

**Система ввода-вывода Fastwel I/O
СРМ713. Контроллер узла сети MODBUS TCP**

Руководство по конфигурированию и программированию сетевых средств

ИМЕС.00300-02 33 03-4

Версия 2.0

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
1. ВВЕДЕНИЕ	7
2. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ СРМ713	8
2.1. ОБЩИЕ СВЕДЕНИЯ	8
2.2. ХАРАКТЕРИСТИКИ ИНТЕРФЕЙСА ВНЕШНЕЙ СЕТИ СРМ713.....	8
2.2.1. <i>Характеристики сервера MODBUS TCP</i>	8
2.2.2. <i>Характеристики клиента MODBUS TCP</i>	8
3. ПРИНЦИП РАБОТЫ СЕРВИСА ВНЕШНЕЙ СЕТИ	10
3.1. ОБЩИЕ СВЕДЕНИЯ	10
3.2. ПРИНЦИП РАБОТЫ И СПОСОБЫ КОНФИГУРИРОВАНИЯ СЕРВЕРА MODBUS	10
3.2.1. <i>Общие сведения</i>	10
3.2.2. <i>Настройка коммуникационных параметров сервера MODBUS TCP</i>	11
3.2.3. <i>Представление входных и выходных данных сервера MODBUS</i>	11
3.2.4. <i>Обмен данными с клиентами MODBUS</i>	12
3.2.5. <i>Формат запроса/ответа на чтение расширенной идентификационной информации</i>	16
3.2.6. <i>Обслуживание сетевых запросов</i>	16
3.2.7. <i>Диагностика</i>	17
3.2.7.1. <i>Диагностические каналы</i>	17
3.2.7.2. <i>Инициализация сервиса внешней сети</i>	17
3.2.7.3. <i>Индикация</i>	17
3.3. ПРИНЦИП РАБОТЫ И СПОСОБЫ КОНФИГУРИРОВАНИЯ МАСТЕРА MODBUS	17
3.3.1. <i>Общие сведения</i>	17
3.3.2. <i>Настройка параметров мастера MODBUS</i>	18
3.3.3. <i>Обмен данными с подчиненными устройствами</i>	19
3.3.4. <i>Диагностика</i>	21
3.4. ДОСТУП К ПОЛЯМ ДАННЫХ КОММУНИКАЦИОННЫХ ОБЪЕКТОВ ИЗ ПРИЛОЖЕНИЯ	21
3.5. БИБЛИОТЕКА FASTWELSYSLIBSOCKETS.LIB	23
3.5.1. <i>Общие сведения</i>	23
3.5.2. <i>Описание функций</i>	23
3.5.2.1. <i>FwSysSockCreate</i>	23
3.5.2.2. <i>FwSysSockClose</i>	23
3.5.2.3. <i>FwSysSockListen</i>	24
3.5.2.4. <i>FwSysSockAccept</i>	24
3.5.2.5. <i>FwSysSockBind</i>	24
3.5.2.6. <i>FwSysSockConnect</i>	25
3.5.2.7. <i>FwSysSockGetOption</i>	25
3.5.2.8. <i>FwSysSockSetOption</i>	26
3.5.2.9. <i>FwSysSockIoctl</i>	27
3.5.2.10. <i>FwSysSockSelect</i>	27
3.5.2.11. <i>FwSysSockRecv</i>	28
3.5.2.12. <i>FwSysSockRecvFrom</i>	28
3.5.2.13. <i>FwSysSockSend</i>	29
3.5.2.14. <i>FwSysSockSendTo</i>	29
3.5.2.15. <i>FwSysSockShutdown</i>	30
3.5.2.16. <i>FwSysSockHtons</i>	30
3.5.2.17. <i>FwSysSockHtonl</i>	30
3.5.2.18. <i>FwSysSockNtohs</i>	31
3.5.2.19. <i>FwSysSockNtohl</i>	31
3.5.2.20. <i>FwSysSockGetLastError</i>	31

3.5.2.21.	FwSysSockGetHostName	31
3.5.2.22.	FwSysSockInetAddr.....	32
3.5.2.23.	FwSysSockInetNtoa	32
3.5.2.24.	FwSysSockSetIPAddress	32
3.5.3.	<i>Описание типов данных</i>	33
3.5.3.1.	INADDR	33
3.5.3.2.	SOCKADDRESS	33
3.5.3.3.	SOCKET_FD_SET	33
3.5.4.	<i>Коды ошибок</i>	33
4.	НАСТРОЙКА КОММУНИКАЦИОННЫХ СРЕДСТВ ВЕРХНЕГО УРОВНЯ.....	35
4.1.	НАСТРОЙКА КОММУНИКАЦИОННОГО ДРАЙВЕРА СЕРВЕРА CODESYS GATEWAY SERVER	35
4.1.1.	<i>Общие сведения</i>	35
4.1.2.	<i>Установка коммуникационного драйвера CoDeSys Gateway Server</i>	35
4.1.3.	<i>Создание логического информационного канала между средой разработки и контроллером</i>	36
4.2.	СОЗДАНИЕ ЛОГИЧЕСКОГО ИНФОРМАЦИОННОГО КАНАЛА МЕЖДУ СРЕДОЙ РАЗРАБОТКИ И КОНТРОЛЛЕРОМ СРМ713 ПО ПОСЛЕДОВАТЕЛЬНОМУ КАНАЛУ СВЯЗИ	37
4.3.	ДОПОЛНИТЕЛЬНЫЕ ЗАМЕЧАНИЯ	38
	ПРИЛОЖЕНИЕ А. ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	39

Авторское право

Фирма-производитель «ООО ФАСТВЕЛ», именуемая в дальнейшем Фаствел®, является официальным владельцем всех авторских прав на комплекс, информация о котором представлена в данном руководстве по эксплуатации. Фаствел® также является владельцем всех авторских прав на примененные оригинальные технические решения и встроенное системное программное обеспечение.

Данное руководство по эксплуатации и содержащаяся в нем информация являются исключительной собственностью Фаствел®.

Право воспроизведения информации

Данное руководство по эксплуатации и содержащаяся в нем информация могут быть воспроизведены каким-либо известным способом без предварительного уведомления и последующего извещения Фаствел®. Ссылка на первоисточник воспроизводимой информации является обязательной.

Право внесения информации

Фаствел® оставляет за собой исключительное право внесения изменений и дополнений в данное руководство по эксплуатации без предварительного уведомления. Все изменения и дополнения включаются в последующие редакции документа и представлены на Web-сайтах Фаствел® и компании «ПРОСОФТ», именуемой в дальнейшем ПРОСОФТ®.

Право обновления спецификации изделия

Фаствел® оставляет за собой исключительное право внесения изменений и дополнений в конструкцию, электрическую схему и программное обеспечение, улучшающие технические и потребительские характеристики изделия, без предварительного уведомления. Все изменения и дополнения включаются в последующие редакции документа и представлены на Web-сайтах Фаствел® и ПРОСОФТ®.

Фирменные и торговые марки

Все товарные знаки и торговые марки, а также зарегистрированные товарные знаки и торговые марки, представленные в руководстве по эксплуатации, являются исключительной собственностью своих законных владельцев.

Контактная информация

Адрес: 119313, Москва, а/я 242;

Телефон: (095) 234–0639;

Факс: (095) 232–1654;

E-mail: info@fastwel.ru;

Web: www.fastwel.ru.

Поставка и техническая поддержка

ПРОСОФТ® осуществляет поставку и техническую поддержку продукции Фаствел®.

Адрес: 119313, Москва, а/я 81;

Телефон: (095) 234–0636;

Факс: (095) 234–0640;

E-mail: info@prosoft.ru;

Web: www.prosoft.ru.

Фаствел® приветствует любые предложения и замечания по улучшению данного руководства по эксплуатации, а также объективную информацию о функционировании представленного изделия и встроенного системного программного обеспечения.

Примечание – Необходимо ознакомиться со сведениями общего характера во Введении до начала использования изделия, представленного в данном руководстве по эксплуатации.

Фаствел® не несет никакой ответственности за возможные повреждения и ущерб, обусловленные несоблюдением основных рекомендаций и требований данного руководства по эксплуатации.

Настоящее руководство содержит указания по настройке и программированию интерфейса внешней сети контроллера СРМ713 комплекса Fastwel I/O System в среде CoDeSys.

Фаствел® является официальным OEM-партнером фирмы 3S Smart Software Solutions и производителем адаптации среды CoDeSys для работы совместно с устройствами серии Fastwel I/O System.

ПРОСОФТ® является официальным дистрибьютором Фаствел®.

Каталог продукции Фаствел® размещен на Web-странице:

<http://www.fastwel.ru/products/catalog/index.htm>.

Каталог продукции Фаствел® размещен также на файл-сервере ПРОСОФТ® по адресу:

<ftp://ftp.prosoft.ru/pub/Hardware/Fastwel/>.

1. ВВЕДЕНИЕ

Настоящее руководство содержит указания по настройке и программированию интерфейса внешней сети контроллера CPM713 комплекса Fastwel I/O System в среде CoDeSys фирмы 3S Smart Software Solutions.

При работе с настоящим документом следует также пользоваться основным руководством программиста на контроллеры серии Fastwel I/O:

Система ввода-вывода Fastwel I/O. Контроллеры CPM711/CPM713/CPM713. Руководство программиста.

Предполагается, что пользователь Fastwel I/O знаком с принципами построения сетей TCP/IP на базе интерфейса Ethernet и функционирования протокола MODBUS TCP.

2. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ СРМ713

2.1. Общие сведения

Общие для всех контроллеров серии Fastwel I/O характеристики приведены в п. 2.4 документа:

Система ввода-вывода Fastwel I/O. Контроллеры СРМ711/СРМ713/СРМ713. Руководство программиста.

2.2. Характеристики интерфейса внешней сети СРМ713

2.2.1. Характеристики сервера MODBUS TCP

Сервис сервера MODBUS TCP реализует функциональность подчиненного узла протокола MODBUS TCP. Перечень основных характеристик сервиса приведен. Перечень основных характеристик сервиса приведен в табл. 1.

Таблица 1

Тип узла протокола MODBUS	Подчиненный	
Скорость обмена, Мбит/с	10, 100 (автоопределение)	
Адресация IP	статическая, по конфигурации контроллера из проекта CoDeSys	
Адрес по умолчанию	10.0.0.1 (при поставке и при запуске со включенным переключателем "1")	
Количество клиентских соединений по MODBUS TCP	не более 32-х	
Операции протокола MODBUS	<i>Тип</i>	<i>Описание</i>
	01	Выдача за один запрос от 1 до 2000 смежных битовых полей, доступных для записи
	02	Выдача за один запрос от 1 до 2000 смежных битовых полей, доступных для чтения
	03	Выдача за один запрос от 1 до 125 смежных регистров, доступных для записи
	04	Выдача за один запрос от 1 до 125 смежных регистров, доступных для чтения
	05	Прием значения одного битового поля, доступного для записи
	06	Прием значения одного регистра, доступного для записи
	15	Прием за один запрос значений до 2000 смежных битовых полей, доступных для записи
	16	Прием за один запрос значений до 125 смежных регистров, доступных для записи
	22	Изменение содержимого заданного регистра, доступного для записи с использованием комбинации масок И, ИЛИ с текущим содержимым регистра для индивидуального сброса или установки бит регистра
23	Прием и выдача за один запрос значений до 125 смежных регистров, доступных для записи	
43	Сервис инкапсулированного транспорта. Функция 128: обмен со средой разработки CoDeSys Функция 14: выдача расширенной идентификационной информации	
Количество регистров	До 4096	
Количество битовых полей	До 65536	

2.2.2. Характеристики клиента MODBUS TCP

Сервис клиента MODBUS TCP реализует функциональность мастера протокола MODBUS TCP. Перечень основных характеристик сервиса приведен в табл. 2.

Таблица 2

Тип узла протокола MODBUS	Мастер	
Скорость обмена, Мбит/с	10, 100 (автоопределение)	
Адресация IP	статическая, по конфигурации контроллера из проекта CoDeSys	
Адрес по умолчанию	10.0.0.1 (при поставке и при запуске со включенным переключателем "1")	
Количество серверных соединений по MODBUS TCP	Ограничено ресурсами конфигурационных данных проекта CoDeSys	
Операции протокола MODBUS	<i>Тип</i>	<i>Описание</i>
	01	Чтение от 1 до 2000 смежных битовых полей, доступных для записи
	02	Чтение от 1 до 2000 смежных битовых полей, доступных для чтения
	03	Чтение от 1 до 125 смежных регистров, доступных для записи
	04	Чтение от 1 до 125 смежных регистров, доступных для чтения
	05	Запись одного битового поля, доступного для записи
	06	Запись значения одного регистра, доступного для записи
	15	Запись значений до 2000 смежных битовых полей, доступных для записи
	16	Запись значений до 125 смежных регистров, доступных для записи
23	Чтение и чтение за один запрос значений до 125 смежных регистров, доступных для записи	

3. ПРИНЦИП РАБОТЫ СЕРВИСА ВНЕШНЕЙ СЕТИ

3.1. Общие сведения

Сервис внешней сети контроллера узла CPM713 реализует функциональность подчиненного узла и мастера сети MODBUS TCP. Наиболее актуальную спецификацию протокола MODBUS TCP можно загрузить с Web-узла <http://www.modbus.org>.

Внешний вид контроллера с расположением соединителей показан на рис. 1.

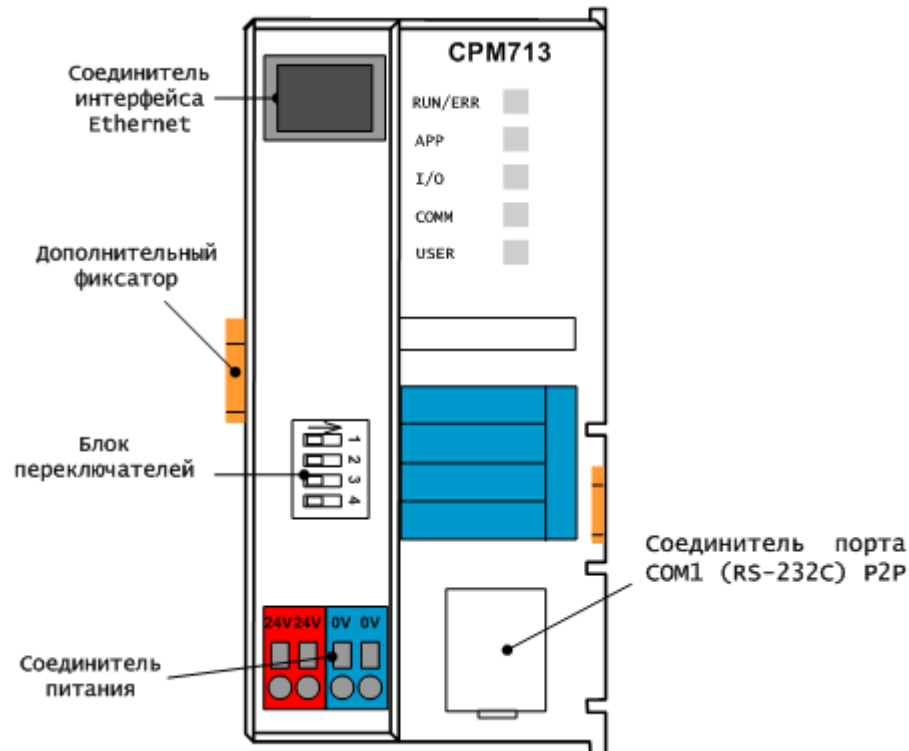


Рис. 1. Внешний вид контроллера CPM713

3.2. Принцип работы и способы конфигурирования сервера MODBUS

3.2.1. Общие сведения

Порт интерфейса Ethernet контроллера представлен в дереве конфигурации ресурса **PLC Configuration** проекта CoDeSys элементом *CPM713 MODBUS TCP Programmable Controller–Ethernet Multiprotocol Port*, и может одновременно использоваться для работы в качестве подчиненного узла (сервера) и мастера (клиента) сети MODBUS TCP. Для настройки параметров и создания коммуникационных объектов сервера MODBUS TCP используется элемент конфигурации *CPM713 MODBUS TCP Programmable Controller–Ethernet Multiprotocol Port–Modbus TCP Slave*, показанный на рис. 2.

Кроме того, имеется возможность миграции проекта CoDeSys, ранее созданного для контроллера CPM703, на платформу CPM713 (см. п. 3.7 документа *Контроллеры CPM711/CPM713/CPM713. Руководство программиста*). В этом случае принципы работы и способы конфигурирования сервера MODBUS совпадают с используемыми для контроллера CPM703. Более подробная информация приведена в документе *Система ввода-вывода Fastwel I/O. CPM703. Контроллер узла сети MODBUS TCP. Руководство по конфигурированию и программированию сетевых средств*.

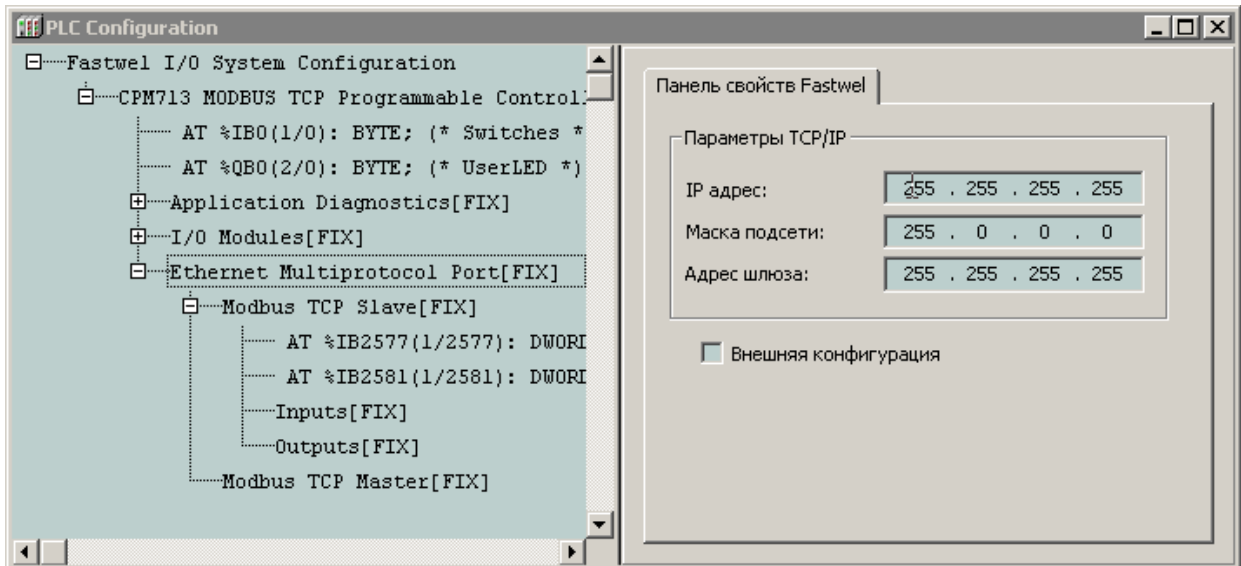


Рис. 2. Элемент конфигурации для настройки параметров TCP/IP

3.2.2. Настройка коммуникационных параметров сервера MODBUS TCP

Для настройки коммуникационных параметров протокола TCP/IP используется **Панель свойств Fastwel**, показанная на рис. 2. Кроме того, для сервера MODBUS TCP в стандартной панели свойств CoDeSys может быть задан идентификатор узла (параметр UnitID), как показано на рис. 3.

Описание параметров протокола TCP/IP приведено в табл. 3.

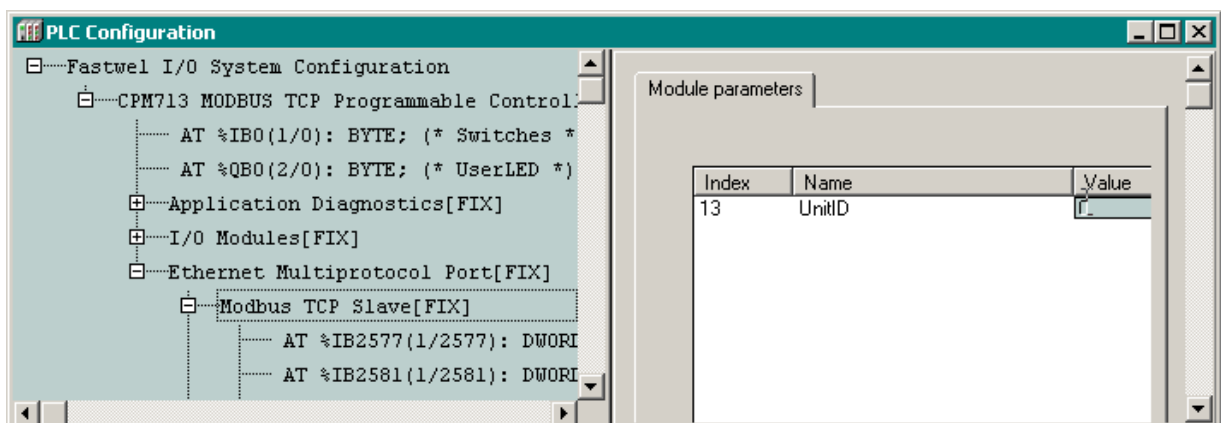


Рис. 3. Параметр UnitID сервера протокола MODBUS

Таблица 3

Параметр	Назначение
IP адрес	IP-адрес контроллера в сети TCP/IP. В принудительном безопасном режиме и при отсутствии загруженного приложения 10.0.0.1
Маска подсети	Маска подсети
Адрес шлюза	Адрес шлюза
Внешняя конфигурация	Признак необходимости использования текущих параметров TCP/IP. ВНИМАНИЕ! Не используйте данный параметр в контроллере CPM713!

Для того, чтобы только что установленные параметры протокола были переданы в дерево конфигурации проекта CoDeSys, необходимо выполнить одно из следующих действий:

- сохранить проект (**File–Save**); или
- выполнить построение проекта (**Project–Build**); или
- выбрать в дереве проекта в ресурсе **PLC Configuration** узел, отличный от *Modbus TCP Slave* (по простому говоря, нужно щелкнуть мышкой на каком-нибудь другом элементе дерева).

3.2.3. Представление входных и выходных данных сервера MODBUS

Области памяти контроллера, отображаемые на множества регистров и битовых полей сервера MODBUS, описываются в секциях *Inputs* и *Outputs* в древовидном списке конфигурации сервера.

Секция *Inputs* содержит список объектов доступа к данным, поступающим по сети от удаленных клиентов. Секция *Outputs* содержит список объектов доступа к данным для передачи в сеть.

Перечень типов и назначение используемых объектов данных приведены в табл. 4.

Доступ к каждому объекту данных по сети или к группам объектов может осуществляться при помощи сетевых запросов чтения или/и записи, имеющихся в протоколе MODBUS. Информация о типе коммуникационного объекта MODBUS (Input Register, Holding Register, Discrete Input или Coil), его начальном адресе и количестве объектов в запросе MODBUS содержится в панели свойств **Modbus Access Properties**, показанной на рис. 4.

Для добавления объекта данных в список сервера щелкните правой кнопкой мыши на позиции соответствующей секции, выберите в контекстном меню команду **Append Subelement** и затем требуемый тип объекта.

Для вставки объекта в список щелкните правой кнопкой мыши на соответствующей позиции в списке объектов данных сервера, выберите в контекстном меню команду **Insert Element** и затем требуемый тип объекта.

Для удаления объекта данных щелкните правой кнопкой мыши на его позиции и выберите в контекстном меню команду **Delete** или выберите его в списке и нажмите клавишу **Del**.

Таблица 4

Тип секции	Тип объекта	Область данных объекта
<i>Outputs</i>	WORD Output	Двухбайтовый выходной канал типа WORD
	DWORD Output	Четырехбайтовый выходной канал типа DWORD
	REAL Output	Четырехбайтовый выходной канал типа REAL
	LREAL Output	Восьмибайтовый выходной канал типа LREAL
	2-Bytes Output	2 однобайтовых выходных канала типа BYTE
<i>Inputs</i>	WORD Input	Двухбайтовый входной канал типа WORD
	DWORD Input	Четырехбайтовый входной канал типа DWORD
	REAL Input	Четырехбайтовый выходной канал типа REAL
	LREAL Input	Восьмибайтовый входной канал типа LREAL
	2-Bytes Input	2 однобайтовых входных канала типа BYTE

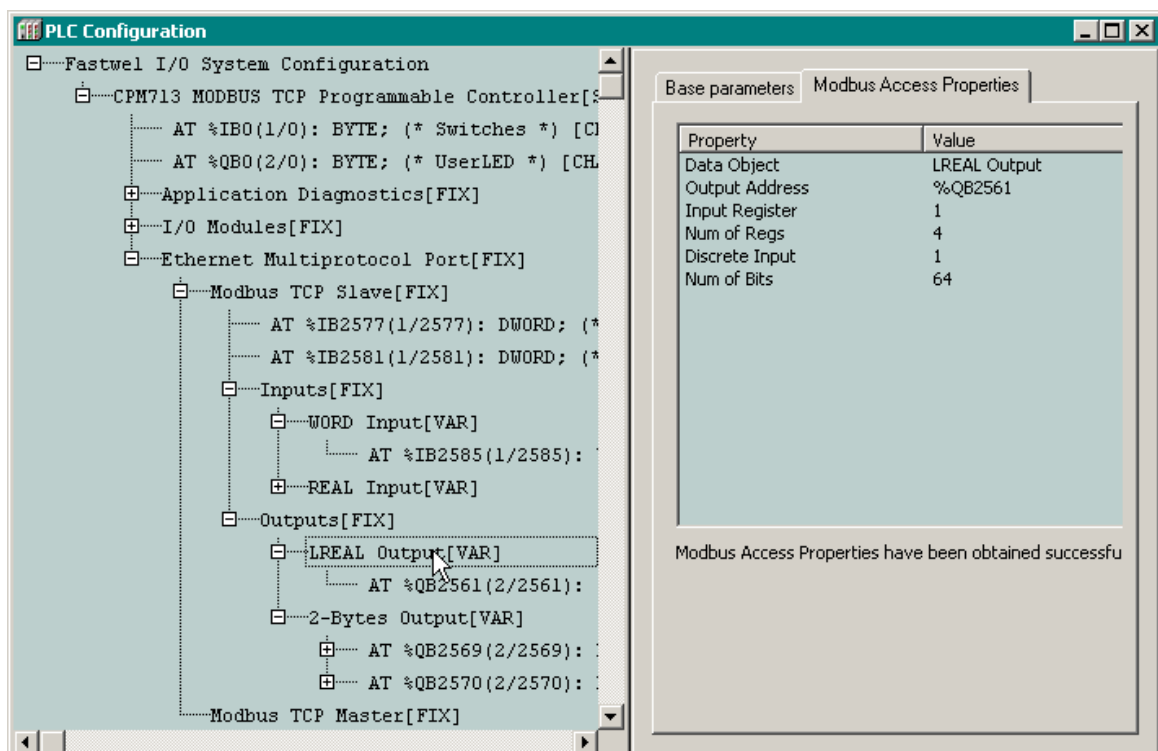


Рис. 4. Конфигурирование коммуникационных объектов сервера MODBUS

3.2.4. Обмен данными с клиентами MODBUS

Стандартные сетевые операции протокола MODBUS, поддерживаемые сервисом сервера MODBUS, перечислены в п. 2.2 настоящего руководства.

Диапазоны регистровых адресов коммуникационных объектов сервера, регистровый адрес, тип и количество регистров для доступа к данным объектов со стороны сети (указывается в сетевом запросе) можно получить в таблице параметров **Modbus Access Properties**, выполнив следующие действия:

1. Выбрать корневой элемент объекта (секция *Inputs* или *Outputs*) в древовидном списке конфигурации сервера, получить параметры доступа ко всем объектам данной секции, как показано на рис. 5.

The screenshot shows a tree view on the left under 'Outputs[FIX]' with the following structure:

- WORD Output
 - AT %Q0
- DWORD Output
 - AT %Q0
- REAL Output
 - AT %QE
- LREAL Output
 - AT %QE
- 2-Bytes Out
 - AT %QE
 - AT %QE

The 'Modbus Access Properties' table on the right is as follows:

Data Object	Output Address	Input Register	Num of Regs	Discrete Input	Num of Bits
WORD Output	%QB64000	1	1	1	16
DWORD Output	%QB64002	2	2	17	32
REAL Output	%QB64006	4	2	49	32
LREAL Output	%QB64010	6	4	81	64
2-Bytes Output	%QB64018	10	1	145	16

Рис. 5. Список и параметры доступа к выходным объектам данных сервера MODBUS

2. Выбрать отдельный объект в дереве конфигурации, получить параметры доступа к интересующему объекту, как показано на рис. 6.

The screenshot shows the same tree view as in Figure 5, but with 'LREAL Output' selected. The 'Modbus Access Properties' table on the right is as follows:

Property	Value
Data Object	LREAL Output
Output Address	%QB64010
Input Register	11
Num of Regs	4
Discrete Input	161
Num of Bits	64

Рис. 6. Параметры доступа к объекту данных сервера MODBUS

Описание назначения отдельных свойств входных и выходных коммуникационных объектов в таблице свойств Modbus Access Properties для секций *Inputs* и *Outputs* приведено в табл. 5.

Таблица 5

Секция <i>Outputs</i>		
Объекты данных, передаваемые в сеть мастеру в ответ на запрос чтения		
Столбец	Наименование	Назначение
<i>Data Object</i>	Тип объекта данных	Название типа объекта данных (см. табл. 5)
<i>Output Address</i>	Начальный адрес в области выходных данных приложения	Смещение в сегменте выходных данных приложения, на которое отображен объект данных
<i>Input Register</i>	Начальный адрес регистра в запросе чтения	Начальный адрес входного регистра (Input Register), используемый для чтения данного объекта или группы объектов, расположенных ниже текущего, по сети. Адрес, передаваемый в запросе MODBUS, должен быть на 1 меньше отображаемого значения.
<i>Num of Regs</i>	Количество регистров в запросе чтения	Количество регистров, которое должно быть указано в сетевом запросе чтения для атомарного получения значения данного объекта с сохранением целостности данных.
<i>Discrete Input</i>	Начальный адрес битового поля в запросе чтения	Начальный адрес битового поля (Discrete Input), доступного для чтения, используемый для чтения данного объекта или группы объектов, расположенных ниже текущего, по сети. Адрес, передаваемый в запросе MODBUS, должен быть на 1 меньше отображаемого значения.
<i>Num of Bits</i>	Количество битовых полей в запросе чтения	Количество битовых полей, которое должно быть указано в сетевом запросе чтения для атомарного получения значения данного объекта с сохранением целостности данных.
Секция <i>Inputs</i>		
Объекты данных, получаемые по сети от мастера в запросе записи		
Столбец	Наименование	Назначение
<i>Data Object</i>	Тип объекта данных	Название типа объекта данных (см. табл. 5)
<i>Input Address</i>	Начальный адрес в области входных данных приложения	Смещение в сегменте входных данных приложения, на которое отображен объект данных
<i>Holding Register</i>	Начальный адрес регистра в запросе записи или чтения	Начальный адрес выходного регистра (Holding Register), используемый для записи или чтения по сети данного объекта или группы объектов, расположенных ниже текущего. Адрес, передаваемый в запросе MODBUS, должен быть на 1 меньше отображаемого значения.
<i>Num of Regs</i>	Количество регистров в запросе записи или чтения	Количество регистров, которое должно быть указано в сетевом запросе записи или чтения для атомарного изменения или получения значения данного объекта с сохранением целостности данных.
<i>Coil</i>	Начальный адрес битового поля в запросе записи или чтения	Начальный адрес битового поля (Coil), доступного для записи или чтения, используемый для записи или чтения данного объекта или группы объектов, расположенных ниже текущего, по сети. Адрес, передаваемый в запросе MODBUS, должен быть на 1 меньше отображаемого значения.
<i>Num of Bits</i>	Количество битовых полей в запросе записи или чтения	Количество битовых полей, которое должно быть указано в сетевом запросе записи или чтения для атомарного изменения или получения значения данного объекта с сохранением целостности данных.

Например, пусть в приложении контроллера требуется иметь 16 переменных типа **BOOL** и одну переменную типа **LREAL**, значения которых должны доставляться контроллеру по сети от мастера протокола MODBUS. При этом для оптимизации сетевого трафика значения переменных типа **BOOL** и **LREAL** должны поступать в одном сетевом запросе записи (тип 3).

Тогда в конфигурацию сервера MODBUS контроллера могут быть добавлены объекты типа *2-Bytes Input* и *LREAL Input*, как показано на рис. 7.

Мастер сети MODBUS может выполнять запись значений в данные объекты следующими регистровыми операциями:

1. Один запрос записи (тип 16) с начальным адресом регистра 0 и количеством регистров 5. Таким образом будет произведено одновременное изменение всех данных, описываемых указанными двумя объектами.
2. Два отдельных запроса записи регистров: для *2-Bytes Input* – с начальным адресом 0 и количеством объектов 1; для *LREAL Input* – с начальным адресом 1 и количеством объектов 4.

Чтение ранее записанных значений при помощи регистровых операций чтения выполняется аналогично.

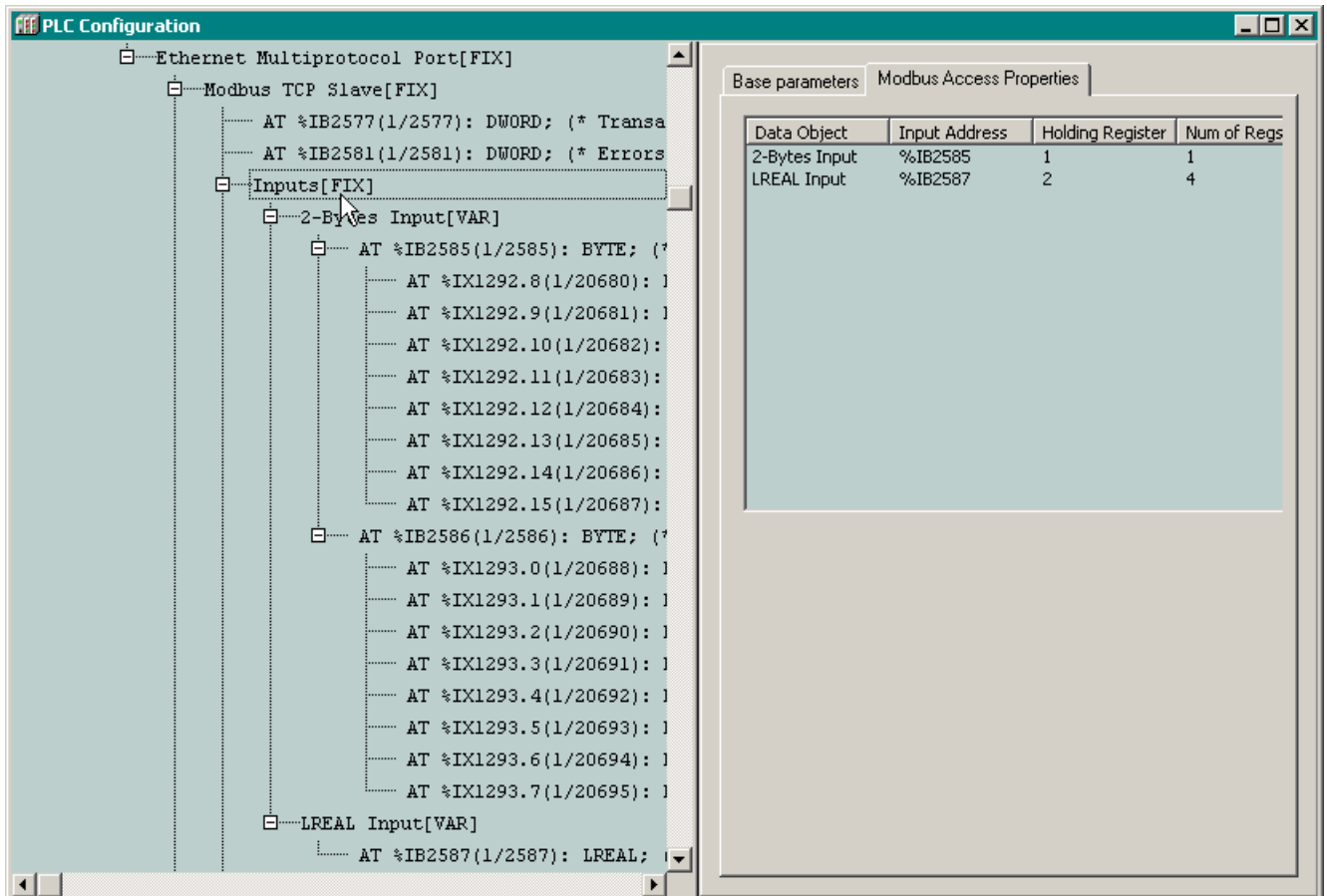


Рис. 7. Пример конфигурации входных объектов сервера

Запись битовых полей объекта *2-Bytes Input* в рассматриваемом примере может выполняться путем запроса записи (тип 15) с адресом 0 и количеством битовых полей 16, либо индивидуально.

Аналогичным образом выполняется чтение мастером MODBUS объектов, расположенных в области конфигурации *Outputs*.

3.2.5. Формат запроса/ответа на чтение расширенной идентификационной информации

Таблица 6

Поле	Длина, байт	Описание		
<i>Запрос</i>				
Код функции	1	43 (2Bh) – инкапсулированный транспорт		
Тип запроса	1	14 (0Eh) – чтение идентификатора устройства		
Код запроса	1	<i>Категория запрашиваемой информации</i>		
		1	базовая информация	
		2	регулярная	
		3	расширенная (не поддерживается)	
		4	доступ к индивидуальным объектам	
Id объекта	1	<i>Код запрашиваемого объекта</i>		
		0	Производитель	Fastwel Co.Ltd.
		1	Код продукта	CPM713 PLC
		2	Код версии продукта	2.52.23926
		3	URL производителя	www.fastwel.com
		4	Наименование продукта	Fastwel CPM713 MODBUS TCP PLC Runtime
		5	Наименование модели	CPM713
6	Имя приложения пользователя	имя файла проекта		
<i>Ответ</i>				
Код функции	1	43 (2Bh) – инкапсулированный транспорт		
Тип запроса	1	14 (0Eh) – чтение идентификатора устройства		
Код запроса	1	<i>Категория запрашиваемой информации</i>		
		1	базовая информация	
		2	регулярная	
		3	расширенная (не поддерживается)	
		4	доступ к индивидуальным объектам	
Уровень совместимости	1	В соответствии со спецификацией		
Признак продолжения	1	=0 – есть продолжение; =255 – нет продолжения		
Число объектов	1	Количество объектов в списке		
<i>Список объектов</i>				
Id объекта	1	...		
Длина объекта	1	...		
Значение	длина объекта	...		

3.2.6. Обслуживание сетевых запросов

Сервис сервера MODBUS активизируется при возникновении прерывания от коммуникационного порта либо при необходимости передать данные в сеть.

Запрос чтения одного или нескольких регистров приводит к тому, что буферизованные значения, ранее выведенные из сегмента выходных данных приложения в участок образа процесса, на который отображены регистры, упаковываются в ответное сообщение и передаются по сети мастеру.

Запрос записи одного или нескольких регистров приводит к тому, что поступившие значения записываются в участок входной части образа процесса, на который отображены соответствующие регистры.

ВНИМАНИЕ!

Одновременно к серверу протокола MODBUS TCP может быть подключено не более 32-х клиентов. При попытке подключения каждого последующего клиента сервер MODBUS TCP контроллера закрывает соединение с клиентом, активность которого не проявлялась в последние 30 секунд, и использует освободившееся соединение для работы с новым клиентом. Клиент, чье соединение было только что разорвано, должен будет повторно его устанавливать перед очередным MODBUS-запросом. Такое поведение поддерживается не всеми клиентами MODBUS TCP, поэтому не следует превышать максимальное количество одновременно подключенных клиентов.

В соответствии с п. 4.2.1 документа "MODBUS MESSAGING ON TCP/IP. IMPLEMENTATION GUIDE" каждый очередной запрос к MODBUS TCP серверу должен передаваться в отдельном сетевом пакете. Поэтому сервер MODBUS TCP контроллера, приняв очередной пакет в сокет порта 502, извлекает сообщение MODBUS TCP (MODBUS ADU) один раз.

Адреса регистров и битовых полей в сетевых запросах к серверу MODBUS должны быть на единицу меньше отображаемых в диалоговых панелях **Modbus Access Properties**.

3.2.7. Диагностика

3.2.7.1. Диагностические каналы

Элемент конфигурации Modbus TCP Slave имеет два входных канала типа DWORD, отображающих количество принятых сетевых запросов и количество сетевых запросов, принятых с ошибками. Описание каналов приведено в табл. 7. Обновление значений данных каналов производится системой исполнения контроллера с периодичностью около 1 с.

Таблица 7

Элемент/канал	Адрес	Тип	Назначение
TransactionsCount	%IB2577	DWORD	Количество принятых входящих сообщений
ErrorsCount	%IB2581	DWORD	Количество входящих сообщений, при приеме которых обнаружены ошибки

При поступлении по сети корректного пакета протокола MODBUS значение на канале *MessagesCount* увеличивается на 1. При обнаружении ошибки в принятом сообщении значение на канале *ErrorsCount* увеличивается на 1.

3.2.7.2. Инициализация сервиса внешней сети

Системное программное обеспечение при включении питания или перезапуске контроллера считывает конфигурацию сервиса внешней сети, задает параметры обмена, создает и связывает регистры, описания которых имеются в конфигурации. Если в процессе конфигурирования сервиса внешней сети произошли критические ошибки, контроллер будет переведен в безопасный режим с индикацией, описание которой приведено в табл. 8.

Таблица 8

Индикация безопасного режима по ошибке в конфигурации сервиса внешней сети	
Индикатор: APP Цвет: Зеленый Частота F _{Hz} : 1 Гц	
N (кол-во включений)	Причина
2	– неправильный тип протокола; – неправильный тип сети; – неподдерживаемый тип сетевого устройства; – отсутствующий номер сетевого устройства
3	недопустимый адрес (идентификатор) узла
4	резерв
5	– неправильный формат конфигурации; – неправильный тип объекта данных; – неподдерживаемый или неправильный параметр коммуникационного объекта
6	резерв

3.2.7.3. Индикация

Индикатор COMM светится красным цветом, если контроллер не подключен к сети Ethernet.

Индикатор COMM не светится, если контроллер подключен к сети Ethernet, но в сети отсутствуют пакеты, поступающие контроллеру.

Индикатор COMM светится зеленым цветом, если контроллер подключен к сети Ethernet, и в сети имеются пакеты, поступающие контроллеру.

3.3. Принцип работы и способы конфигурирования мастера MODBUS

3.3.1. Общие сведения

Контроллер CPM713, наряду с функциональностью сервера MODBUS TCP, реализует функциональность мастера (клиента) протокола MODBUS TCP. Настройка параметров клиента MODBUS TCP выполняется в дереве конфигурации контроллера в отношении элемента *CPM713 MODBUS TCP Programmable Controller–Ethernet Multiprotocol Port–Modbus TCP Master*, как показано на рис. 8.

Конфигурирование клиента MODBUS состоит в выполнении следующих действий:

1. Установка параметров узла мастера.
2. Создание списка подчиненных узлов, с которыми мастер должен осуществлять обмен данными и настройка параметров соединений;

3. Создание списков и редактирование параметров коммуникационных объектов подчиненных узлов, описывающих области считываемых и записываемых данных и расписание обмена.

3.3.2. Настройка параметров мастера MODBUS

Для настройки параметров мастера протокола MODBUS RTU или ASCII используется **Панель свойств Fastwel**, показанная на рис. 8.

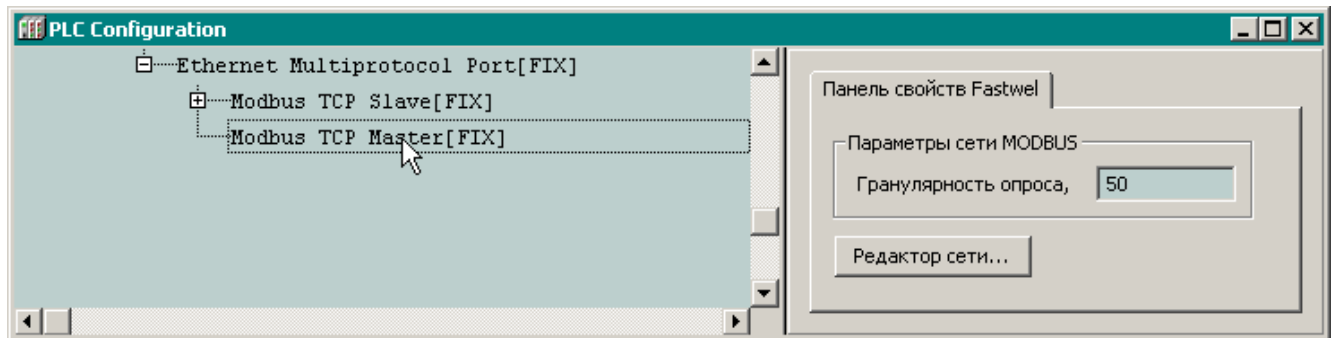


Рис. 8. Параметры мастера протокола MODBUS

Единственный параметр **Гранулярность опроса** определяет временной квант периода обмена с подчиненными узлами (в миллисекундах). Чтение и запись коммуникационных объектов, описанных в конфигурации каждого подчиненного узла данного мастера, будут выполняться с периодами, определяемыми произведением значения в поле **Гранулярность опроса** на значения, заданные для параметров **Тип передачи** коммуникационных объектов.

В безопасном режиме контроллера сервис клиента MODBUS отключен от коммуникационного интерфейса!

В конфигурацию мастера MODBUS должны быть добавлены описания отдельных подчиненных узлов и коммуникационных объектов, которые будут запрашиваться мастером у каждого подчиненного узла.

Для добавления подчиненного узла в список мастера щелкните правой кнопкой мыши на его имени и выберите команду **Append TCP Slave** в появившемся контекстном меню. Для вставки узла в список щелкните правой кнопкой мыши на соответствующей позиции в списке подчиненных узлов и выберите в контекстном меню команду **Insert TCP Slave**.

Для удаления ранее добавленного подчиненного узла щелкните правой кнопкой мыши на его имени и выберите в контекстном меню команду **Delete** или выберите подчиненный узел в списке и нажмите клавишу **Del**.

Параметры подчиненного узла мастера MODBUS становятся доступными для редактирования в панели свойств Fastwel после щелчка мышью над элементом *TCP Slave* в соответствующей позиции списка конфигурации, как показано на рис. 9.

Параметр **Гранулярность опроса** определяет временной квант периода обмена с подчиненными узлами (в миллисекундах). Чтение и запись коммуникационных объектов, описанных в конфигурации каждого подчиненного узла данного мастера, будут выполняться с периодами, определяемыми произведением значения в поле **Гранулярность опроса** на значения, заданные для параметров **Тип передачи** коммуникационных объектов.

Перечень и назначение параметров подчиненного узла MODBUS приведены в табл. 9.

Таблица 9

Обозначение	Назначение
IP адрес	IP-адреса подчиненного узла
Порт	TCP-порт сервера MODBUS TCP. По умолчанию 502.
ID адресата	Идентификатор узла
Таймаут ответа	Максимальное время ожидания ответа на запрос
Переставлять слова	Флаг перестановки байт при передаче/приеме слов данных узлу. Клиент использует принятое стандартом Big-Endian представление для адресов и слов данных.. Для серверов, использующих Little-Endian представление, отметить флажок.

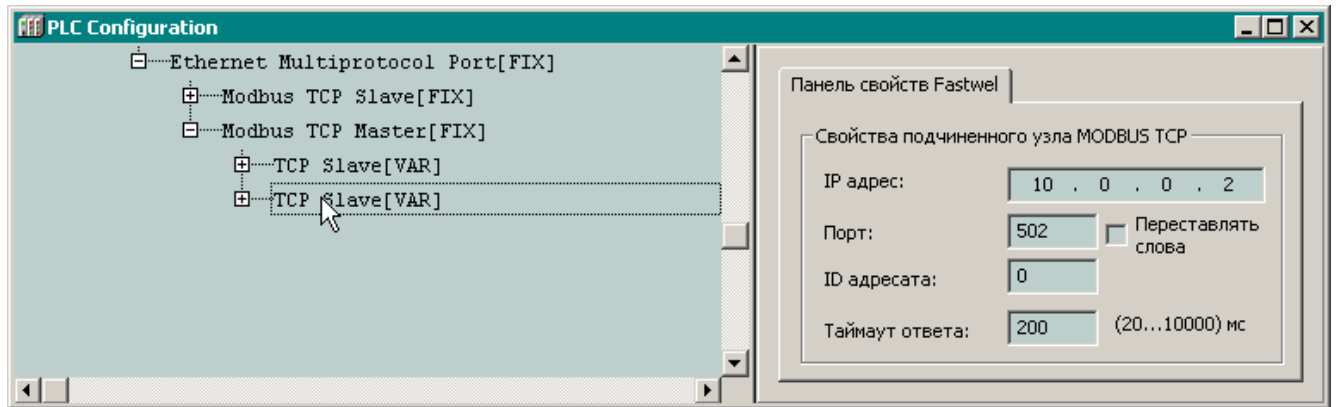


Рис. 9. Параметры мастера протокола MODBUS

3.3.3. Обмен данными с подчиненными устройствами

Конфигурирование обмена данными с подчиненным устройством (сервером MODBUS) осуществляется посредством описания его коммуникационных объектов, показанных на рис. 10. Коммуникационный объект определяет области данных на стороне клиента и сервера, расписание и направление операций доступа к данным (чтение/запись/запись-чтение).

Перечень типов и назначение используемых коммуникационных объектов приведены в табл.10.

Таблица 10

Обозначение	Назначение
Read-Write Coil	Используется для доступа по записи-чтению к данным типа Coil сервера
Read-Write Holding Register	Используется для доступа по записи-чтению к данным типа Holding Registers сервера
Write-Only Coil	Используется для доступа по записи к данным типа Coil сервера
Write-Only Holding Register	Используется для доступа по записи к данным типа Holding Registers сервера
Read-Only Input	Используется для доступа по чтению к данным типа Discrete Input сервера
Read-Only Register	Используется для доступа по чтению к данным типа Input Registers сервера
Read-Only Coil	Используется для доступа по чтению к данным типа Coil сервера
Read-Only Holding Register	Используется для доступа по чтению к данным типа Holding Registers сервера

Для добавления коммуникационного объекта в список сервера щелкните правой кнопкой мыши на позиции подчиненного узла, выберите в контекстном меню команду **Append Subelement** и затем требуемый тип объекта.

Для вставки коммуникационного объекта в список щелкните правой кнопкой мыши на соответствующей позиции в списке коммуникационных объектов сервера, выберите в контекстном меню команду **Insert Element** и затем требуемый тип объекта.

Для удаления коммуникационного объекта щелкните правой кнопкой мыши на его позиции и выберите в контекстном меню команду **Delete** или выберите объект в списке и нажмите клавишу **Del**.

Параметры коммуникационного объекта становятся доступными для редактирования в панели свойств Fastwel после щелчка мышью над его именем, как показано на рис. 10. Перечень и назначение параметров приведены в табл. 11.

Таблица 11

Обозначение	Назначение	
Адрес объекта	Указывается MODBUS адрес области данных на сервере. Данное значение должно быть на единицу больше передаваемого в сетевом запросе серверу MODBUS.	
Тип передачи	0	Указывает на то, что клиент не проводит сетевые операции с данными объекта (не активный объект)
	1...250	Значение N от 1 до 250 указывает на то, что клиент проводит сетевые операции с данными объекта с периодом $P = N * T_0$, где T_0 – значение параметра Гранулярность опроса , заданного в конфигурации мастера.
	251...254	не используются
	255	Для объектов с доступом только по чтению не используется. Для объектов с доступом по записи указывает на то, что сетевые операции будут производиться по изменению данных, передаваемых серверу

Тип (Read-Only/Write-Only/Read-Write) и адрес коммуникационного объекта определяют адрес области данных (регистр MODBUS) на стороне сервера и тип операции доступа (чтение/запись/запись-чтение). Параметр **Тип передачи** задает расписание обмена.

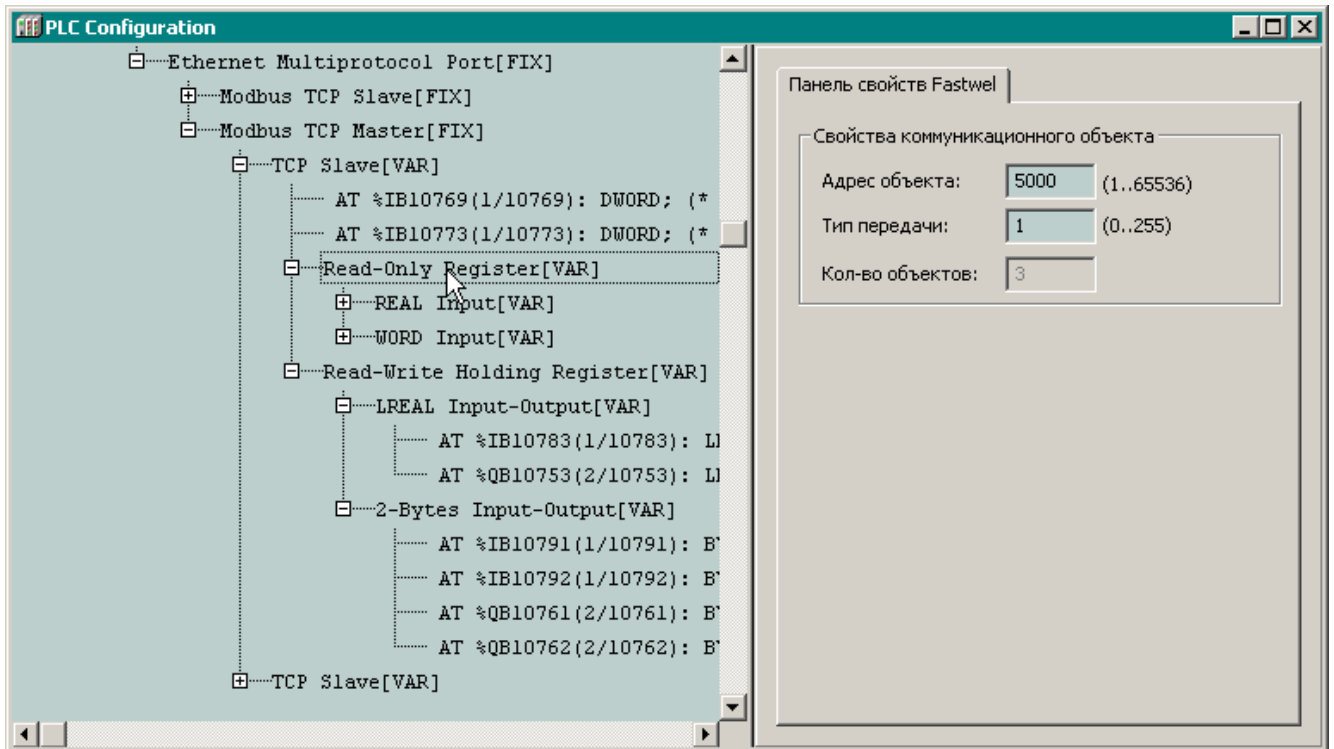


Рис. 10. Коммуникационные объекты мастера MODBUS

Область данных со стороны приложения задается списком переменных объекта. Перечень используемых переменных и схема адресации в зависимости от типа объекта приведены в табл. 12.

Таблица 12

Тип объекта	Тип переменной	Область данных переменной
Read-Write Coil	8-Bit Input-Output	Однобайтовый входной и выходной канал типа BYTE
	16-Bit Input-Output	Двухбайтовый входной и выходной канал типа WORD
	32-Bit Input-Output	Четырехбайтовый входной и выходной канал типа DWORD
Read-Write Holding Register	WORD Input-Output	Двухбайтовый входной и выходной канал типа WORD
	DWORD Input-Output	Четырехбайтовый входной и выходной канал типа DWORD
	REAL Input-Output	Четырехбайтовый входной и выходной канал типа REAL
	LREAL Input-Output	Восьмибайтовый входной канал типа LREAL
	2-Bytes Input-Output	2 однобайтовых входных и выходных канала типа BYTE
Write-Only Coil	8-Bit Output	Однобайтовый выходной канал типа BYTE
	16-Bit Output	Двухбайтовый выходной канал типа WORD
	32-Bit Output	Четырехбайтовый выходной канал типа DWORD
Write-Only Holding Register	WORD Output	Двухбайтовый выходной канал типа WORD
	DWORD Output	Четырехбайтовый выходной канал типа DWORD
	REAL Output	Четырехбайтовый выходной канал типа REAL
	LREAL Output	Восьмибайтовый выходной канал типа LREAL
	2-Bytes Output	2 однобайтовых выходных канала типа BYTE
Read-Only Input	8-Bit Input	Однобайтовый входной канал типа BYTE
	16-Bit Input	Двухбайтовый входной канал типа WORD
	32-Bit Input	Четырехбайтовый входной канал типа DWORD
Read-Only Register	WORD Input	Двухбайтовый входной канал типа WORD
	DWORD Input	Четырехбайтовый входной канал типа DWORD
	REAL Input	Четырехбайтовый выходной канал типа REAL
	LREAL Input	Восьмибайтовый входной канал типа LREAL
	2-Bytes Input	2 однобайтовых входных канала типа BYTE
Read-Only Coil	8-Bit Input	Однобайтовый входной канал типа BYTE
	16-Bit Input	Двухбайтовый входной канал типа WORD
	32-Bit Input	Четырехбайтовый входной канал типа DWORD
Read-Only Holding Register	WORD Input	Двухбайтовый входной канал типа WORD
	DWORD Input	Четырехбайтовый входной канал типа DWORD
	REAL Input	Четырехбайтовый выходной канал типа REAL
	LREAL Input	Восьмибайтовый входной канал типа LREAL
	2-Bytes Input	2 однобайтовых входных канала типа BYTE

Для добавления переменной к коммуникационному объекту щелкните правой кнопкой мыши на его позиции, выберите в контекстном меню команду **Append Subelement** и затем требуемый тип переменной.

Для вставки переменной в список щелкните правой кнопкой мыши на соответствующей позиции в списке переменных коммуникационного объекта, выберите в контекстном меню команду **Insert Element** и затем требуемый тип переменной.

Для удаления переменной щелкните правой кнопкой мыши на ее позиции и выберите в контекстном меню команду **Delete** или выберите ее в списке и нажмите клавишу **Del**.

Сервис клиента MODBUS активизируется при наступлении события, требующего выполнения сетевой операции доступа к данным подчиненного устройства.

Выполнение операции записи в удаленный узел приводит к тому, что буферизованные значения, ранее выведенные из сегмента выходных данных приложения в участок образа процесса, на который отображены переменные коммуникационных объектов, передаются по сети подчиненному узлу.

Выполнение операции чтения из удаленного узла приводит к тому, что поступившие по сети из подчиненного устройства значения записываются в участок входной части образа процесса, на который отображены соответствующие переменные коммуникационного объекта.

Стандартные сетевые операции протокола MODBUS, используемые сервисами клиентов MODBUS для выполнения транзакций обмена данными с подчиненными узлами, перечислены в п. 2.2.2 настоящего руководства.

3.3.4. Диагностика

В конфигурации подчиненного узла мастера MODBUS *TCP Slave* имеется секция, в которой определены два диагностических входных канала, позволяющих приложению во время выполнения получить общее количество инициированных мастером транзакций обмена данными с удаленным узлом и общее количество транзакций завершившихся с ошибкой. Описание каналов приведено в табл. 13.

Таблица 13

Элемент/канал	Тип	Назначение
TransactionsCount	DWORD	Количество инициированных мастером транзакций обмена данными с сервером
ErrorsCount	DWORD	Количество транзакций, при выполнении которых обнаружены ошибки

3.4. Доступ к полям данных коммуникационных объектов из приложения

Доступ к данным для передачи в сеть осуществляется посредством выходных переменных программы, ссылающихся на выходные каналы объектов, как показано в табл. 14.

Таблица 14

Объект выходных данных	Выходные каналы				Выходные переменные
DWORD Output	%QX0.0–%QX0.7	%QB0	%QW0	%QD0	VAR dwOut AT%QD0 : DWORD; wOut0 AT%QW1 : WORD; wOut1 AT%QW1 : WORD; byteOut0 AT%QB0 : BYTE; byteOut1 AT%QB1 : BYTE; byteOut2 AT%QB2 : BYTE; byteOut3 AT%QB3 : BYTE; bitOut25 AT%QX1.9 : BOOL; END_VAR
	%QX0.8–%QX0.15	%QB1			
	%QX1.0–%QX1.7	%QB2	%QW1		
	%QX1.8–%QX1.15	%QB3			
REAL Output	%QX2.0–%QX2.7	%QB4	%QW2	%QD1	VAR realOut AT%QB4 : REAL; END_VAR
	%QX2.8–%QX2.15	%QB5			
	%QX3.0–%QX3.7	%QB6	%QW3		
	%QX3.8–%QX3.15	%QB7			

Доступ к данным, поступающим по сети, осуществляется через входные переменные приложения, ссылающиеся на каналы объектов данных, как показано в табл. 15.

Таблица 15

Объект входных данных	Входные каналы			Входные переменные
2-Bytes Input	%IX0.0–%IX0.7	%IB0	%IW0	VAR wIn AT%IW0 : WORD; byteIn0 AT%IB0 : BYTE; byteIn1 AT%IB1 : BYTE; bitIn1 AT%QX0.1 : BOOL; END_VAR
	%IX0.8–%IX0.15	%IB1		
REAL Input	%IX1.0–%IX1.7	%IB2	%IW1	VAR realIn AT%IB2 : REAL; END_VAR
	%IX1.8–%IX1.15	%IB3		
	%IX2.0–%IX2.7	%IB4	%IW2	
	%IX2.8–%IX2.15	%IB5		

Имеется возможность создания символических ссылок на каналы объектов данных в ресурсе **PLC Configuration**, как показано на рис. 11.

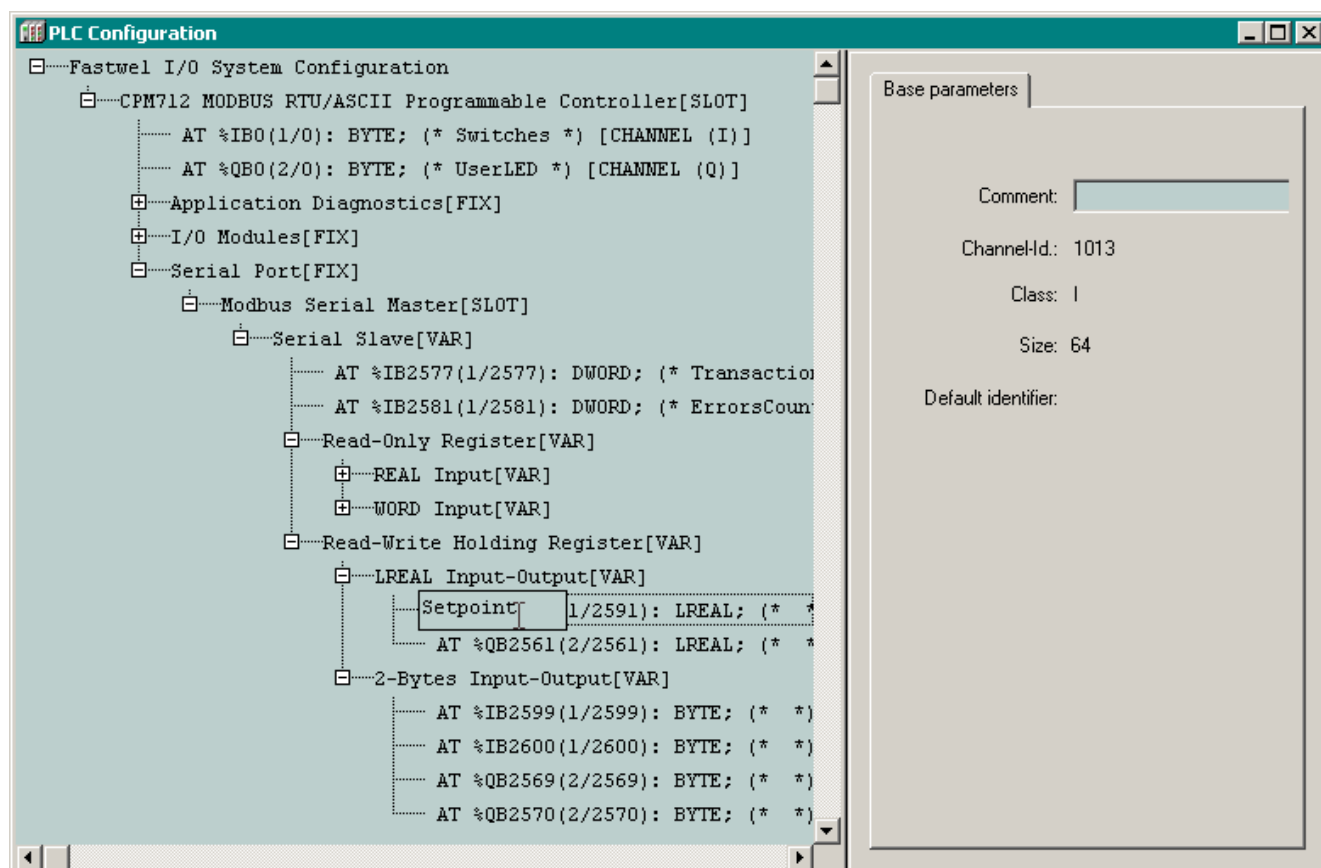


Рис. 11. Создание символических ссылок

ВНИМАНИЕ!

Пусть в конфигурации сервисов внешней сети имеется входной (выходной) объект данных, канал которого ссылается на некоторый адрес в области выходных (входных) данных приложения. При последующей вставке или удалении в конфигурации контроллера описаний секций, имеющих выходные (входные) каналы, канал данного объекта будет ссылаться на другой адрес в области выходных (входных) данных программы, что потребует ручной коррекции адресов выходных (входных) переменных, ссылающихся на канал данного объекта данных.

Данная проблема может быть частично решена путем использования символических ссылок на каналы отдельных каналов регистров.

3.5. Библиотека FastwelSysLibSockets.lib

3.5.1. Общие сведения

Данная библиотека поддерживает работу с сокетами для коммуникаций по TCP/IP и UDP, по сути являясь аналогом исходной библиотеки SysLibSockets.lib, входящей в стандартную среду исполнения CoDeSys.

Следует обратить внимание на тот факт, что в библиотеке FastwelSysLibSockets.lib реализованы только неблокирующиеся сокеты, и выполнение функций асинхронно по отношению к потоку исполнения, из которого вызываются функции. Таким образом, функции библиотеки не используют таймаутов ожидания и не приостанавливают выполнение вызывающей программы CoDeSys.

Пример использования библиотеки имеется в проектах *fsyslibsock_srv_cpm713.pro* и *fsyslibsock_clnt_cpm713.pro*, которые входят в пакет адаптации CoDeSys для Fastwel I/O.

3.5.2. Описание функций

3.5.2.1. FwSysSockCreate

Функция создает новый сокет. Возвращает дескриптор сокета, используемый другими функциями (например, FwSysSockBind, FwSysSockConnect).

```
FUNCTION FwSysSockCreate : DINT
VAR_INPUT
    diAddressFamily:DINT;
    diType:DINT;
    diProtocol:DINT;
END_VAR
;
```

Входные параметры:

diAddressFamily:DINT

Идентификатор формата адреса. Единственное допустимое значение: SOCKET_AF_INET.

diType:DINT

Идентификатор типа создаваемого сокета. Одно из двух допустимых значений:

SOCKET_STREAM – используется при создании сокета для TCP соединения;

SOCKET_DGRAM – используется при создании сокета для UDP соединения.

diProtocol:DINT

Идентификатор протокола. Одно из двух допустимых значений:

SOCKET_IPPROTO_TCP – используется при создании сокета для TCP соединения;

SOCKET_IPPROTO_UDP – используется при создании сокета для UDP соединения.

Возвращаемый результат:

Системный идентификатор сокета, который в дальнейшем используется в вызовах других функций библиотеки. В случае ошибки (если сокет не может быть создан) возвращается значение: SOCKET_INVALID. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.2.2. FwSysSockClose

Функция закрывает сокет, разрывая соединение с удаленным узлом.

```
FUNCTION FwSysSockClose : BOOL
VAR_INPUT
    diSocket:DINT;
END_VAR
;
```

Входные параметры:

diSocket:DINT

Системный идентификатор сокета.

Возвращаемый результат:

Возвращает TRUE в случае успеха, иначе FALSE. Код ошибки может быть уточнен вызовом функции **FwSysSockGetLastError**.

3.5.2.3. FwSysSockListen

Функция устанавливает сокет в режим сервера, ожидающего запросы на установление соединений с удаленными клиентами.

```
FUNCTION FwSysSockListen : BOOL
VAR_INPUT
    diSocket:DINT;
    diMaxConnections:DINT;
END_VAR
;
END_FUNCTION
```

Входные параметры:

diSocket:DINT

Системный идентификатор сокета, созданного функцией **FwSysSockCreate**.

diMaxConnections:DINT

Максимальное число запросов в очереди на соединение. Допустимое значение от 1 до 5.

Возвращаемый результат:

Возвращает TRUE в случае успеха, иначе FALSE. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.2.4. FwSysSockAccept

Функция принимает входящее соединение, ожидаемое на серверном сокете и, в случае успеха, создает сокет соединения с клиентом. Функция не блокирует выполнение вызывающей программы.

```
FUNCTION FwSysSockAccept : DINT
VAR_INPUT
    diSocket:DINT;
    pSockAddr:DWORD;
    piSockAddrSize:DWORD;
END_VAR
END_FUNCTION
```

Входные параметры:

diSocket:DINT

Идентификатор серверного сокета, установленного в режим ожидания входящих соединений функцией **FwSysSockListen**.

pSockAddr:DWORD

Указатель на переменную типа SOCKADDR, которая будет заполнена адресом вызывающего клиента. Если передается 0 информация о клиенте не возвращается.

piSockAddrSize:DWORD

Указатель на переменную типа DINT, содержащую размер структуры SOCKADDR (может быть получен оператором SIZEOF). Если передается 0, информация о клиенте не возвращается.

Возвращаемый результат:

Системный идентификатор сокета нового клиентского соединения. В случае ошибки возвращается значение SOCKET_INVALID. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.2.5. FwSysSockBind

Функция ассоциирует сокет с локальным IP адресом и портом. Используется на вновь созданном сокете до вызова **FwSysSockConnect** или **FwSysSockListen**.

```
FUNCTION FwSysSockBind : BOOL
VAR_INPUT
    diSocket:DINT;
    pSockAddr:DWORD;
    diSockAddrSize:DINT;
END_VAR
END_FUNCTION
```

Входные параметры:

diSocket:DINT

Системный идентификатор сокета созданного функцией **FwSysSockCreate**.

pSockAddr:DWORD

Указатель на переменную типа SOCKADDR.

diSockAddrSize:DINT

Размер структуры SOCKADDR (может быть получен оператором SIZEOF).

Возвращаемый результат:

Возвращает TRUE в случае успеха, иначе FALSE. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.2.6. FwSysSockConnect

Функция устанавливает соединение с удаленным сервером. Функция не блокирует выполнение вызывающей программы.

```
FUNCTION FwSysSockConnect : BOOL
VAR_INPUT
    diSocket:DINT;
    pSockAddr:DWORD;
    diSockAddrSize:DINT;
END_VAR
END_FUNCTION
```

Входные параметры:

diSocket:DINT

Системный идентификатор сокета созданного функцией **FwSysSockCreate**.

pSockAddr:DWORD

Указатель на переменную типа SOCKADDR.

diSockAddrSize:DINT

Размер структуры SOCKADDR (может быть получен оператором SIZEOF).

Возвращаемый результат:

Возвращает TRUE в случае успеха, иначе FALSE. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**. Значение кода ошибки SOCKET_EWOULDBLOCK указывает на то, что процесс установки соединения инициирован, но не завершен к настоящему времени. В этом случае фактический результат установки соединения с сервером должен быть в дальнейшем получен проверкой статусов «записи» и «ошибок» для данного сокета функцией **FwSysSockSelect**.

3.5.2.7. FwSysSockGetOption

Функция получает значение соответствующей опции сокета.

```
FUNCTION FwSysSockGetOption : BOOL
VAR_INPUT
    diSocket:DINT;
    diLevel:DINT;
    diOption:DINT;
    pOptionValue:DWORD;
    piOptionLength:DWORD;
END_VAR
;
END_FUNCTION
```

Входные параметры:

diSocket:DINT

Системный идентификатор сокета, созданного функцией **FwSysSockCreate**.

diLevel:DINT

Специфический уровень протокола:

SOCKET_SOL – уровень сокетов,

SOCKET_IPPROTO_TCP – уровень протокола TCP.

diOption:DINT

Идентификатор имени опции. **FwSysSockSetOption** поддерживаются следующие опции:

Уровень SOCKET_IPPROTO_TCP:

Код	Тип	Значение
SOCKET_TCP_NODELAY	BOOL	Выключает Nagle алгоритм (запрещает передачу новых TCP сегментов при поступлении новых выходных данных пользователя, если имеются ранее отправленные неподтвержденные данные).

Уровень SOCKET_SOL:

Код	Тип	Значение
SOCKET_SO_BROADCAST	BOOL	Позволяет передавать и принимать широковещательные сообщения.
SOCKET_SO_KEEPAIVE	BOOL	Посылает сообщения “keep-alive”.

SOCKET_SO_ACCEPTCONN	BOOL	Режим приема входящих соединений.
SOCKET_SO_ERROR	DINT	Чтение и очистка статуса ошибки.
SOCKET_SO_TYPE	DINT	Тип сокета.
SOCKET_SO_RCVBUF	DINT	Размер буфера приемной очереди.
SOCKET_SO_REUSEADDR	BOOL	Позволяет связать сокет с локальным адресом, который уже используется на другом открытом сокете.
SOCKET_SO_SNDBUF	DINT	Размер буфера выходной очереди.

pOptionValue:DWORD

Указатель на буфер, в который будет записано значение опции.

piOptionLength:DWORD

Указатель на размер буфера.

Возвращаемый результат:

Возвращает TRUE в случае успеха, иначе FALSE. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.2.8. FwSysSockSetOption

Функция устанавливает значение соответствующей опции сокета.

FUNCTION FwSysSockSetOption : BOOL

VAR_INPUT

diSocket:DINT;

diLevel:DINT;

diOption:DINT;

pOptionValue:DWORD;

diOptionLength:DINT;

END_VAR

END_FUNCTION

Входные параметры:

diSocket:DINT

Системный идентификатор сокета созданного функцией **FwSysSockCreate**.

diLevel:DINT

Специфический уровень протокола:

SOCKET_SOL – уровень сокетов,

SOCKET_IPPROTO_TCP – уровень протокола TCP.

diOption:DINT

Идентификатор имени опции. **FwSysSockSetOption** поддерживаются следующие опции:

Уровень SOCKET_SOL:

Код	Тип	Значение
SOCKET_SO_BROADCAST	BOOL	Позволяет передавать и принимать широковещательные сообщения.
SOCKET_SO_KEEPAIVE	BOOL	Посылает сообщения “keep-alive”.
SOCKET_SO_RCVBUF	DINT	Устанавливает размер буфера приемной очереди для сокета.
SOCKET_SO_REUSEADDR	BOOL	Позволяет связать сокет с локальным адресом, который уже используется на другом открытом сокете.
SOCKET_SO_SNDBUF	DINT	Устанавливает размер буфера выходной очереди для сокета.

Уровень SOCKET_IPPROTO_TCP:

Код	Тип	Значение
SOCKET_TCP_NODELAY	BOOL	Выключает Nagle алгоритм (запрещает передачу новых TCP сегментов при поступлении новых выходных данных пользователя, если на соединении остаются неподтвержденными ранее отправленные данные).

pOptionValue:DWORD

Указатель на буфер, в который записано значение опции.

diOptionLength:DINT

Размер буфера значения опции.

Возвращаемый результат:

Возвращает TRUE в случае успеха, иначе FALSE. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.2.9. FwSysSockIoctl

Функция управления режимом ввода/вывода сокета.

```
FUNCTION FwSysSockIoctl : DINT
VAR INPUT
    diSocket:DINT;
    diCommand:DINT;
    piParameter:DWORD;
END_VAR
;
END_FUNCTION
```

Входные параметры:

diSocket:DINT

Системный идентификатор сокета созданного функцией **FwSysSockCreate**.

diCommand:DINT

Команда, которую нужно применить к сокету. Допустимые значения: SOCKET_FIONBIO, SOCKET_FIONREAD.

piParameter:DWORD

Указатель на параметр команды.

Команда	Тип параметра	Значение
SOCKET_FIONBIO	DINT	Не 0 – устанавливает неблокирующий режим ввода/вывода для сокета. 0 – устанавливает блокирующий режим. Внимание! В текущей реализации поддерживается только неблокирующий режим, который устанавливается по умолчанию при создании сокета. Попытка задания блокирующего режима заканчивается неудачей.
SOCKET_FIONREAD	DINT	Используется для получения числа байт, находящихся в приемном буфере сокета и готовых для считывания.

Возвращаемый результат:

Возвращает 0 в случае успеха, иначе SOCKET_INVALID. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.2.10. FwSysSockSelect

Функция проверки одного или нескольких сокетов на готовность к определенным коммуникационным действиям. Функция не блокирует выполнение вызывающей программы.

```
FUNCTION FwSysSockSelect : DINT
VAR INPUT
    diWidth:DINT;
    fdRead:DWORD;
    fdWrite:DWORD;
    fdExcept:DWORD;
    ptvTimeout:DWORD;
END_VAR
;
END_FUNCTION
```

Входные параметры:

diWidth:DINT

Единственное допустимое значение SOCKET_FD_SETSIZE.

fdRead:DWORD

Опциональный указатель на структуру, определяющую сокеты, для которых необходимо проверить статус операции чтения. Может быть 0.

fdWrite:DWORD

Опциональный указатель на структуру, определяющую сокеты, для которых необходимо проверить статус операции записи. Может быть 0.

fdExcept:DWORD

Опциональный указатель на структуру, определяющую сокеты, для которых необходимо проверить статус ошибок. Может быть 0.

ptvTimeout:DWORD

Игнорируется.

Возвращаемый результат:

В случае успеха возвращается суммарное число сокетов, удовлетворяющих условиям проверки. В случае ошибки возвращается значение `SOCKET_INVALID`. Код ошибки может быть получен вызовом функции `FwSysSockGetLastError`.

3.5.2.11. FwSysSockRecv

Функция считывает данные, полученные от удаленного узла соединения. Функция не блокирует выполнение вызывающей программы.

```
FUNCTION FwSysSockRecv : DINT
VAR_INPUT
    diSocket:DINT;
    pbyBuffer:DWORD;
    diBufferSize:DINT;
    diFlags:DINT;
END_VAR
;
```

Входные параметры:

diSocket:DINT

Системный идентификатор сокета с установленным соединением.

pbyBuffer:DWORD

Адрес буфера данных.

diBufferSize:DINT

Размер буфера данных.

diFlags:DINT

Опции вызова функции скомбинированные битовым оператором ИЛИ:

Имя	Значение
SOCKET_MSG_PEEK	Данные копируются в буфер пользователя, но не удаляются из входной очереди. Возвращает число байт данных, которое может быть считано одной операцией чтения.

Возвращаемый результат:

Возвращает число принятых байт (если установлена опция `SOCKET_MSG_PEEK`, возвращает число байт, которое может быть считано одной операцией чтения), или `SOCKET_INVALID` в случае ошибки. Код ошибки может быть получен вызовом функции `FwSysSockGetLastError`. Если соединение было закрыто удаленным узлом, возвращаемое значение равно 0.

3.5.2.12. FwSysSockRecvFrom

Функция считывает данные, полученные от удаленного узла соединения. Для сокетов UDP соединений (тип `SOCKET_DGRAM`) сохраняет адрес узла, являющегося источником данных. Функция не блокирует выполнение вызывающей программы.

```
FUNCTION FwSysSockRecvFrom : DINT
VAR_INPUT
    diSocket:DINT;
    pbyBuffer:DWORD;
    diBufferSize:DINT;
    diFlags:DINT;
    pSockAddr:DWORD;
    diSockAddrSize:DINT;
END_VAR
;
```

Входные параметры:

diSocket:DINT

Системный идентификатор сокета с установленным соединением.

pbyBuffer:DWORD

Адрес буфера данных.

diBufferSize:DINT

Размер буфера данных.

diFlags:DINT

Опции вызова функции скомбинированные битовым оператором ИЛИ:

Имя	Значение
SOCKET_MSG_PEEK	Данные копируются в буфер пользователя, но не удаляются из входной очереди. Возвращает число байт данных, которое может быть считано одной операцией чтения.

pSockAddr : **DWORD**

Указатель на переменную типа SOCKADDR, в которую будет записан адрес узла, отправителя данных.

diSockAddrSize : **DINT**

Размер структуры SOCKADDR (может быть получен оператором SIZEOF).

Возвращаемый результат:

Возвращает число принятых байт (если установлена опция SOCKET_MSG_PEEK возвращает число байт, которое может быть считано одной операцией чтения), или SOCKET_INVALID в случае ошибки. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**. Если соединение было закрыто удаленным узлом, возвращаемое значение равно 0.

3.5.2.13. FwSysSockSend

Функция записывает данные в буфер передачи сокета. Функция не блокирует выполнение вызывающей программы.

```
FUNCTION FwSysSockSend : DINT
```

```
VAR_INPUT
```

```
  diSocket:DINT;
```

```
  pbyBuffer:DWORD;
```

```
  diBufferSize:DINT;
```

```
  diFlags:DINT;
```

```
END_VAR
```

```
;
```

```
END_FUNCTION
```

Входные параметры:

diSocket : **DINT**

Системный идентификатор сокета с установленным соединением.

pbyBuffer : **DWORD**

Адрес буфера данных.

diBufferSize : **DINT**

Размер буфера данных.

diFlags : **DINT**

Опции вызова функции. В текущей реализации значение аргумента игнорируется.

Возвращаемый результат:

Возвращает число записанных байт, которое может быть меньше переданного числа байт, или SOCKET_INVALID в случае ошибки. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.2.14. FwSysSockSendTo

Функция записывает данные в буфер передачи сокета для отправки данных по указанному адресу. Функция не блокирует выполнение вызывающей программы.

```
FUNCTION FwSysSockSendTo : DINT
```

```
VAR_INPUT
```

```
  diSocket:DINT;
```

```
  pbyBuffer:DWORD;
```

```
  diBufferSize:DINT;
```

```
  diFlags:DINT;
```

```
  pSockAddr:DWORD;
```

```
  diSockAddrSize:DINT;
```

```
END_VAR
```

```
;
```

```
END_FUNCTION
```

Входные параметры:

diSocket : **DINT**

Системный идентификатор сокета с установленным соединением.

pbyBuffer : **DWORD**

Адрес буфера данных.

diBufferSize : **DINT**

Размер буфера данных.

diFlags : **DINT**

Опции вызова функции. В текущей реализации значение аргумента игнорируется.

pSockAddr : **DWORD**

Указатель на переменную типа SOCKADDR, в которую записан адрес узла, являющегося получателем данных.

diSockAddrSize : **DINT**

Размер структуры SOCKADDR (может быть получен оператором SIZEOF). Для сокетов TCP соединений (тип SOCKET_STREAM) аргументы **pSockAddr** и **pSockAddr** игнорируются. Вызов **FwSysSockSendTo** эквивалентен вызову **FwSysSockSend**.

Возвращаемый результат:

Возвращает число записанных байт, которое может быть меньше переданного числа байт, или SOCKET_INVALID в случае ошибки. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.2.15. FwSysSockShutdown

Функция запрещает операции передачи и/или чтения данных для сокета.

```
FUNCTION FwSysSockShutdown : BOOL
VAR_INPUT
    diSocket:DINT;
    diHow:DINT;
END_VAR
;
END_FUNCTION
```

Входные параметры:

diSocket:DINT

Системный идентификатор сокета созданного функцией **FwSysSockCreate**.

diHow:DINT

Тип запрещаемых действий. Допустимые значения: SOCKET_SD_RECEIVE, SOCKET_SD_SEND, SOCKET_SD_BOTH.

Имя	Значение
SOCKET_SD_RECEIVE	Запрещает последующие вызовы операций чтения данных.
SOCKET_SD_SEND	Запрещает последующие вызовы операций передачи данных.
SOCKET_SD_BOTH	Запрещает последующие вызовы операций чтения и передачи данных.

Возвращаемый результат:

Возвращает TRUE в случае успеха, иначе FALSE. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.2.16. FwSysSockHtons

Функция конвертирует переменную типа WORD в соответствии с порядком байт в сетях TCP/IP.

```
FUNCTION FwSysSockHtons : WORD
VAR_INPUT
    wHost:WORD;
END_VAR
;
END_FUNCTION
```

Входные параметры:

wHost:WORD

Значение для конвертирования.

Возвращаемый результат:

Возвращает конвертированное значение.

3.5.2.17. FwSysSockHtonl

Функция конвертирует переменную типа DWORD в соответствии с порядком байт в сетях TCP/IP.

```
FUNCTION FwSysSockHtonl : DWORD
VAR_INPUT
    dwHost:DWORD;
END_VAR
;
END_FUNCTION
```

Входные параметры:

dwHost:DWORD

Значение для конвертирования.

Возвращаемый результат:

Возвращает конвертированное значение.

3.5.2.18. FwSysSockNtohs

Функция конвертирует переменную типа WORD, представленную в соответствии с порядком байт в сетях TCP/IP в порядок хоста.

```
FUNCTION FwSysSockNtohs : WORD
VAR_INPUT
    wNet:WORD;
END_VAR
;
END_FUNCTION
```

Входные параметры:

wNet:WORD

Значение для конвертирования.

Возвращаемый результат:

Возвращает конвертированное значение.

3.5.2.19. FwSysSockNtohl

Функция конвертирует переменную типа DWORD, представленную в соответствии с порядком байт в сетях TCP/IP в порядок хоста.

```
FUNCTION FwSysSockNtohl : DWORD
VAR_INPUT
    dwNet:DWORD;
END_VAR
;
END_FUNCTION
```

Входные параметры:

dwNet:DWORD

Значение для конвертирования.

Возвращаемый результат:

Возвращает конвертированное значение.

3.5.2.20. FwSysSockGetLastError

Функция возвращает код ошибки последней операции.

```
FUNCTION FwSysSockGetLastError : INT
VAR_INPUT
END_VAR
;
END_FUNCTION
```

3.5.2.21. FwSysSockGetHostName

Функция получает имя хоста.

```
FUNCTION FwSysSockGetHostName : BOOL
VAR_INPUT
    stHostName:POINTER TO STRING;
    diNameLength:DINT;
END_VAR
;
END_FUNCTION
```

Входные параметры:

stHostName:POINTER TO STRING

Указатель на переменную типа STRING, в которую будет записано имя хоста.

diNameLength:DINT

Размер буфера строки.

Возвращаемый результат:

Возвращает TRUE в случае успеха, иначе FALSE. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.2.22. FwSysSockInetAddr

Функция конвертирует строку, содержащую интернет адрес, в адрес, используемый в структуре INADDR.

```
FUNCTION FwSysSockInetAddr : DWORD
VAR_INPUT
    stIPAddr: STRING;
END_VAR
;
END_FUNCTION
```

Входные параметры:

stIPAddr: STRING
IP адрес.

Возвращаемый результат:

Возвращает конвертированное значение.

3.5.2.23. FwSysSockInetNtoa

Функция конвертирует сетевой интернет адрес в строку стандартного формата.

```
FUNCTION FwSysSockInetNtoa : BOOL
VAR_INPUT
    pInAddr: DWORD;
    stIPAddr: STRING;
    diIPAddrSize: DINT;
END_VAR
;
END_FUNCTION
```

Входные параметры:

pInAddr: DWORD
Указатель на структуру INADDR, содержащую Интернет адрес.
stIPAddr: STRING
Указатель на строку, в которую будет записано конвертированное значение.
diIPAddrSize: DINT
Размер буфера строки.

Возвращаемый результат:

Возвращает TRUE в случае успеха, иначе FALSE. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.2.24. FwSysSockSetIPAddress

Функция устанавливает IP адрес заданной сетевой платы.

```
FUNCTION FwSysSockSetIPAddress : BOOL
VAR_INPUT
    stCardName: STRING;
    stIPAddress: STRING;
    stMask: STRING;
END_VAR
;
END_FUNCTION
```

Входные параметры:

stCardName: STRING
Имя сетевой платы: "Ethernet1" или "Ethernet2".
stIPAddress: STRING
IP адрес.
stMask: STRING
Маска подсети.

Возвращаемый результат:

Возвращает TRUE в случае успеха, иначе FALSE. Код ошибки может быть получен вызовом функции **FwSysSockGetLastError**.

3.5.3. Описание типов данных

3.5.3.1. INADDR

```
TYPE INADDR :
STRUCT
  S_addr:DWORD;
END_STRUCT
END_TYPE
```

Поля структуры:

```
S_addr:DWORD;
```

Интернет адрес.

3.5.3.2. SOCKADDRESS

Тип данных SOCKADDRESS используется для представления информации об адресе IP соединения.

```
TYPE SOCKADDRESS :
STRUCT
  sin_family:INT;
  sin_port:UINT;
  sin_addr:UDINT;
  sin_zero:ARRAY [0..7] OF SINT;
END_STRUCT
END_TYPE
```

Поля структуры:

```
sin_family:INT;
```

Идентификатор формата адреса. Единственное возможное значение при работе с функциями библиотеки – SOCKET_AF_INET.

```
sin_port:UINT;
```

Порт. Значение поля отображается в соответствии с порядком байт в сетях TCP/IP.

```
sin_addr:UDINT;
```

IP-адрес. Значение поля отображается в соответствии с порядком байт в сетях TCP/IP.

```
sin_zero:ARRAY [0..7] OF SINT;
```

Буфер.

3.5.3.3. SOCKET_FD_SET

Тип данных SOCKET_FD_SET описывает некоторое множество (набор) сокетов. Используется с функцией **FwSysSockSelect** с целью проверки сокетов набора на готовность к определенным коммуникационным действиям.

```
TYPE SOCKET_FD_SET :
STRUCT
  fd_count:UDINT;
  fd_array:ARRAY [0..MAX_SOCKET_FD_SETSIZE] OF DINT;
END_STRUCT
END_TYPE
```

Поля структуры:

```
fd_count:UDINT
```

Число сокетов в наборе.

```
fd_array:ARRAY [0..MAX_SOCKET_FD_SETSIZE] OF DINT
```

Массив идентификаторов сокетов. Идентификаторы находятся в первых **fd_count** ячейках массива.

3.5.4. Коды ошибок

В списке приведены коды ошибок возвращаемые функцией **FwSysSockGetLastError**. Ошибки перечислены в порядке возрастания их числового значения:

Имя / значение	Описание
SOCKET_EIO 5	<i>Ошибка ввода/вывода.</i> Ошибка инициализации библиотеки или другая ошибка сетевого уровня.
SOCKET_EBADF 9	<i>Неверное значение идентификатора.</i> В качестве идентификатора сокета передано неверное или неизвестное значение.
SOCKET_EWOULDBLOCK 11	<i>Ресурс временно не доступен.</i> Поскольку библиотека поддерживает только неблокирующие сокет, ошибка возникает на операциях, которые не могут быть завершены без ожидания. Например, при вызове

	функции FwSysSockRecv в случае отсутствия принятых данных. Или при вызове FwSysSockConnect для установки TCP соединения, т.к. требуется время для выполнения фактического соединения с удаленным узлом.
SOCKET_ENOMEM 12	<i>Не хватает памяти.</i> Операция на сокете не была выполнена из-за недостаточного размера буфера памяти, или переполнения очереди заданий.
SOCKET_EFAULT 14	<i>Неверное значение адреса.</i> Неправильное значения указателя на объекта или неправильное значение размера объекта при передаче объекта в параметрах вызова.
SOCKET_EINVAL 22	<i>Недопустимое значение аргумента.</i> Неправильное или недопустимое значение параметра при вызове. Например, указан недостаточный размер буфера объекта при записи/чтении опции сокета в вызове FwSysSockSetOption/FwSysSockGetOption. Или недопустимое использование вызова. Например, вызов функции FwSysSockAccept для сокета, не находящегося в состоянии listen.
SOCKET_ENFILE 23	<i>Нет свободных дескрипторов сокетов.</i> Закончилось зарезервированное число сокетов (для CPM713 - 64).
SOCKET_ENOSYS 38	<i>Операция не поддерживается.</i> Функция, команда или опция управления не поддерживается реализацией библиотеки для данного контроллера.
SOCKET_ENOPROTOOPT 92	<i>Неверная опция протокола.</i> Неизвестная или неподдерживаемая опция протокола при вызове FwSysSockSetOption, FwSysSockGetOption.
SOCKET_EAFNOSUPPORT 97	<i>Формат адреса не поддерживается.</i> Единственное допустимое значение для сокетов библиотеки - SOCKET_AF_INET.
SOCKET_EADDRINUSE 98	<i>Адрес используется.</i> Например, при попытке связать сокет с локальным адресом (значением адрес/порт) которое уже используется на другом открытом сокете.
SOCKET_EADDRNOTAVAIL 99	<i>Недопустимое значение адреса.</i> Например, при попытке связать сокет с несуществующим локальным адресом.
SOCKET_ECONNABORTED 103	<i>Соединение разорвано.</i> Установленное соединение было разорвано, возможно, из-за ошибок или истечения таймаутов протокола
SOCKET_ECONNRESET 104	<i>Соединение разорвано удаленным узлом.</i> Установленное соединение было разорвано по инициативе удаленного узла.
SOCKET_ENOBUFS 105	<i>Нет свободных дескрипторов соединений.</i>
SOCKET_EISCONN 106	<i>Соединение уже установлено.</i> Запрос операции, требующей соединения на сокете с уже установленным соединением.
SOCKET_ENOTCONN 107	<i>Соединение не установлено.</i> Вызов функций чтения/передачи данных для сокета с неустановленным соединением.
SOCKET_ESHUTDOWN 108	<i>Операция отключена.</i> Попытка чтения/передачи данных для сокета с выключенной функцией приема/передачи.
SOCKET_EALREADY 114	<i>Операция выполняется.</i>
SOCKET_EINPROGRESS 115	<i>Предыдущая операция с сокетом не завершена.</i> Выполняется операция с сокетом, требующая блокировки. Данная ошибка также может проявляться в случае конкурентного доступа к сокету из разных задач, что не рекомендуется.

4. НАСТРОЙКА КОММУНИКАЦИОННЫХ СРЕДСТВ ВЕРХНЕГО УРОВНЯ

4.1. Настройка коммуникационного драйвера сервера CoDeSys Gateway Server

4.1.1. Общие сведения

CoDeSys Gateway Server предназначен для организации информационного обмена между средой разработки CoDeSys, функционирующей на компьютере, и средой исполнения CoDeSys на удаленном контроллере через интерфейс внешней сети, поддерживаемый контроллером.

Поддержка той или иной сети реализуется при помощи коммуникационных драйверов, с которыми взаимодействует Gateway Server, как показано на рис. 12.

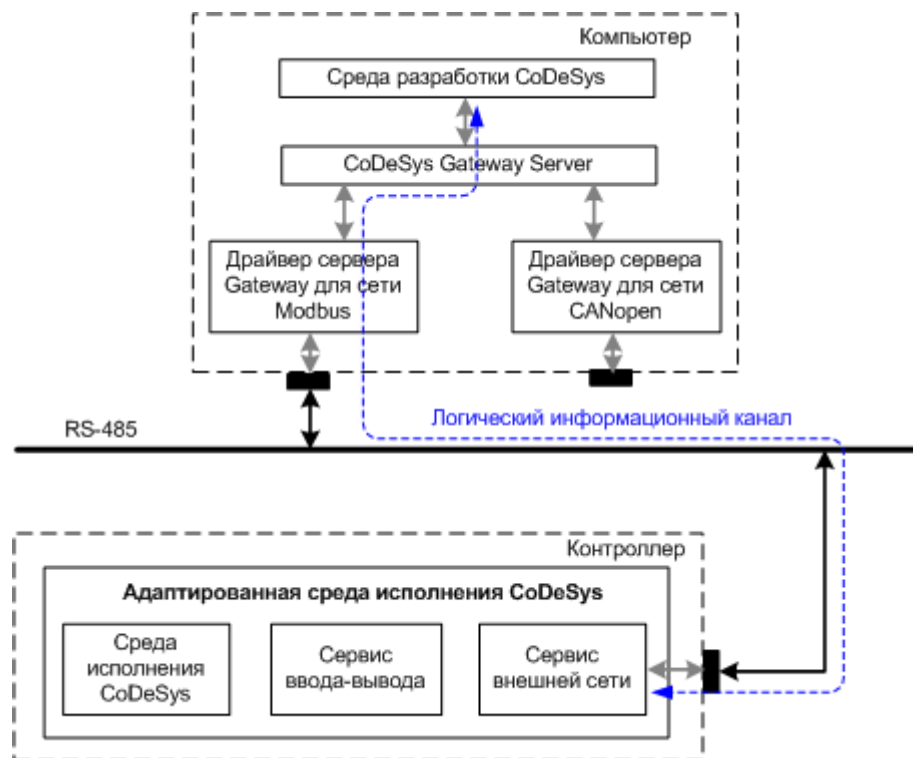


Рис. 12. Архитектура информационного обмена между средой разработки и средой исполнения CoDeSys

Коммуникационный драйвер GDrvFastwel.dll имеет встроенную поддержку протоколов MODBUS over Serial Line и MODBUS TCP.

При установлении логического информационного канала между средой разработки и удаленным контроллером по сети MODBUS используется интерфейс инкапсуляции транспорта протокола MODBUS (Encapsulated Interface Transport, код функции 2Bh, тип 80h).

Настоящий подраздел содержит указания по настройке параметров коммуникационного драйвера GDrvFastwel.dll, поставляемого в комплекте с адаптированной средой CoDeSys для Fastwel I/O System.

4.1.2. Установка коммуникационного драйвера CoDeSys Gateway Server

Коммуникационный драйвер CoDeSys Gateway Server включает в себя два компонента (GDrvFastwel.dll и modbusDLL.dll), автоматически устанавливаемые в подкаталог \System32 каталога установки Windows в процессе установки пакета адаптации среды CoDeSys для Fastwel I/O.

4.1.3. Создание логического информационного канала между средой разработки и контроллером

Логический информационный канал между средой разработки и средой исполнения CoDeSys на удаленном контроллере служит для выполнения операций по сети, инициируемых из меню **Online** среды разработки CoDeSys.

Для создания информационного канала выполните следующие действия:

1. Запустите среду разработки CoDeSys
2. Выберите команду меню **Online–Communication Parameters...** На экран будет выведена диалоговая панель **Communication Parameters**, показанная на рис. 13

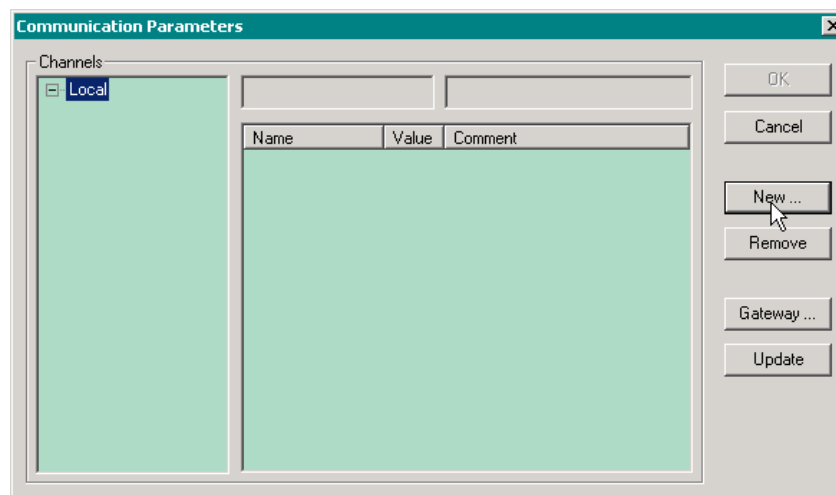


Рис. 13. Диалоговая панель параметров коммуникационного сервера связи с удаленным контроллером

3. Для создания логического информационного канала нажмите кнопку **New** и в появившейся диалоговой панели введите имя создаваемого канала, а в списке **Device** выберите строку *ModbusTCP: Fastwel Modbus TCP driver*, как показано на рис. 14 и закройте диалоговую панель нажатием кнопки **OK**. В древовидном списке **Channels** диалоговой панели **Communication Parameters** появится элемент, соответствующий созданному каналу, а в таблице параметров канала справа – параметры созданного канала, как показано на рис. 15.

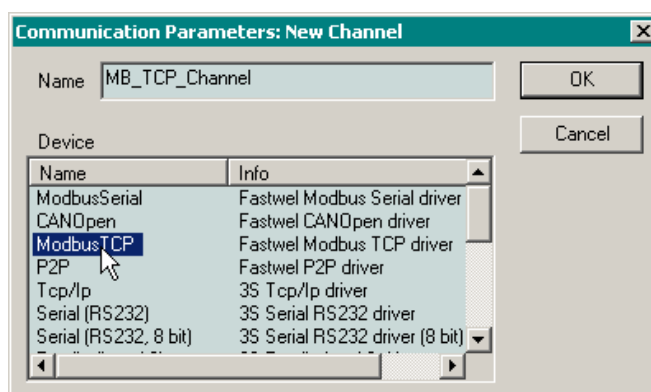


Рис. 14. Создание канала с использованием драйвера Fastwel Modbus driver

4. Дважды щелкните на значении параметра *Address* и введите значение IP-адреса подчиненного узла, с которым предполагается установить связь.
5. Закройте диалоговую панель **Communication Parameters** нажатием кнопки **OK**.

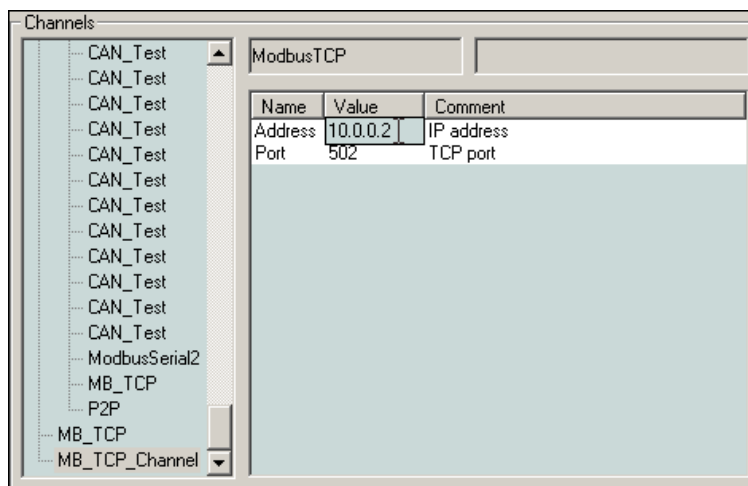


Рис. 15. Настройка параметров канала

4.2. Создание логического информационного канала между средой разработки и контроллером CPM713 по последовательному каналу связи

Для создания информационного канала выполните следующие действия:

1. Соедините последовательный порт ПК с портом консоли контроллера, расположенным под пластмассовой защитной крышкой на передней панели контроллера, при помощи кабеля соединительного ACS00019
2. Запустите среду разработки CoDeSys
3. Выберите команду меню **Online–Communication Parameters...** На экран будет выведена диалоговая панель **Communication Parameters**, показанная на рис. 13.
4. Для создания логического информационного канала через последовательный порт нажмите кнопку **New** и в появившейся диалоговой панели введите имя создаваемого канала, а в списке **Device** выберите строку **P2P: Fastwel P2P driver**, как показано на рис. 16, введите имя канала в поле **Name** и закройте диалоговую панель нажатием кнопки **OK**. В древовидном списке **Channels** диалоговой панели **Communication Parameters** появится элемент, соответствующий созданному каналу, а в таблице параметров канала справа – параметры созданного канала, как показано на рис. 17.

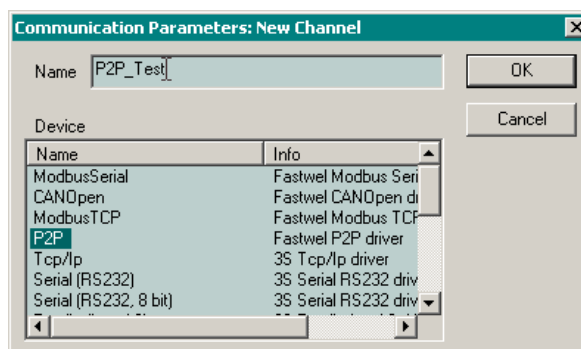


Рис. 16. Создание канала с использованием драйвера Fastwel P2P driver

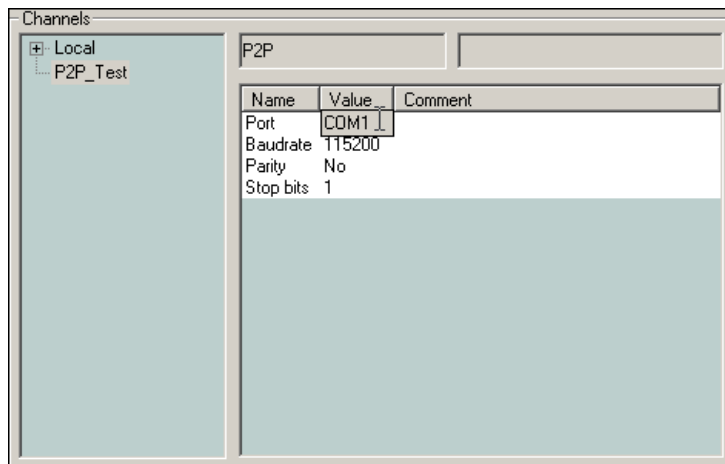


Рис. 17. Настройка параметров канала

5. Если для связи с контроллером используется последовательный порт, отличный от COM1, дважды щелкните левой кнопкой мыши над именем *COM1* в таблице параметров и клавишами ↑ ("стрелка вверх") или ↓ ("стрелка вниз") выберите требуемый последовательный порт компьютера, через который будет осуществляться взаимодействие с контроллером в режиме "точка-точка", и нажмите клавишу Enter.
6. Закройте диалоговую панель **Communication Parameters** нажатием кнопки **OK**

4.3. Дополнительные замечания

ВНИМАНИЕ!

Иногда после копирования файла проекта CoDeSys с расширением *.pro с одного компьютера на другой или после удаления вспомогательных файлов проекта при попытке выполнить команду **Online–Login** среда разработки CoDeSys по неизвестной причине выводит на экран монитора сообщение, показанное на рис. 18.

Для решения данной проблемы нажмите кнопку **Gateway...** в диалоговой панели **Communication Parameters**, после чего в появившейся диалоговой панели **Communication Parameters: Gateway** установите опцию **Connection : Local**, затем нажмите **OK**, и всё будет в порядке.



Рис. 18. Сообщение о невозможности установить соединение с удаленным сервером Gateway

ПРИЛОЖЕНИЕ А . ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Версия	Дата	Ссылка	Статус	Примечания
2.52.23926	15.03.2011	Документ	создан	
2.61.23940	25.06.2013	Документ	изменен	Устранены дефекты оформления