



CODESYS V3.5

Протокол OBEN



Руководство пользователя

25.02.2020

версия 2.1

Оглавление

Глоссарий.....	3
1 Цель и структура документа.....	3
2 Основные сведения о протоколе ОВЕН.....	4
3 Описание библиотеки OwenNet	5
3.1 Установка библиотеки	5
3.2 Добавление библиотеки в проект CODESYS.....	6
3.3 Описание библиотеки.....	8
3.3.1 Блок COM_CONTROL (библиотека OwenCommunication)	8
3.3.2 Блок OWEN_GET_DINT	9
3.3.3 Блок OWEN_SET_DINT	11
3.3.4 Блок OWEN_GET_REAL.....	12
3.3.5 Блок OWEN_SET_REAL	13
3.3.6 Блок OWEN_UNI_IO.....	14
3.3.7 Блок OWEN_LISTEN	16
3.3.8 Блок OwenPoolMaster	17
3.3.9 Структура OwenPool	18
3.3.10 Перечисление INT_FRM.....	19
3.3.11 Перечисление REAL_FRM	19
3.3.12 Перечисление OWEN_FRM.....	19
4 Пример: СПК1xx [M01] + TPM212	21
4.1 Описание реализации на языке CFC	23
4.2 Описание реализации на языке ST	25
4.3 Запуск примера	29
Приложение А. Список кодов ошибок при обмене по протоколу ОВЕН	30
Приложение Б. Листинг программы из п. 4	32

Глоссарий

ПЛК – программируемый логический контроллер.

ФБ – функциональный блок.

1 Цель и структура документа

Настоящее руководство описывает настройку обмена данными с использованием протокола **ОВЕН** для контроллеров **ОВЕН**, программируемых в среде **CODESYS V3.5**. Руководство предназначено для пользователей, которые обладают базовыми навыками работы с **CODESYS** и ПЛК, поэтому общие вопросы (например, создание и загрузка проектов) в данном документе не рассматриваются. Основная информация приведена в документах **CODESYS V3.5. Первый старт** и **CODESYS V3.5. FAQ**, которые доступны на сайте [ОВЕН](#) в разделе **CODESYS V3/Документация**.

Протокол **ОВЕН** поддерживается такими устройствами, как TPM, СИ, модули Mx110 и др.

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

Руководство содержит описание библиотеки и пример ее использования для опроса **TPM212** (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**.

Ключевые моменты, которые необходимы для понимания настоящего руководства:

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; восьмиканальный ТРМ138 – восемь адресов. В настройках прибора устанавливается его **базовый адрес**, который соответствует адресу первого канала. Таким образом, **ТРМ200** с базовым адресом 10 займет в сети адреса 10 и 11: адрес 10 будет соответствовать его первому каналу измерения, адрес 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.
Обычно каждый канал прибора имеет идентичный набор оперативных параметров. При чтении/записи конкретного параметра следует указывать **адрес** соответствующего канала (см. пп. 2).
Конфигурационные параметры содержат информацию о настройках прибора. В случае наличия нескольких однотипных конфигурационных параметров (например, коэффициентов ПИД-регуляторов многоканального прибора) для обращения к ним следует 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 из комплекта поставки и сайте компании [ОВЕН](#) в разделе **CODESYS V3/Библиотеки**.

Для установки библиотеки в **CODESYS** в меню **Инструменты** следует выбрать пункт **Репозиторий библиотек**, нажать кнопку **Установить** и указать путь к файлу библиотеки:

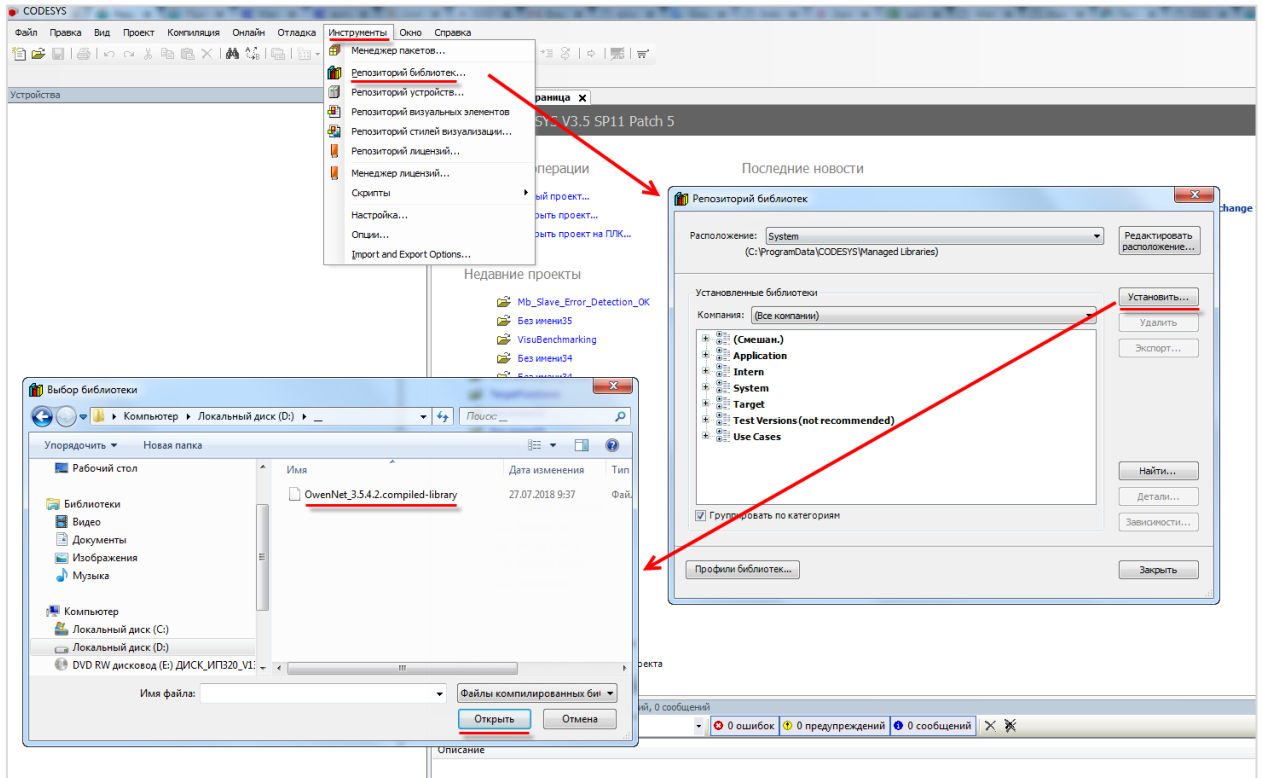


Рисунок 3.1 – Установка библиотеки OwenNet в среду CODESYS

3.2 Добавление библиотеки в проект CODESYS

Для добавления библиотеки **OwenNet** в проект **CODESYS**, в **Менеджере библиотек** следует нажать кнопку **Добавить библиотеку** и в строке поиска ввести **owennet**, после чего выбрать из списка нужную библиотеку и нажать **ОК**.

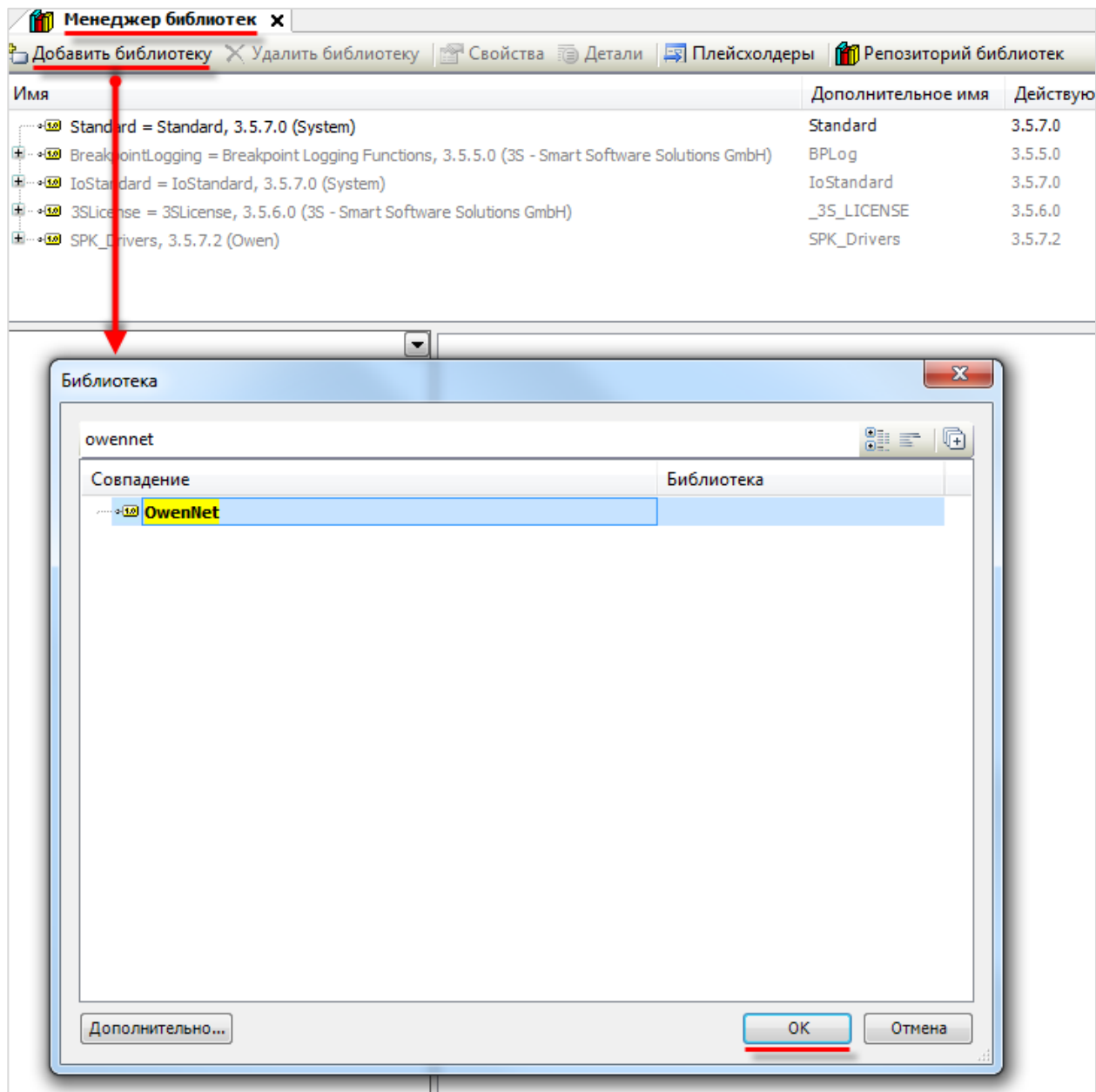


Рисунок 3.2 – Добавление библиотеки OwenNet

В библиотеку **OwenNet** не входит функциональный блок (ФБ) открытия COM-порта, поэтому для этой цели необходимо воспользоваться другой библиотекой. Рекомендуется использовать библиотеку **OwenCommunication**.

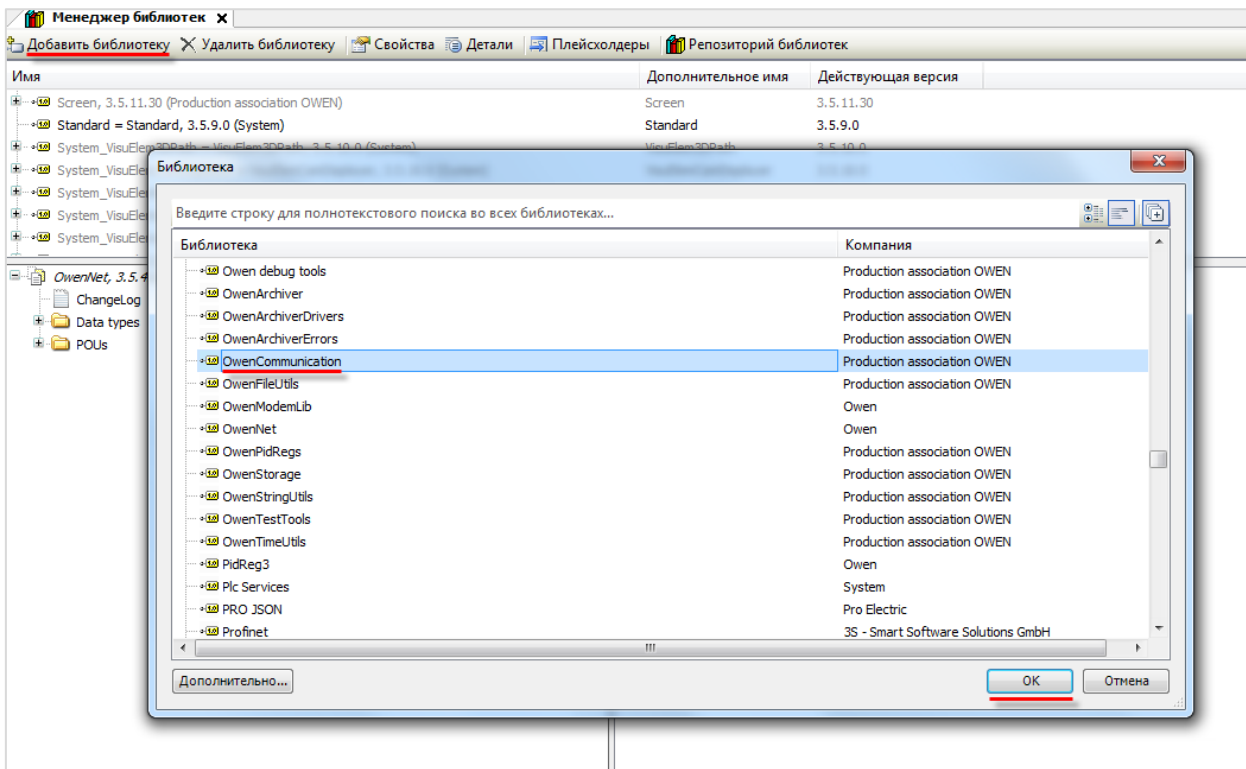


Рисунок 3.3 – Добавление библиотеки OwenCommunication

После добавления библиотеки появятся в списке Менеджера библиотек:

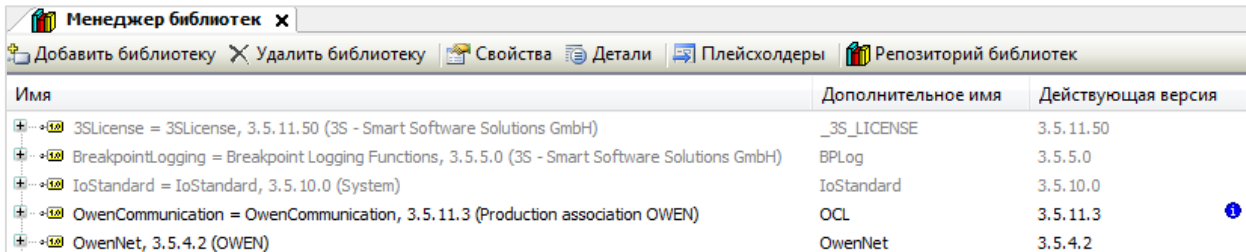


Рисунок 3.4 – Список библиотек проекта

3.3 Описание библиотеки

3.3.1 Блок COM_CONTROL (библиотека OwenCommunication)

Функциональный блок **COM_CONTROL**, входящий в библиотеку **OwenCommunication**, используется для настройки и открытия/закрытия COM-порта.

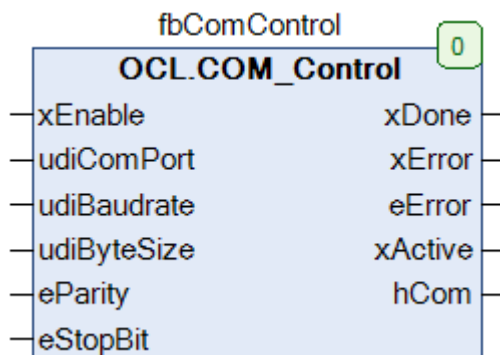


Рисунок 3.5 – Внешний вид ФБ COM_CONTROL на языке CFC

Таблица 3.1 – Описание входов и выходов ФБ COM_CONTROL

Название	Тип	Описание
Входы		
xEnable	BOOL	По переднему фронту происходит открытие COM-порта, по заднему – закрытие
udiComPort	UDINT	Номер COM-порта
udiBaudrate	UDINT	Скорость обмена в бодах. Стандартные возможные значения: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
udiByteSize	UDINT(7..8)	Число бит данных (7 или 8)
eParity	OCL.COM_PARITY	Режим контроля четности
eStopBit	OCL.COM_STOPBIT	Число стоп-бит
Выходы		
xDone	BOOL	Принимает TRUE на один цикл ПЛК при успешном открытии порта
xError	BOOL	Принимает значение TRUE в случае возникновения ошибки
eError	OCL.ERROR	Статус работы ФБ (или код ошибки)
xActive	BOOL	Пока порт открыт, данный выход имеет значение TRUE
hCom	CAA.HANDLE	Дескриптор COM-порта

3.3.2 Блок OWEN_GET_DINT

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

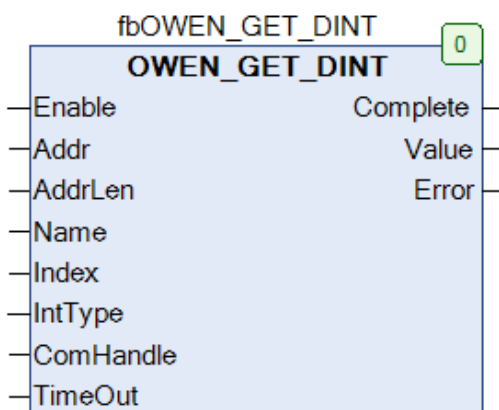


Рисунок 3.6 – Внешний вид ФБ OWEN_GET_DINT на языке CFC

Таблица 3.2 – Описание входов и выходов ФБ OWEN_GET_DINT

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему фронту
Addr	WORD	Адрес slave-устройства (для многоканального устройства – адрес канала)
AddrLen	ADR_LEN	Длина сетевого адреса slave-устройства. Возможные значения: 8 – 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 COM CONTROL (или аналогичного)
TimeOut	TIME	Таймаут ответа slave-устройства. Если в течение этого времени устройство не отвечает, то ПЛК 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 . В случае наличия ошибок обмена выход не принимает значение TRUE , а на выход 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 slave-устройства по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А . Список

3 Описание библиотеки OwenNet

		кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор
--	--	--

3.3.3 Блок OWEN_SET_DINT

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

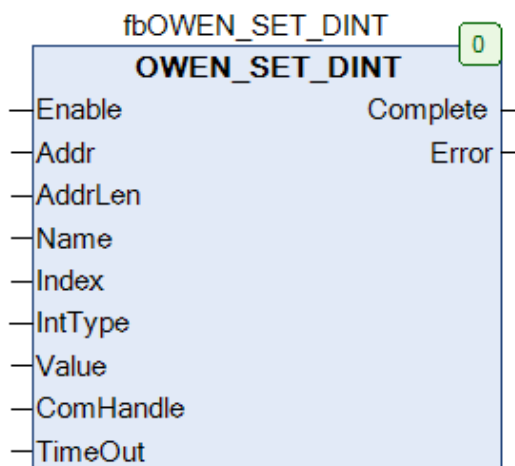


Рисунок 3.7 – Внешний вид ФБ OWEN_SET_DINT на языке CFC

Таблица 3.3 – Описание входов и выходов ФБ OWEN_SET_DINT

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему фронту
Addr	WORD	Адрес slave-устройства (для многоканального устройства – адрес канала)
AddrLen	ADR_LEN	Длина сетевого адреса slave-устройства. Возможные значения: 8 – 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 ПЛК 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 . В случае наличия ошибок обмена выход не принимает значение TRUE , а на выход Error подается код ошибки
Error	WORD	Код ошибки. Значение 0 соответствует отсутствию ошибок. Код ошибки 16#FFFF (255) характеризует отсутствие ответа от slave-устройства по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А . Список кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор

3.3.4 Блок OWEN_GET_REAL

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

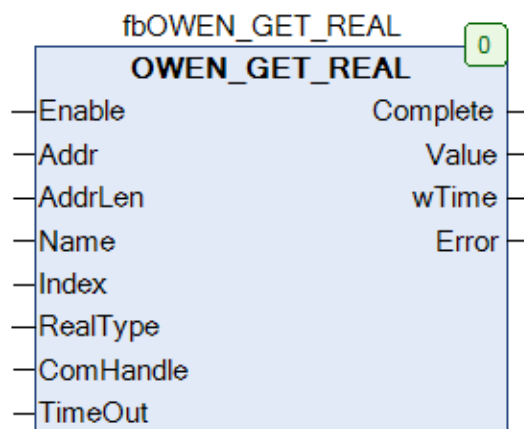


Рисунок 3.8 – Внешний вид ФБ OWEN_GET_REAL на языке CFC

Таблица 3.4 – Описание входов и выходов ФБ OWEN_GET_REAL

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему фронту
Addr	WORD	Адрес slave-устройства (для многоканального устройства – адрес канала)
AddrLen	ADR_LEN	Длина сетевого адреса slave-устройства. Возможные значения: 8 – 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 ПЛК 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 . В случае наличия ошибок обмена выход не принимает значение TRUE , а на выход 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 slave-устройства по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А . Список кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор

3.3.5 Блок OWEN_SET_REAL

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

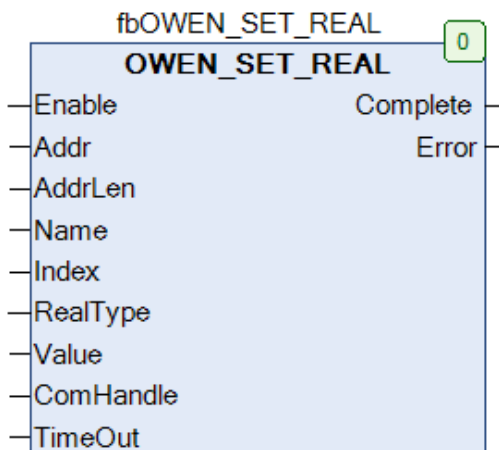


Рисунок 3.9 – Внешний вид ФБ OWEN_SET_REAL на языке CFC

Таблица 3.5 – Описание входов и выходов ФБ OWEN_SET_REAL

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему фронту
Addr	WORD	Адрес slave-устройства (для многоканального устройства – адрес канала)
AddrLen	ADR_LEN	Длина сетевого адреса slave-устройства. Возможные значения: 8 – 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 ПЛК 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 . В случае наличия ошибок обмена выход не принимает значение TRUE , а на выход Error подается код ошибки
Error	WORD	Код ошибки. Значение 0 соответствует отсутствию ошибок. Код ошибки 16#FFFF (255) характеризует отсутствие ответа от slave-устройства по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А . Список кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор

3.3.6 Блок OWEN_UNI_IO

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

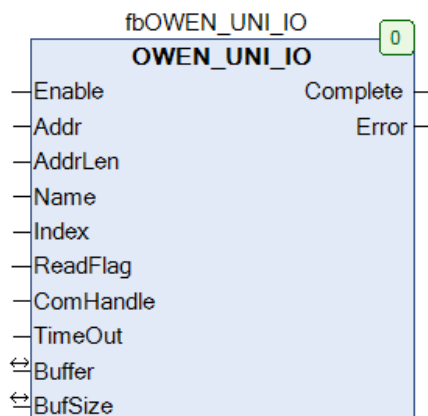


Рисунок 3.10 – Внешний вид ФБ OWEN_UNI_IO на языке CFC

Таблица 3.6 – Описание входов и выходов ФБ OWEN_UNI_IO

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему фронту
Addr	WORD	Адрес slave-устройства (для многоканального устройства – адрес канала)
AddrLen	ADR_LEN	Длина сетевого адреса slave-устройства. Возможные значения: 8 – 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 ПЛК 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 . В случае наличия ошибок обмена выход не принимает значение TRUE , а на выход Error подается код ошибки

Error	WORD	Код ошибки. Значение 0 соответствует отсутствию ошибок. Код ошибки 16#FFFF (255) характеризует отсутствие ответа от slave-устройства по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А . Список кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор
<i>Входные-выходные переменные (VAR_IN_OUT)</i>		
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 в случае необходимости считывать данные в сети, в которой уже находится master-устройство.

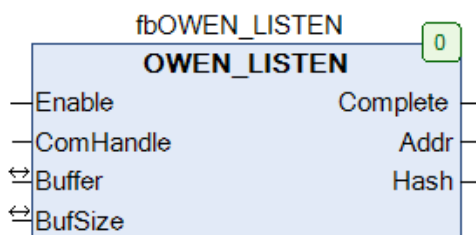


Рисунок 3.11 – Внешний вид ФБ OWEN_LISTEN на языке CFC

Таблица 3.7 – Описание входов и выходов ФБ OWEN_LISTEN

Имя переменной	Тип	Описание
Входные переменные		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему фронту
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 . В случае наличия ошибок обмена выход не принимает значение TRUE , а на выход 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 slave-устройства по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А . Список кодов ошибок прибора приведен в руководстве по эксплуатации 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).

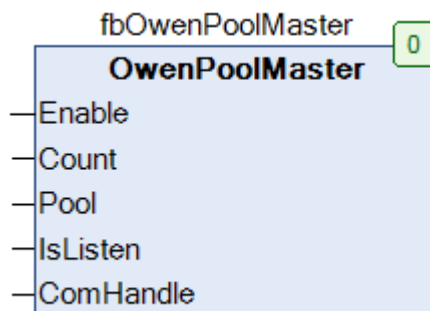


Рисунок 3.12 – Внешний вид ФБ OwenPoolMaster на языке CFC

Таблица 3.8 – Описание входов и выходов ФБ OwenPoolMaster

Имя переменной	Тип	Описание
<i>Входные переменные</i>		
Enable	BOOL	Переменная работы блока. Опрос начинается по переднему фронту переменной и прекращается по заднему фронту
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)

3.3.9 Структура OwenPool

Структура **OwenPool** используется при работе с ФБ [OwenPoolMaster](#). Каждый экземпляр структуры представляет собой команду, выполняемую блоком.

Таблица 3.9 – Описание элементов структуры OwenPool

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 . В случае наличия ошибок обмена выход не принимает значение TRUE , а на выход Error подается код ошибки
TimeOut	TIME	Таймаут ответа slave-устройства. Если в течение этого времени устройство не отвечает, то ПЛК 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 slave-устройства по истечению таймаута опроса. Список кодов сетевых ошибок приведен в приложении А . Список кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор

3.3.10 Перечисление INT_FRM

Перечисление **INT_FRM** описывает форматы целочисленных данных. Используется во время работы с ФБ [OWEN_GET_DINT](#) и [OWEN_SET_DINT](#).

Таблица 3.10 – Описание элементов перечисления INT_FRM

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

3.3.11 Перечисление REAL_FRM

Перечисление **REAL_FRM** описывает форматы данных с плавающей точкой. Используется во время работы с ФБ [OWEN_GET_REAL](#) и [OWEN_SET_REAL](#).

Таблица 3.11 – Описание элементов перечисления REAL_FRM

Имя	Тип	Значение	Описание
FLOAT32T	INT	0	Значение с плавающей точкой по стандарту IEEE 754 + время относительного измерения в мс (6 байта)
FLOAT32		1	Значение с плавающей точкой по стандарту IEEE 754 (4 байта)
FLOAT24		2	Укороченное значение с плавающей точкой по стандарту IEEE 754 (3 байта)
FIX_BIN		3	Знаковое число с фиксированной точкой в двоичном виде
FIX_BCD		4	Знаковое число с фиксированной точкой в двоично-десятичном виде

3.3.12 Перечисление OWEN_FRM

Перечисление **OWEN_FRM** описывает форматы данных, используемых во время работы с ФБ [OwenPoolMaster](#).

Таблица 3.12 – Описание элементов перечисления OWEN_FRM

Имя	Тип	Значение	Описание
OW_FL32T	INT	0	Значение с плавающей точкой по стандарту IEEE 754 + время относительного измерения в мс (6 байта)
OW_FL32		1	Значение с плавающей точкой по стандарту IEEE 754 (4 байта)
OW_FL24		2	Укороченное значение с плавающей точкой по стандарту IEEE 754 (3 байта)
OW_FX_BIN		3	Знаковое целое с фиксированной точкой в двоичном виде
OW_FX_BCD		4	Знаковое целое с фиксированной точкой в двоично-десятичном виде
OW_SINT1		5	Знаковое целое (1 байт)
OW_SINT2		6	Знаковое целое (2 байта)
OW_SINT4		7	Знаковое целое (4 байта)

3 Описание библиотеки OwenNet

OW_UINT		8	Беззнаковое целое (4 байта)
OW_NTYPE		9	Нетипизированное значение

4 Пример: СПК1хх [M01] + TPM212

В данном разделе приведен пример настройки обмена между контроллером **СПК1хх [M01]** (master) и **TPM212** (slave) с использованием библиотеки **OwenNet**.

Сетевые настройки приборов приведены в таблице ниже.

Таблица 4.1 – Сетевые параметры СПК и ТРМ

Параметр	СПК1хх [M01]	TPM212
COM-порт ПЛК, к которому подключен TPM212	COM2	-
Базовый адрес TPM212	-	2
Скорость обмена	115200	
Количество бит данных	8	
Контроль четности	Отсутствует	
Количество стоп-бит	1	

В рамках примера считывается измеренное значение первого канала и записывается значение уставки ПИД-регулятора. Информация о параметрах приведена в документе **Параметры, передаваемые по RS**, который доступен на диске с ПО из комплекта поставки и 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						
<u>PV*</u> (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
<u>LuPV*</u> (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 evt; 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 dP10 =0; -199.9...999.9 для dP0 =1 или dP10=1; -19.99...99.99 для dP =2; -1.999...9.999 для dP =3.
<u>SP*</u>		9107	F24	<u>Уставка регулятора</u>		Определяется параметрами SL.L и SL.H

Рисунок 4.1 – Список параметров TPM212 по протоколу ОВЕН

Характеристики параметров приведены в таблице ниже.

Таблица 4.2 – Характеристики параметров TPM212, используемых в примере

Параметр	Имя	Тип	Формат	Адрес	Индекс
Измеренное значение 1 канала	PV	Оперативный	Float24	2	не используется
Измеренное значение 2 канала (<i>не используется в примере</i>)	PV	Оперативный	Float24	3	не используется
Уставка регулятора	SP	Конфигурационный	Float24	2	не используется



ПРИМЕЧАНИЕ

В рамках примера параметр Измеренное значение второго канала *не используется*. Он приведен только для того, чтобы продемонстрировать принцип адресации оперативных параметров. Так как базовый адрес прибора – **2** (см. таблицу 4.1), то адрес для всех параметров первого канала прибора равен **2**, а адрес для параметров второго канала – **3**. Индексация параметров в данном приборе не используется (иначе это было бы отражено в списке параметров).

Пример создан в среде **CODESYS 3.5 SP11 Patch 5** и подразумевает запуск на **СПК1xx [M01]** с таргет-файлом **3.5.11.x**. В случае необходимости запуска проекта на другом устройстве следует изменить таргет-файл в проекте (**ПКМ** на узел **Device** – **Обновить устройство**).

Пример содержит два проекта – **Device_CFC** и **Device_ST**. Каждый проект содержит библиотеку **OwenNet** и **OwenCommunication**, 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.

Пример доступен для скачивания: [Example_TRM212_3511v2.projectarchive](#)

4.1 Описание реализации на языке CFC

1. Переменные программы PLC_PRG:

```

1  PROGRAM PLC_PRG
2  VAR
3      fbComControl:      OCL.COM_Control;      // фБ настройки и открытия порта COM2
4
5      xStep1:            BOOL;                  // переменная завершения шага 1
6      xStep2:            BOOL;                  // переменная завершения шага 2
7
8      TRM212_GetReal:    OWEN_GET_REAL;        // фБ считывания REAL значения
9      TRM212_SetReal:    OWEN_SET_REAL;        // фБ записи REAL значения
10
11     TRM212_PV1:         REAL;                  // значение, считываемое с TPM (измеренная величина 1-го входа)
12     TRM212_SP:         REAL;                  // значение, записываемое в TPM (уставка регулятора)
13
14     xTrigger:           BOOL;                  // триггер записи уставки
15 END_VAR

```

Рисунок 4.2 – Объявление программы PLC_PRG (CFC)

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

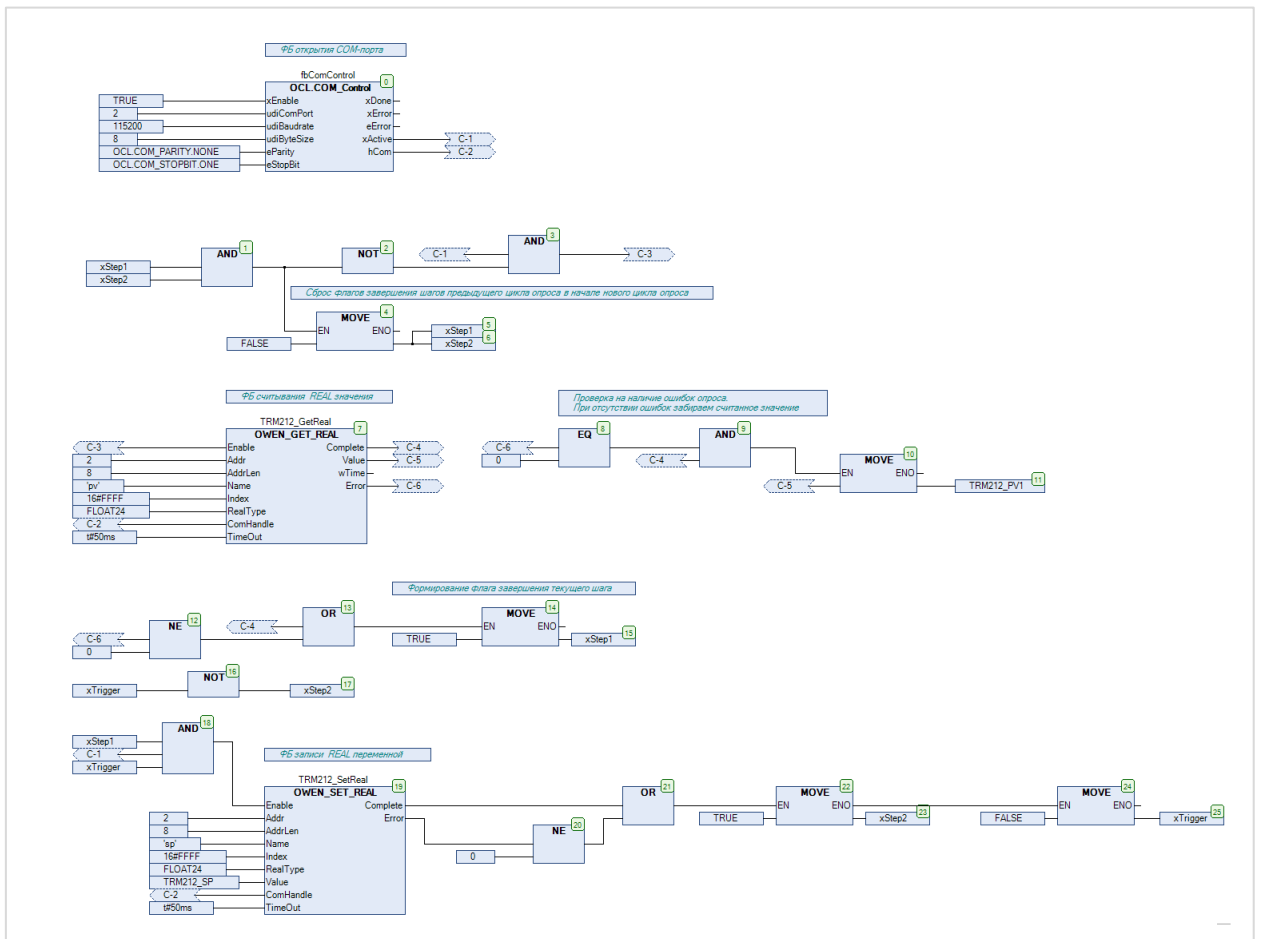


Рисунок 4.4 – Код программы PLC_PRG (CFC)

Алгоритм работы программы:

- **Блок 0.** Открытие COM-порта с настройками в соответствии с таблицей 4.1. Нумерация COM-портов приведена в документа **CODESYS V3.5. FAQ**.
- **Блоки 1–3.** Формируется сигнал инициализации первого шага опроса (чтение переменной).
Условие начала опроса:
 - COM-порт успешно открыт (**xActive=TRUE**);
 - предыдущий цикл опроса завершен (**xStep1=TRUE** и **xStep2=TRUE**).
 Флаги завершения предыдущего цикла опроса инвертируются, так как опрос модулей начинается по переднему фронту входа **Enable** – соответственно, перед началом нового цикла опроса они должны принять значение **FALSE**.
- **Блоки 4–6.** Сброс флага завершения предыдущего цикла опроса.
- **Блок 7.** Считывание значения с плавающей точкой (измеренное значение первого канала) с TPM212 с помощью ФБ [OWEN_GET_REAL](#).
На вход **ComHandle** данного ФБ передается значение выхода **hCom** ФБ [COM_CONTROL](#).
На входе **Addr** задается адрес канала (т. к. параметр является **оперативным**) – согласно таблице 4.1, TPM212 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 таблице 4.1, данный параметр имеет формат **FLOAT24**.
- **Блоки 8–11.** 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**.



ПРИМЕЧАНИЕ

Забирать данные с выхода ФБ [OWEN_GET_REAL](#) можно только по флагу **Complete**.

- **Блоки 12–15.** 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).
- **Блоки 16–17.** Сброс взведенного ранее флага завершения записи в случае поступления новой команды записи.
- **Блок 18.** Формируется сигнал инициализации второго шага опроса (запись переменной).
Условия записи:
 - COM-порт успешно открыт (**xActive=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 TPM.

- **Блок 19.** Запись значения с плавающей точкой (уставка регулятора) в TPM212 с помощью ФБ [OWEN_SET_REAL](#).
На вход **ComHandle** данного ФБ передается значение выхода **hCom** ФБ [COM_CONTROL](#).
На входе **Addr** задается базовый адрес устройства (т. к. параметр является **конфигурационным**) – согласно таблице 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 таблице 4.1, данный параметр имеет формат **FLOAT24**. К входу **Value** привязывается переменная, значение которой будет записано в TPM212.

- **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 опроса завершен, то выполняется сброс триггера записи.

**ПРИМЕЧАНИЕ**

В случае необходимости количество шагов можно увеличить – для этого следует увеличить число переменных **xStep№**.

**ПРИМЕЧАНИЕ**

Приступать к выполнению следующего шага можно только после завершения текущего (т. е. после того, как переменная **xStep№** текущего шага принимает значение **TRUE**).

**ПРИМЕЧАНИЕ**

Выполнение каждого шага занимает несколько циклов программы.

4.2 Описание реализации на языке ST

1. Переменные программы PLC_PRG:

```

1  PROGRAM PLC_PRG
2  VAR
3      fbComControl:      OCL.COM_Control;           // #Б настройки и открытия порта COM2
4
5      iStep:             INT;                       // счетчик шагов
6
7      TRM212_GetReal:    OWEN_GET_REAL;            // #Б считывания REAL значения
8      TRM212_SetReal:    OWEN_SET_REAL;            // #Б записи REAL значения
9
10     TRM212_FV1:        REAL;                       // значение, считываемое с TPM (измеренная величина 1-го входа)
11     TRM212_SP:         REAL;                       // значение, записываемое в TPM (уставка регулятора)
12     xTrigger:          BOOL;                       // триггер записи данных
13 END_VAR
14

```

Рисунок 4.4 – Объявление программы PLC_PRG (ST)

2. Код программы PLC_PRG на языке ST (рисунок хорошо масштабируется; листине программы приведен в [приложении Б](#)):

```

1 // [1] открываем COM-порт
2 fbComControl
3 (
4   xEnable      := TRUE,
5   udiComPort   := 2,
6   udiBaudrate  := 115200,
7   udiByteSize  := 8,
8   eParity      := OCL.COM_PARITY.NONE,
9   eStopBit     := OCL.COM_STOPBIT.ONE
10 );
11
12
13 // [2] xStep определяет номер текущего шага опроса
14 CASE iStep OF
15
16   0:
17
18     // [2.0.1] запускаем ФБ чтения REAL переменной
19     TRM212_GetReal
20     (
21       Enable      := fbComControl.xActive,
22       Addr        := 2,
23       AddrLen     := 8,
24       Name        := 'pv',
25       Index       := 16#FFFF,
26       RealType    := FLOAT24,
27       ComHandle   := fbComControl.hCom,
28       Timeout     := T#50MS,
29     );
30
31     // [2.0.2] если ФБ чтения завершил работу (успешно или с ошибкой)...
32     IF TRM212_GetReal.Complete OR TRM212_GetReal.Error <> 0 THEN
33
34       // ...и ошибки отсутствуют, то забираем считанное значение
35       IF TRM212_GetReal.Error = 0 THEN
36         TRM212_FV1 := TRM212_GetReal.Value;
37       END_IF
38
39       // завершаем работу блока
40       TRM212_GetReal(Enable := FALSE);
41
42       // переходим к следующему шагу
43       iStep := 1;
44     END_IF
45
46   1:
47
48     // [2.1.0] если триггер записи активен...
49     IF xTrigger THEN
50
51       //... то запускаем ФБ записи REAL переменной
52       TRM212_SetReal
53       (
54         Enable:=fbComControl.xActive,
55         Addr      := 2,
56         AddrLen   := 8,
57         Name      := 'sp',
58         Index     := 16#FFFF,
59         RealType  := FLOAT24,
60         Value     := TRM212_SP,
61         ComHandle := fbComControl.hCom,
62         Timeout   := T#50MS,
63       );
64
65       // [2.1.1] если ФБ записи завершил работу (успешно или с ошибкой)...
66       IF TRM212_SetReal.Complete OR TRM212_SetReal.Error <> 0 THEN
67
68         // завершаем работу ФБ
69         TRM212_SetReal(Enable := FALSE);
70
71         // сбрасываем триггер записи
72         xTrigger := FALSE;
73
74         // переходим к следующему шагу
75         iStep := 2;
76       END_IF
77
78     ELSE
79       iStep := 2; // если триггер записи не активирован - то сразу переходим к следующему шагу
80     END_IF
81
82   2:
83     // [2.2.0]
84     // читаем/записываем другие параметры этого устройства или других устройств (пошагово)
85     // после конечной операции переходим к начальному шагу
86
87     iStep := 0;
88   END_CASE
89 END_CASE
90

```

Рисунок 4.5 – Код программы PLC_PRG (ST)

Алгоритм работы программы:

- **Блок [1].** Открытие COM-порта с настройками в соответствии с таблицей 4.1. Нумерация COM-портов приведена в документа **CODESYS V3.5. FAQ**.
- **Блок [2].** Реализация последовательного выполнения шагов опроса через оператор **CASE**:
 - в случае **iStep=0** происходит считывание измеренного значения первого канала;
 - в случае **iStep=1** происходит запись уставки регулятора.
- **Блок [2.0.1].** Считывание значения с плавающей точкой (измеренное значение первого канала) с TPM212 с помощью ФБ [OWEN_GET_REAL](#).
На вход **ComHandle** данного ФБ передается значение выхода **hCom** ФБ [COM_CONTROL](#).
На входе **Addr** задается адрес канала (т. к. параметр является **оперативным**) – согласно таблице 4.1, TPM212 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 таблице 4.1, данный параметр имеет формат **FLOAT24**.
- **Блок [2.0.2].** 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**.



ПРИМЕЧАНИЕ

Забирать данные с выхода ФБ [OWEN_GET_REAL](#) можно только по флагу **Complete**.

Независимо от наличия ошибок следует завершить данный шаг опроса и перейти к следующему.

- **Блок [2.1.0].** Если триггер записи активирован (**xTrigger=TRUE**), то производится запись значения с плавающей точкой (уставка регулятора) в TPM212 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 TPM.

На вход **ComHandle** данного ФБ передается значение выхода **hCom** ФБ [COM_CONTROL](#).
На входе **Addr** задается базовый адрес устройства (т. к. параметр является **конфигурационным**) – согласно таблице 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 таблице 4.1, данный параметр имеет формат **FLOAT24**. К входу **Value** привязывается переменная, значение которой будет записано в TPM212.

- **Блок [2.1.1].** 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 происходит завершение текущего шага опроса, сброс триггер записи и переход к следующему шагу. Если триггер записи на данном цикле не взводился, то также происходит переход к следующему шагу.
- **Блок [2.2.0].** После завершения последнего шага опроса происходит переход к начальному шагу.



ПРИМЕЧАНИЕ

В случае необходимости количество шагов можно увеличить – для этого следует увеличить количество действий в операторе **CASE**.



ПРИМЕЧАНИЕ

Перед переходом к следующему шагу опроса следует завершить текущий, вызвав ФБ опроса с **Enable=FALSE**.



ПРИМЕЧАНИЕ

Выполнение каждого шага занимает несколько циклов программы.

4.3 Запуск примера

Проект следует загрузить в контроллер и запустить. Подразумевается, что у TPM212 заданы сетевые настройки в соответствии с таблицей 4.1. На верхний цифровой индикатор следует вывести измеренное значение первого канала, на нижний – значение уставки регулятора.

Затем следует изменить значение переменной **TRM212_SP** и присвоить переменной **xTrigger** значение **TRUE** – это приведет к изменению уставки TPM.

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

Выражение	Тип	Значение	Подготовленное ...	Адрес	Комментарий
fbComControl	OCL.COM_Control				ФБ настройки и открытия порта COM2
iStep	INT	0			счетчик шагов
TRM212_GetReal	OWEN_GET_REAL				ФБ считывания REAL значения
TRM212_SetReal	OWEN_SET_REAL				ФБ записи REAL значения
TRM212_PV1	REAL	23.5			значение, считываемое...PM (измеренная вели...
TRM212_SP	REAL	23.5			значение, записываемое в TPM (уставка регуля...
xTrigger	BOOL	FALSE			триггер записи данных

Рисунок 4.6 – Измеренное значение первого канала TPM212 и ввод уставки в CODESYS



Рисунок 4.7 – Отображение измеренного значения первого канала TPM212 и его уставки на индикаторах

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 <code>modific</code>		
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 <code>ModAllPermis()</code> и <code>ModEditPermis()</code>		
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

Код ошибки	Имя ошибки	Описание
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)
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

Приложение Б. Листинг программы из п. 4

```

PROGRAM PLC_PRG
VAR
    fbComControl:    OCL.COM_Control;    // ФБ настройки и открытия 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 (величина первого входа)
    TRM212_SP:       REAL;              // значение, записываемое в ТРМ (уставка регулятора)

    xTrigger:        BOOL;              // триггер записи данных
END_VAR

// [1] открываем COM-порт
fbComControl
(
    xEnable          := TRUE,
    udiComPort       := 2,
    udiBaudrate      := 115200,
    udiByteSize      := 8,
    eParity          := OCL.COM_PARITY.NONE,
    eStopBit         := OCL.COM_STOPBIT.ONE
);

// [2] xStep определяет номер текущего шага опроса
CASE iStep OF

    0:
        // [2.0.1] запускаем ФБ чтения REAL переменной
        TRM212_GetReal
        (
            Enable:= fbComControl.xActive,
            Addr:=2,
            AddrLen:=8,
            Name:='pv',
            Index:=16#FFFF,
            RealType:=FLOAT24,
            ComHandle:=fbComControl.hCom,
            TimeOut:=T#50MS,
            Complete=> ,
            Value=> ,
            wTime=> ,
            Error=>
        );

        // [2.0.2] 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
END_CASE

```



```
1:
// [2.1.0] если триггер записи активен...
IF xTrigger THEN

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

// [2.1.1] 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:
// [2.2.0]
// читаем/записываем другие параметры этого устройства или других устройств
// (пошагово)
// после конечной операции переходим к начальному шагу

iStep:=0;

END_CASE
```