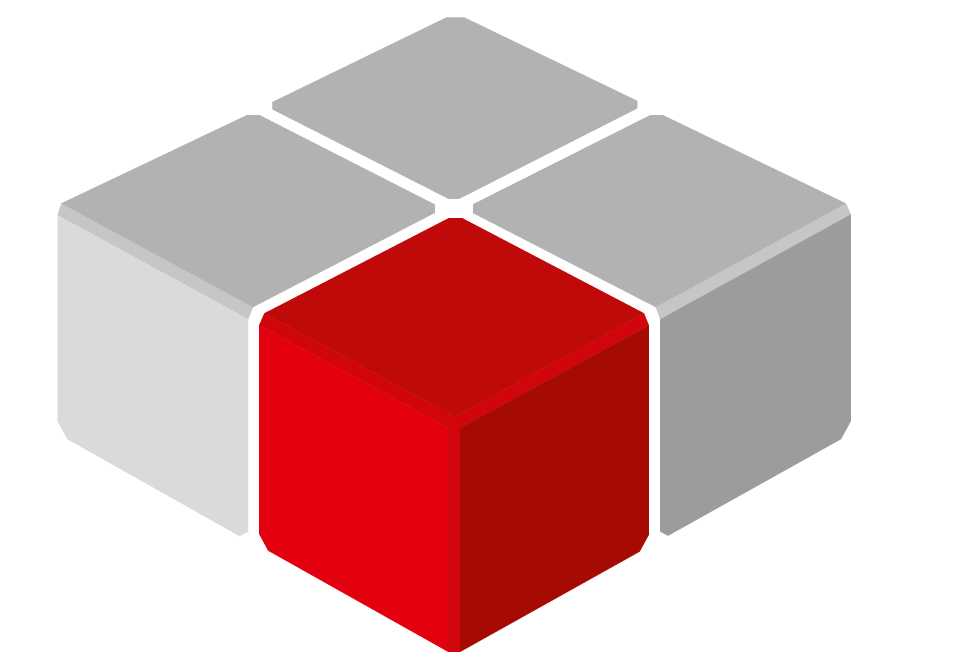


**CODESYS V3.5**

**Протокол ОВЕН**



Руководство пользователя

25.02.2020

версия 2.1

Оглавление

[Глоссарий 3](#_Toc32914175)

[1 Цель и структура документа 3](#_Toc32914176)

[2 Основные сведения о протоколе ОВЕН 4](#_Toc32914177)

[3 Описание библиотеки OwenNet 5](#_Toc32914178)

[3.1 Установка библиотеки 5](#_Toc32914179)

[3.2 Добавление библиотеки в проект CODESYS 6](#_Toc32914180)

[3.3 Описание библиотеки 8](#_Toc32914181)

[3.3.1 Блок COM\_CONTROL (библиотека OwenCommunication) 8](#_Toc32914182)

[3.3.2 Блок OWEN\_GET\_DINT 9](#_Toc32914183)

[3.3.3 Блок OWEN\_SET\_DINT 10](#_Toc32914184)

[3.3.4 Блок OWEN\_GET\_REAL 11](#_Toc32914185)

[3.3.5 Блок OWEN\_SET\_REAL 12](#_Toc32914186)

[3.3.6 Блок OWEN\_UNI\_IO 13](#_Toc32914187)

[3.3.7 Блок OWEN\_LISTEN 15](#_Toc32914188)

[3.3.8 Блок OwenPoolMaster 16](#_Toc32914189)

[3.3.9 Структура OwenPool 17](#_Toc32914190)

[3.3.10 Перечисление INT\_FRM 18](#_Toc32914191)

[3.3.11 Перечисление REAL\_FRM 18](#_Toc32914192)

[3.3.12 Перечисление OWEN\_FRM 18](#_Toc32914193)

[4 Пример: СПК1хх [М01] + ТРМ212 19](#_Toc32914194)

[4.1 Описание реализации на языке CFC 21](#_Toc32914195)

[4.2 Описание реализации на языке ST 23](#_Toc32914196)

[4.3 Запуск примера 27](#_Toc32914197)

[Приложение A. Список кодов ошибок при обмене по протоколу ОВЕН 29](#_Toc32914199)

[Приложение Б. Листинг программы из п. 4 31](#_Toc32914200)

# Глоссарий

**ПЛК** – программируемый логический контроллер.

**ФБ** – функциональный блок.

# 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](http://www.owen.ua/) 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**.

Руководство содержит описание библиотеки и пример ее использования для опроса **ТРМ212** (на языках **CFC** и **ST**).

# Основные сведения о протоколе ОВЕН

Протокол **ОВЕН** основан на архитектуре **Master-Slave** (ведущий-ведомый) и реализуется поверх последовательного интерфейса [RS-485](https://ru.wikipedia.org/wiki/RS-485).

Спецификация протокола доступна на сайте компании [ОВЕН](http://www.owen.ua/) 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](#_4._Пример:_СПК207)) 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.).

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

# Описание библиотеки OwenNet

## Установка библиотеки

Библиотека **OwenNet** доступна на диске с ПО из комплекта поставки и сайте компании [ОВЕН](http://www.owen.ua/) 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:

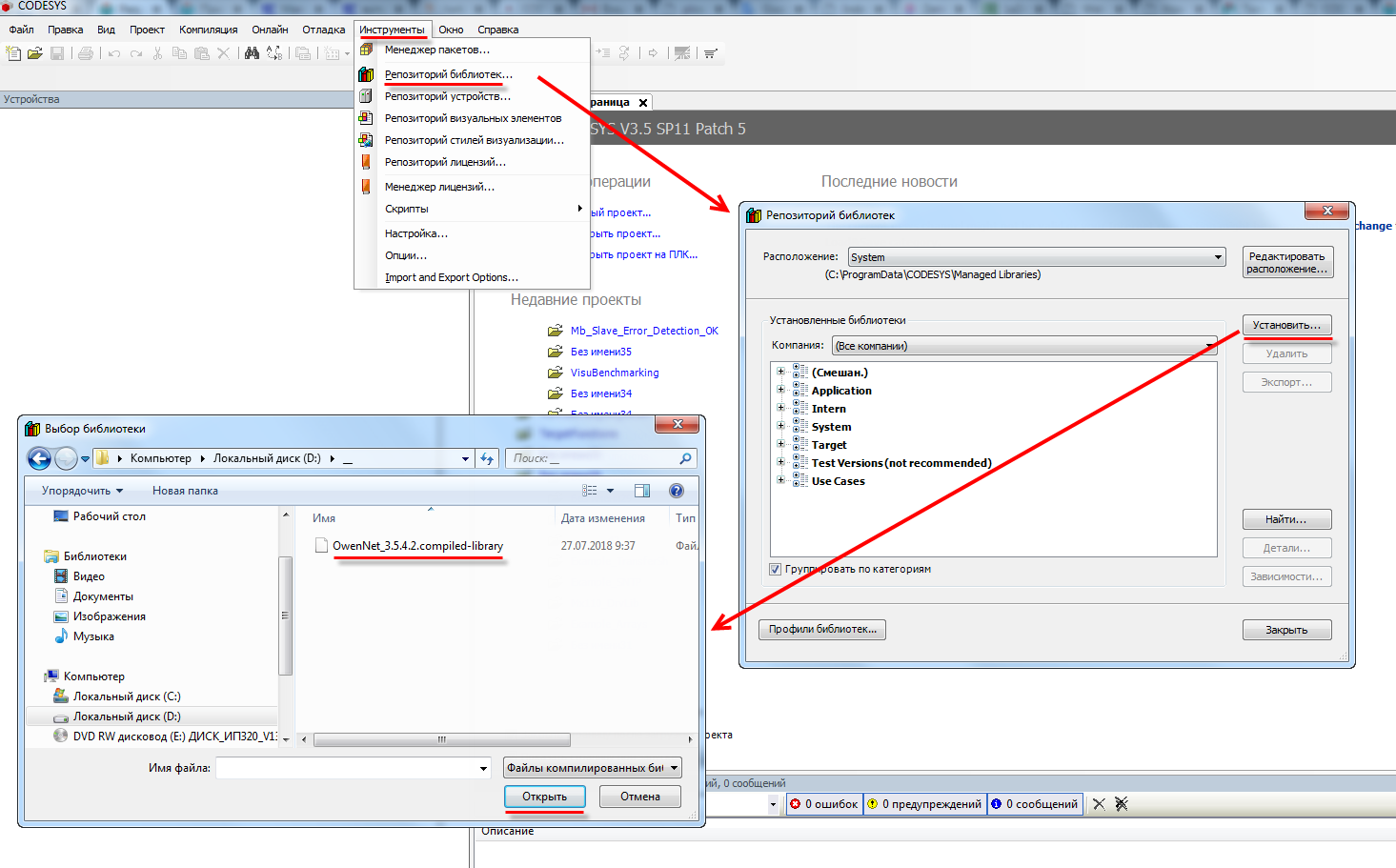


Рисунок 3.1 – Установка библиотеки OwenNet в среду CODESYS

## 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**.

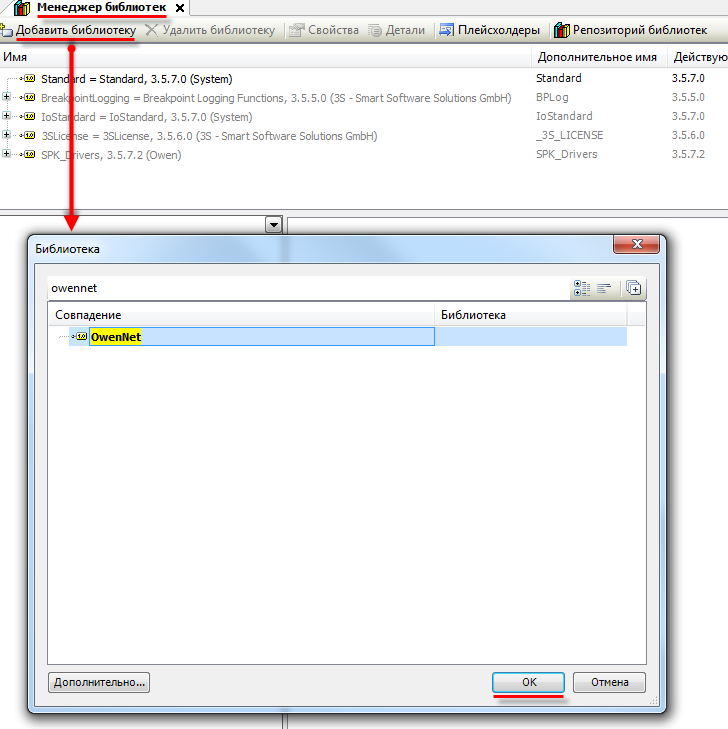


Рисунок 3.2 – Добавление библиотеки OwenNet

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

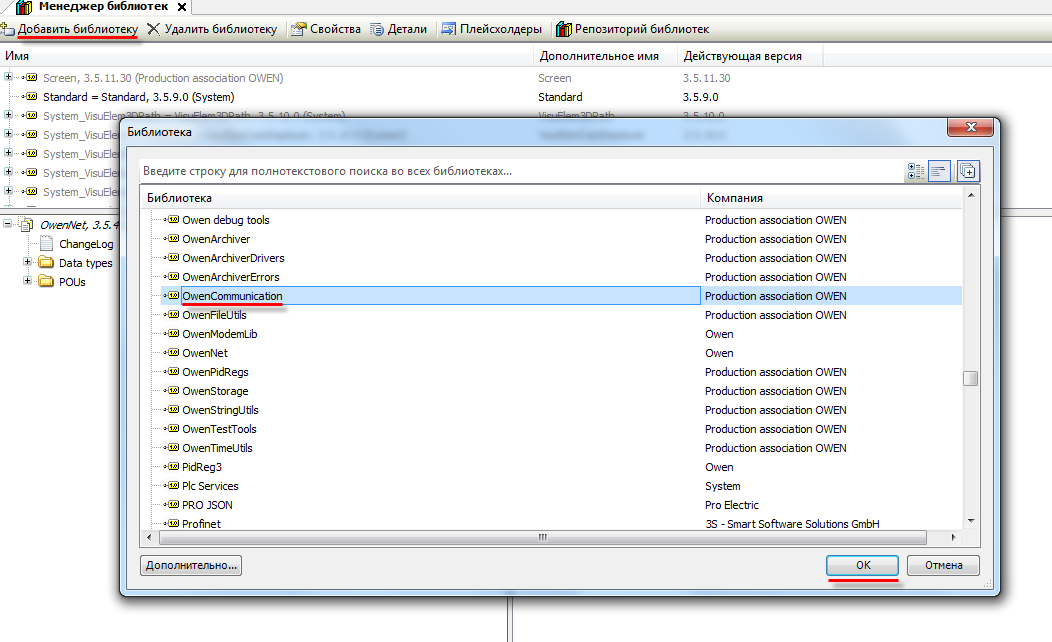
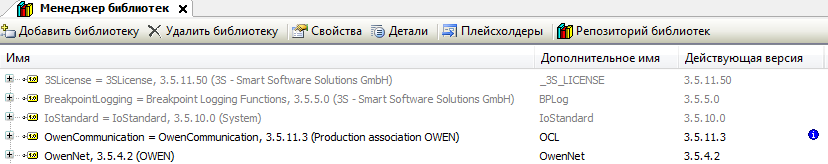


Рисунок 3.3 – Добавление библиотеки OwenCommunication

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

Рисунок 3.4 – Список библиотек проекта

## Описание библиотеки

### Блок COM\_CONTROL (библиотека OwenCommunication)

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

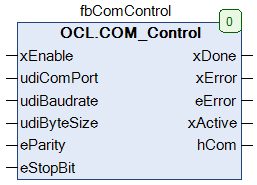


Рисунок 3.5 – Внешний вид ФБ COM\_CONTROL на языке CFC

**Таблица 3.1 – Описание входов и выходов ФБ СOM\_CONTROL**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| **Входы** | | |
| xEnable | BOOL | По переднему фронту происходит открытие COM-порта, по заднему – закрытие |
| udiComPort | UDINT | [Номер COM-порта](#_3.1._Соответствие_нумерации) |
| 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 |

### Блок OWEN\_GET\_DINT

Функциональный блок **OWEN\_GET\_DINT** используется для считывания целочисленного значения.

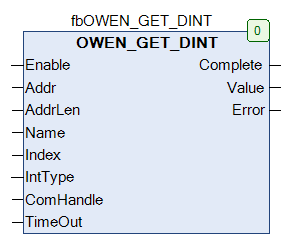


Рисунок 3.6 – Внешний вид ФБ OWEN\_GET\_DINT на языке CFC

**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. Cписок доступных форматов приведен в перечислении [INT\_FRM](#_3.3.11._Перечисление_INT_FRM). |
| СomHandle | DWORD | Идентификатор порта, поступающий с выхода блока [COM\_CONTROL](#_6.3.1._Блок_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 | DINT | Cчитанное 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](#_A._Список_кодов). Список кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор |

### Блок OWEN\_SET\_DINT

Функциональный блок **OWEN\_SET\_DINT** используется для записи целочисленного значения.

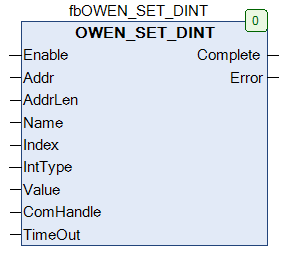


Рисунок 3.7 – Внешний вид ФБ OWEN\_SET\_DINT на языке CFC

**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. Cписок доступных форматов приведен в перечислении [INT\_FRM](#_3.3.11._Перечисление_INT_FRM) |
| Value | DINT | Записываемое значение |
| СomHandle | DWORD | Идентификатор порта, поступающий с выхода блока [ComService](#_6.3.1._Блок_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](#_A._Список_кодов). Список кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор |

### Блок OWEN\_GET\_REAL

Функциональный блок **OWEN\_GET\_REAL** используется для считывания значения с плавающей точкой.

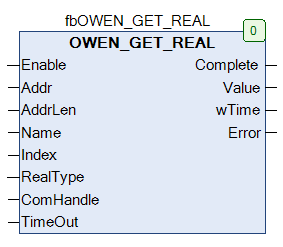


Рисунок 3.8 – Внешний вид ФБ OWEN\_GET\_REAL на языке CFC

**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. Cписок доступных форматов приведен в перечислении [REAL\_FRM](#_3.3.12._Перечисление_REAL_FRM) |
| СomHandle | DWORD | Идентификатор порта, поступающий с выхода блока [ComService](#_6.3.1._Блок_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 | Cчитанное 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](#_A._Список_кодов). Список кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор |

### Блок OWEN\_SET\_REAL

Функциональный блок **OWEN\_SET\_REAL** используется для записи значения с плавающей точкой.

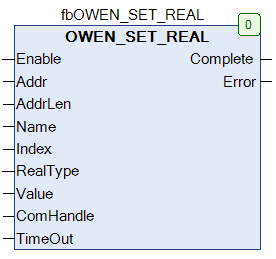


Рисунок 3.9 – Внешний вид ФБ OWEN\_SET\_REAL на языке CFC

**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. Cписок доступных форматов приведен в перечислении [REAL\_FRM](#_3.3.12._Перечисление_REAL_FRM) |
| Value | REAL | Записываемое значение |
| СomHandle | DWORD | Идентификатор порта, поступающий с выхода блока [ComService](#_6.3.1._Блок_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](#_A._Список_кодов). Список кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор |

### Блок OWEN\_UNI\_IO

Функциональный блок **OWEN\_UNI\_IO** используется для чтения или записи нетипизированных данных.

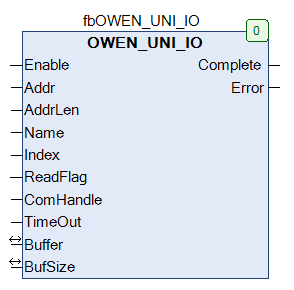


Рисунок 3.10 – Внешний вид ФБ OWEN\_UNI\_IO на языке CFC

**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 |
| СomHandle | DWORD | Идентификатор порта, поступающий с выхода блока [ComService](#_6.3.1._Блок_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](#_A._Список_кодов). Список кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор |
| ***Входные-выходные переменные (VAR\_IN\_OUT)*** | | |
| Buffer | ARRAY [0..14] OF BYTE | Cчитанные данные или данные, подготовленные для записи |
| BufSize | BYTE | Количество считанных/переданных байт |

### 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.

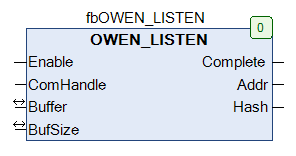


Рисунок 3.11 – Внешний вид ФБ OWEN\_LISTEN на языке CFC

**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 |
| СomHandle | DWORD | Идентификатор порта, поступающий с выхода блока [ComService](#_6.3.1._Блок_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](http://www.owen.ru/) 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](#_A._Список_кодов). Список кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор |
| ***Входные-выходные переменные (VAR\_IN\_OUT)*** | | |
| Buffer | ARRAY [0..14] OF BYTE | Cчитанные данные |
| BufSize | BYTE | Количество считанных байт |

### Блок OwenPoolMaster

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

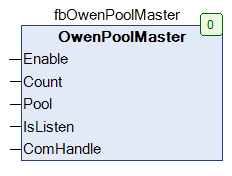


Рисунок 3.12 – Внешний вид ФБ OwenPoolMaster на языке CFC

**Таблица 3.8 – Описание входов и выходов ФБ OwenPoolMaster**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| ***Входные переменные*** | | |
| 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](#_3.3.10_Структура_OwenPool). Каждая структура представляет собой выполняемую команду |
| IsListen | BOOL | При значении **TRUE** блок работает в режиме прослушивания сети |
| СomHandle | DWORD | Идентификатор порта, поступающий с выхода блока [ComService](#_6.3.1._Блок_ComService) (или аналогичного) |

### Структура OwenPool

Структура **OwenPool** используется при работе с ФБ [OwenPoolMaster](#_3.3.8._Блок_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. Cписок доступных форматов приведен в перечислении [OWEN\_FRM](#_3.3.13._Перечисление_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](#_A._Список_кодов). Список кодов ошибок прибора приведен в руководстве по эксплуатации на конкретный прибор |

### Перечисление INT\_FRM

Перечисление **INT\_FRM** описывает форматы целочисленных данных. Используется во время работы с ФБ [OWEN\_GET\_DINT](#_3.3.2._Блок_OWEN_GET_DINT) и [OWEN\_SET\_DINT](#_3.3.3._Блок_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) |

### Перечисление REAL\_FRM

Перечисление **REAL\_FRM** описывает форматы данных с плавающей точкой. Используется во время работы с ФБ [OWEN\_GET\_REAL](#_3.3.4._Блок_OWEN_GET_REAL) и [OWEN\_SET\_REAL](#_3.3.5._Блок_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](https://ru.wikipedia.org/wiki/IEEE_754-2008) + время относительного измерения в мс (6 байта) |
| FLOAT32 | 1 | Значение с плавающей точкой по стандарту [IEEE 754](https://ru.wikipedia.org/wiki/IEEE_754-2008) (4 байта) |
| FLOAT24 | 2 | Укороченное значение с плавающей точкой по стандарту [IEEE 754](https://ru.wikipedia.org/wiki/IEEE_754-2008) (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](https://ru.wikipedia.org/wiki/%D0%94%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D0%BE-%D0%B4%D0%B5%D1%81%D1%8F%D1%82%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) |

### Перечисление OWEN\_FRM

Перечисление **OWEN\_FRM** описывает форматы данных, используемых во время работы с ФБ [OwenPoolMaster](#_3.3.8._Блок_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](https://ru.wikipedia.org/wiki/IEEE_754-2008) + время относительного измерения в мс (6 байта) |
| OW\_FL32 | 1 | Значение с плавающей точкой по стандарту [IEEE 754](https://ru.wikipedia.org/wiki/IEEE_754-2008) (4 байта) |
| OW\_FL24 | 2 | Укороченное значение с плавающей точкой по стандарту [IEEE 754](https://ru.wikipedia.org/wiki/IEEE_754-2008) (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](https://ru.wikipedia.org/wiki/%D0%94%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D0%BE-%D0%B4%D0%B5%D1%81%D1%8F%D1%82%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) |
| OW\_SINT1 | 5 | Знаковое целое (1 байт) |
| OW\_SINT2 | 6 | Знаковое целое (2 байта) |
| OW\_SINT4 | 7 | Знаковое целое (4 байта) |
| OW\_UINT | 8 | Беззнаковое целое (4 байта) |
| OW\_NTYPE | 9 | Нетипизированное значение |

# Пример: СПК1хх [М01] + ТРМ212

В данном разделе приведен пример настройки обмена между контроллером **СПК1хх [М01]** (master) и **ТРМ212** (slave) с использованием библиотеки **OwenNet**.

Сетевые настройки приборов приведены в таблице ниже.

**Таблица 4.1 – Сетевые параметры СПК и ТРМ**

|  |  |  |
| --- | --- | --- |
| **Параметр** | **СПК1хх [М01]** | **ТРМ212** |
| *COM-порт ПЛК, к которому подключен ТРМ212* | *COM2* | *-* |
| *Базовый адрес ТРМ212* | *-* | *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](http://www.owen.ru/) на странице соответствующего прибора:

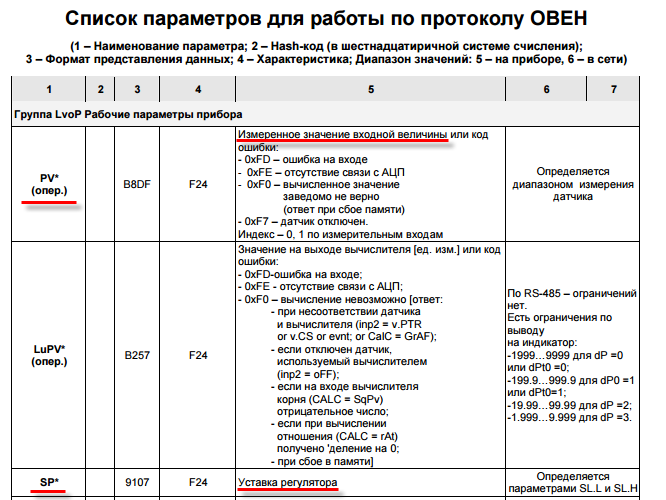


Рисунок 4.1 – Cписок параметров ТРМ212 по протоколу ОВЕН

Характеристики параметров приведены в таблице ниже.

**Таблица 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 |

|  |  |
| --- | --- |
|  | **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 **СПК1хх [М01]** 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](https://owen.ua/uploads/115/example_trm212_3511v2.projectarchive)

## Описание реализации на языке CFC

1. Переменные программы **PLC\_PRG**:

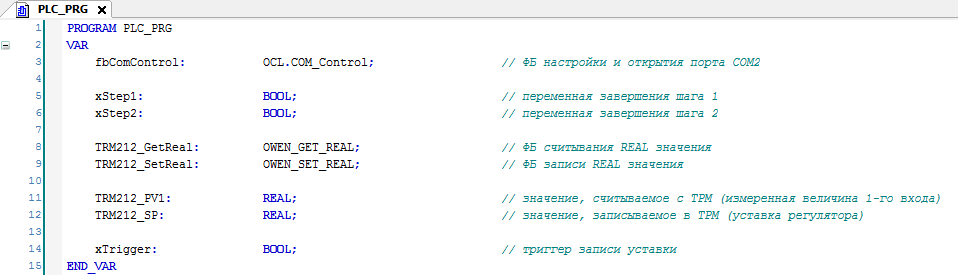


Рисунок 4.2 – Объявление программы PLC\_PRG (CFC)

1. Код программы **PLC\_PRG** на языке **CFC** (*рисунок хорошо масштабируется*):

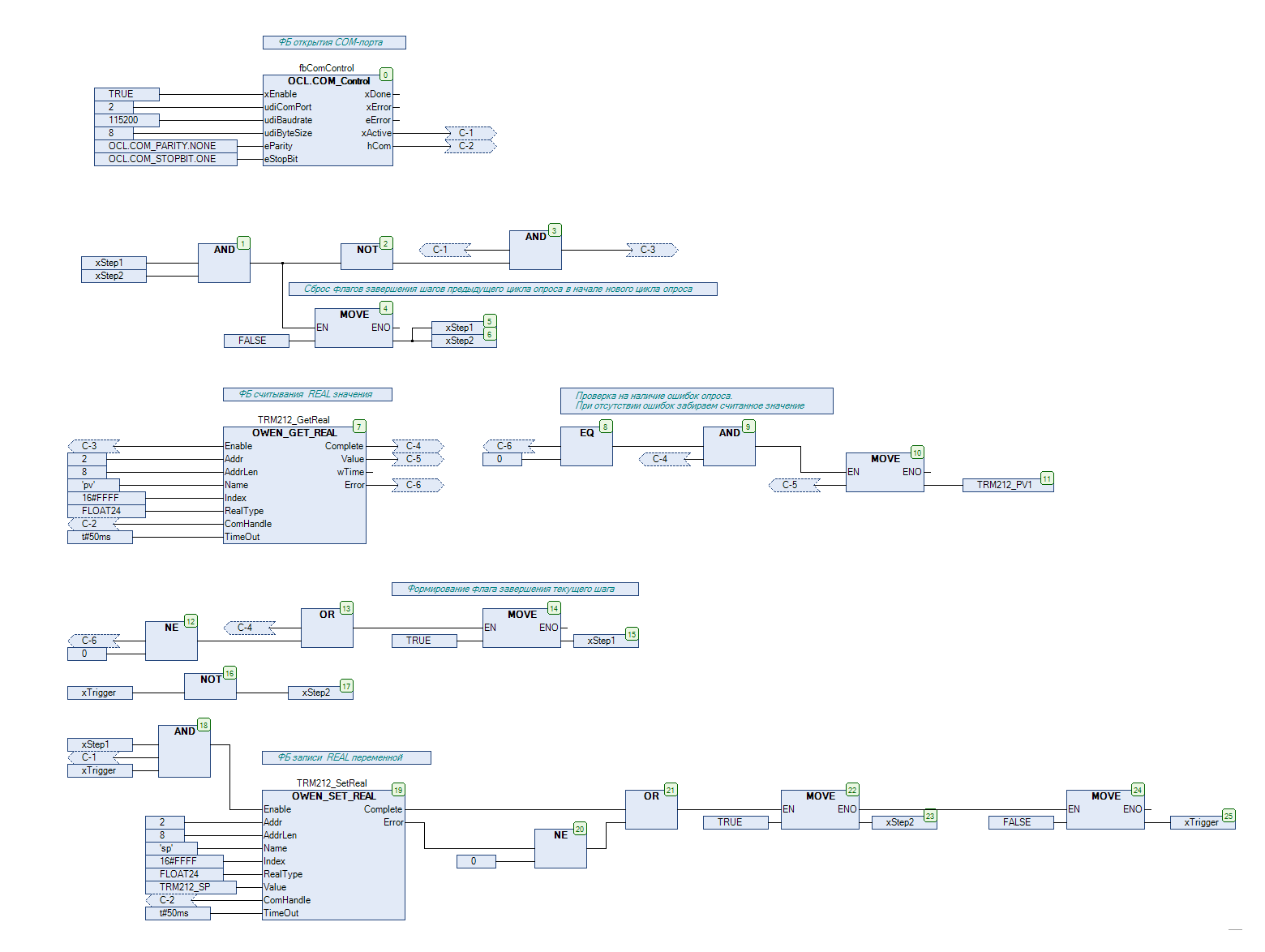


Рисунок 4.4 – Код программы PLC\_PRG (CFC)

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 ТРМ212 с помощью ФБ [OWEN\_GET\_REAL](#_3.3.4._Блок_OWEN_GET_REAL).

На вход **СomHandle** данного ФБ передается значение выхода **hCom** ФБ [COM\_CONTROL](#_Блок_COM_CONTROL_(библиотека). На входе **Addr** задается адрес канала (т. к. параметр является **оперативным**) – согласно таблице 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 таблице 4.1, данный параметр имеет формат **FLOAT24**.

* **Блоки 8–11**. Если первый шаг опросазавершен (об этом сигнализирует импульс по переднему фронту на выходе **Сomplete)** без ошибок (**Error=0**), то считанное значение копируется в переменную программы **TRM212\_PV1**.

|  |  |
| --- | --- |
|  | **ПРИМЕЧАНИЕ**  Забирать данные с выхода ФБ [OWEN\_GET\_REAL](#_3.3.4._Блок_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 ТРМ.

* **Блок 19**. Запись значения с плавающей точкой (уставка регулятора) в ТРМ212 с помощью ФБ [OWEN\_SET\_REAL](#_3.3.5._Блок_OWEN_SET_REAL).

На вход **СomHandle** данного ФБ передается значение выхода **hCom** ФБ [COM\_CONTROL](#_Блок_COM_CONTROL_(библиотека). 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 ТРМ212.

* **Блоки 20–23**. Если второй шаг опросазавершен (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. |

## Описание реализации на языке ST

1. Переменные программы **PLC\_PRG**:

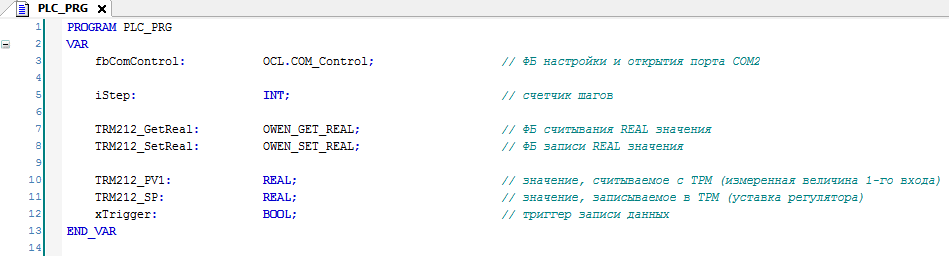


Рисунок 4.4 – Объявление программы PLC\_PRG (ST)

1. Код программы **PLC\_PRG** на языке **ST** (*рисунок хорошо масштабируется; листинг программы приведен в* [*приложении Б*](#_Б._Листинг_программы)):



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 ТРМ212 с помощью ФБ [OWEN\_GET\_REAL](#_3.3.4._Блок_OWEN_GET_REAL).

На вход **СomHandle** данного ФБ передается значение выхода **hCom** ФБ [COM\_CONTROL](#_Блок_COM_CONTROL_(библиотека). На входе **Addr** задается адрес канала (т. к. параметр является **оперативным**) – согласно таблице 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 таблице 4.1, данный параметр имеет формат **FLOAT24**.

* **Блок [2.0.2]**. Если данный шаг опросауспешно завершен (об этом сигнализирует импульс по переднему фронту на выходе **Сomplete**) без ошибок (**Error=0**), то считанное значение копируется в переменную программы **TRM212\_PV1**.

|  |  |
| --- | --- |
|  | **ПРИМЕЧАНИЕ**  Забирать данные с выхода ФБ [OWEN\_GET\_REAL](#_3.3.4._Блок_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 ТРМ212 с помощью ФБ [OWEN\_SET\_REAL](#_3.3.5._Блок_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 **СomHandle** данного ФБ передается значение выхода **hCom** ФБ [COM\_CONTROL](#_Блок_COM_CONTROL_(библиотека). 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 ТРМ212.

* **Блок [2.1.1]**. Если данный шаг опросазавершен (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. |

## Запуск примера

Проект следует загрузить в контроллер и запустить. Подразумевается, что у ТРМ212 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 ТРМ.

В переменную **TRM212\_PV1** будет считано измеренное значение первого канала.

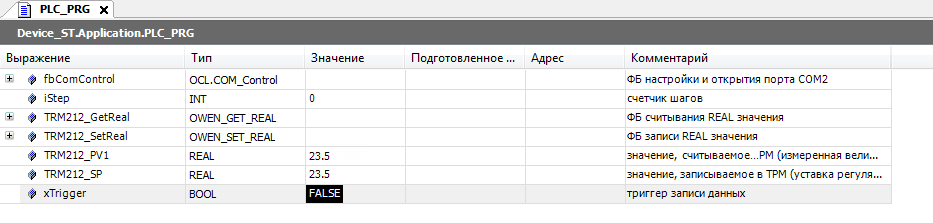


Рисунок 4.6 – Измеренное значение первого канала ТРМ212 и ввод уставки в CODESYS



Рисунок 4.7 – Отображение измеренного значения первого канала ТРМ212 и его уставки на индикаторах

# Приложение A. 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 |

|  |  |  |
| --- | --- | --- |
| **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

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