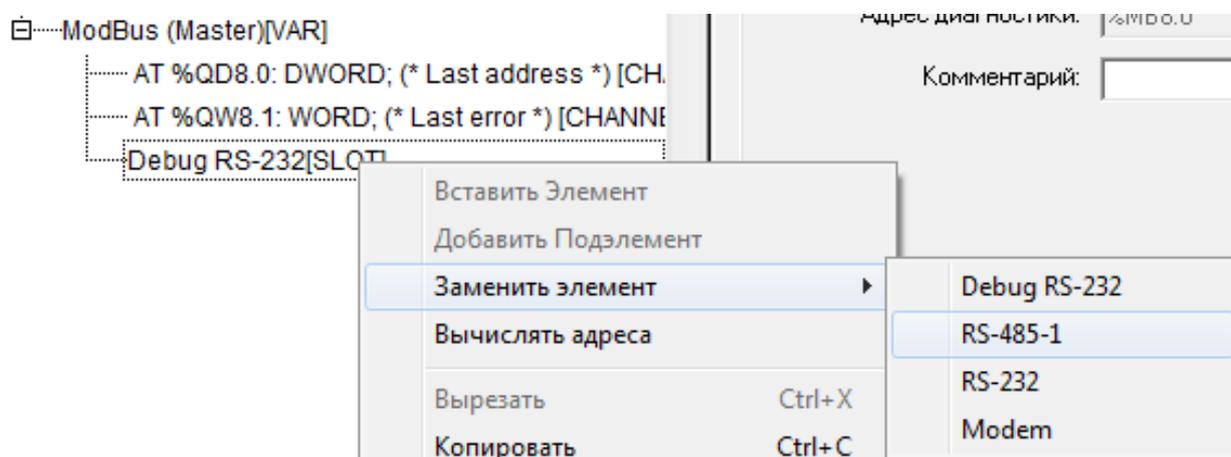


Рис. 3.18. Выбор порта связи с модулями ввода/вывода

Индекс	Имя	Значение	По умолч.	Мин.	Макс.
1	Communication speed	11520	11520		
2	Parity	NO PARITY...	NO PARITY C...		
3	Data bits	8 bits	8 bits		
4	Stop length	One stop bit	One stop bit		
5	Interface Type	RS485	RS485		
6	Frame oriented	RTU	ASCII		
7	Framing time ms	0	0	0	32000
8	Visibility	No	No		

**Рис. 3.19. Параметры связи Modbus (Master)**

7. Добавить новое устройство (подэлемент) – Universal Modbus Device. Для этого устройства устанавливается основной параметр (вкладка «Параметры модуля») – Module Slave Address (значение устанавливается согласно адресу заданному модулю ввода/вывода при его конфигурировании). Для модуля MB110-16ДН устанавливается адрес 3 (рис. 3.20). Остальные параметры модулей оставить заданными по умолчанию. На вкладке «Базовые параметры» рекомендуется описать комментарии по модулю (например, MB110-16ДН – адр. 3), чтобы в будущем не ошибиться.

8. Для первого «Universal Modbus Device» (MB110-16ДН) – модуля дискретного ввода добавить подэлемент «Register Input Module». Работу с дискретными входами этого модуля будем осуществлять через битовую маску входов. В соответствии с руководством по эксплуатации этого модуля этот регистр имеет адрес 51 (см. табл. 3.1).

Индекс	Имя	Значение	По умолч.	Мин.	Макс.
1	ModuleIP	10:0:0:223	10:0:0:223		
2	Max timeout	150	150	10	
3	TCPport	502	502		
4	NetMode	Serial	Serial		
5	ModuleSlaveAddress	3	1	0	255
6	Work mode	By poll time	By poll time		
7	Polling time ms	100	100	10	10000
8	Visibility	No	No		
9	Amount Repeat	0	0	0	100
10	Byte Sequence	Trace_mo...	Trace_mode		

**Рис. 3.20. Установка адреса для Modbus Slave**

9. Настроить параметры этого подэлемента: на вкладке «Параметры модуля» установить адрес регистра («Register Adress») 51 и выбрать команду («Command») 0x03 или 0x04 (рис. 3.21).

10. В конфигурации ПЛК присвоить имя этому каналу (Register Input Module), например, Reg51.

11. Добавить новый модуль МУ110-16Р (адрес 4) – аналогично п. 7.

12. Для второго «Universal Modbus Device» (МУ110-16Р) добавить подэлемент «Register Output Module». Работу с дискретными выходами этого модуля будем осуществлять через битовую маску выходов. В соответствии с руководством по эксплуатации модуля этот регистр имеет адрес 50.

13. В конфигурации ПЛК присвоить имя этому каналу (Register Output Module), например, Reg50 (рис. 3.22).

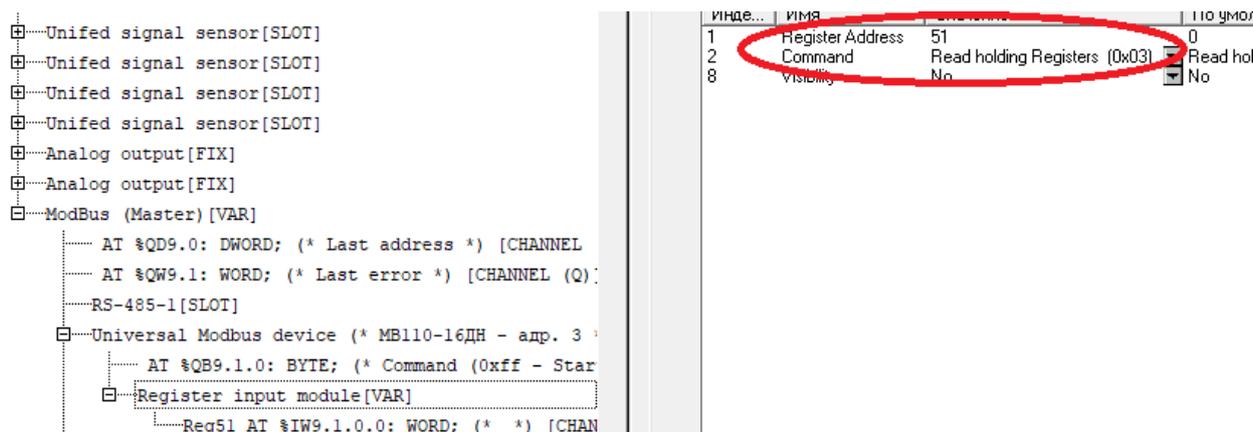


Рис. 3.21. Редактирование адреса регистра и кода функции

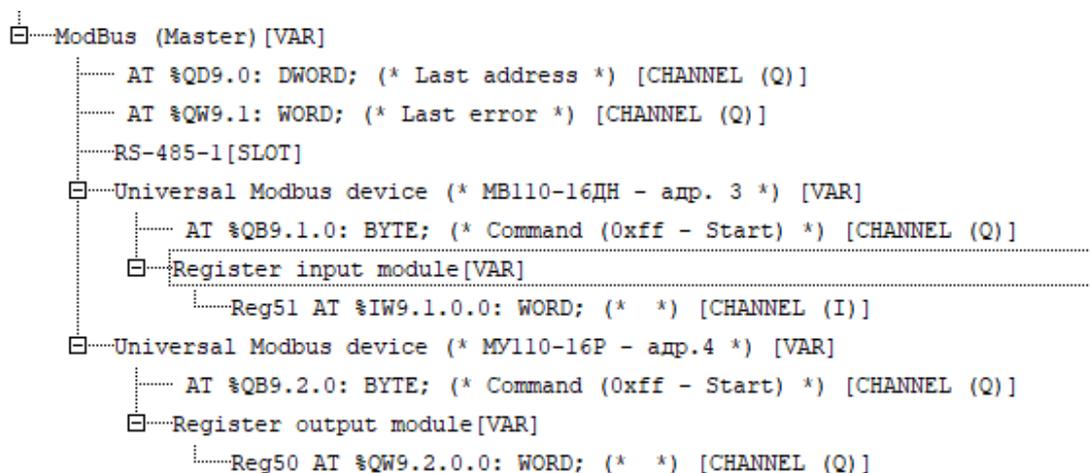


Рис. 3.22. Описание конфигурации ПЛК

14. Настроить параметры этого подэлемента: на вкладке «Параметры модуля» установить адрес регистра («Register Adress») 50 и выбрать команду («Command») 0x10 (Write Multiple Registers).

15. Отредактировать программу PLC\_PRG в соответствии с рис. 3.23.

16. Установить параметры связи с ПЛК, используя команду «Параметры связи» меню «Онлайн». В лабораторной работе связь ПЛК с компьютером осуществляется по сети Ethernet, поэтому необходимо установить IP-адрес контроллера (10.0.6.10, порт 1200) и подтвердить действия, нажав кнопку «Ок» (рис. 3.24).

17. Откомпилировать и загрузить проект в контроллер, воспользовавшись командой «Подключения» меню Онлайн» или кнопкой .

18. Проверить функционирование программы.

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     x1: BOOL;
0004     x2: BOOL;
0005     y1: BOOL;
0006     y2: BOOL;
0007 END_VAR
0008
0001 x1:=reg51.0;
0002 x2:=reg51.1;
0003
0004 y1:=x1 OR x2;
0005 y2:=x1 AND x2;
0006
0007 reg50.0:= y1;
0008 reg50.1:=y2;
0009
```

Рис. 3.23. Программа PLC\_PRG

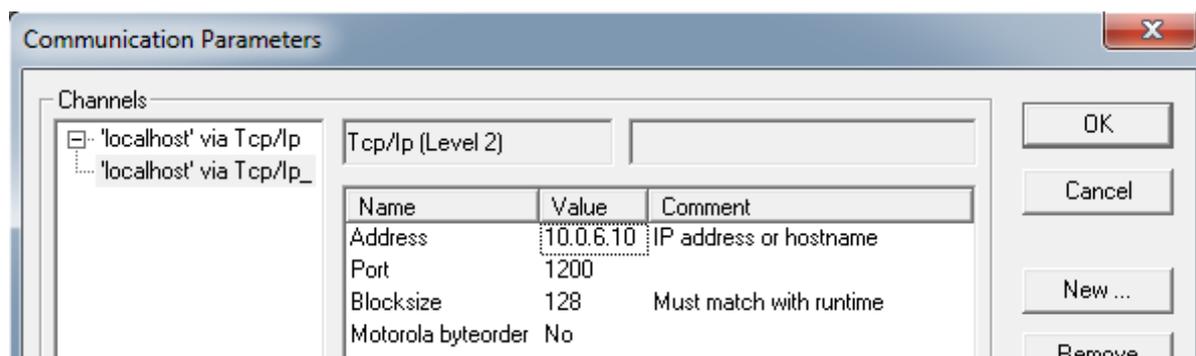


Рис. 3.24. Установка IP-адреса контроллера в параметрах связи

## Контрольные вопросы

1. Какой принцип доступа к шине заложен в протоколы Modbus RTU и Modbus ASCII?
2. Какое максимальное количество ведомых устройств может присутствовать в сети Modbus RTU?
3. Сколько ведущих устройств может присутствовать в сети Modbus RTU?
4. Какие параметры для Universal Modbus Device необходимо установить при настройке для работы по протоколу Modbus RTU?
5. Что такое регистр Modbus?
6. Что такое карта адресов Modbus-регистров?
7. Как можно узнать адрес или номер регистра требуемого параметра у конфигурируемых устройств?
8. Какие коды Modbus функций поддерживает модуль дискретного ввода MB110-16ДН?
9. Какие коды Modbus функций поддерживает модуль дискретного вывода MB110-16Р?

**ОРГАНИЗАЦИЯ СВЯЗИ КОНТРОЛЛЕРОВ ОВЕН ПЛК1XX  
С МОДУЛЯМИ АНАЛОГОВОГО ВВОДА-ВЫВОДА  
ПО ПРОТОКОЛУ MODBUS RTU**

*Цель работы:* получение навыков организации обмена данными между контроллером и модулями аналогового ввода компании ОВЕН с использованием протокола Modbus RTU.

*Задание на лабораторную работу.* Имеется контроллер ПЛК-100 (сетевой адрес 10.0.6.10), к которому по сети RS-485 подключен модуль ввода-вывода: модуль аналогового ввода MB110-8AC (адрес в сети RS-485 – 1). Имеется объект управления (рис. 3.25), в котором требуется регулировать уровень и температуру. В ходе выполнения работы необходимо:

1. Получить информацию от модуля MB110-8AC (адрес 1) в виде кодов АЦП о поданном на его первый вход сигнале от датчика уровня (LE) и поданном на второй вход сигнале от датчика температуры (TE). Значения кодов АЦП 0...10 000 соответствуют диапазонам измерения уровня 0...100% и температуры 0...200 °С.

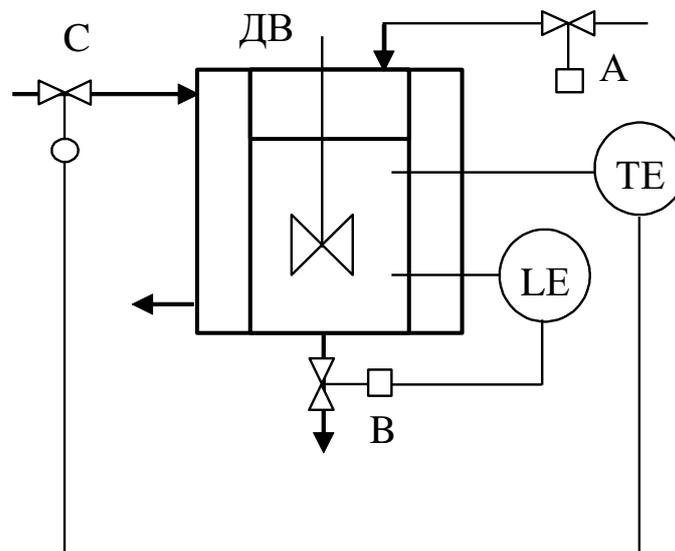


Рис. 3.25. Функциональная схема объекта управления

2. Осуществить преобразование кодов АЦП в физические величины измеренного уровня и температуры.

3. Разработать алгоритм позиционного регулирования уровня в емкости: при достижении уровня величины  $L_{\min}$  – происходит закрытие сливного клапана В, при достижении уровня величины  $L_{\max}$  – происходит открытие сливного клапана В. Управление сливным клапаном осуществляется с помощью первого дискретного выхода контроллера ПЛК-110.

4. Разработать алгоритм ПИД-регулирования температуры в емкости с ограничением выходного сигнала регулятора от 0 до 100. Регулирование осуществляется за счет изменения степени открытия клапана С подачи теплоносителя в нагревающую рубашку емкости.

5. Осуществить преобразование выходного сигнала регулятора в код ЦАП модуля аналогового вывода МУ110-6У (диапазон изменения кода ЦАП – 0...1000).

6. Вывести сигнал управления регулирующим клапаном С с первого выхода модуля аналогового вывода МУ110-6У.

### **Карты адресов Modbus-регистров модулей аналогового ввода/вывода**

В таблицах 3.3 и 3.4 приведены распределения некоторых Modbus-регистров модуля МВ110-8АС и некоторых параметров модуля МУ110-6У.

Чтение регистров осуществляется командами 03 или 04 (прибор поддерживает обе команды).

Запись в регистры осуществляется командой 16 (0x10), чтение – командами 03 или 04 (прибор поддерживает обе команды).

В соответствии с заданием на лабораторную работу к модулю МВ110-8АС подключаются сигналы от датчика уровня (канал 1) и датчика температуры (канал 2). По карте адресов modbus-регистров (табл. 3.3) каналам 1 и 2 соответствуют адреса 0x100 и 0x101. В десятичном представлении эти адреса будут равны соответственно 256 и 257.

### 3.3. Карта адресов Modbus-регистров модуля MB110-8AC

Команда	Адрес регистра	Данные записи	Данные чтения	Примечание
<b>Оперативные параметры</b>				
Измеренное значение <b>iRD</b>	1к - 0x100 2к - 0x101 ... 8к - 0x107	Нет	int_16 (рез. изм)	Только чтение
Измеренное значение <b>iRDt</b>	1к - 0x108...0x109 2к - 0x10a...0x10b ... 8к - 0x116...0x117	Нет	int_16 - рез. изм + Word_16 ( метка относительного времени время, дискретность 10 мс)	Только чтение
Чтение статуса результатов измерения. <b>SRD</b>	0x118... 0x11f	Нет	int_16: Кодирование статуса в таблице 2.2	Только чтение
Измеренное значение <b>Read</b>	1к - 0x120...0x122 2к - 0x123...0x125 ... 8к - 0x135...0x137	Нет	Float_32 (IEEE 754) - рез. изм + Word_16 ( метка относительного времени время, дискретность 10 мс)	Только чтение

### 3.4. Карта адресов Modbus-регистров модуля МУ110-6У

Параметр	Значение [ед. изм.]	Тип	Адрес регистра	
			(Hex)	(Dec)
Значение на выходе № 1	0...1000 [0,1 %]	Uint16	0000	0000
Значение на выходе № 2	0...1000 [0,1 %]	Uint16	0001	0001
Значение на выходе № 3	0...1000 [0,1 %]	Uint16	0002	0002
Значение на выходе № 4	0...1000 [0,1 %]	Uint16	0003	0003
Значение на выходе № 5	0...1000 [0,1 %]	Uint16	0004	0004
Значение на выходе № 6	0...1000 [0,1 %]	Uint16	0005	0005
Аварийное значение на выходе № 1	0...1000 [0,1 %]	Uint16	0010	0016
Аварийное значение на выходе № 2	0...1000 [0,1 %]	Uint16	0011	0017
Аварийное значение на выходе № 3	0...1000 [0,1 %]	Uint16	0012	0018
Аварийное значение на выходе № 4	0...1000 [0,1 %]	Uint16	0013	0019
Аварийное значение на выходе № 5	0...1000 [0,1 %]	Uint16	0014	0020
Аварийное значение на выходе № 6	0...1000 [0,1 %]	Uint16	0015	0021
Максимальный сетевой тайм-аут	0...600 [с]	Uint16	0030	0048

## Порядок выполнения работы

1. Запустить CoDeSys и выбрать целую платформу контроллера, на котором будет исполняться проект, например, PLC100.R-L.
2. При создании нового программного компонента (POU) выбрать пункт «Программа», в поле названия программы – оставить имя PLC\_PRG и выбрать язык «ST».
3. Перейти на вкладку «Ресурсы» и открыть окно «Конфигурация ПЛК».
4. Добавить поддержку портов интерфейса связи с модулями ввода/вывода – подэлемент Modbus (Master) (рис. 3. 26).
5. Выбрать для связи с модулями ввода/вывода порт RS-485-1 (рис. 3.18).

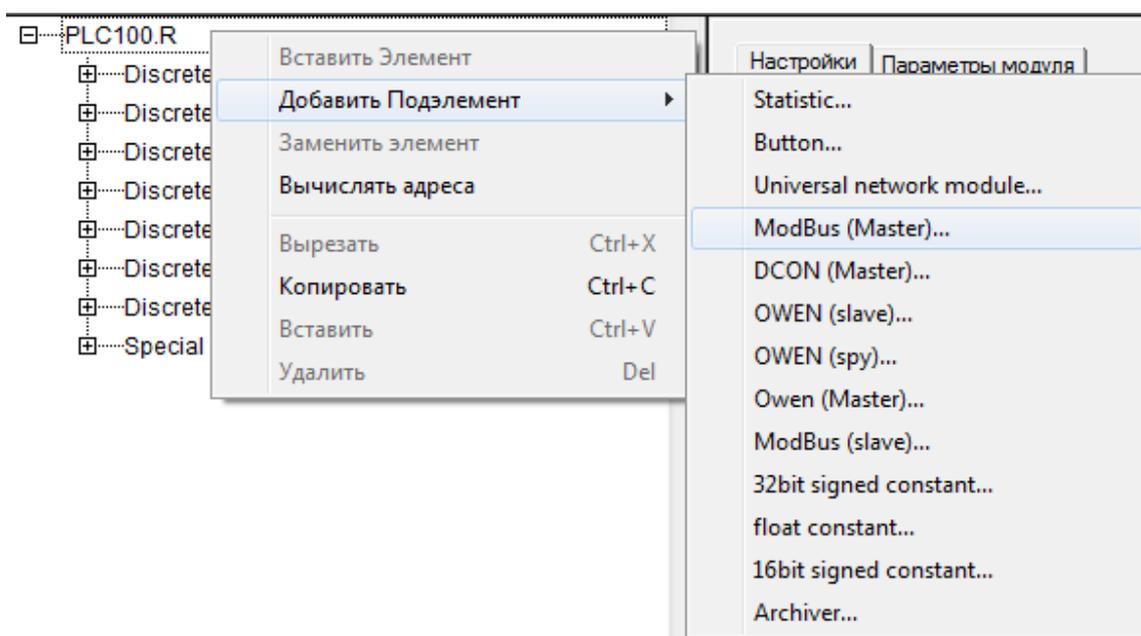


Рис. 3.26. Добавление подэлемента Modbus (Master)

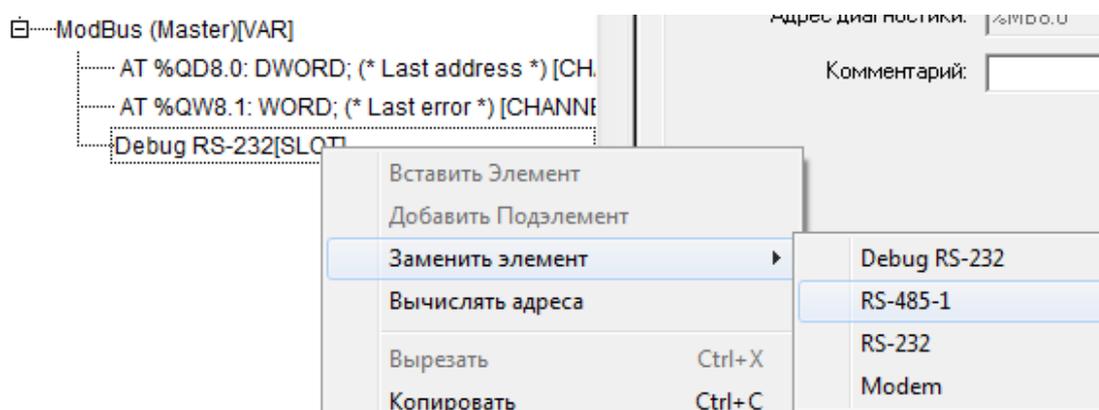


Рис. 3.18. Выбор порта связи с модулями ввода/вывода

6. Настроить параметры связи порта RS-485-1 на вкладке «Параметры модуля». При этом выбрать (рис. 3.27): скорость обмена (Communication Speed) 115200, проверка на четность (Parity) – NO PARITY CHECK; количество бит даны (Data Bits) – 8; количество стоп-бит (Stop Length) – One Stop Bit, тип интерфейса (Interface Type) – RS485; вид протокола (Frame Oriented) – RTU.

7. Добавить новое устройство (подэлемент) – Universal Modbus Device. Это устройство будет отвечать за связь с модулем аналогового ввода MB110-8AC, для этого на вкладке «Параметры модуля» устанавливается параметр – Module Slave Address (адрес Modbus Slave) равным 1 (рис. 3.28).

8. Аналогично добавить в конфигурацию второе устройство – Universal Modbus Device. Оно будет отвечать за связь с модулем аналогового вывода MB110-6У, и оно будет иметь адрес 2 (Module Slave Address – 2).

Индекс	Имя	Значение	По умолч.	Мин.	Макс.
1	Communication speed	11520	11520		
2	Parity	NO PARITY...	NO PARITY C...		
3	Data bits	8 bits	8 bits		
4	Stop length	One stop bit	One stop bit		
5	Interface Type	RS485	RS485		
6	Frame oriented	RTU	ASCII		
7	Framing time ms	0	0	0	32000
8	Visibility	No	No		

Рис. 3.27. Параметры связи Modbus (Master)

Индекс	Имя	Значение	По умолч.	Мин.	Макс.
1	ModuleIP	10:0:0:223	10:0:0:223		
2	Max timeout	150	150	10	
3	TCPport	502	502		
4	NetMode	Serial	Serial		
5	ModuleSlaveAddress	1	1	0	255
6	Work mode	By poll time	By poll time		
7	Polling time ms	100	100	10	10000
8	Visibility	No	No		
9	Amount Repeat	0	0	0	100
10	Byte Sequence	Trace_mo...	Trace_mode		

Рис. 3.28. Установка адреса устройства в сети RS-485

9. Для первого Universal Modbus Device (MB110-8AC) добавить два подэлемента «Register Input Module» (рис. 3.29). Присвоить имена этим каналам соответственно LE\_emk и TE\_emk.

10. Для первого «Register Input Module» (LE\_emk) – на вкладке «Параметры модуля» установить адрес регистра – 256 и выбрать функцию modbus – «Read Holding Registers (0x03)» (рис. 3.30). Для второго «Register Input Module» (TE\_emk) установить аналогично – адрес 257.

11. Для второго Universal Modbus Device (МУ110-6У) добавить подэлемент «Register Output Module» (рис. 3.29). Присвоить имя этому каналу k1\_C (управление клапаном С).

12. Для этого канала «Register Output Module» (Out\_Reg\_T) на вкладке «Параметры модуля» установить адрес регистра – 0 и выбрать команду – «Write Multiple Registers (0x10)» (рис. 3.31).

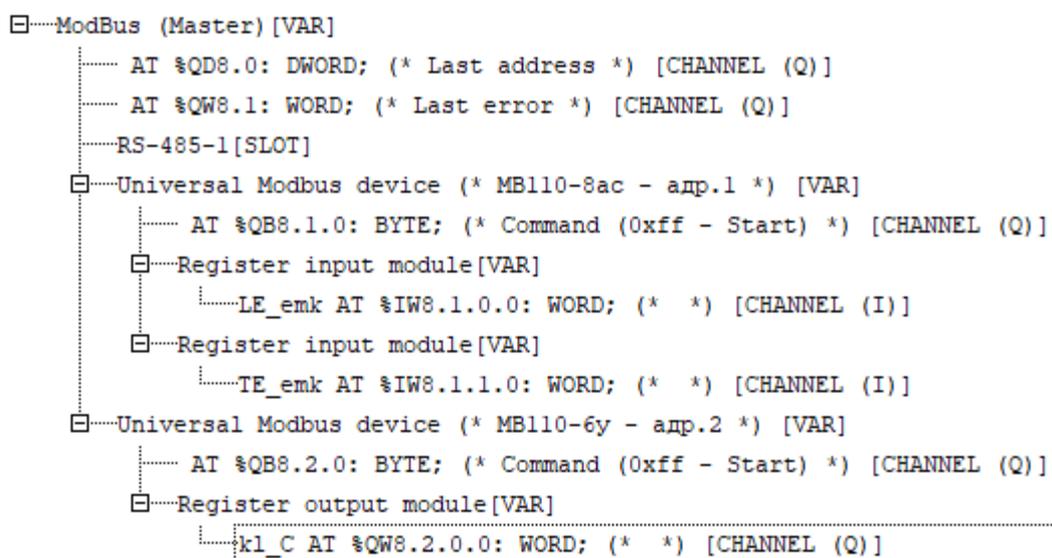


Рис. 3.29. Добавление регистров в конфигурацию

Инде...	Имя	Значение	По ум
1	Register Address	256	0
2	Command	Read holding Registers (0x03)	Read I
8	Visibility	No	No

Инде...	Имя	Значение	По ум
1	Register Address	257	0
2	Command	Read holding Registers (0x03)	Read I
8	Visibility	No	No

Рис. 3.30. Установка адресов modbus-регистров модуля MB110-8AC

Инде...	Имя	Значение	По y
1	Register Address	0	0
2	Command	Write multiple registers(0x10)	Prese
8	Visibility	No	No

Рис. 3.31. Установка адреса modbus-регистра модуля MB110-6У

```

1 PROGRAM PLC_PRG
2 VAR
3   LE_real: REAL; (* Значения датчика в физических единицах измерения - 0...100% *)
4   TE_real: REAL; (* Значения датчика в физических единицах измерения - 0...200гр.С *)
5   LE_max: REAL := 80; (* Настройка позиционного регулятора уровня *)
6   LE_min: REAL := 20; (* Настройка позиционного регулятора уровня *)
7
8   eps: REAL;
9   TE_zad: REAL := 110; (*Задание регулятору температуры*)
10  Xint: REAL;
11  del_t: REAL := 0.01; (*шаг интегрирования - цикл контроллера*)
12  Ti: REAL := 20; (*Постоянная времени*)
13  Kp: REAL := 0.5; (*коэффициент пропорциональности*)
14  PI_reg: REAL; (*выходной сигнал ПИ-регулятора*)
15 END VAR
16
17 (* Масштабирование входных сигналов *)
18 LE_real := LE_emk * 100.0 / 10000; (* 0...100 - шкала датчика, 0...10000 - диапазон кода АЦП *)
19 TE_real := TE_emk * 200.0 / 10000; (* 0...200 - шкала датчика, 0...10000 - диапазон кода АЦП *)
20
21 (* Позиционный регулятор уровня *)
22 IF LE_real >= LE_max
23 THEN k1_B := TRUE;
24 END_IF
25 IF LE_real <= LE_min
26 THEN k1_B := FALSE;
27 END_IF
28
29 (* ПИ-регулятор температуры (упрощенный) *)
30 eps := TE_zad - TE_real;
31 Xint := Xint + eps * del_t / Ti;
32 PI_reg := Kp * (eps + Xint);
33 IF PI_reg < 0 THEN PI_reg := 0;
34 ELSIF PI_reg > 100 THEN PI_reg := 100;
35 END_IF
36
37 (* Масштабирование выходного сигналов: сигнал 0...100 % -> ЦАП 0...1000 *)
38 k1_C := REAL_TO_INT(PI_reg / 100.0 * 1000);

```

Рис. 3.32. Программа PLC\_PRG

13. Первому дискретному выходу контроллера назначить переменную k1\_B (рис. 3.32), которая будет отвечать за управление клапаном В.

14. Отредактировать программу PLC\_PRG в соответствии с рис. 3.32.

```

□ PLC100.R
  □ Discrete input 8 bit[FIX]
  □ Discrete output - relay[FIX]
    □ k1_B AT %QX1.0: BOOL; (* relay *) [CHANNEL (Q)]
  □ Discrete output - relay[FIX]

```

Рис. 3.32. Конфигурация дискретного выхода контроллера ПЛК-100

15. Установить параметры связи с ПЛК, используя команду «Параметры связи» меню «Онлайн». В лабораторной работе связь ПЛК с компьютером осуществляется по сети Ethernet, поэтому необходимо установить IP-адрес контроллера (10.0.6.10, порт 1200) и подтвердить действия нажав кнопку «Ok».

16. Откомпилировать и загрузить проект в контроллер, воспользовавшись командой «Подключения» меню «Онлайн» или кнопкой .

17. Проверить функционирование программы.

### **Контрольные вопросы**

1. В данных какого типа хранятся измеренные значения каналов в модуле аналогового ввода MB110-8AC?

2. Какие функции Modbus могут быть использованы для чтения измеренных значений в модуле аналогового ввода MB110-8AC?

3. Какие функции используются для чтения регистров модуля аналогового вывода МУ110-6У?

4. Какие функции используются для записи регистров модуля аналогового вывода МУ110-6У?

5. Какие сетевые параметры узлов должны быть одинаковыми и какие отличаться при организации сети обмена данными по сети с интерфейсом RS-485 и протоколом Modbus RTU?

## ОРГАНИЗАЦИЯ СВЯЗИ КОНТРОЛЛЕРОВ ОВЕН ПЛК-1XX С ПРИБОРАМИ ТРМ-200 ПО ПРОТОКОЛУ MODBUS RTU

*Цель работы:* получение навыков организации обмена данными между контроллером и цифровыми приборами ТРМ-200 по сети RS-485 с использованием протокола Modbus RTU.

*Задание на лабораторную работу.* Имеется контроллер ПЛК-100, к которому по сети RS-485 подключен прибор ТРМ-200. В ходе выполнения работы необходимо в контроллере получить значение температуры от термометра сопротивления Pt100, подключенному к первому каналу прибора ТРМ-200.

### Краткие сведения о доступе к данным прибора ТРМ-200

В таблице 3.5 приведены некоторые доступные Modbus-регистры прибора ТРМ-200. Полный список доступных Modbus-регистров ТРМ-200 можно найти на официальном сайте производителя этого прибора – компании «ОВЕН» [www.owen.ru](http://www.owen.ru).

### 3.5. Карта Modbus-регистров ТРМ-200

Параметр Имя ОВЕН	Назначение	Адрес Modbus (hex)	Тип данных	Кол-во знаков после запятой	Диапазон значений (dec)
<b>Группа LvoP. Оперативные параметры (только чтение: Modbus-функция 0x03)</b>					
<b>STAT</b>	Регистр статуса	0x0000	binary	–	16 бит <sup>1)</sup>
<b>PV1</b>	Измеренная величина на входе 1	0x0001	Signed Int16	*	диапазон изм. датчика
<b>PV2</b>	Измеренная величина на входе 2	0x0002	Signed Int16	**	диапазон изм. датчика
<b>LUPV1</b>	Величина на верхнем ЦИ	0x0003	Signed Int16	*	
<b>LUPV2</b>	Величина на нижнем ЦИ	0x0004	Signed Int16	**	
<b>Группа LvoP. Оперативные параметры (только чтение: Modbus-функция 0x03)</b>					
<b>PV1</b>	Измеренная величина на входе 1	0x 1009 0x 100A	Float32	–	
<b>PV2</b>	Измеренная величина на входе 2	0x 100B 0x 100C	Float32	–	
<b>LUPV1</b>	Величина на верхнем ЦИ	0x100D 0x100E	Float32	–	
<b>LUPV2</b>	Величина на нижнем ЦИ	0x100F 0x1010	Float32	–	

Параметр Имя ОВЕН	Назначение	Адрес Modbus (hex)	Тип данных	Кол-во знаков после запятой	Диапазон значений (dec)
<b>Группа Lvin. Параметры входов (чтение: Modbus-функция 0x03 / запись: Modbus-функция 0x10)</b>					
<b>in.t1</b>	Тип входного датчика или сигнала для входа 1	0x 0200	Int16	0	1... 26
<b>dPt1</b>	Точность вывода температуры на входе 1	0x 0201	Int16	0	0,1
<b>dP1</b>	Положение десятичной точки для входа 1	0x 0202	Int16	0	0,1,2,3
<b>in.L1</b>	Нижняя граница диапазона измерения для входа 1	0x 0203	Signed Int16	*	-1999...9999
<b>in.H1</b>	Верхняя граница диапазона измерения для входа 1	0x 0204	Signed Int16	*	-1999...9999
<b>SH1</b>	Сдвиг характеристики для входа 1	0x 0205	Signed Int16	*	-500...+500
<b>KU1</b>	Наклон характеристики для входа 1	0x 0206	Int16	3	0.500...2.000
<b>Fb1</b>	Полоса фильтра для входа 1	0x 0207	Int16	*	0...9999
<b>inF1</b>	Постоянная времени цифрового фильтра для входа 1	0x 0208	Int16	0	0...999
<b>Sqr1</b>	Вычислитель квадратного корня для аналогового входа 1	0x 0209	Int16	0	0 – выкл; 1 – вкл
<b>iLU1</b>	Величина на верхнем ЦИ	0x 020A	Int16	0	0 – PV1; 1 – PV2; 2 – dPV
<b>in.t2</b>	Тип входного датчика или сигнала для входа 2	0x 020B	Int16	0	1... 26
<b>dPt2</b>	Точность вывода температуры на входе 2	0x 020C	Int16	0	0,1
<b>dP2</b>	Положение десятичной точки для входа 2	0x 020D	Int16	0	0,1,2,3
<b>in.L2</b>	Нижняя граница диапазона измерения для входа 2	0x 020E	Signed Int16	**	-1999...9999
<b>in.H2</b>	Верхняя граница диапазона измерения для входа 2	0x 020F	Signed Int16	**	-1999...9999
<b>SH2</b>	Сдвиг характеристики для входа 2	0x 0210	Signed Int16	**	-500...+500
<b>KU2</b>	Наклон характеристики для входа 2	0x 0211	Int16	3	0.500...2.000
<b>Fb2</b>	Полоса фильтра для входа 2	0x 0212	Int16	**	0...9999
<b>inF2</b>	Постоянная времени цифрового фильтра для входа 2	0x 0213	Int16	0	0...999
<b>Sqr2</b>	Вычислитель квадратного корня для аналогового входа 2	0x 0214	Int16	0	0 – выкл; 1 – вкл
<b>iLU2</b>	Величина на нижнем ЦИ	0x 0215	Int16	0	0 – PV1; 1 – PV2; 2 – dPV

Получить значение об измеренной величине на входе 1 можно двумя путями:

1. Считать значение этого параметра (PV1 – см. табл. 3.5) из регистров 0x1009 и 0x100A как вещественное значение с плавающей точкой;

2. Считать значение этого параметра (PV1) из регистра 0x0001 как целое значение и скорректировать его с учетом параметра dPt1 – точности вывода температуры на входе 1.

### **Порядок выполнения работы**

1. Подключить термометр сопротивления к входу 1 прибора ТРМ-200 по трехпроводной схеме, соединить клеммы RS-485 интерфейса прибора ТРМ-200 с соответствующими клеммами контроллера ПЛК-100 (для уточнения использовать руководство по эксплуатации на это оборудование).

2. Подать питание на контроллер и прибор ТРМ-200.

3. Произвести конфигурирование (программирование) прибора ТРМ-200: установить тип датчика Pt100 (параметр in.t1), точность вывода температуры (параметр dPt1), протокол обмена – Modbus RTU, скорость обмена – 115.2, адрес прибора в сети – 5. Программирование ТРМ-200 осуществлять в соответствии с руководством по эксплуатации.

4. Запустить CoDeSys. При создании нового программного компонента (POU) выбрать пункт «Программа», в поле названия программы – оставить имя PLC\_PRG и выбрать язык «ST».

5. Перейти на вкладку «Ресурсы» и открыть окно «Конфигурация ПЛК».

6. Добавить поддержку портов интерфейса связи с прибором ТРМ – подэлемент Modbus (Master).

7. Выбрать для связи с прибором ТРМ порт RS-485-1.

8. Настроить параметры связи порта RS-485-1 на вкладке «Параметры модуля». При этом выбрать вид протокола (Frame Oriented) – RTU. Остальные параметры оставить без изменения.

9. Добавить новое устройство (подэлемент) – Universal Modbus Device. Для этого устройства устанавливается основной параметр (вкладка «Параметры модуля») – Module Slave Address. Адрес прибора ТРМ-200 в сети Modbus запрограммировали равным 5.

10. Для «Universal Modbus Device» добавить подэлемент «Real Input Module», который будет считывать информацию о температуре в формате REAL. Параметры типа REAL занимают два регистра. Информация о темпера-

туре хранится в регистрах 0x1009 и 0x100A (в шестнадцатеричной системе) или 4105 и 4106 (в десятичной).

11. Настроить параметры этого подэлемента: на вкладке «Параметры модуля» установить адрес регистра («Register Adress»), указав адрес первого регистра, т.е. 4105 и выбрать команду («Command») 0x03.

12. В конфигурации ПЛК присвоить имя этому каналу (Real Input Module), например, T\_real\_1.

13. Для «Universal Modbus Device» добавить подэлемент «Register Input Module», который будет считывать информацию о температуре в формате INT.

14. Настроить параметры этого подэлемента: на вкладке «Параметры модуля» установить адрес регистра 1, и выбрать команду («Command») 0x03.

15. В конфигурации ПЛК присвоить имя этому каналу, например, T\_int.

16. Для «Universal Modbus Device» добавить еще один подэлемент «Register Input Module», который будет считывать информацию о точности вывода температуре на входе 1.

17. Настроить параметры этого подэлемента: на вкладке «Параметры модуля» установить адрес регистра 513 (0x201), и выбрать команду («Command») 0x03.

18. В конфигурации ПЛК присвоить имя этому каналу, например, Tochnost.

19. Отредактировать программу PLC\_PRG в соответствии с рис. 3.33.

20. Установить параметры связи с ПЛК, используя команду «Параметры связи» меню «Онлайн». В лабораторной работе связь ПЛК с компьютером осуществляется по сети Ethernet, поэтому необходимо установить IP-адрес контроллера (по умолчанию 10.0.6.10, порт 1200) и подтвердить действия нажав кнопку «Ок».

21. Откомпилировать и загрузить проект в контроллер, воспользовавшись командой «Подключения» меню «Онлайн» или кнопкой .

22. Правильность работы программы проверяется тем, что значения переменных T\_real\_1 и T\_real\_2 должны быть равны значению, отображаемому на 1 канале прибора ТРМ-200.

```

0001 PROGRAM PLC_PRG
0002 VAR
0003     T_real_2: REAL;
0004 END_VAR
0005
0001 CASE Tochnost OF
0002 0: T_real_2:=INT_TO_REAL(T_int);
0003 1: T_real_2:=INT_TO_REAL(T_int)*0.1;
0004 END_CASE;
0005

```

**Рис. 3.33. Программа PLC\_PRG**

### **Контрольные вопросы**

1. К какому классу микропроцессорных приборов относится прибор ТРМ-200?
2. Какие протоколы обмена поддерживает прибор ТРМ-200?
3. Расскажите про назначение пунктов меню при конфигурировании прибора ТРМ-200.
4. Как осуществляется конфигурирование сетевых параметров прибора ТРМ-200?
5. В данных какого типа хранятся измеренные значения каналов в приборе ТРМ-200?
6. Какие функции Modbus могут быть использованы для чтения регистров прибора ТРМ-200?
7. Какие функции Modbus могут быть использованы для записи регистров прибора ТРМ-200?

### **Контрольное задание**

Разработать программу для контроллера ПЛК-100, позволяющее осуществить перепрограммирование прибора ТРМ-200 через сеть RS-485 с использованием протокола Modbus RTU в реальном времени. Программа должна обеспечивать:

- 1) возможность изменения точности отображения измеренной температуры;
- 2) возможность отображения измеренной температуры на верхнем цифровом индикаторе (ЦИ), на нижнем индикаторе или одновременно на двух индикаторах.

**ОРГАНИЗАЦИЯ МЕЖКОНТРОЛЛЕРНОГО ОБМЕНА ДАННЫМИ  
В СЕТИ КОНТРОЛЛЕРОВ ОВЕН ПЛК-1XX  
ПО ИНТЕРФЕЙСУ RS-485 И ПРОТОКОЛУ MODBUS RTU**

*Цель работы:* получение навыков организации обмена данными между контроллерами ОВЕН ПЛК с использованием протокола Modbus RTU.

*Задание на лабораторную работу.* Имеются два контроллера ПЛК-100, которые соединены между собой по интерфейсу RS-485, причем контроллер № 1 является Master-узлом, а контроллер № 2 – Slave-узлом. Также контроллер № 1 обслуживает модули ввода-вывода, также подключенные по RS-485: модуль дискретного ввода MB110-16ДН (адрес в сети RS-485 – 1) и модуль дискретного вывода МУ110-16Р (адрес 4). Оба контроллера подключены в сеть Ethernet. Контроллер № 1 имеет сетевой адрес 10.0.6.10, контроллер № 2 – 10.0.6.9. В ходе выполнения работы необходимо:

1) в контроллере № 1 получить информацию от модуля MB110-16ДН (адрес 3) о состоянии двух дискретных входов  $x_1$  и  $x_2$ ;

2) переслать из контроллера № 1 в контроллер № 2 информации о состоянии двух дискретных входов  $x_1$  и  $x_2$ ;

3) в контроллере № 2 осуществить вычисление следующих выражений:

$$y_1 = x_1 \cup x_2, \quad y_2 = x_1 \cap x_2;$$

4) переслать результаты вычисления выражений из контроллера № 2 в контроллер № 1;

5) вывести через модуль МУ110-16Р (адрес 4) полученные результаты выражений  $y_1$  и  $y_2$ .

**Краткие сведения о настройке каналов модуля Modbus (Slave)**

Возможные типы каналов (переменных) приведены в табл. 3.6.

При работе по протоколу Modbus обращение устройства-мастера к используемым переменным производится по адресу переменной в памяти модуля «Modbus (Slave)».

### 3.6. Типы каналов (переменных) модуля «Modbus (Slave)»

Типы передаваемых переменных	Канал	Размер в памяти	Команды считывания	Команды записи
1...8 бит	8 bit	8 бит	01, 02	05
Word, INT, 16 бит	2 byte	2 байта	03, 04	6, 16
Dword, DINT, 32 бита	4 byte	4 байта	03, 04	16
Real	Float	32 бита	03,04	16
Файл	File	*)	20	-

Адресация переменных автоматически формируется системой CoDeSys при программировании контроллера, но при этом не отображается в интерфейсе программы. Поэтому для обращения к переменной ее адрес следует вычислить, учитывая при этом особенности распределения адресов ячеек памяти («выравнивание» переменных).

Особенности распределения адресов ячеек памяти («выравнивание» переменных) заключаются в следующем. Все используемые переменные хранятся в одном адресном пространстве, и считывать данные из этого пространства можно как побитно, так и целыми регистрами (регистр численно равен двум байтам). При этом биты нумеруются от 0 до  $n$  (побитно), а регистры – также от 0 до  $n$ , но по регистрам (т.е. по два байта). Пример адресации приведен в табл. 3.7.

Автоматическая адресация переменных производится последовательно, начиная с нулевого адреса (как для битовых переменных, так и для переменных, передаваемых регистрами).

Таким образом, если в модуле используются переменные одного типа, то при запросе устройством – Мастером регистра с адресом «0», модуль считывает первые два байта, для регистра с адресом «1» – вторые два байта и так далее.

### 3.7. Пример адресации битов и регистров в памяти модуля

Адрес бита	Адрес байта	Адрес регистра
0... 7	0x0000	0x00
8... 15	0x0001	
16... 23	0x0002	0x01
24... 31	0x0003	
32... 39	0x0004	0x02
40... 47	0x0005	

Если переменные имеют длину более двух байт, то при запросе регистра с адресом «0», модуль считывает первые два байта первой переменной, для регистра с адресом «1» – вторые два байта первой переменной и так далее.

Но если в модуле используются переменные разных типов (например, одновременно восьмибитный, двухбайтный и четырехбайтный), то при распределении адресов система CoDeSys производит «выравнивание» адресов переменных – упорядочение адресов переменных в памяти модуля. Такое упорядочение заключается в организации памяти таким образом, что переменные размером 8 бит, 2 байта и 4 байта располагаются только по определенным адресам: четырехбайтным переменным присваиваются адреса, кратные 4; двухбайтным – кратные 2; однобайтным (или восьмибитным) – кратные 1. То есть, независимо от порядка задания переменных, выравнивание назначает переменным адреса, кратные их длине.

### **Порядок выполнения работы**

#### *Часть 1 – проект для контроллера № 2 – Slave*

Передачу дискретных сигналов (сигналов типа BOOL) будем осуществлять через битовую маску, упаковкой и распаковкой битовых сигналов в число типа INT.

1. Запустить CoDeSys. При создании нового программного компонента (POU) выбрать пункт «Программа», в поле названия программы – оставить имя PLC\_PRG и выбрать язык «ST».

2. Сохранить проект под названием «Contr\_2».

3. Перейти на вкладку «Ресурсы» и открыть окно «Конфигурация ПЛК».

4. Добавить подэлемент Modbus (Slave).

5. Перейти на вкладку «Параметры модуля» у подэлемента Modbus (Slave) и установить адрес устройства в сети Modbus, например 6.

6. Выбрать порт ПЛК, который будет использоваться для связи с СПК. Для этого нажать правой кнопкой мыши на элемент Modbus (FIX) и добавить подэлемент RS-485-1.

7. Настроить параметры связи порта RS-485-1 на вкладке «Параметры модуля». При этом выбрать вид протокола (Frame Oriented) – RTU. Остальные параметры оставить без изменения (рис. 3.34).

8. Нажать правую клавишу мыши на элемент Modbus (Slave) и добавить два канала «2 byte». Первый канал обозначить Reg\_X, второй – Reg\_Y (рис. 3.35). Регистр Reg\_X будет содержать упакованные значения дискретных входов  $x_1$  и  $x_2$ , а Reg\_Y – упакованные значения дискретных выходов  $y_1$  и  $y_2$ .

9. Отредактировать программу PLC\_PRG в соответствии с рис. 3.36.

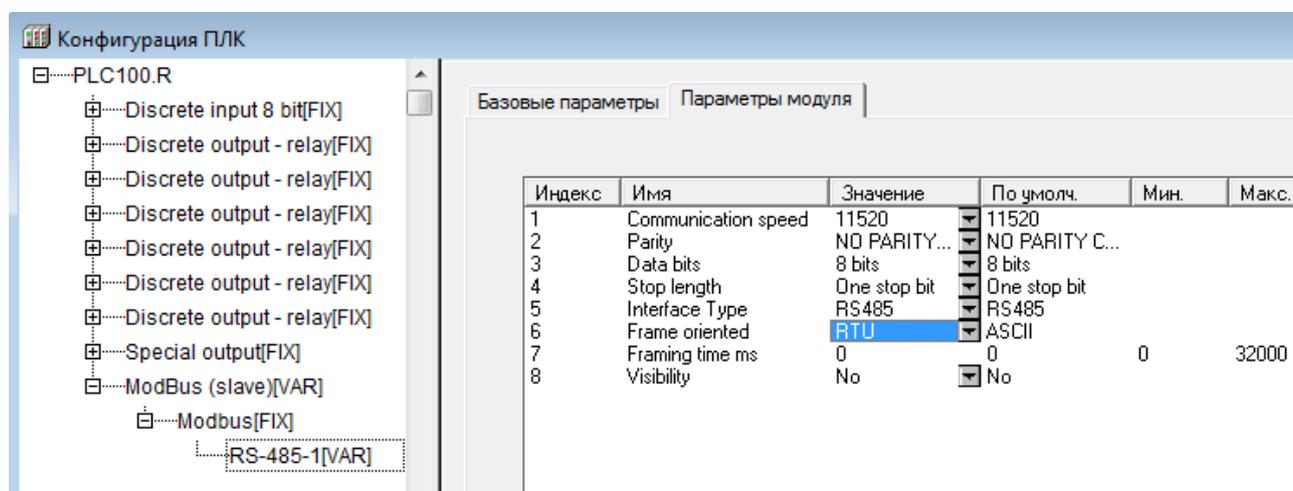


Рис. 3.34. Конфигурация ПЛК. Настройки подэлемента RS-485-1

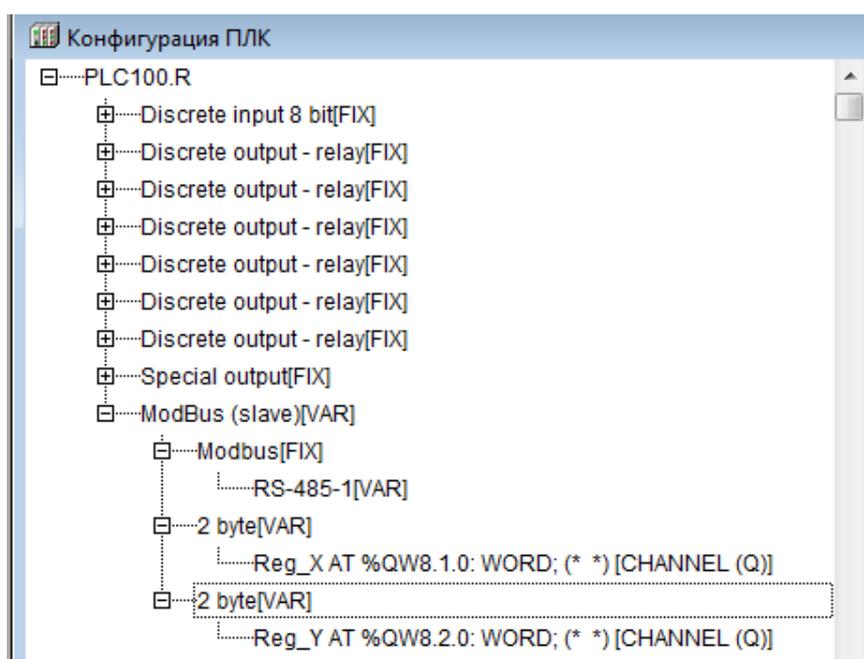


Рис. 3.35. Внешний вид Modbus (Slave) с добавленными подэлементами

```

0001 PROGRAM PLC_PRG
0002 VAR
0003     x1: BOOL;
0004     x2: BOOL;
0005     y1: BOOL;
0006     y2: BOOL;
0007 END_VAR
0001 x1:=Reg_X.0;
0002 x2:=Reg_X.1;
0003
0004 y1:=x1 OR x2;
0005 y2:=x1 AND x2;
0006
0007 Reg_Y.0:=y1;
0008 Reg_Y.1:=y2;

```

Рис. 3.36. Программа PLC\_PRG для контроллера № 2

10. Установить параметры связи с ПЛК, используя команду «Параметры связи» меню «Онлайн». Для контроллера № 2 установить IP-адрес 10.0.6.9, порт 1200 и подтвердить действия нажав кнопку «Ok».

11. Откомпилировать и загрузить проект в контроллер, воспользовавшись командой «Подключения» меню «Онлайн» или кнопкой .

*Часть 2 – проект для контроллера №1 – Master*

1. Запустить новую копию CoDeSys. При создании нового программного компонента (POU) выбрать пункт «Программа», в поле названия программы – оставить имя PLC\_PRG и выбрать язык «ST».

2. Сохранить проект под названием «Contr\_1».

3. Перейти на вкладку «Ресурсы» и открыть окно «Конфигурация ПЛК».

4. Добавить подэлемент ModBus (Master).

5. Выбрать для связи с модулями ввода/вывода и контроллером № 2 порт RS-485-1.

6. Настроить параметры связи порта RS-485-1 на вкладке «Параметры модуля». При этом выбрать вид протокола (Frame oriented) – RTU. Остальные параметры оставить без изменения.

7. Добавить новое устройство (подэлемент) – Universal Modbus Device. Для этого устройства устанавливается основной параметр (вкладка «Параметры модуля») – ModuleSlaveAddress (значение устанавливается согласно адресу заданному модулю ввода/вывода при его конфигурировании). Для модуля

МВ110-16ДН устанавливается адрес 1. Остальные параметры модулей оставить заданными по умолчанию. На вкладке «Базовые параметры» рекомендуется описать комментарии по модулю (например, МВ110-16ДН – адр. 1), чтобы в будущем не ошибиться.

8. Для первого «Universal Modbus Device» (МВ110-16ДН) – модуля дискретного ввода – добавить подэлемент «Register Input Module». Работу с дискретными входами этого модуля будем осуществлять через битовую маску входов. В соответствии с руководством по эксплуатации этого модуля этот регистр имеет адрес 51 (см. табл. 3.1, лаб. работа № 3).

9. Настроить параметры этого подэлемента: на вкладке «Параметры модуля» установить адрес регистра («Register Adress») 51 и выбрать команду («Command») 0x03 или 0x04.

10. В конфигурации ПЛК присвоить имя этому каналу (Register Input Module), например, Reg51.

11. Добавить новый модуль МУ110-16Р (адрес 4) – аналогично п. 7.

12. Для второго «Universal Modbus Device» (МУ110-16Р) добавить подэлемент «Register output module». Работу с дискретными выходами этого модуля будем осуществлять через битовую маску выходов. В соответствии с руководством по эксплуатации модуля этот регистр имеет адрес 50.

13. В конфигурации ПЛК присвоить имя этому каналу (Register Output Module), например, Reg50.

14. Настроить параметры этого подэлемента: на вкладке «Параметры модуля» установить адрес регистра («Register Adress») 50 и выбрать команду («Command») 0x10.

15. Добавить новое устройство (подэлемент) – Universal Modbus Device. Это устройство будет отвечать за связь с контроллером № 2. Для удобства работы описать комментарий для этого устройства – например, «Контроллер 2». На вкладке «Параметры модуля» установить значение Module Slave Address равным 6.

16. Для третьего «Universal Modbus Device» (Контроллер 2) добавить подэлемент «Register output module» и на вкладке «Параметры модуля» установить адрес регистра («Register Adress») – 0, и выбрать команду («Command»)

0x10. В конфигурации ПЛК присвоить имя этому каналу (Register Output Module) Reg\_X.

17. Также для третьего «Universal Modbus Device» (Контроллер 2) добавить подэлемент «Register Input Module» и на вкладке «Параметры модуля» установить адрес регистра («Register Adress») – 1, и выбрать команду («Command») 0x03 или 0x04. В конфигурации ПЛК присвоить имя этому каналу (Register Output Module) Reg\_Y.

18. Отредактировать программу PLC\_PRG:

Reg\_X:= Reg51;

Reg50:=Reg\_Y.

19. Установить параметры связи с ПЛК, используя команду «Параметры связи» меню «Онлайн». Для контроллера № 1 установить IP-адрес 10.0.6.10, порт 1200 и подтвердить действия нажав кнопку «Ok».

20. Откомпилировать и загрузить проект в контроллер, воспользовавшись командой «Подключения» меню «Онлайн» или кнопкой .

### Контрольное задание

Имеются два контроллера ПЛК-100, которые посредством межконтроллерного обмена по Modbus RTU совместно реализуют функции калькулятора. При этом контроллер № 1 (IP 10.0.6.10) реализует только функции ввода данных и отображения результатов, за вычисления отвечает контроллер № 2 (IP 10.0.6.9).

В контроллере № 1 в глобальной области памяти хранятся операнды выражений A и B, а также результат выполнения операций Y. Вид операции (сложение, вычитание, умножение, деление) определяется включением тумблера на первом, втором, третьем или четвертом входе контроллера.

В контроллере № 2 осуществляется вычисление результата операций над операндами A и B:

$$Y = \begin{cases} A + B, & \text{если } x_1 = TRUE; \\ A - B, & \text{если } x_2 = TRUE; \\ A \cdot B, & \text{если } x_3 = TRUE; \\ A / B, & \text{если } x_4 = TRUE. \end{cases}$$

**ОРГАНИЗАЦИЯ СВЯЗИ КОНТРОЛЛЕРА ОВЕН ПЛК2XX  
С МОДУЛЯМИ ВВОДА-ВЫВОДА ОВЕН МХ210  
ПО ПРОТОКОЛУ MODBUS TCP**

*Цель работы:* получение навыков организации обмена данными с модулями ввода-вывода Мх210 с использованием протокола Modbus TCP.

*Задание на лабораторную работу.* Имеется контроллер ПЛК210 (IP адрес в сети 192.168.0.10), к которому по Ethernet подключены модули ввода-вывода: модуль аналогового ввода МВ210-101 (IP адрес – 192.168.0.1) и модуль дискретного ввода-вывода МК210-301 (IP адрес – 192.168.0.3). Параметры Modbus TCP: ModbusSlave ID=1, Port=502.

В ходе выполнения работы необходимо разработать проект, реализующий «линейный вольтметр», при этом:

1) сконфигурировать модуль аналогового ввода МВ210-101 (в соответствии с лабораторной работой №2);

2) получить информацию от модуля аналогового ввода МВ210-101 о поданном на его вход напряжении. Значение напряжения 0...1 В соответствует значению измерительного канала (0...100.0);

3) осуществить преобразование значения выходного сигнала измерительного канала в величину измеренного напряжения;

4) сформировать команды управления светодиодами  $y_1...y_8$  линейного вольтметра по следующему правилу: при напряжении менее 0,1 В все светодиоды выключены; при достижении напряжения 0,1В – включается первый светодиод; при достижении напряжения 0,2В – подключается также второй светодиод, и так далее; при напряжении 0,8В и выше – горят все восемь светодиодов.

5) вывести через модуль МК210-301 команды управления  $y_1...y_8$  светодиодами, которые физически подключены к дискретным выходам этого модуля.

## Краткие сведения о ПЛК210 и CoDeSys 3.5

Контроллеры ОВЕН ПЛК2хх – новая линейка современных моноблочных контроллеров с расширенными коммуникационными возможностями и дополнительными функциями надежности. Программирование контроллеров серии ПЛК2хх осуществляется в среде CoDeSys 3.5. В ПЛК2хх реализована поддержка промышленных протоколов Modbus RTU/ASCII/TCP, OPC UA (Server), MQTT (client), SNMP (Manager/Agent).

Программный комплекс CoDeSys 3.5 – это современный продукт, который широко используется в качестве среды программирования контроллеров. Среда CoDeSys предоставляет пользователю возможность программирования на всех языках программирования в соответствии со стандартом IEC (МЭК) 61131-3. Помимо программирования контроллеров среда CoDeSys используется для разработки человеко-машинного интерфейса.

Для программирования контроллера ОВЕН ПЛК210 используется версия CODESYS V3.5 SP17 Patch 3, для ранних прошивок ПЛК (выпущенных до 06.20022) – версия CODESYS V3.5 SP14 Patch 3. Прошивка – это внутреннее системное программное обеспечение контроллера, которое управляет его работой на аппаратном уровне.

Для начала работы со средой программирования CoDeSys v3.5 требуется:

1. Скачать и установить необходимую версию CoDeSys. Скачать можно с сайта компании ОВЕН ([https://owen.ru/product/codesys\\_v3](https://owen.ru/product/codesys_v3)).

2. Скачать и установить таргет-файлы контроллеров. Таргет-файл (target) – файл целевой платформы, в котором в специальном формате описываются ресурсы контроллера и организуется его связь с CoDeSys. Каждая модификация контроллера, который поддерживается системой CoDeSys, имеет свой таргет-файл. Скачать таргет-файлы контроллеров компании ОВЕН можно непосредственно с сайта компании [https://owen.ru/product/codesys\\_v3](https://owen.ru/product/codesys_v3).

Карта некоторых адресов modbus-регистров для использующихся в лабораторной работе модулей ввода-вывода имеет вид:

Параметр	Адреса регистров MB210-101	Адреса регистров MK210-301
Чтение значений аналоговых входов (1 – 8)	4000-4001 (вход 1), 4003-4004 (вход 2), 4006-4007 (вход 3), 4009-4010 (вход 4), 4012-4013 (вход 5), 4015-4016 (вход 6), 4018-4019 (вход 7), 4021-4022 (вход 8)	
Запись маски дискретных выходов		470

Модули поддерживают следующие функции Modbus: 03 (0x03) – Read Holding Registers; 04 (0x04) – Read Input Registers; 06 (0x06) – Write Single Register; 16 (0x10) – Write Multiple Registers.

### Порядок выполнения работы

1. Конфигурация модулей ввода-вывода. Произвести конфигурирование сетевых параметров модулей MB210-101 и MK210-301, а также параметров аналоговых входов MB210-101 (см. лабораторную работу № 2).

2. Создание нового проекта. После запуска среды появляется стартовое окно, которое позволяет создать новый проект ( New Project...), открыть существующий на компьютере проект ( Open Project...), открыть проект, сохраненный в контроллере ( Open Project from PLC...) или выбрать один из ранее активных проектов. Все эти пункты доступны также через меню «File» («Файл»). Для создания нового проекта нажимаем на пиктограмме  New Project... или выбираем аналогичный пункт меню «File» («Файл»). Появляется окно создания нового проекта (рис. 3.37). В этом окне можно выбрать категорию проекта: «Libraries» («Библиотеки») или «Projects» («Проекты»). Выбираем «Projects» («Проекты»). Далее в соответствующих полях указываем имя проекта («Name:») и его расположение («Location»). В заключение выбираем шаблон проекта «Template» и подтверждаем создание нажатием на кнопку «ОК». Для примера будем создавать проект для контроллера ПЛК210 модификации 01, для этого используем шаблон PLC210-01. Этот шаблон автоматически подключает к проекту таргет-файл контроллера ПЛК210-01.

Возможен альтернативный способ выбора таргет-файла контроллера при создании проекта. Для этого выбираем шаблон «Standard project» («Стандартный проект») и в открывшемся окне (рис. 3.38) выбираем необходимый таргет-файл контроллера (PLC210), а также язык программирования для главной программы проекта PLC\_PRG (в примере выбран язык программирования ST).

3. Настройка сетевого интерфейса компьютера. Установить IP-адрес компьютера с протоколом IPv4, например, 192.168.0.200, а также маску сети IPv4 – 255.255.0.0. При установке IP-адресов следует помнить, что в локальных сетях не должно возникать конфликта IP-адресов, т.е. разные устройства не должны обладать совпадающими адресами.

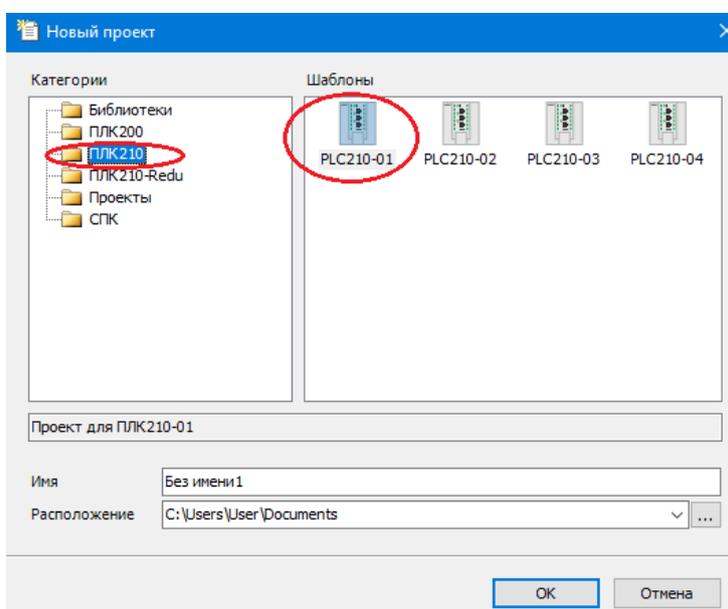


Рис. 3.37. Создание нового проекта с использованием шаблона

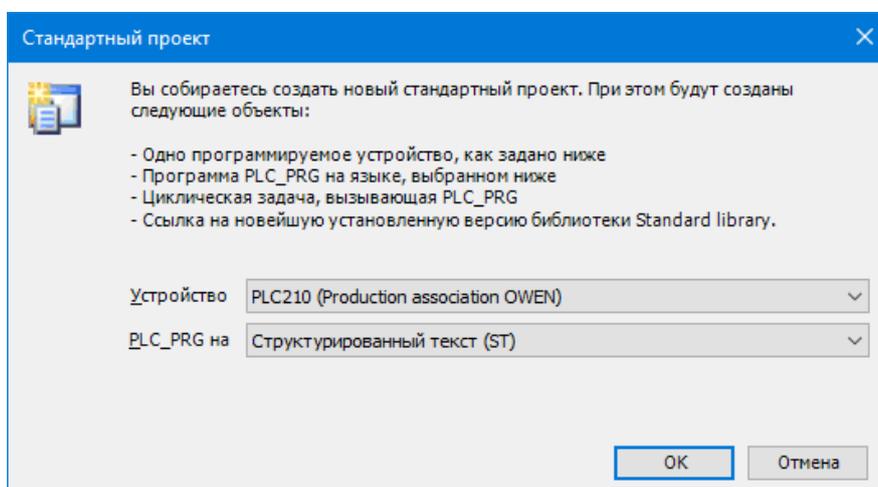


Рис. 3.38. Альтернативный способ создания нового проекта

4. Проверка на соответствие таргет-файла и модели контроллера. Компонент Device который определяется соответствующим таргет-файлом, должен соответствовать модели контроллера. В случае необходимости тип устройства можно изменить, выбрав в дереве проекта компонент Device (рис. 3.39) и, нажав на него правой клавишей мыши, открыть окно «Update Device» («Обновить устройство»). Затем следует выбрать устройство, соответствующее модели контроллера. Следует помнить про соответствие версии прошивки контроллера, среды программирования CODESYS и таргет-файла. При отсутствии в списке требуемой версии устройства можно осуществить поиск в расширенном списке, установив галочку в поле  Display all versions (for experts only) . После выбора устройства следует нажать кнопку  и закрыть окно.

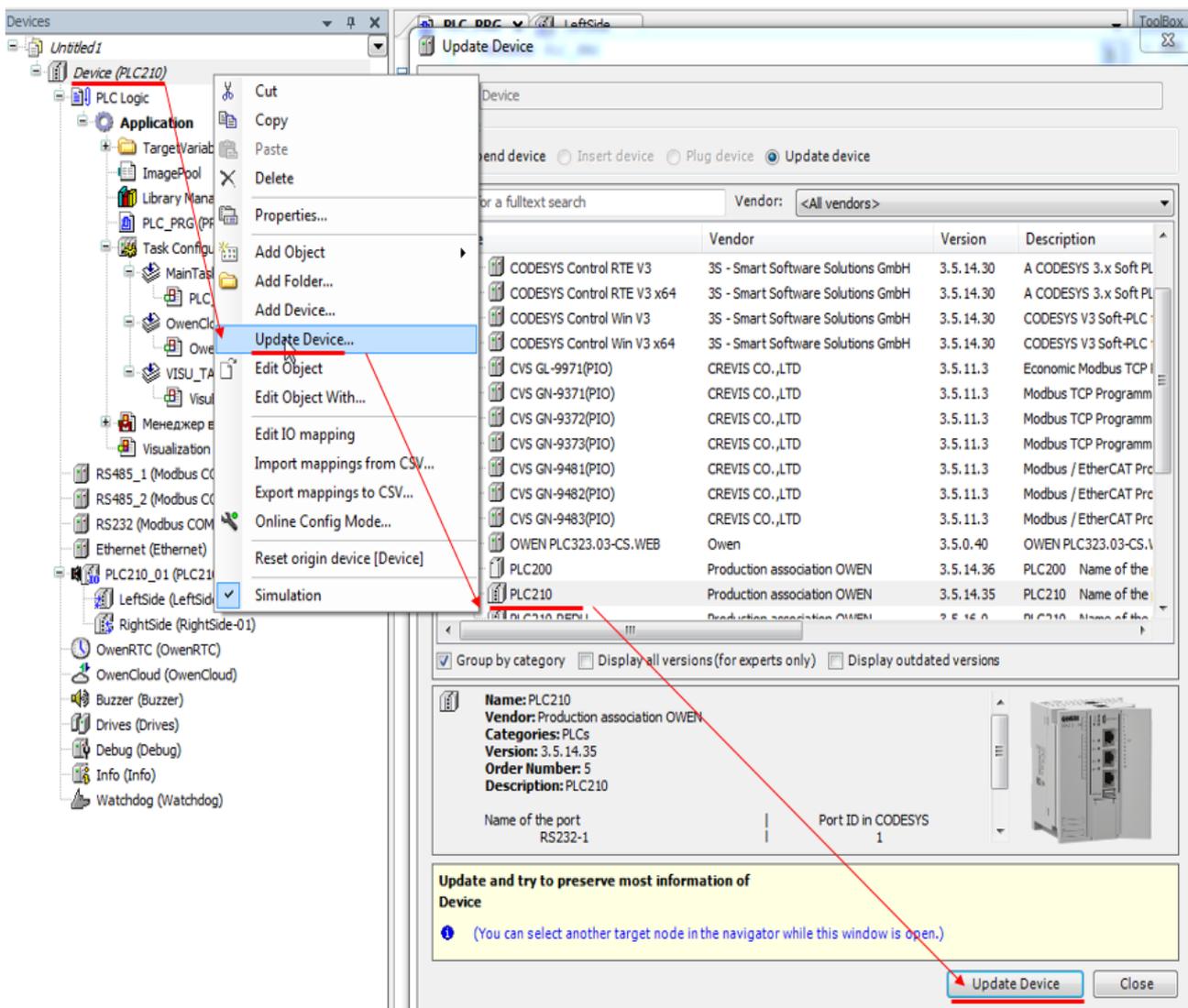


Рис. 3.39. Окно «Update Device»

4. Настройка шлюза («Gateway») и установка связи с контроллером. Для этого необходимо дважды кликнуть на поле «Device» в дереве объектом и в открывшемся окне (рис. 3.40) выбрать пункт «Communication Setting». При отсутствии установленных шлюзов следует добавить новый шлюз («Add New Gateway»). Настройки рекомендуется оставить по умолчанию (имя – Gateway-1, IP-адрес – localhost). Затем закрыть окно настроек шлюза и нажать кнопку «Scan Network». В появившемся списке следует выбрать нужный контроллер и установить связь, нажав кнопку ОК. Связь с контроллером можно непосредственно по его IP-адресу. Для этого в поле контроллера необходимо ввести его адрес (например, 192.168.0.10) и нажав кнопку клавишу Enter. В случае успешной установки связи индикаторы шлюза и контроллера будут гореть зеленым цветом.

5. Добавление адаптера Ethernet в дерево проекта. Выделить «Device (PLC210)» и вызвать контекстное меню, нажав правую кнопку мыши. Далее выбрать «Add Device». В появившемся окне «Add Device» (рис.3.41) выбрать требуемую версию сетевого адаптера Ethernet. Для возможности осуществления выбора версий устройства должна быть отмечено () поле «Display all Versions».

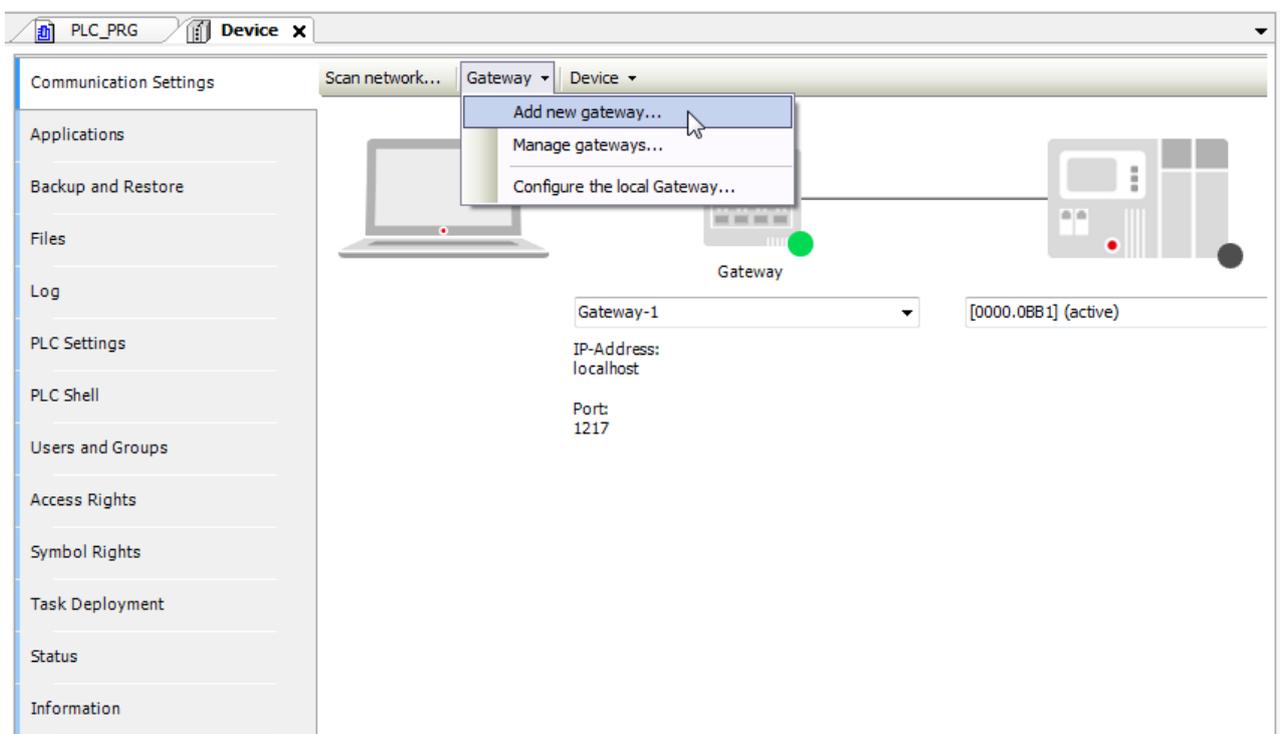


Рис. 3.40. Окно для установки связи с контроллером

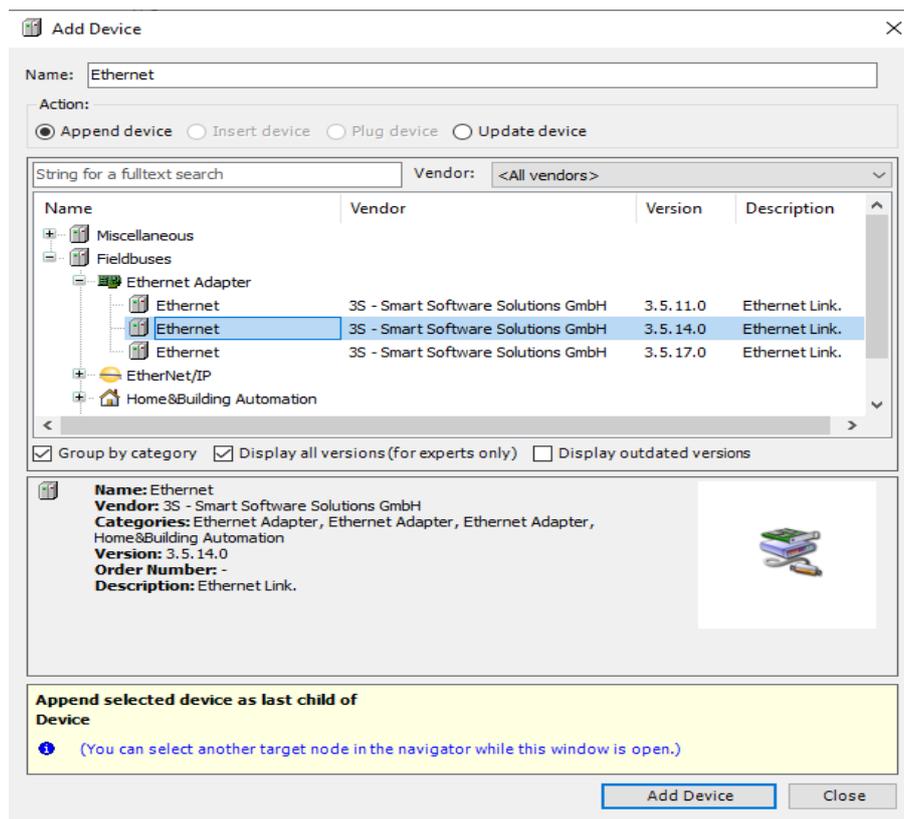
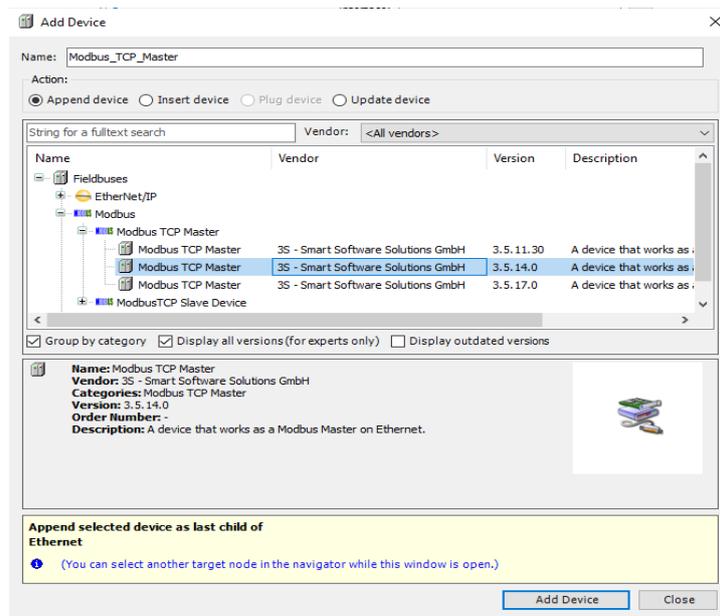


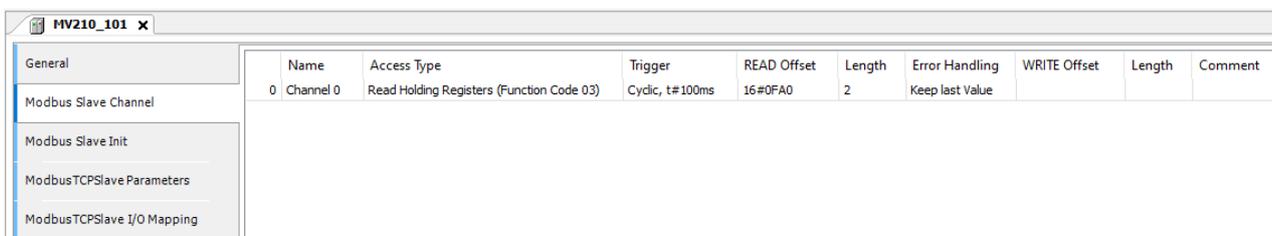
Рис. 3.41. Добавление адаптера Ethernet

6. Добавление в дерево проекта мастера Modbus TCP. Для этого в дереве проекта выделить устройство Ethernet, правой кнопкой мыши вызвать контекстное меню и нажать «Add Device». В появившемся окне осуществить выбор требуемой версии устройства «Modbus TCP Master». На вкладке «Общее» («General») отметить () поле «Auto-reconnect» («Автоподключение») и установить сетевые настройки сетевого адаптера контроллера (IP адрес 192.168.0.10) и маску подсети. При уже установленной ранее в п. 4 связи сетевые параметры можно установить автоматически, выбрав соответствующий интерфейс в поле «Interface» при нажатии на кнопку .

7. Добавление в дерево проекта Modbus TCP Slave устройств. Добавление устройств осуществляется аналогично через контекстное меню и пункт «Add Device». В соответствии с заданием на лабораторную работу необходимо добавить два Modbus TCP Slave устройства: первое соответствует модулю аналогового ввода MB210-101 (имеет IP адрес 192.168.0.1), второе – модулю дискретного ввода-вывода МК210-301 (IP адрес 192.168.0.3). Задание IP-адресов осуществляется вкладке «Общее» («General»). Для улучшения читабельности проекта переименовать Modbus TCP Slave устройства: первое назвать MV210\_101, второе – МК210\_301. Также требуется



**Рис. 3.42. Добавление Modbus TCP мастера**



**Рис. 3.43. Добавление канала Modbus TCP Slave для MV210-101**

8. Добавление каналов Modbus Slave. Для устройства MV210\_101 на вкладке «Modbus Slave Channel» нажать кнопку «Add Channel» и отредактировать в соответствии с рис.3.43.

В результате с модуля одним групповым запросом будут считаны 2 регистра – начиная с регистра 0x0FA0 (HEX) = 4000 (DEC). В этих двух регистрах хранятся значения первого аналогового входа модуля в представлении с плавающей точкой (так как каждое 4-байтовое значение занимает 2 регистра).

На вкладке «ModbusTCP Slave I/O Mapping» привязать к регистрам канала переменные wAI1\_1 и wAI1\_2 (рис. 3.44). Также необходимо в нижней части этого окна Для параметра «Всегда обновлять переменные установите значение Вкл. 2 (Всегда в задаче цикла шины)» («Always Update Variable» = «Enable 2(always in bus cycle task)»).

Аналогично для модуля МК210-301 добавить канал для записи в регистр маски дискретных выходов (рис. 3.45) и привязать переменную wMaskaDO к регистру этого канала (рис. 3.46). Регистр с маской дискретных выходов расположен по адресу 470 (0x1D6).

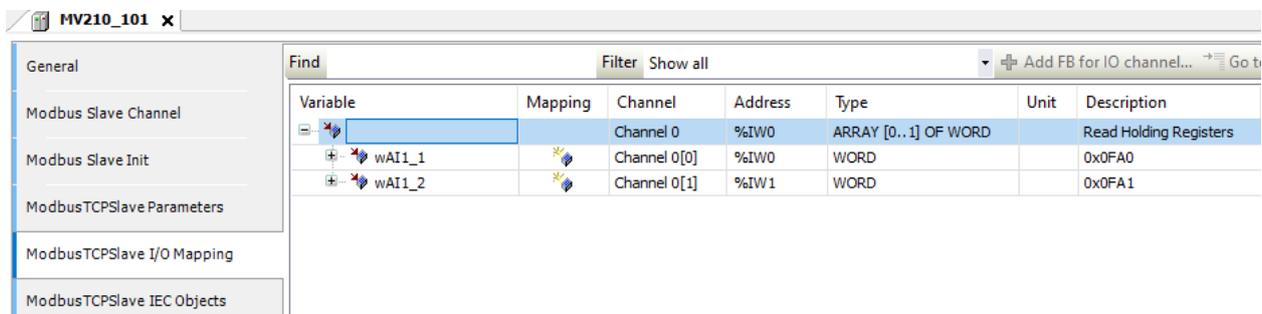


Рис. 3.44. Привязка переменных к регистрам канала ModusTCP Slave модуля MB210-101

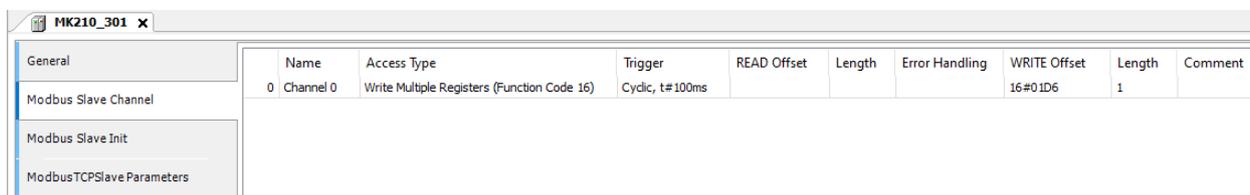


Рис. 3.45. Добавление канала Modbus TCP Slave для МК210-301

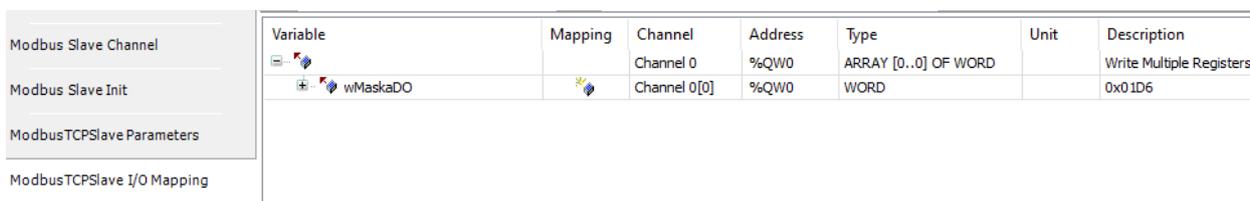


Рис. 3.46. Привязка переменной к регистру канала ModusTCP Slave модуля МК210-301

9. Обработка вещественных значений для передачи через Modbus. К каналам компонента Modbus TCP Slave можно привязать только переменные типа WORD. Поэтому в коде для каждого аналогового входа потребуется выполнить преобразование двух переменных типа WORD в одну переменную типа REAL. Для этого нажмите правой клавишей мыши на узел Application и выберите команду «Add Object» → «DUT» → «Union». Создайте объединение с названием WORD2\_AS\_REAL и следующим содержимым:

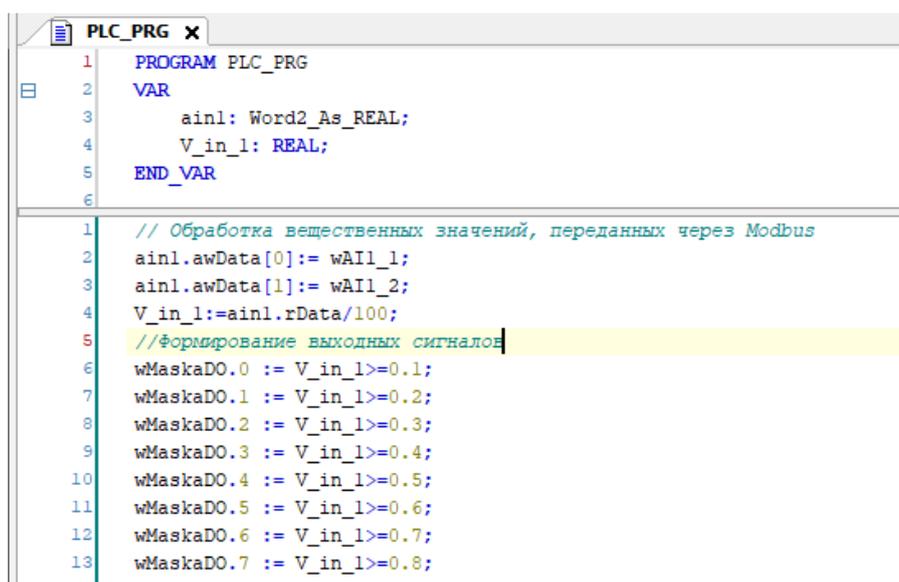
```

1  TYPE WORD2_AS_REAL :
2  UNION
3      awData: ARRAY [0..1] OF WORD;
4      rData: REAL;
5  END_UNION
6  END_TYPE

```

10. Отредактировать программу PLC\_PRG в соответствии с рис. 3.47.

11. Откомпилировать и загрузить проект в контроллер. Загрузка откомпилированного проекта в контроллер осуществляется через пункт «Login» («Логин») меню «Online» («Онлайн») или при нажатии горячих клавиш «Alt + F8» или нажатием на пиктограмме . После загрузки проекта в контроллер, его следует запустить. Это осуществляется нажатием функциональной клавиши F5 или пиктограммы , воспользовавшись командой «Подключения» меню «Онлайн» или кнопкой.



```
1  PROGRAM PLC_PRG
2  VAR
3      ain1: Word2_As_REAL;
4      V_in_1: REAL;
5  END_VAR
6
7  // Обработка вещественных значений, переданных через Modbus
8  ain1.awData[0] := wAI1_1;
9  ain1.awData[1] := wAI1_2;
10 V_in_1 := ain1.rData/100;
11 //Формирование выходных сигналов
12 wMaskaDO.0 := V_in_1>=0.1;
13 wMaskaDO.1 := V_in_1>=0.2;
14 wMaskaDO.2 := V_in_1>=0.3;
15 wMaskaDO.3 := V_in_1>=0.4;
16 wMaskaDO.4 := V_in_1>=0.5;
17 wMaskaDO.5 := V_in_1>=0.6;
18 wMaskaDO.6 := V_in_1>=0.7;
19 wMaskaDO.7 := V_in_1>=0.8;
```

Рис. 3.47. Код программы PLC\_PRG

## ЗАКЛЮЧЕНИЕ

Необходимость изучения дисциплины «Промышленные вычислительные сети» связана с современными взглядами на построение автоматизированных систем управления, в которых посредством промышленных сетей осуществляется коммуникационное взаимодействие между устройствами человеко-машинного интерфейса, программируемыми логическими контроллерами, интеллектуальными датчиками и исполнительными механизмами. Данное пособие ориентировано на подготовку специалистов, область профессиональных приложений которых связана с автоматизацией и управлением техническими системами в промышленности.

## СПИСОК ЛИТЕРАТУРЫ

---

1. Кругляк, К. Промышленные сети: цели и средства / К. Кругляк // Современные технологии автоматизации. – 2002. – № 4. – С. 6 – 17.
2. Елизаров, И. А. Технические средства автоматизации. Программно-технические комплексы и контроллеры : учебное пособие / И. А. Елизаров, Ю. Ф. Мартемьянов, А. Г. Схиртладзе. – Старый Оскол, 2021. – 236 с.
3. Барбасова, Т. А. Промышленные сети и системы связи : учебное пособие / Т. А. Барбасова, Е. А. Канашев. – Челябинск : Издательский центр ЮУрГУ, 2020. – 144 с.
4. Сахнюк, А. А. Промышленные сети / А. А. Сахнюк, А. М. Литвин // ПиКАД. – 2004. – № 2. – С. 6 – 8.
5. Денисенко, В. В. Компьютерное управление технологическим процессом, экспериментом, оборудованием / В. В. Денисенко. – М. : Горячая линия – Телеком», 2009 г. – 608 с.
6. CODESYS V3.5 Настройка обмена по протоколу Modbus: Руководство пользователя [Электронный ресурс]. – URL : [https://ftp.owen.ru/CoDeSys3/11\\_Documentation/03\\_3.5.11.5/CDSv3.5\\_Modbus\\_v3.2.pdf](https://ftp.owen.ru/CoDeSys3/11_Documentation/03_3.5.11.5/CDSv3.5_Modbus_v3.2.pdf)
7. Modbus Application Protocol Specification V1.1b3 [Электронный ресурс]. – URL : [https://www.modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b3.pdf](https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf)
8. Обзор протокола Modbus TCP [Электронный ресурс]. – URL : [https://www.hcfa-russia.ru/company/news/obzor\\_protokola\\_modbus\\_tcp/](https://www.hcfa-russia.ru/company/news/obzor_protokola_modbus_tcp/)

# ОГЛАВЛЕНИЕ

---

ВВЕДЕНИЕ .....	3
1. ОБЩИЕ СВЕДЕНИЯ О ПРОМЫШЛЕННЫХ СЕТЯХ .....	4
1.1. Место промышленных сетей в современной автоматизированной системе управления .....	4
1.2. Базовые термины и определения .....	6
1.3. Общие требования к промышленным сетям .....	7
1.4. Модель взаимодействия открытых систем OSI .....	8
1.5. Топологии промышленных сетей .....	10
1.6. Физические интерфейсы .....	13
2. СЕТИ MODBUS .....	22
2.1. Общие сведения .....	22
2.2. Разновидности протокола Modbus .....	33
2.2.1. Modbus RTU .....	34
2.2.2. Modbus ASCII .....	36
2.2.3. Modbus TCP .....	38
2.3. Особенности использования протоколов Modbus .....	40
<i>Лабораторная работа № 1. Конфигурирование модулей ввода-вывода Овен Mx110 для работы в сети по протоколу Modbus RTU .....</i>	<i>44</i>
<i>Лабораторная работа № 2. Конфигурирование модулей ввода-вывода Овен Mx210 для работы в сети по протоколу Modbus TCP .....</i>	<i>51</i>
<i>Лабораторная работа № 3. Организация связи контроллеров Овен ПЛК1xx с модулями дискретного ввода-вывода Mx110 по протоколу Modbus RTU .....</i>	<i>57</i>
<i>Лабораторная работа № 4. Организация связи контроллеров Овен ПЛК1xx с модулями аналогового ввода-вывода по протоколу Modbus RTU .....</i>	<i>65</i>
<i>Лабораторная работа № 5. Организация связи контроллеров Овен ПЛК-1xx с приборами ТРМ-200 по протоколу Modbus RTU .....</i>	<i>73</i>
<i>Лабораторная работа № 6. Организация межконтроллерного обмена данными в сети контроллеров ОВЕН ПЛК-1xx по интерфейсу RS-485 и протоколу Modbus RTU .....</i>	<i>78</i>
<i>Лабораторная работа № 7. Организация связи контроллера ОВЕН ПЛК2xx с модулями ввода-вывода ОВЕН Mx210 по протоколу Modbus TCP .....</i>	<i>85</i>
ЗАКЛЮЧЕНИЕ .....	94
СПИСОК ЛИТЕРАТУРЫ .....	95

Учебное электронное издание

ЕЛИЗАРОВ Игорь Александрович  
НАЗАРОВ Виктор Николаевич  
ТРЕТЬЯКОВ Александр Александрович  
ПОГОНИН Василий Александрович

# ПРОМЫШЛЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

*Учебное пособие*

Редактирование Е. С. Мордасовой  
Графический и мультимедийный дизайнер Т. Ю. Зотова  
Обложка, упаковка, тиражирование Е. С. Мордасовой

ISBN 978-5-8265-2794-8



Подписано к использованию 19.08.2024.

Тираж 50 шт. Заказ № 86

Издательский центр ФГБОУ ВО «ТГТУ»  
392000, г. Тамбов, ул. Советская, д. 106, к. 14

Телефон: (4752) 63-81-08

E-mail: izdatelstvo@tstu.ru