2017

СПК

Настройка обмена по протоколу ОВЕН

Руководство для начинающих пользователей

Версия: 1.2
Дата: 21.04.2017

Оглавление

1. Цель и структура документа

Этот документ представляет собой руководство по настройке обмена данными с использованием протокола **Овен** для панельных контроллеров Овен СПК в среде **CODESYS V3.5**. Подразумевается, что читатель обладает базовыми навыками работы с **CODESYS** и **СПК**, поэтому общие вопросы (например, создание и загрузка проектов) в данном документе не рассматриваются; они подробно описаны в документах **СПК. Первый старт** и **СПК. FAQ**, которые доступны на сайте Овен в разделе **CODESYS V3/Документация**.

Протокол **Овен** поддерживается такими устройствами, как ТРМ, СИ, модули Mx110 и др.

Работа с протоколом в **CODESYS** реализована в библиотеке **OwenNet**.

Документ содержит описание библиотеки, а также пример ее использования для опроса **ТРМ212** (на языках **CFC** и **ST**).

2. Основные сведения о протоколе Овен

Протокол **Овен** основан на архитектуре **Master-Slave** (ведущий-ведомый) и реализуется поверх последовательного интерфейса RS-485.

Спецификация протокола доступна на сайте компании Овен в разделе **Документация/Сетевые протоколы обмена по RS-485**. Начинающим пользователям также рекомендуется ознакомиться со статьей **Протокол ОВЕН для чайников**, опубликованной в журнале **АиП** №29 (2007/1), который доступен на сайте в разделе **Пресс-центр/Журнал АиП.**

Ниже перечислены ключевые моменты, которые необходимы для понимания последующего текста:

1. Сеть имеет единственное **ведущее (master)** устройство, инициирующее процесс обмена. В примере (п. 4) таким устройством будет являться **СПК**. Все остальные устройства сети являются **ведомыми (slave).**

2. Каждое slave-устройство занимает в сети число адресов, равное количеству его каналов. Например, **ТРМ200** имеет два канала измерения температуры, и, соответственно, занимает в сети два адреса, а восьмиканальный **ТРМ138** – восемь адресов. В настройках прибора устанавливается его **базовый адрес**, который соответствует адресу первого канала. Иными словами, **ТРМ200** с базовым адресом 10 займет в сети адреса 10 и 11, причем адрес 10 будет соответствовать его первому каналу измерения, а 11 – второму.

3. Slave-устройство имеет два типа параметров: **оперативные** и **конфигурационные**.

Каждый параметр имеет уникальное символьное имя (например, **PV**, **t.dn** и т.п.).

Оперативные параметры содержат информацию о текущем состоянии прибора и объекта регулирования. К ним относятся измеренные и вычисленные значения, выходные мощности регуляторов, номера запущенных в данный момент программ и т.д.

Поскольку обычно каждый канал прибора имеет набор одинаковых параметров, то при чтении/записи оперативных параметров необходимо указывать **адрес** соответствующего канала (см. пп. 2)

Конфигурационные параметры содержат информацию о настройках прибора. При наличии нескольких однотипных конфигурационных параметров (например, к таким относятся коэффициенты ПИД-регуляторов многоканального прибора) для обращения к ним необходимо указывать **линейный индекс параметра** (0, 1, 2 и т.д.).

4. Каждый параметр имеет свой тип (например, значение с плавающей точкой) и формат (например, укороченное значение с плавающей точкой, которое занимает 24 бита).

3. Описание библиотеки OwenNet

3.1. Установка библиотеки

Библиотека **OwenNet** доступна на диске с ПО, входящем в комплект поставки, а также на сайте компании ОВЕН в разделе **CODESYS V3/Библиотеки**. Она распространяется как отдельно, так и в составе пакета библиотек Овен **LibInstall.**

Для установки пакета в **CODESYS** в меню **Инструменты** выберите пункт **Менеджер пакетов**, после чего укажите путь к файлу пакета и нажмите **Установить**:

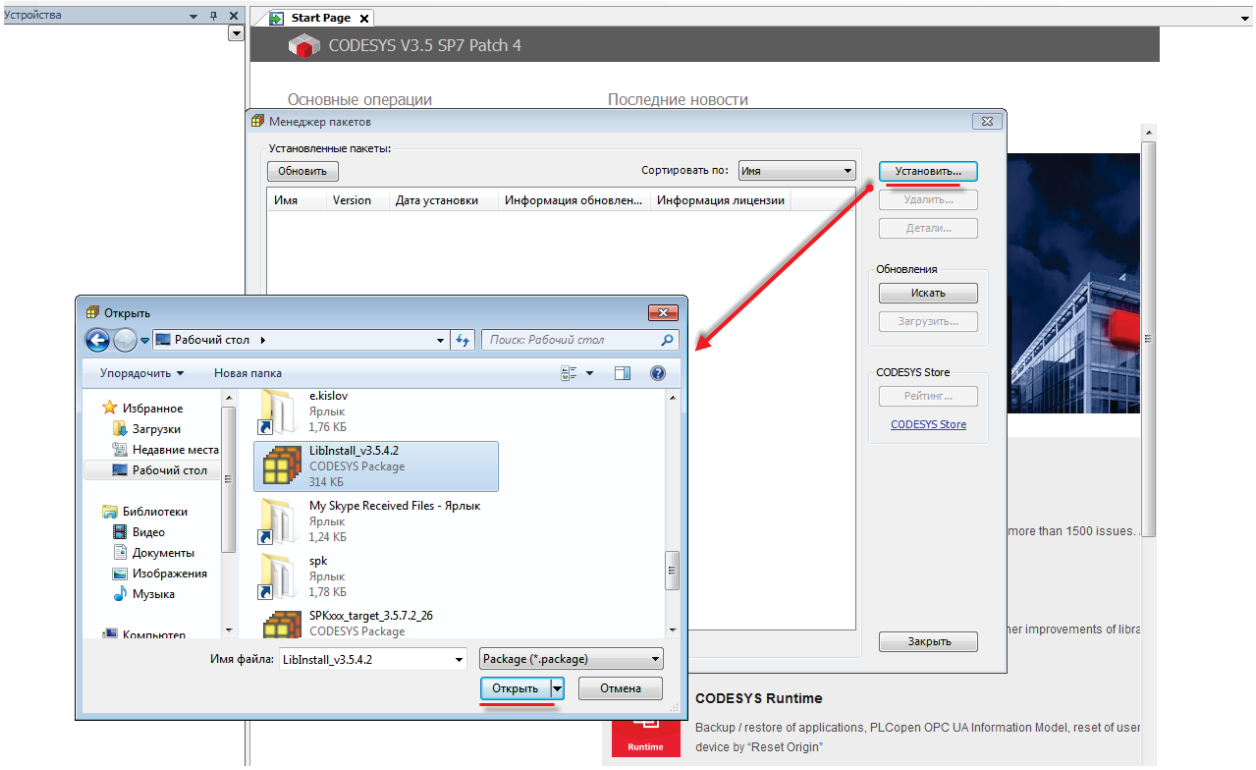

Рис. 3.1. Установка пакета библиотек Овен в среду **CODESYS**

В появившемся диалоговом окне выберите пункт **Полная установка**, после чего нажмите кнопку **Next**:

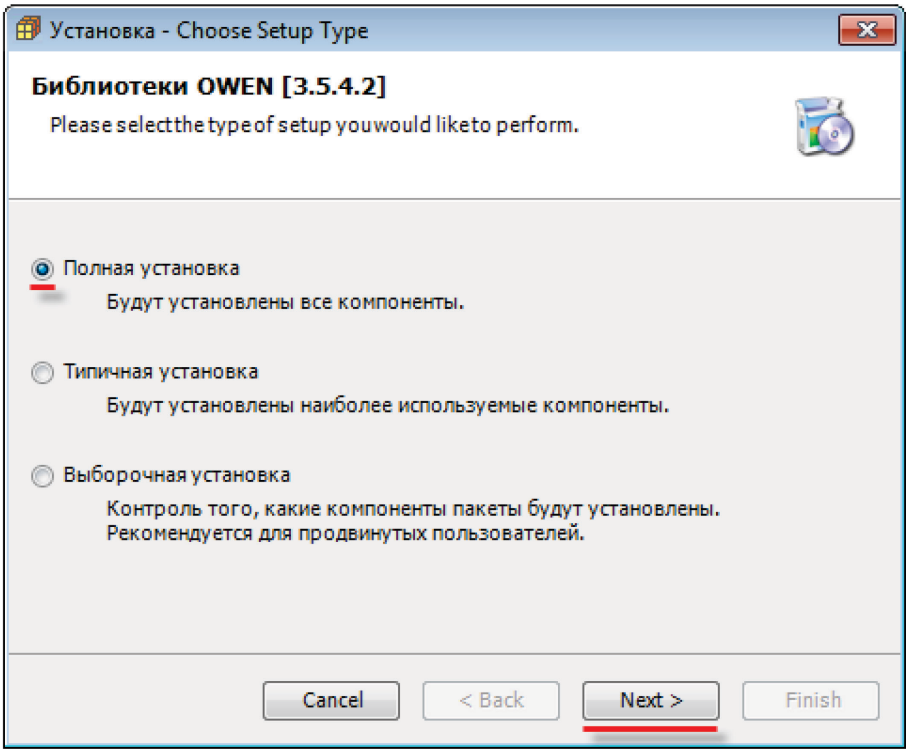

Рис. 3.2. Начало установки пакета библиотек

После завершения установки закройте диалоговое окно с помощью кнопки **Finish**:

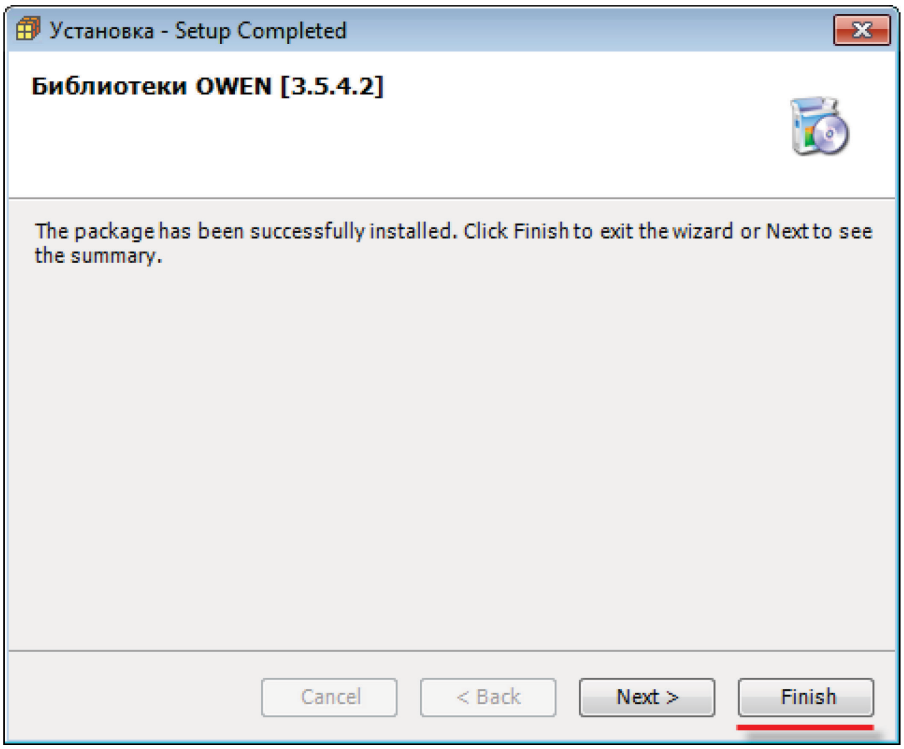

Рис. 3.3. Завершение установки пакета библиотек

3.2. Добавление библиотеки в проект CODESYS

Для добавления библиотеки **OwenNet** в проект **CODESYS**, в **Менеджере библиотек** нажмите кнопку **Добавить библиотеку** и в строке поиска введите **owennet**, после чего выберите из списка нужную библиотеку и нажмите **ОК**.

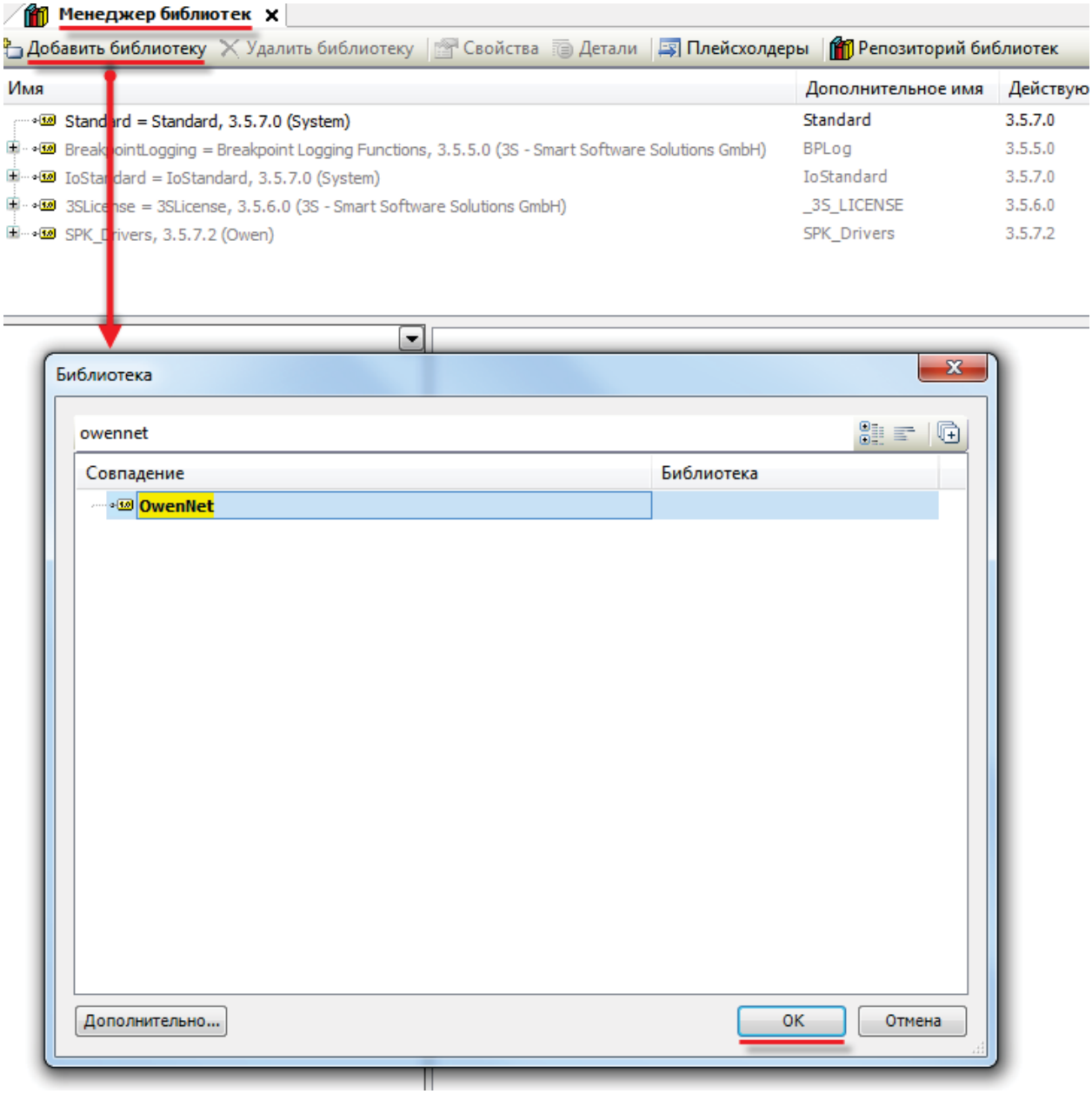

Рис. 3.4. Добавление библиотеки **OwenNet**

В библиотеку **OwenNet** не входит ФБ открытия COM-порта, поэтому для этой цели необходимо воспользоваться другой библиотекой. Рекомендуется использовать библиотеку **ComService**.

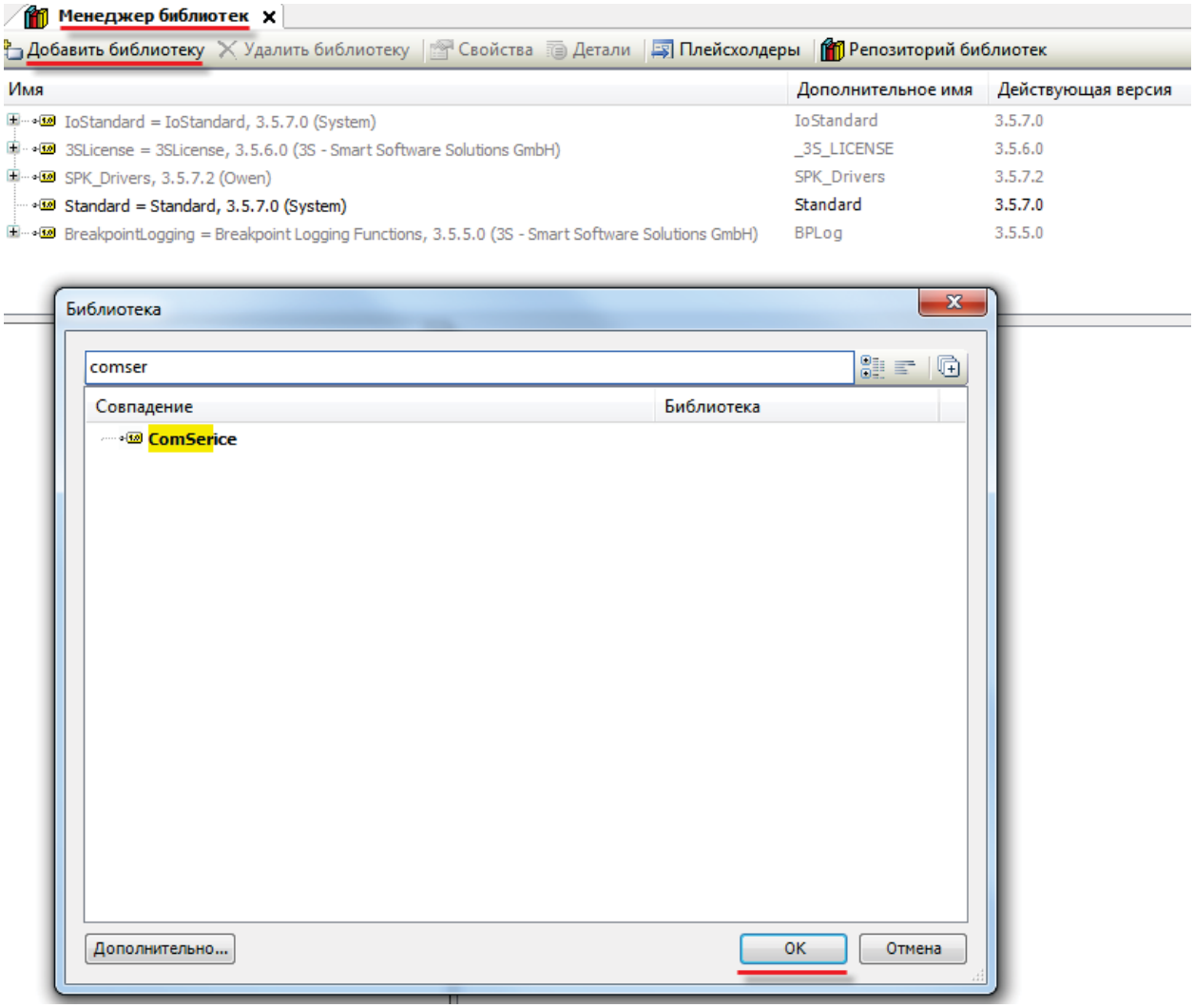

Рис. 3.5. Добавление библиотеки **ComService**

После добавления библиотеки появятся в списке **Менеджера библиотек**:

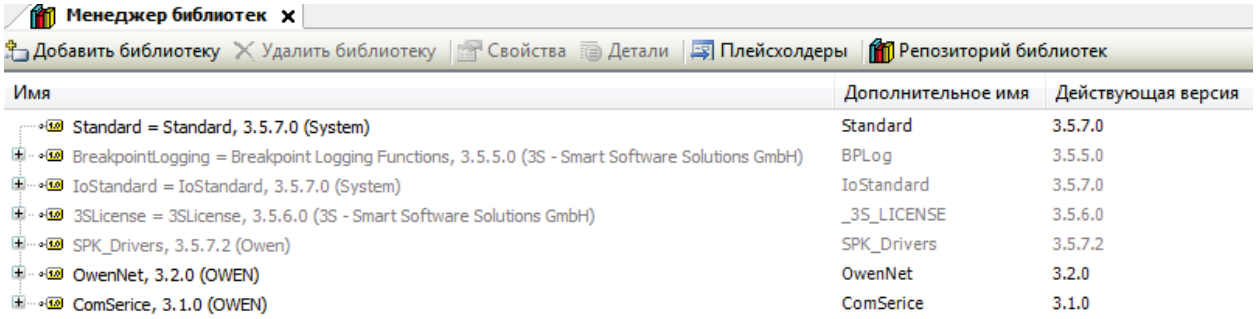

Рис. 3.6. Список библиотек проекта

3.3. Описание библиотеки

3.3.1. Блок COM_SERVICE (библиотека ComService)

Функциональный блок **COM_SERVICE**, входящий в библиотеку **ComService**, используется для открытия/закрытия COM-порта, а также его настройки.

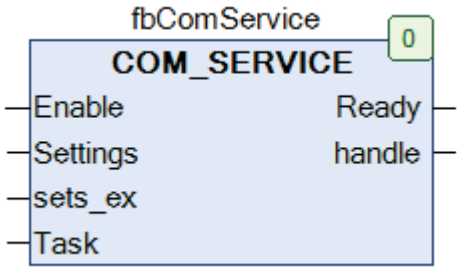

Рис. 3.7. Внешний вид ФБ **COM_SERVICE** на языке CFC

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная действия с портом. По ее переднему фронту с COM-портом номер **sPort** совершается действие, определяемое переменной **Task**.
Settings	SysCom.Com_Settings	Структура настроек COM-порта.
sPort	SysCom.COM_Ports	Номер порта СПК.
byStopBits	SysCom.COM_StopBits	Количество стоп-бит. Возможные значения: **1** – 1 бит; **2** – 1.5 бит; **3** – 2 бит.
byParity	SysCom.COM_Parity	Режим контроля паритета. Возможные значения: **0** – отсутствие контроля паритета (**NONE**); **1** – четный (**EVEN**); **2** – нечетный (**ODD**).
uiBaudrate	SysCom.COM_Baudrate	Скорость обмена, бод. Возможные значения: **4800/9600/19200/38400/57600/115200**.
uiTimeout	SysCom.COM_Timeout	Время ожидания следующего символа. В СПК параметр не используется.
uiBufferSize	UDINT	Размер FIFO буфера COM-порта. В СПК параметр не используется.
sets_ex	SysCom.Com_SettingsEx	Структура расширенных настроек COM-порта. Описание всех переменных доступно в документации библиотеки **SysCom**.
byByteSize	BYTE	Количество информационных бит в передаваемых/принимаемых байтах. Обычно выбирается значение **8** для **Modbus RTU** и **7** для **Modbus ASCII**.

Task	COM_TSK	Действие, совершаемое с портом. Возможные значения: **OPEN_TSK** – открытие порта; **RESET_TSK** – перезапуск порта; **CLOSE_TSK** – закрытие порта.
Выходные переменные		
Ready	BOOL	Переменная состояния порта. Принимает значение **TRUE**, если порт открыт. Принимает значение **FALSE**, если порт закрыт.
Handle	SysCom.RTS_IEC_HANDLE	Идентификатор открытого порта, используется для обращения к ФБ опроса модулей.

3.3.2. Блок OWEN_GET_DINT

Функциональный блок **OWEN_GET_DINT** используется для считывания целочисленного значения.

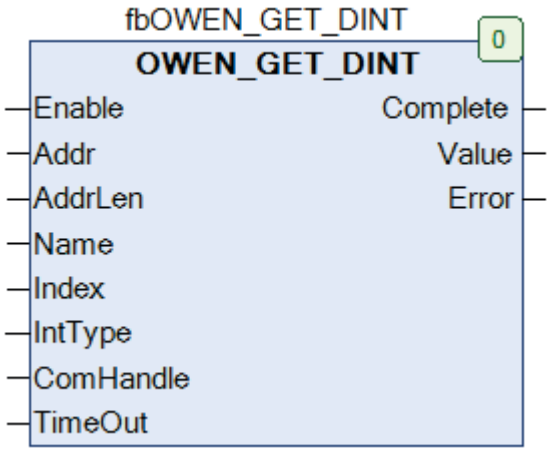

Рис. 3.8. Внешний вид ФБ **OWEN_GET_DINT** на языке CFC

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему.
Addr	WORD	Адрес slave-устройства (для многоканального устройства – адрес канала).
AddrLen	ADR_LEN	Длина сетевого адреса slave-устройства. Возможные значения: **8** – бит; **11** – 11 бит.

Name	STRING(9)	Имя считываемого параметра.
Index	WORD	Линейный индекс опрашиваемого параметра. Если индекс не используется, то выставляется значение **0xFFFF**.
IntType	INT_FRM	Формат считываемого значения. Список доступных форматов приведен в перечислении INT_FRM.
ComHandle	DWORD	Идентификатор порта, поступающий с выхода блока ComService (или аналогичного).
TimeOut	TIME	Таймаут ответа slave-устройства. Если в течение этого времени устройство не отвечает, то СПК переходит к опросу следующего slave-устройства. Рекомендуемое значение – 50 мс.
Выходные переменные		
Complete	BOOL	Флаг успешного опроса slave-устройства. Принимает значение **TRUE** после успешного завершения опроса, на следующем цикле сбрасывается в **FALSE**. При наличии ошибок обмена выход не принимает значение **TRUE**, а на выходе **Error** подается код ошибки.
Value	DINT	Считанное значение.
Error	WORD	Код ошибки. Значение **0** соответствует отсутствию ошибок. Код ошибки **16#FFFF (255)** характеризует отсутствие ответа от слэйва по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А. Список кодов ошибок прибора приведен в РЭ на конкретный прибор.

3.3.3. Блок OWEN_SET_DINT

Функциональный блок **OWEN_SET_DINT** используется для записи целочисленного значения.

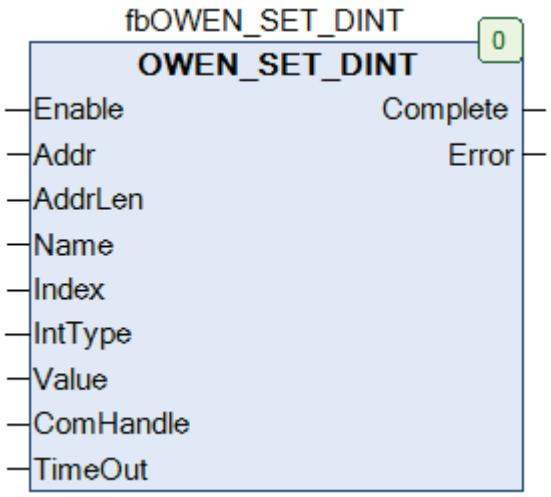

Рис. 3.9. Внешний вид ФБ **OWEN_SET_DINT** на языке CFC

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему.
Addr	WORD	Адрес slave-устройства (для многоканального устройства – адрес канала).
AddrLen	ADR_LEN	Длина сетевого адреса slave-устройства. Возможные значения: **8** – бит; **11** – 11 бит.
Name	STRING(9)	Имя записываемого параметра.
Index	WORD	Линейный индекс записываемого параметра. Если индекс не используется, то выставляется значение **0xFFFF**.
IntType	INT_FRM	Формат считываемого значения. Список доступных форматов приведен в перечислении INT_FRM.
Value	DINT	Записываемое значение.
ComHandle	DWORD	Идентификатор порта, поступающий с выхода блока ComService (или аналогичного).
TimeOut	TIME	Таймаут ответа slave-устройства. Если в течение этого времени устройство не отвечает, то СПК переходит к опросу следующего slave-устройства. Рекомендуемое значение – 50 мс.

Выходные переменные		
Complete	BOOL	Флаг успешного опроса slave-устройства. Принимает значение **TRUE** после успешного завершения опроса, на следующем цикле сбрасывается в **FALSE**. При наличии ошибок обмена выход не принимает значение **TRUE**, а на выходе **Error** подается код ошибки.
Error	WORD	Код ошибки. Значение **0** соответствует отсутствию ошибок. Код ошибки **16#FFFF (255)** характеризует отсутствие ответа от слэйва по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А. Список кодов ошибок прибора приведен в РЭ на конкретный прибор.

3.3.4. Блок OWEN_GET_REAL

Функциональный блок **OWEN_GET_REAL** используется для считывания значения с плавающей точкой.

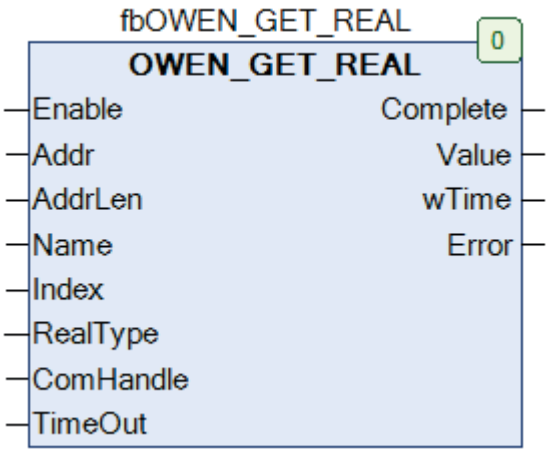

Рис. 3.9. Внешний вид ФБ **OWEN_GET_REAL** на языке CFC

Имя переменной	**Тип**	**Описание**
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему.
Addr	WORD	Адрес slave-устройства (для многоканального устройства – адрес канала).
AddrLen	ADR_LEN	Длина сетевого адреса slave-устройства. Возможные значения: **8** – бит; **11** – 11 бит.

Name	STRING(9)	Имя считываемого параметра.
Index	WORD	Линейный индекс опрашиваемого параметра. Если индекс не используется, то выставляется значение **0xFFFF**.
RealType	REAL_FRM	Формат считываемого значения. Список доступных форматов приведен в перечислении REAL_FRM.
ComHandle	DWORD	Идентификатор порта, поступающий с выхода блока ComService (или аналогичного).
TimeOut	TIME	Таймаут ответа slave-устройства. Если в течение этого времени устройство не отвечает, то СПК переходит к опросу следующего slave-устройства. Рекомендуемое значение – 50 мс.
Выходные переменные		
Complete	BOOL	Флаг успешного опроса slave-устройства. Принимает значение **TRUE** после успешного завершения опроса, на следующем цикле сбрасывается в **FALSE**. При наличии ошибок обмена выход не принимает значение **TRUE**, а на выходе **Error** подается код ошибки.
Value	REAL	Считанное значение.
wTime	WORD	Значение относительного времени измерения в мс (для формата **FLOAT32T**).
Error	WORD	Код ошибки. Значение **0** соответствует отсутствию ошибок. Код ошибки **16#FFFF (255)** характеризует отсутствие ответа от слэйва по истечению таймаута опроса. Список кодов остальных ошибок приведен в приложении А.

3.3.5. Блок OWEN_SET_REAL

Функциональный блок **OWEN_SET_REAL** используется для записи значения с плавающей точкой.

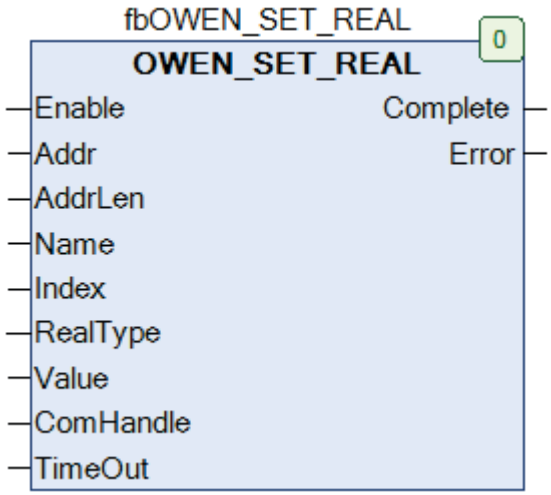

Рис. 3.10. Внешний вид ФБ **OWEN_SET_REAL** на языке CFC

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему.
Addr	WORD	Адрес slave-устройства (для многоканального устройства – адрес канала).
AddrLen	ADR_LEN	Длина сетевого адреса slave-устройства. Возможные значения: **8** – бит; **11** – 11 бит.
Name	STRING(9)	Имя записываемого параметра.
Index	WORD	Линейный индекс записываемого параметра. Если индекс не используется, то выставляется значение **0xFFFF**.
RealType	REAL_FRM	Формат считываемого значения. Список доступных форматов приведен в перечислении REAL_FRM.
Value	REAL	Записываемое значение.
ComHandle	DWORD	Идентификатор порта, поступающий с выхода блока ComService (или аналогичного).
TimeOut	TIME	Таймаут ответа slave-устройства. Если в течение этого времени устройство не отвечает, то СПК переходит к опросу следующего slave-устройства. Рекомендуемое значение – 50 мс.

Выходные переменные		
Complete	BOOL	Флаг успешного опроса slave-устройства. Принимает значение **TRUE** после успешного завершения опроса, на следующем цикле сбрасывается в **FALSE**. При наличии ошибок обмена выход не принимает значение **TRUE**, а на выходе **Error** подается код ошибки.
Error	WORD	Код ошибки. Значение **0** соответствует отсутствию ошибок. Код ошибки **16#FFFF (255)** характеризует отсутствие ответа от слэйва по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А. Список кодов ошибок прибора приведен в РЭ на конкретный прибор.

3.3.6. Блок OWEN_UNI_IO

Функциональный блок **OWEN_UNI_IO** используется для чтения или записи нетипизированных данных.

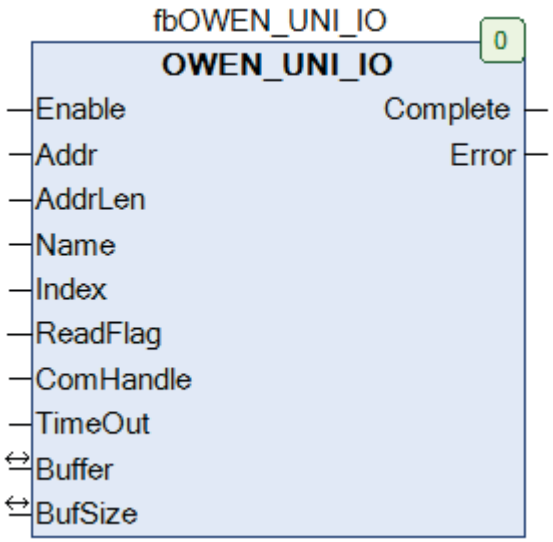

Рис. 3.11. Внешний вид ФБ **OWEN_UNI_IO** на языке CFC

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему.
Addr	WORD	Адрес slave-устройства (для многоканального устройства – адрес канала).

AddrLen	ADR_LEN	Длина сетевого адреса slave-устройства. Возможные значения: **8** – бит; **11** – 11 бит.
Name	STRING(9)	Имя считываемого/записываемого параметра.
Index	WORD	Линейный индекс опрашиваемого параметра. Если индекс не используется, то выставляется значение **0xFFFF**.
ReadFlag	BOOL	Режим работы блока (тип доступа к параметру): **FALSE** – запись; **TRUE** – чтение.
ComHandle	DWORD	Идентификатор порта, поступающий с выхода блока ComService (или аналогичного).
TimeOut	TIME	Таймаут ответа slave-устройства. Если в течение этого времени устройство не отвечает, то СПК переходит к опросу следующего slave-устройства. Рекомендуемое значение – 50 мс.
Выходные переменные		
Complete	BOOL	Флаг успешного опроса slave-устройства. Принимает значение **TRUE** после успешного завершения опроса, на следующем цикле сбрасывается в **FALSE**. При наличии ошибок обмена выход не принимает значение **TRUE**, а на выходе **Error** подается код ошибки.
Error	WORD	Код ошибки. Значение **0** соответствует отсутствию ошибок. Код ошибки **16#FFFF (255)** характеризует отсутствие ответа от слэйва по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А. Список кодов ошибок прибора приведен в РЭ на конкретный прибор.
Входные-выходные переменные (VAR_IN_OUT)		
Buffer	ARRAY [0..14] OF BYTE	Считанные данные или данные, подготовленные для записи.
BufSize	BYTE	Количество считанных/переданных байт.

3.3.7. Блок OWEN_LISTEN

Функциональный блок **OWEN_LISTEN** используется для прослушивания (сниффинга) сети. Это может потребоваться при необходимости считывать данные в сети, в которой уже находится master-устройство.

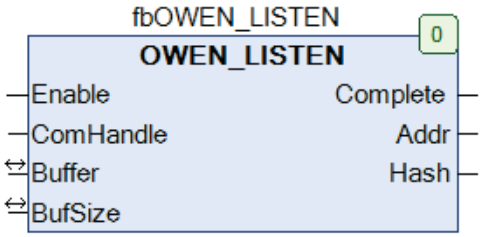

Рис. 3.12. Внешний вид ФБ **OWEN_LISTEN** на языке CFC

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему.
ComHandle	DWORD	Идентификатор порта, поступающий с выхода блока ComService (или аналогичного).
Выходные переменные		
Complete	BOOL	Флаг успешного опроса slave-устройства. Принимает значение **TRUE** после успешного завершения опроса, на следующем цикле сбрасывается в **FALSE**. При наличии ошибок обмена выход не принимает значение **TRUE**, а на выходе **Error** подается код ошибки.
Addr	WORD	Адрес slave-устройства (для многоканального устройства – адрес канала).
Hash	WORD	Хэш параметра. Информация о хэшировании приведена в спецификации протокола, доступной на сайте компании Овен в разделе **Документация/Сетевые протоколы обмена по RS-485**.
Error	WORD	Код ошибки. Значение **0** соответствует отсутствию ошибок. Код ошибки **16#FFFF (255)** характеризует отсутствие ответа от слэйва по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А. Список кодов ошибок прибора приведен в РЭ на конкретный прибор.
Входные-выходные переменные (VAR_IN_OUT)		
Buffer	ARRAY [0..14] OF BYTE	Считанные данные.
BufSize	BYTE	Количество считанных байт.

3.3.8. Блок OwenPoolMaster

Функциональный блок **OwenPoolMaster** используется для циклического выполнения набора команд (чтение/запись/прослушивание сети).

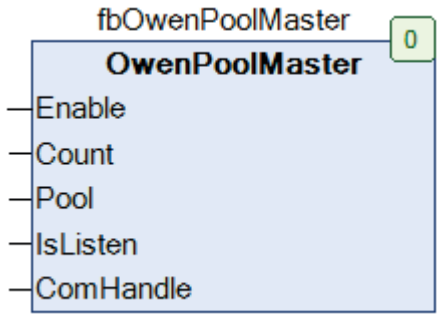

Рис. 3.13. Внешний вид ФБ **OwenPoolMaster** на языке CFC

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему.
Count	BYTE	Количество выполняемых команд.
Pool	POINTER TO OwenPool	Указатель на первый элемент массива структур типа OwenPool. Каждая структура представляет собой выполняемую команду.
IsListen	BOOL	При значении **TRUE** блок работает в режиме прослушивания сети.
ComHandle	DWORD	Идентификатор порта, поступающий с выхода блока ComService (или аналогичного).

3.3.10. Структура OwenPool

Структура **OwenPool** используется при работе с ФБ OwenPoolMaster. Каждый экземпляр структуры представляет собой команду, выполняемую блоком.

Имя переменной	Тип	Описание
Every	TIME	Период выполнения команды.
IsRead	BOOL	Режим работы блока: **FALSE** – запись; **TRUE** – чтение.
EncodeType	OWEN_FRM	Формат считываемого/записываемого значения. Список доступных форматов приведен в перечислении OWEN_FRM.
Addr	WORD	Адрес slave-устройства (для многоканального устройства – адрес канала).
AddrLen	ADR_LEN	Длина сетевого адреса slave-устройства. Возможные значения: **8** – бит; **11** – 11 бит.
Name	STRING(9)	Имя считываемого/записываемого параметра.
Index	WORD	Линейный индекс считываемого/записываемого параметра. Если индекс не используется, то выставляется значение **0xFFFF**.
Complete	BOOL	Флаг успешного опроса slave-устройства. Принимает значение **TRUE** после успешного завершения опроса, на следующем цикле сбрасывается в **FALSE**. При наличии ошибок обмена выход не принимает значение **TRUE**, а на выходе **Error** подается код ошибки.
TimeOut	TIME	Таймаут ответа slave-устройства. Если в течение этого времени устройство не отвечает, то СПК переходит к опросу следующего slave-устройства. Рекомендуемое значение – 50 мс.
ValReal	REAL	Считанное/записываемое значение с плавающей точкой – в случае выбора соответствующего формата (см. вход **EncodeType**).
MeasTime	WORD	Значение относительного времени измерения в мс в – случае выбора соответствующего формата (см. вход **EncodeType**).
ValDint	DINT	Считанное/записываемое целочисленное значение – в случае выбора соответствующего формата (см. вход **EncodeType**).
buf	ARRAY [0..14] OF BYTE	Считанное/записываемое нетипизированное значение – в случае выбора соответствующего формата (см. вход **EncodeType**).
buf_sz	BYTE	Количество считанных/записанных байт нетипизированного значения – в случае выбора соответствующего формата (см. вход **EncodeType**).
Error	WORD	Код ошибки. Значение **0** соответствует отсутствию ошибок. Код ошибки **16#FFFF (255)** характеризует отсутствие ответа от слэйва по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А. Список кодов ошибок прибора приведен в РЭ на конкретный прибор.

3.3.11. Перечисление INT_FRM

Перечисление **INT_FRM** описывает форматы целочисленных данных. Используется при работе с ФБ OWEN_GET_DINT и OWEN_SET_DINT.

Имя	Тип	Значение	Описание
UINT_FRM	INT	0	Целочисленный знаковый (4 байта).
SINT1_FRM	INT	1	Целочисленный беззнаковый (1 байт).
SINT2_FRM	INT	2	Целочисленный беззнаковый (2 байта).
SINT4_FRM	INT	3	Целочисленный беззнаковый (4 байта).

3.3.12. Перечисление REAL_FRM

Перечисление **REAL_FRM** описывает форматы данных с плавающей точкой. Используется при работе с ФБ OWEN_GET_REAL и OWEN_SET_REAL.

Имя	Тип	Значение	Описание
FLOAT32T	INT	0	Значение с плавающей точкой по стандарту IEEE754 + время относительного измерения в мс (6 байта).
FLOAT32	INT	1	Значение с плавающей точкой по стандарту IEEE754 (4 байта).
FLOAT24	INT	2	Укороченное значение с плавающей точкой по стандарту IEEE754 (3 байта).
FIX_BIN	INT	3	Знаковое число с фиксированной точкой в двоичном виде.
FIX_BCD	INT	4	Знаковое число с фиксированной точкой в двоично-десятичном виде.

3.3.13. Перечисление OWEN_FRM

Перечисление **OWEN_FRM** описывает форматы данных, используемых при работе с ФБ OwenPoolMaster.

Имя	Тип	Значение	Описание
OW_FL32T	INT	0	Значение с плавающей точкой по стандарту IEEE754 + время относительного измерения в мс (6 байта).
OW_FL32	INT	1	Значение с плавающей точкой по стандарту IEEE754 (4 байта).
OW_FL24	INT	2	Укороченное значение с плавающей точкой по стандарту IEEE754 (3 байта).
FX_BIN	INT	3	Знаковое число с фиксированной точкой в двоичном виде.
FX_BCD	INT	4	Знаковое число с фиксированной точкой в двоично-десятичном виде.
OW_SINT1	INT	5	Целочисленный беззнаковый (1 байт).
OW_SINT2	INT	6	Целочисленный беззнаковый (2 байта).
OW_SINT4	INT	7	Целочисленный беззнаковый (4 байта).
OW_UINT	INT	8	Целочисленный знаковый (4 байта).
OW_NTYPE	INT	9	Нетипизированное значение.

4. Пример: СПК207 + ТРМ212

Рассмотрим пример настройки обмена с использованием библиотеки **OwenNet**. В нем мы наладим связь между контроллером **СПК207.03** (master) и **ТРМ212** (slave).

Сетевые настройки приборов приведены в табл. 4.1:

Табл. 4.1. Сетевые параметры СПК и ТРМ

Параметр	СПК207.03	ТРМ212
COM-порт СПК, к которому подключен ТРМ212	*COM2*	-
Базовый адрес ТРМ212	-	*2*
Скорость обмена	115200	
Количества бит данных	8	
Контроль четности	отсутствует	
Количество стоп-бит	1	

В примере мы будем считывать измеренное значение 1-го канала и записывать значение уставки ПИД-регулятора. Информация о параметрах приведена в документе **Параметры, передаваемые по RS**, который доступен на диске с ПО из комплекта поставки, а также на сайте компании ОВЕН на странице соответствующего прибора:

Список параметров для работы по протоколу ОВЕН

(1 – Наименование параметра; 2 – Hash-код (в шестнадцатиричной системе счисления); 3 – Формат представления данных; 4 – Характеристика; Диапазон значений: 5 – на приборе, 6 – в сети)

1	2	3	4	5	6	7
Группа LvoP Рабочие параметры прибора						
PV* (опер.)		B8DF	F24	Измеренное значение входной величины или код ошибки: - 0xFD – ошибка на входе - 0xFE – отсутствие связи с АЦП - 0xF0 – вычисленное значение заведомо не верно (ответ при сбое памяти) - 0xF7 – датчик отключен. Индекс – 0, 1 по измерительным входам	Определяется диапазоном измерения датчика	
LuPV* (опер.)		B257	F24	Значение на выходе вычислителя [ед. изм.] или код ошибки: - 0xFD-ошибка на входе; - 0xFE - отсутствие связи с АЦП; - 0xF0 – вычисление невозможно [ответ: - при несоответствии датчика и вычислителя (inp2 = v.PTR or v.CS or evnt; or CalC = GrAF); - если отключен датчик, используемый вычислителем (inp2 = oFF); - если на входе вычислителя корня (CALC = SqPv) отрицательное число; - если при вычислении отношения (CALC = rAt) получено 'деление на 0; - при сбое в памяти]	По RS-485 – ограничений нет. Есть ограничения по выводу на индикатор: -1999…9999 для dP =0 или dPt0 =0; -199.9…999.9 для dP0 =1 или dPt0=1; -19.99…99.99 для dP =2; -1.999…9.999 для dP =3.	
SP*		9107	F24	Уставка регулятора	Определяется параметрами SL.L и SL.H	

Рис. 4.1. Список параметров ТРМ212 по протоколу **Овен**

Характеристики параметров приведены в табл. 4.2. ***Обратите внимание***, что параметр Измеренное значение 2-го канала в примере ***использоваться не будет*** – он приведен только для того, чтобы показать принцип адресации оперативных параметров. Поскольку базовый адрес прибора – **2** (см. табл. 4.1), то адрес всех параметров 1-го канала прибора – **2**, а адрес параметров 2-го канала – **3**. Индексация параметров в данном приборе не используется (иначе это было бы отражено в списке параметров).

Табл. 4.2. Характеристики параметров ТРМ212, используемых в примере

Параметр	Имя	Тип	Формат	Адрес	Индекс
Измеренное значение 1 канала	PV	оперативный	Float24	2	не используется
Измеренное значение 2 канала (*не используется в примере*)	PV	оперативный	Float24	3	не используется
Уставка регулятора	SP	конфигурационный	Float24	2	не используется

Пример создан в среде **CODESYS 3.5 SP7 Patch4** и подразумевает запуск на **СПК207.03.CS(-WEB)** с таргет-файлом **3.5.4.20 (023).**

Пример доступен для скачивания: Example_TRM212owen.projectarchive

Пример содержит два проекта – **Device_CFC** и **Device_ST**. Каждый проект содержит библиотеку **OwenNet** и **ComService**, в каждом из них реализовано чтение параметра **PV** и запись параметра **SP** на соответствующем языке программирования. Время цикла задачи, к которой привязана программа – 10 мс.

4.1. Описание реализации на языке CFC

1. Переменные программы **PLC_PRG**:

PLC_PRG

```
PROGRAM PLC_PRG
VAR
    COM_SERVICE_COM2:    COM_SERVICE;          // ФБ настройки и открытия порта COM2

    xStep1:              BOOL;                 // переменная завершения шага 1
    xStep2:              BOOL;                 // переменная завершения шага 2

    TRM212_GetReal:      OWEN_GET_REAL;        // ФБ считывания REAL значения
    TRM212_SetReal:      OWEN_SET_REAL;        // ФБ записи REAL значения

    TRM212_PV1:          REAL;                 // значение, считываемое с ТРМ (измеренная величина 1-го входа)
    TRM212_SP:           REAL;                 // значение, записываемое в ТРМ (уставка регулятора)

    xTrigger:            BOOL;                 // триггер записи уставки
END_VAR
```

Рис. 4.2. Объявление программы **PLC_PRG (CFC)**

2. Код программы **PLC_PRG** на языке **CFC** (*рисунок хорошо масштабируется*):

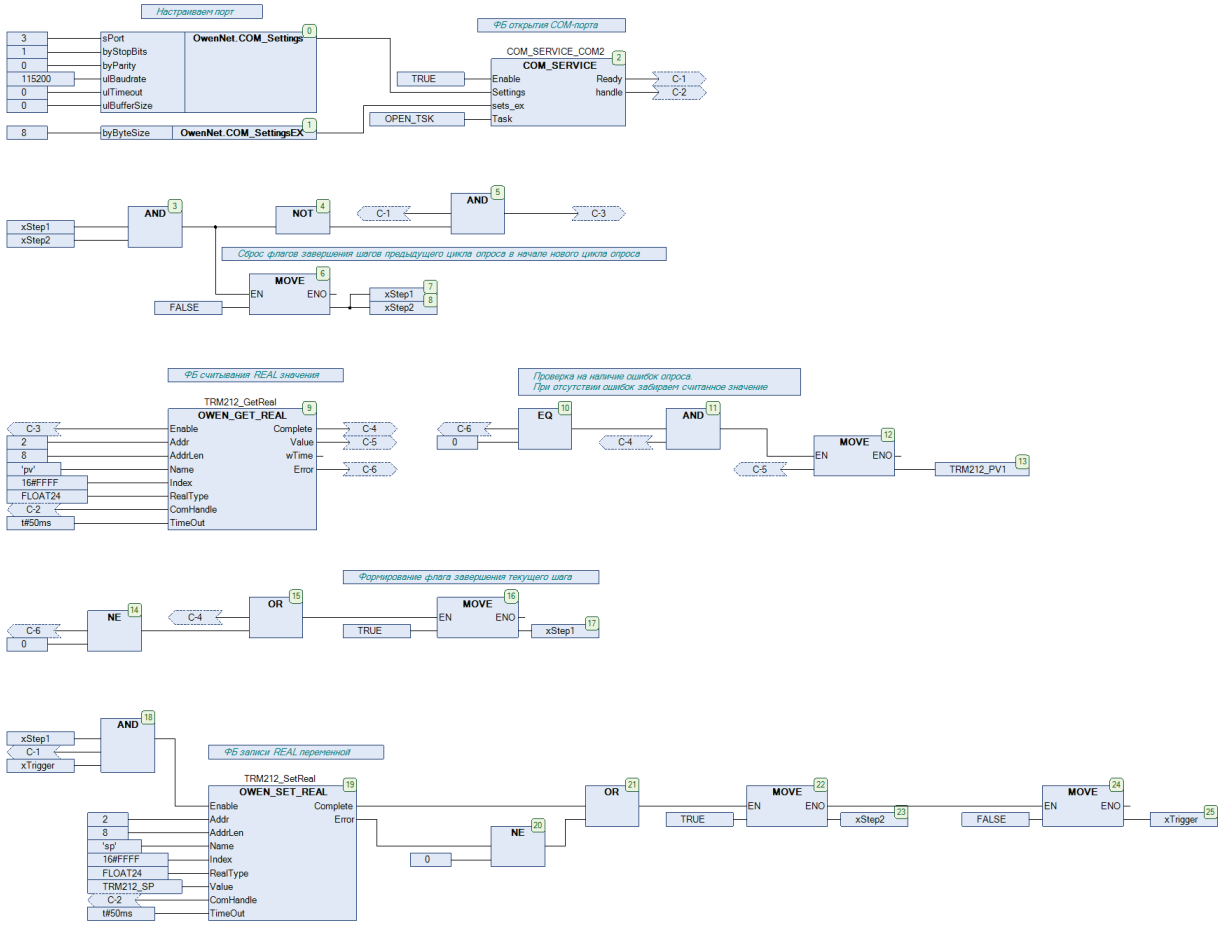

Рис. 4.3. Код программы **PLC_PRG (CFC)**

Программа работает следующим образом:

Блоки 0-1. Настройка COM-порта в соответствии с табл. 4.1. ***Обратите внимание***, что COM-порту **COM2** соответствует номер **3** (номера портов в **CODESYS** смещены на +1 по сравнению с маркировкой на корпусе прибора).

Блок 2. При первом запуске программы с помощью ФБ ComService открывается COM-порт **COM3** с сетевыми настройками, заданными в блоках 0-1. ***Обратите внимание***, что открытие порта занимает 200 циклов программы.

Блоки 3-5. Формируется сигнал инициализации первого шага опроса (чтение переменной). Опрос будет начат только в том случае, если:

- ФБ ComService завершил работу (**Ready=TRUE**);

- предыдущий цикл опроса завершен. ***Обратите внимание***, что флаги завершения предыдущего цикла опроса инвертируются. Это необходимо, так как опрос модулей начинается по переднему фронту входа **Enable** – соответственно, перед началом нового цикла опроса они должны принять значение **FALSE**.

Блоки 6-8. Сброс флага завершения предыдущего цикла опроса.

Блок 9. Считывание значения с плавающей точкой (измеренное значение 1-го канала) с ТРМ212 с помощью ФБ OWEN_GET_REAL. На вход **ComHandle** данного ФБ поступает значение выхода **Handle** ФБ ComService. На вход **Addr** поступает адрес канала (т.к. параметр является **оперативным**) – согласно табл. 4.1, ТРМ212 имеет базовый адрес **2** и, соответственно, его первый канал также имеет адрес **2**. На входе **Name** указано имя считываемого параметра (**'pv'**), на входе **Index** – его линейный индекс (поскольку данный параметр не имеет линейного индекса, то на входе указывается значение **16#FFFF**). На входе **RealType** указывается формат считываемого значения – согласно табл. 4.1, данный параметр имеет формат **FLOAT24**.

Блоки 10-13. Если первый шаг опроса завершен (об этом сигнализирует импульс по переднему фронту на выходе **Complete)** без ошибок (**Error=0**), то считанное значение копируется в переменную программы **TRM212_PV1**. ***Обратите внимание***, что забирать данные с выхода ФБ OWEN_GET_REAL можно только по флагу **Complete.**

Блоки 14-17. Формирование флага завершения первого шага опроса (опрос считается завершенным, если был импульс на выходе **Complete** или на выходе **Error** появилось значение, отличное от нуля – т.е. произошла ошибка).

Блок 18. Формируется сигнал инициализации второго шага опроса (запись переменной). Запись будет произведена только в том случае, если:

- ФБ ComService завершил работу (**Ready=TRUE**);

- предыдущий шаг опроса завершен (**xStep1=TRUE**);

- триггер записи активирован (**xTrigger=TRUE**). В данном примере подразумевается ручное управление триггерной переменной в режиме онлайн-мониторинга; в общем случае работа с триггером должна осуществляться в пользовательской

программе. Использование записи по триггеру обусловлено ограничением на количество перезаписей памяти ТРМ.

Блок 19. Запись значения с плавающей точкой (уставка регулятора) в ТРМ212 с помощью ФБ OWEN_SET_REAL. На вход **ComHandle** данного ФБ поступает значение выхода **Handle** ФБ ComService. На вход **Addr** поступает базовый адрес устройства (т.к. параметр является **конфигурационным**) – согласно табл. 4.1, он равен **2**. На входе **Name** указано имя записываемого параметра (**'sp'**), на входе **Index** – его линейный индекс (поскольку данный параметр не имеет линейного индекса, то на входе указывается значение **16#FFFF**). На входе **RealType** указывается формат записываемого значения – согласно табл. 4.1, данный параметр имеет формат **FLOAT24**. На вход **Value** подается переменная, значение которой будет записано в ТРМ212.

Блоки 20-23. Если второй шаг опроса завершен (опрос считается завершенным, если был импульс на выходе **Complete** или на выходе **Error** появилось значение, отличное от нуля – т.е. произошла ошибка), то формируется флаг завершения второго шага опроса.

Блоки 24-25. Если второй шаг опрос завершен, то происходит сброс триггера записи.

Обратите внимание:

1.при необходимости количество шагов можно увеличить – для этого, соответственно, потребуется увеличить число переменных **xStep№;**

2. приступать к выполнению следующего шага можно только после завершения предыдущего (т.е. после того, когда переменная **xStep№** предыдущего шага принимает значение **TRUE)**;

3. выполнение **каждого шага** занимает несколько **циклов программы**.

4.2. Описание реализации на языке ST

1. Переменные программы **PLC_PRG**:

PLC_PRG

```
PROGRAM PLC_PRG
VAR
    COM_SERVICE_COM2:       COM_SERVICE;                    // ФБ настройки и открытия порта COM2
    Settings_COM2:          OwenNet.SysCom.COM_SETTINGS;    // Структура настроек порта COM2
    SettingsEX_COM2:        OwenNet.SysCom.COM_SETTINGSex;  // Структура расширенных настроек порта COM2

    iStep:                  INT;                            // счетчик шагов

    TRM212_GetReal:         OWEN_GET_REAL;                  // ФБ считывания REAL значения
    TRM212_SetReal:         OWEN_SET_REAL;                  // ФБ записи REAL значения

    TRM212_PV1:             REAL;                           // значение, считываемое с ТРМ (измеренная величина 1-го входа)
    TRM212_SP:              REAL;                           // значение, записываемое в ТРМ (уставка регулятора)

    xTrigger:               BOOL;                           // триггер записи данных
END_VAR
```

Рис. 4.4. Объявление программы **PLC_PRG (ST)**

2. Код программы **PLC_PRG** на языке **CFC** (*рисунок хорошо масштабируется; листинг программы приведен в приложении Б*):

```
// [1] настраиваем COM-порт
Settings_COM2.sPort:=3;
Settings_COM2.byStopBits:=1;
Settings_COM2.byParity:=0;
Settings_COM2.ulBaudrate:=115200;
Settings_COM2.ulTimeout:=0;
Settings_COM2.ulBufferSize:=0;

SettingsEX_COM2.byByteSize:=8;


// [2] открываем COM-порт
COM_SERVICE_COM2
(
    Enable:=TRUE,
    Settings:=Settings_COM2,
    Sets_Ex:=SettingsEX_COM2,
    Task:=OPEN_TSK,
);

// [3] xStep определяет номер текущего шага опроса
CASE iStep OF

    0:

        // [3.0.1] запускаем ФБ чтения REAL переменной
        TRM212_GetReal
            (
            Enable:=COM_SERVICE_COM2.Ready,
            Addr:=2,
            AddrLen:=8,
            Name:='pv',
            Index:=16#FFFF,
            RealType:=FLOAT24,
            ComHandle:=COM_SERVICE_COM2.handle,
            TimeOut:=T#50MS,
            Complete=> ,
            Value=> ,
            wTime=> ,
            Error=>
            );


        // [3.0.2] если ФБ чтения завершил работу (успешно или с ошибкой)...
        IF TRM212_GetReal.Complete OR TRM212_GetReal.Error<>0 THEN

            // ...и ошибки отсутствуют, то забираем считанное значение
            IF TRM212_GetReal.Error=0 THEN
                TRM212_PV1:=TRM212_GetReal.Value;
            END_IF

            // завершаем работу блока
            TRM212_GetReal(Enable:=FALSE);

            // переходим к следующему шагу
            iStep:=1;

        END_IF


    1:

        // [3.1.0] если триггер записи активен...
        IF xTrigger THEN

            //... то запускаем ФБ записи REAL переменной
            TRM212_SetReal
                (
                Enable:=COM_SERVICE_COM2.Ready,
                Addr:=2,
                AddrLen:=8,
                Name:='sp',
                Index:= 16#FFFF,
                RealType:=FLOAT24,
                Value:=TRM212_SP,
                ComHandle:=COM_SERVICE_COM2.handle,
                TimeOut:=T#50MS,
                Complete=> ,
                Error=>
                );



        // [3.1.1] если ФБ записи завершил работу (успешно или с ошибкой)...
        IF TRM212_SetReal.Complete OR TRM212_SetReal.Error<>0  THEN


            // завершаем работу ФБ
            TRM212_SetReal(Enable:=FALSE);

            // сбрасываем триггер
            xTrigger:=FALSE;

            // переходим к следующему шагу
            iStep:=2;

        END_IF

        ELSE
            iStep:=2; // если триггер записи не поднят - то переходим к следующему шагу
        END_IF

    2:      // [3.2.0]
            // читаем/записываем другие параметры этого устройства или других устройств (пошагово)
            // после конечной операции переходим к начальному шагу

            iStep:=0;
END_CASE
```

Рис. 4.5. Код программы **PLC_PRG (ST)**

Программа работает следующим образом:

Блок [1]. Настройка COM-порта в соответствии с табл. 4.1. ***Обратите внимание***, что COM-порту **COM2** соответствует номер **3** (номера портов в **CODESYS** смещены на +1 по сравнению с маркировкой на корпусе прибора).

Блок [2]. При первом запуске программы с помощью ФБ ComService открывается COM-порт **COM3** с сетевыми настройками, заданными в блоках 0-1. ***Обратите внимание***, что открытие порта занимает 200 циклов программы.

Блок [3]. Реализация последовательного выполнения шагов опроса через оператор **CASE**:

- в случае **iStep=0** происходит считывание измеренного значения 1-го канала;

- в случае **iStep=1** происходит запись уставки регулятора.

Блок [3.0.1]. Считывание значения с плавающей точкой (измеренное значение 1-го канала) с ТРМ212 с помощью ФБ OWEN_GET_REAL. На вход **ComHandle** данного ФБ поступает значение выхода **Handle** ФБ ComService. На вход **Addr** поступает адрес канала (т.к. параметр является **оперативным**) – согласно табл. 4.1, ТРМ212 имеет базовый адрес **2** и, соответственно, его первый канал также имеет адрес **2**. На входе **Name** указано имя считываемого параметра (**'pv'**), на входе **Index** – его линейный индекс (поскольку данный параметр не имеет линейного индекса, то на входе указывается значение **16#FFFF**). На входе **RealType** указывается формат считываемого значения – согласно табл. 4.1, данный параметр имеет формат **FLOAT24**.

Блок [3.0.2]. Если данный шаг опроса успешно завершен (об этом сигнализирует импульс по переднему фронту на выходе **Complete**) без ошибок (**Error=0**), то считанное значение копируется в переменную программы **TRM212_PV1**. ***Обратите внимание***, что забирать данные с выхода ФБ OWEN_GET_REAL можно только по флагу **Complete.**

Независимо от наличия ошибок завершаем данный шаг опроса и переходим к следующему.

Блок [3.1.0]. Если триггер записи активирован (**xTrigger=TRUE**), то производим запись значения с плавающей точкой (уставка регулятора) в ТРМ212 с помощью ФБ OWEN_SET_REAL.

В данном примере подразумевается ручное управление триггерной переменной в режиме онлайн-мониторинга; в общем случае работа с триггером должна осуществляться в пользовательской программе. Использование записи по триггеру обусловлено ограничением на количество перезаписей памяти ТРМ.

На вход **ComHandle** данного ФБ поступает значение выхода **Handle** ФБ ComService. На вход **Addr** поступает базовый адрес устройства (т.к. параметр является **конфигурационным**) – согласно табл. 4.1, он равен **2**. На входе **Name** указано имя записываемого параметра (**'sp'**), на входе **Index** – его линейный индекс (поскольку данный параметр не имеет линейного индекса, то на входе указывается значение **16#FFFF**). На входе **RealType** указывается формат записываемого значения – согласно табл. 4.1, данный параметр имеет формат **FLOAT24**. На вход **Value** подается переменная, значение которой будет записано в ТРМ212.

Блок [3.1.1]. Если данный шаг опроса завершен (опрос считается завершенным, если был импульс на выходе **Complete** или на выходе **Error** появилось значение, отличное от нуля – т.е. произошла ошибка), то завершаем данный шаг опроса, сбрасываем триггер записи и переходим к следующему шагу. Если триггер записи на данном цикле не взводился, то также переходим к следующему шагу.

Блок [3.2.0]. После завершения последнего шага опроса, переходим к начальному.

Обратите внимание:

1. при необходимости количество шагов можно увеличить – для этого, соответственно, потребуется увеличить количество действий в операторе **CASE;**

2. перед переходом к следующему шагу опроса, необходимо завершить текущий, вызвав ФБ опроса с **Enable=FALSE**;

3. выполнение **каждого шага** занимает несколько **циклов программы**.

4.3. Запуск примера

Загрузите проект в СПК и запустите его. Подразумевается, что у ТРМ212 заданы сетевые настройки в соответствии с табл. 4.1. На верхний цифровой индикатор выведите измеренное значение 1-го канала, на нижний – значение уставки регулятора.

В переменную **TRM212_PV1** будет считано измеренное значение 1 канала.

Измените значение переменной **TRM212_SP** и присвойте переменной **xTrigger** значение **TRUE** – это приведет к изменению уставки ТРМ.

PLC_PRG

Device_CFC.Application.PLC_PRG

Выражение	Тип	Значение	Подготовленное ...	Адрес
COM_SERVICE_COM2	COM_SERVICE			
Settings_COM2	ComSerice.COM_SET...			
SettingsEX_COM2	ComSerice.COM_SET...			
xStep1	BOOL	TRUE		
xStep2	BOOL	FALSE		
TRM212_GetReal	OWEN_GET_REAL			
TRM212_SetReal	OWEN_SET_REAL			
TRM212_PV1	REAL	23.5		
TRM212_SP	REAL	23.5		
xTrigger	BOOL	FALSE		

Рис. 4.6. Отображение измеренного значения 1-го канала ТРМ212 и уставки его регулятора в CODESYS и на индикаторах ТРМ

Приложение

А. Список кодов ошибок при обмене по протоколу Овен

Код ошибки	Имя ошибки	Описание
Определение констант ошибок приема		
0x0	OK	Отсутствие ошибок обмена.
Ошибки записи параметров и атрибутов функцией modific		
0x2	PDOT	Задано положение точки, превышающее 3.
0x3	EROM	Попытка модификации ROM-параметра.
0x4	ESTR	Не целое число при записи индекса строки или времени.
0x5	EDOT	Неверно задано положение десятичной точки (при фиксированной точке).
0x6	ERNG	Значение мантиссы превышает ограничение дескриптора.
Ошибки записи атрибутов функциями ModAllPermis() и ModEditPermis()		
0x7	EOWNER	Несанкционированная попытка редактирования атрибутов (попытка изменения атрибута пользователем, который не является хозяином параметра).
0x8	EPERM	У запрошенного параметра отсутствуют признаки.
Стандартные ошибки протокола обмена		
0x21	AFE	Аппаратная ошибка кадрирования.
0x22	B8E	Ошибка в 8-м бите посылки.
0x23	B9E	Ошибка в 9-м бите посылки.
0x24	SBE	Ошибка приема стоп-байта (стоп пришел не вовремя).
0x25	OVB	Ошибка переполнения буфера.
0x26	ERS	Принят недопустимый символ.
0x27	CRCE	Неверная контрольная сумма кадра.
0x28	EDESC	Не найден дескриптор.
0x29	NFNC	Не найдена сетевая функция при наличии дескриптора.
Стандартные ошибки модулей		
0x30	EDGT	Мантисса двоично-десятичного параметра содержит ошибку.
0x31	SZE	Размер поля данных не соответствует ожидаемому.
0x32	EASK	Значение бита запроса не соответствует ожидаемому.
0x33	EACC	Редактирование параметра запрещено индивидуальным атрибутом.
0x34	IDXOVF	Недопустимо большой линейный индекс.
0x35	IDXLIM	Индекс параметра превышает ограничитель индекса.
0x36	EXTROM	Данный код не используется.
0x37	RESERVED	Данный код не используется.

Запрещение и записи групповым атрибутом		
0x38	LEVGRATT	Запрещающий групповой атрибут находится на уровне 0 (в корне).
0x39	LEVGRATT1	Запрещающий групповой атрибут находится на уровне 1.
0x3A	LEVGRATT2	Запрещающий групповой атрибут находится на уровне 2.
0x3B	LEVGRATT3	Запрещающий групповой атрибут находится на уровне 3.
0x3C	LEVGRATT4	Запрещающий групповой атрибут находится на уровне 4.
0x3D	LEVGRATT5	Запрещающий групповой атрибут находится на уровне 5.
0x3E	LEVGRATT6	Запрещающий групповой атрибут находится на уровне 6.
0x3F	LEVGRATT7	Запрещающий групповой атрибут находится на уровне 7.
Состояния COMMON-сегмента		
0x41	__LOCKSEG	Выполняется другая задача (сегмент COMMON занят).
0x42	__FREESEG	Задача еще не запущена (сегмент COMMON свободен).
0x43	__READYSEG	Запрошенная задача уже выполняется.
0x44	__DEBUGSEG	Программе неизвестна запрошенная функция.
0x45	__NOWHATCOM	В программе стоит заглушка функции WhatCOMState()
0x46	__NORUNCOM	В программе стоит заглушка функции RunCOMTask()
0x47		Недопустимое сочетание значений параметров (изменение параметра было запрещено функцией Valid).
0x48		
Ошибки при редактировании графиков		
0x49		Нарушена упорядоченность узлов X по возрастанию.
0x4A		Попытка записи X при ненулевом числе узлов графика.
0x4B		Ошибка выполнения функции PrevWriteActions().
Ошибки мостов и ретрансляторов		
0x50	GATE_OVR	Переполнение буфера моста или ретранслятора.
0x51	GATE_DERR	Превышение таймаута ответа, потеря пакета в дочерней сети (сети, в которую ретранслируется пакет).
0x52	GATE_NONET	Запрошенная дочерняя подсеть недоступна (в случае ретрансляции в одну из нескольких дочерних подсетей).
0x53	GATE_MERR	Ответ из дочерней сети не может быть ретранслирован в материнскую сеть.

Б. Листинг программы из п. 4

```
PROGRAM PLC_PRG
VAR
	COM_SERVICE_COM2: COM_SERVICE;		// ФБ настройки и открытия порта COM2
	Settings_COM2:	ComSerice.COM_SETTINGS;	// Структура настроек порта COM2
	SettingsEX_COM2:	ComSerice.COM_SETTINGSex;	// Структура расширенных настроек порта COM2

	iStep:		INT;	// счетчик шагов

	TRM212_GetReal:	OWEN_GET_REAL;		// ФБ считывания REAL значения
	TRM212_SetReal:	OWEN_SET_REAL;		// ФБ записи REAL значения

	TRM212_PV1:	REAL;	// значение, считываемое с ТРМ (измеренная величина 1-го входа)
	TRM212_SP:	REAL;	// значение, записываемое в ТРМ (уставка регулятора)

	xTrigger:	BOOL;	// триггер записи данных
END_VAR


// [1] настраиваем COM-порт
Settings_COM2.sPort:=3;
Settings_COM2.byStopBits:=1;
Settings_COM2.byParity:=0;
Settings_COM2.ulBaudrate:=115200;
Settings_COM2.ulTimeout:=0;
Settings_COM2.ulBufferSize:=0;

SettingsEX_COM2.byByteSize:=8;


// [2] открываем COM-порт
COM_SERVICE_COM2
(
	Enable:=TRUE,
	Settings:=Settings_COM2,
	Sets_Ex:=SettingsEX_COM2,
	Task:=OPEN_TSK,
);

// [3] xStep определяет номер текущего шага опроса
CASE iStep OF

	0:
		// [3.0.1] запускаем ФБ чтения REAL переменной
		TRM212_GetReal
			(
			Enable:=COM_SERVICE_COM2.Ready,
			Addr:=2,
			AddrLen:=8,
			Name:='pv',
			Index:=16#FFFF,
			RealType:=FLOAT24,
			ComHandle:=COM_SERVICE_COM2.handle,
			TimeOut:=T#50MS,
			Complete=> ,
			Value=> ,
			wTime=> ,
			Error=>
			);


		// [3.0.2] если ФБ чтения завершил работу...
		IF TRM212_GetReal.Complete OR TRM212_GetReal.Error<>0 THEN

			// ...и ошибки отсутствуют, то забираем считанное значение
			IF TRM212_GetReal.Error=0 THEN
				TRM212_PV1:=TRM212_GetReal.Value;
			END_IF

			// завершаем работу блока
			TRM212_GetReal(Enable:=FALSE);

			// переходим к следующему шагу
			iStep:=1;

		END_IF
```

```
1:

        // [3.1.0] если триггер записи активен...
        IF xTrigger THEN

                //... то запускаем ФБ записи REAL переменной
                TRM212_SetReal
                        (
                        Enable:=COM_SERVICE_COM2.Ready,
                        Addr:=2,
                        AddrLen:=8,
                        Name:='sp',
                        Index:= 16#FFFF,
                        RealType:=FLOAT24,
                        Value:=TRM212_SP,
                        ComHandle:=COM_SERVICE_COM2.handle,
                        TimeOut:=T#50MS,
                        Complete=> ,
                        Error=>
                        );



        // [3.1.1] если ФБ записи завершил работу...
        IF TRM212_SetReal.Complete OR TRM212_SetReal.Error<>0 THEN


                // завершаем работу ФБ
                TRM212_SetReal(Enable:=FALSE);

                // сбрасываем триггер
                xTrigger:=FALSE;

                // переходим к следующему шагу
                iStep:=2;


        END_IF

        ELSE
                iStep:=2; // если триггер записи не поднят - переходим к следующему шагу

        END_IF



2:              // [3.2.0]
                // читаем/записываем другие параметры этого устройства или других устройств
                // (пошагово)
                // после конечной операции переходим к начальному шагу

                iStep:=0;
END_CASE
```