



# **CODESYS V3.5**

**Настройка обмена по протоколу SNMP**



**Руководство пользователя**

27.05.2026

версия 2.2

# Оглавление

<b>1</b>	<b>Цель документа.....</b>	<b>3</b>
<b>2</b>	<b>Описание библиотеки OwenSnmp .....</b>	<b>4</b>
2.1	Установка библиотеки .....	4
2.2	Добавление библиотеки в проект CODESYS .....	5
2.3	Типы данных.....	6
2.3.1	Перечисление ERROR.....	6
2.3.2	Перечисление REQUEST_PDU_TYPE .....	8
2.3.3	Перечисление TRAP_PDU_TYPE .....	8
2.3.4	Перечисление VALUE_TYPE .....	8
2.3.5	Структура VARBIND .....	9
2.3.6	Список параметров Settings .....	10
2.4	Функциональные блоки .....	11
2.4.1	ФБ Manager.....	11
2.4.2	ФБ TrapReceiver .....	13
2.4.3	ФБ SendTrap .....	15
2.4.4	ФБ Agent.....	17
2.5	Функции .....	19
2.5.1	Функция REAL_TO_OPAQUE .....	19
2.5.2	Функция LREAL_TO_OPAQUE .....	20
2.5.3	Функция OPAQUE_TO_REAL .....	21
2.5.4	Функция OPAQUE_TO_LREAL.....	22
2.5.5	Функция SWAP_DATA.....	23
<b>3</b>	<b>Примеры.....</b>	<b>24</b>

## 1 Цель документа

Настоящее руководство описывает настройку обмена данными по протоколу **SNMP** для контроллеров ОВЕН, программируемых в среде **CODESYS V3.5**, с использованием библиотеки **OwenSnmplib**. 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**.

**SNMP** (Simple Network Management Protocol) – прикладной протокол, используемый в системах мониторинга сетевого оборудования. Протокол основан на архитектуре «Клиент/Сервер», при этом в терминологии протокола клиенты называются **менеджерами**, а серверы – **агентами**. К числу устройств с поддержкой SNMP относятся коммутаторы, ИБП, рабочие станции (APM), модули ввода-вывода и другие устройства.

Менеджеры могут производить чтение (**GET**) и запись (**SET**) параметров агентов. Агенты могут отправлять менеджерам асинхронные уведомления (**трапы**) – например, о переходе оборудования в аварийное состояние. По спецификации протокола для отправки запросов используется порт **161**, для отправки трапов – **162**.

Каждый параметр агента имеет уникальный идентификатор (**OID**), представляющий собой последовательность цифр, разделенных точками. Для упрощения настройки обмена производители устройств-агентов обычно предоставляют MIB-файлы, которые включают в себя список параметров прибора с их названиями и OID'ами.

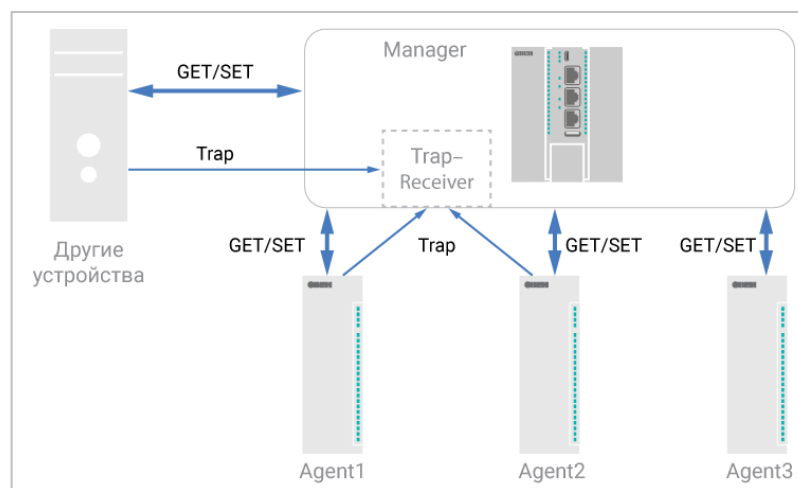


Рисунок 1.1 – Пример структурной схемы обмена по протоколу SNMP

Библиотека **OwenSnmplib** позволяет использовать контроллер в качестве SNMP-менеджера или агента. Поддерживается версия протокола **SNMPv2c**. Поддерживаются запросы GET/SET, а также прием и отправка трапов.



### ПРИМЕЧАНИЕ

Для использования библиотеки требуется версия **CODESYS V3.5 SP14 Patch 3** или выше.



### ПРИМЕЧАНИЕ

Работа библиотеки поддерживается только на контроллерах ОВЕН и виртуальном контроллере **CODESYS Control Win V3**.

## 2 Описание библиотеки OwenSnmr

### 2.1 Установка библиотеки

Библиотека **OwenSnmr** доступна для загрузки на сайте компании [OBEH](#) 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:

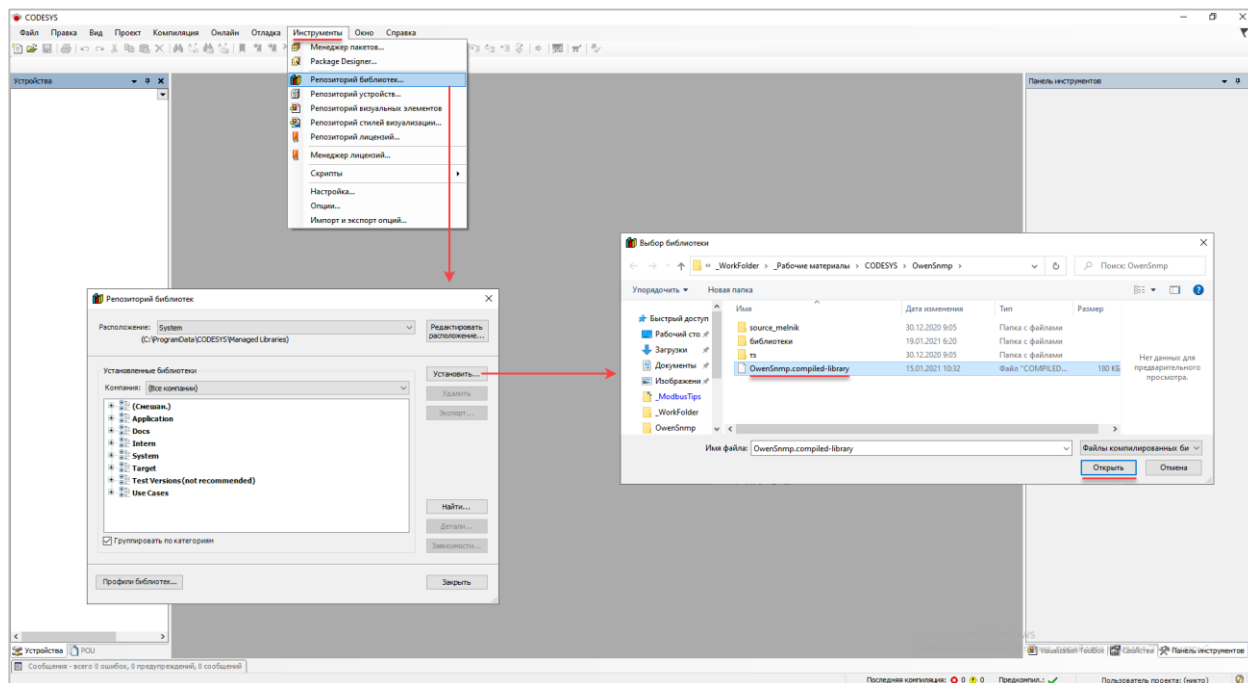


Рисунок 2.1 – Установка библиотеки OwenSNMP в среду CODESYS

## 2.2 Добавление библиотеки в проект CODESYS

Для добавления библиотеки **OwenSNMP** в проект **CODESYS** в **Менеджере библиотек** следует нажать кнопку **Добавить библиотеку**, в появившемся списке выбрать библиотеку **OwenSnmp** и нажать **ОК**.

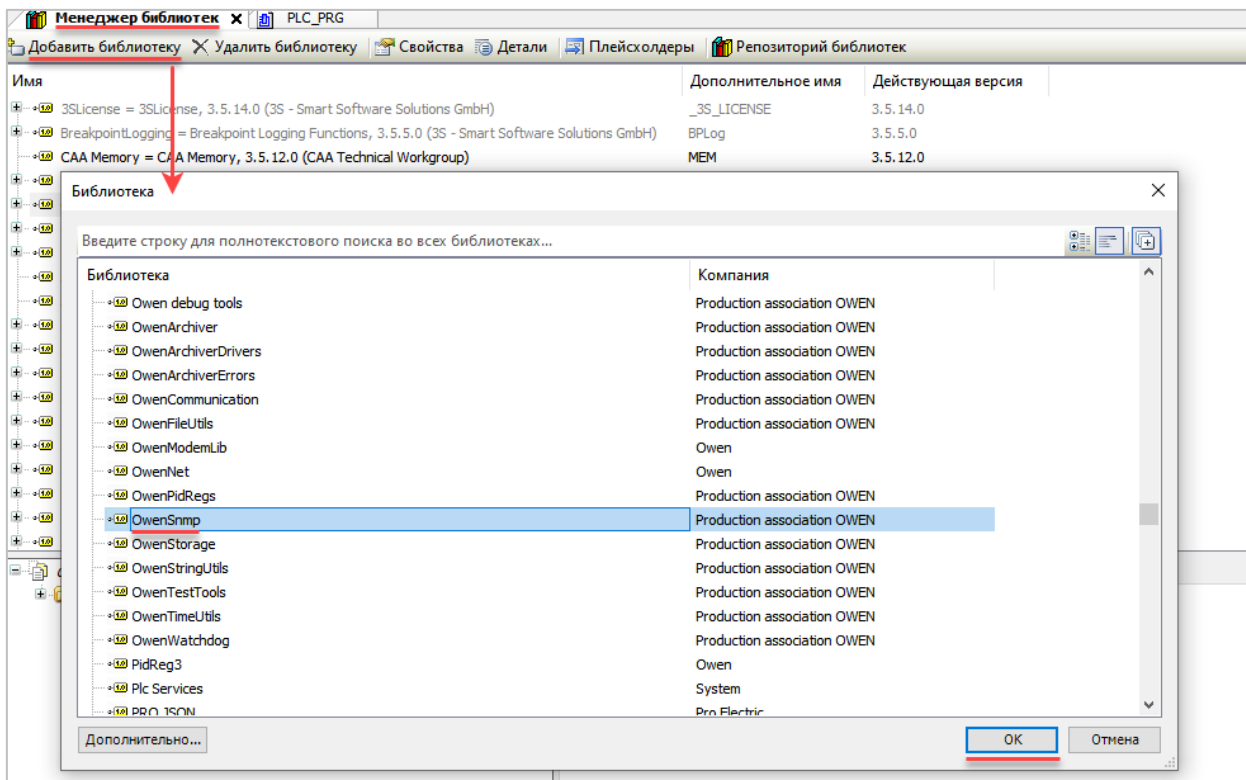


Рисунок 2.2 – Добавление библиотеки OwenSNMP

После добавления библиотека появится в списке **Менеджера библиотек**:

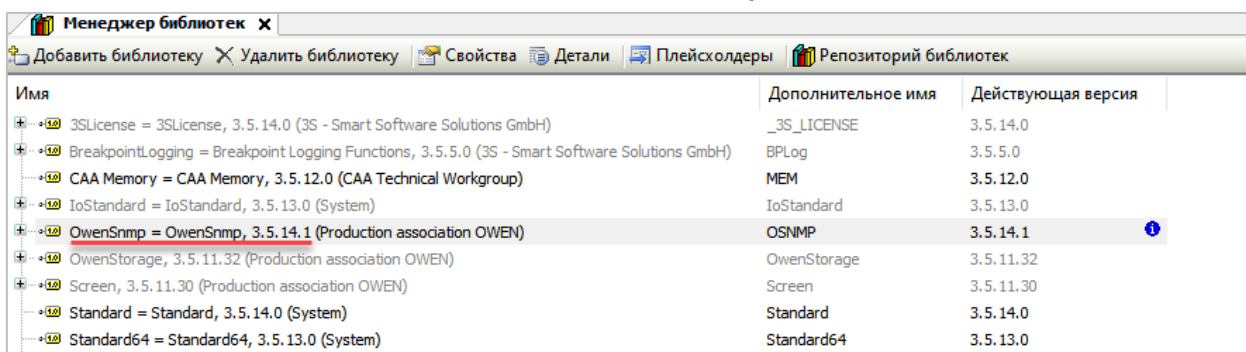


Рисунок 2.3 – Список библиотек проекта



### ПРИМЕЧАНИЕ

При обращении к типам данных и POU библиотеки следует перед их названием указывать префикс **OSNMP** (пример: **OSNMP.Manager**).

## 2.3 Типы данных

### 2.3.1 Перечисление ERROR

Перечисление **OSNMP.ERROR** описывает ошибки, которые могут возникнуть во время вызова ФБ и функций библиотеки.

Таблица 2.3.1 – Описание элементов перечисления OSNMP.ERROR

Название	Значение	Описание
NO_ERROR	0	Нет ошибок
TOO_BIG	1	Слишком большой кадр SNMP (следует уменьшить количество связанных переменных – см. вход usiCntVarBind)
NO_SUCH_NAME	2	Ошибка SNMPv1 (при использовании прокси-серверов SNMPv1/SNMPv2). Соответствует ошибкам SNMPv2: NO_ACCESS, NOT_WRITABLE, NO_CREATION, INCONSISTENT_NAME, AUTHORIZATION_ERROR
BAD_VALUE	3	Ошибка SNMPv1 (при использовании прокси-серверов SNMPv1/SNMPv2). Соответствует ошибкам SNMPv2: WRONG_VALUE, WRONG_ENCODING, WRONG_TYPE, WRONG_LENGTH, INCONSISTENT_VALUE
GENERR	5	Прочие специфичные ошибки SNMPv2 (см. описание соответствующего агента) или ошибка SNMPv1, соответствующая ошибкам SNMPv2: RESOURCE_UNAVAILABLE, COMMIT_FAILED, UNDO_FAILED
NO_ACCESS	6	Нет доступа к переменной для записи (для SetRequest)
WRONG_TYPE	7	Неправильный тип устанавливаемого значения переменной (для SetRequest)
WRONG_LENGTH	8	Неправильный размер устанавливаемого значения переменной (для SetRequest)
WRONG_ENCODING	9	Неправильная кодировка устанавливаемого значения переменной (для SetRequest)
WRONG_VALUE	10	Устанавливаемое значение не входит в разрешенные значения переменной (для SetRequest)
NO_CREATION	11	Переменная не может быть создана (для SetRequest)
INCONSISTENT_VALUE	12	Значение переменной не может быть присвоено при текущих обстоятельствах (например, при текущей конфигурации устройства)
RESOURCE_UNAVAILABLE	13	Значение переменной не может быть присвоено, так как необходимый ресурс занят
COMMIT_FAILED	14	Не удалось присвоить значение переменной
UNDO_FAILED	15	Не удалось отменить присвоенное значение переменной (то есть произошла ошибка COMMIT_FAILED)
AUTHORIZATION_ERROR	16	Ошибка авторизации (указано неверное сообщение)
NOT_WRITABLE	17	Переменная недоступна для записи (для SetRequest)
INCONSISTENT_NAME	18	Переменная не может быть создана при текущих обстоятельствах (для SetRequest)

TIME_OUT	50	Ошибка таймаута. Кадр SNMP, вероятно, не доставлен (проверьте связь с устройством или увеличьте таймаут)
NOT_OWEN_DEVICE	51	Данное устройство не является контроллером OBEH или виртуальным контроллером CODESYS Control Win V3
SIPADDR_EMPTY	52	В строке slpAddr не найдено ни одного SIP
SIPADDR_NOT_SIP	53	В строке slpAddr пропущен SIP между двумя точками
SIPADDR_MISSED_SIP	54	В строке slpAddr пропущен SIP между двумя точками
SIPADDR_OVER_MAXCHAR_SIP	55	В строке slpAddr значение SIP превышает 3 символа (максимальное значение SIP - 255)
SIPADDR_OVER_MAXVAL_SIP	56	В строке slpAddr значение SIP превышает максимальное (максимальное значение SIP - 255)
INVALID_REQUEST_TYPE	57	Указан неподдерживаемый тип запроса (см. вход eTypeRequest)
INVALID_TRAP_TYPE	58	Указан неподдерживаемый тип трапа (см. вход eTypeTrap)
INVALID_VARBIND_POINTER	59	Передан некорректный указатель на массив связанных переменных (см. вход pArrayVarBind)
INVALID_VARBIND_CNT	60	Указанное количество связанных переменных не входит в допустимые ограничения (usiCntVarBind не соответствует Settings.gc_usiMaxCntVarbindRequest)
INVALID_DATAPOINTER	61	Передан некорректный указатель на буфер данных переменной (см. VARBIND.pData)
INVALID_DATASIZE	62	Указан некорректный размер буфера данных (см. VARBIND.szData)
INVALID_VALUE_TYPE	63	Строка sOid пустая (см. VARBIND.sOid)
INVALID_GET_VALUE_TYPE	64	Указан тип данных, несоответствующий принятому типу данных, при запросе GetRequest (см. VARBIND.eType)
INVALID_GET_VALUE_SIZE	65	Указан размер данных, который меньше принятого размера данных, при запросе GetRequest (см. VARBIND.szData)
SOID_EMPTY	66	Строка sOid пустая (см. VARBIND.sOid)
SOID_NOT_SID	67	В строке sOid не найдено ни одного SID
SOID_MISSED_SID	68	В строке sOid пропущен SID между двумя точками
SOID_OVER_MAXCHAR_SID	69	В строке sOid значение SID превышает 10 символов (максимальное значение SID 4294967295)
SOID_OVER_MAXVAL_SID	70	В строке sOid значение SID превышает максимальное (максимальное значение SID 4294967295)
SOID_NOT_ONE_SID1	71	В строке sOid значение SID1 не равно 1 (для SNMP значение SID1 должно быть 1)
SOID_NOT_THREE_SID2	72	В строке sOid значение SID2 не равно 3 (для SNMP значение SID2 должно быть 3)
WRONG_PARAMETER	73	Как минимум один из аргументов ФБ имеет некорректное значение
RESPONSE_FAILURE	74	Пришел некорректный ответ на запрос
INVALID_TEXT_CONVENTION	75	Данные не соответствуют текстовому соглашению (проверьте входные указатели функции и данные в Oraque)

INVALID_SYSUPTIME_SOID	76	Строка VARBIND.sOid не соответствует sysUpTime.0 ('1.3.6.1.2.1.1.3.0') в ФБ <a href="#">SendTrap</a> в первой переменной
INVALID_SYSUPTIME_TYPE	77	Тип VARBIND.eType не соответствует sysUpTime.0 (TimeTicks) в ФБ <a href="#">SendTrap</a> в первой переменной
INVALID_SNMPTRAPOID_SOID	78	Строка VARBIND.sOid не соответствует snmpTrapOID.0 ('1.3.6.1.6.3.1.1.4.1.0') в ФБ <a href="#">SendTrap</a> во второй переменной
INVALID_SNMPTRAPOID_TYPE	79	Тип VARBIND.eType не соответствует snmpTrapOID.0 (OID) в ФБ <a href="#">SendTrap</a> во второй переменной
NO_SUCH_OBJECT	80	Объект с таким OID отсутствует
NO_SUCH_INSTANCE	81	Переменная с таким OID отсутствует

### 2.3.2 Перечисление REQUEST\_PDU\_TYPE

Перечисление **OSNMP.REQUEST\_PDU\_TYPE** описывает тип запроса, отправляемого SNMP-менеджером (см. ФБ [Manager](#)).

Таблица 2.3.2 – Описание элементов перечисления OSNMP.REQUEST\_PDU\_TYPE

Название	Значение	Описание
GetRequest	16#A0	Запрос на чтение данных
SetRequest	16#A3	Запрос на запись данных

### 2.3.3 Перечисление TRAP\_PDU\_TYPE

Перечисление **OSNMP.TRAP\_PDU\_TYPE** описывает тип трапа, отправляемого SNMP-агентом (см. ФБ [SendTrap](#)).

Таблица 2.3.3 – Описание элементов перечисления OSNMP.TRAP\_PDU\_TYPE

Название	Значение	Описание
InformRequest	16#A6	InformRequest (требуется подтверждения о получении)
SNMPv2_Trap	16#A7	Трап (не требует подтверждения о получении)

### 2.3.4 Перечисление VALUE\_TYPE

Перечисление **OSNMP.VALUE\_TYPE** описывает типы данных протокола SNMP, поддерживаемые библиотекой.

Таблица 2.3.4 – Описание элементов перечисления OSNMP.VALUE\_TYPE

Название	Значение	Тип в CODESYS	Описание
INTEGER	16#2	DINT	32-битное целое со знаком
OCTET_STRING	16#4	STRING(1..255)	Строка ASCII-символов
OID	16#6	STRING(255)	Идентификатор объекта SNMP
IpAddress	16#40	STRING(15)	Строковое представление IP-адреса
Counter32	16#41	UDINT	Значение счетчика (32-битное целое без знака)

Unsigned32	16#42	UDINT	32-битное целое без знака (также соответствует типу SNMP Gauge32)
TimeTicks	16#43	UDINT	Значение счетчика времени (32-битное целое без знака, 1 тик = 10 мс)
Opaque	16#44	ARRAY [1...255] OF BYTE	Тип для передачи произвольных данных
Counter64	16#46	ULINT	Значение счетчика (64-битное целое без знака)
OCTET_STRING _RFC	16#FE	ARRAY [1..255] OF BYTE	Набор бинарных данных. В запросе используется код типа 16#4 (как для OCTET_STRING), но при обработке учитывается, что данные являются бинарными и могут включать в себя нули (для STRING байт со значением 0 является признаком конца строки)
Auto_Detect	16#FF	-	Специальный тип данных библиотеки, не описанный в спецификации SNMP. Используется для автоопределения типа при запросе GetRequest в ФБ <a href="#">Manager</a> , если пользователь знает только OID. Необходимо выделить для значения буфер размером как минимум 81 байт (ARRAY [1..81] OF BYTE). После получения ответа тип и размер в переменной VARBIND будут заменены на полученные в ответе. Если принятый тип может быть равен OCTET_STRING, Opaque или OID, тогда пользователю необходимо выделить максимальный буфер, который необходим для данного значения (см. MIB-файл устройства), но не более 255 байт (ARRAY [1..255] OF BYTE) для Opaque или 255 символов (STRING(255)) для OCTET_STRING и OID. Данный тип желательно использовать только в отладочных целях для идентификации типа и размера переменной, для того чтобы в релизной версии проекта указать правильный тип и размер буфера значения в переменной VARBIND

### 2.3.5 Структура VARBIND

Структура **OSNMP.VARBIND** описывает один параметр протокола SNMP.

**Таблица 2.3.5 – Описание переменных структуры OSNMP.VARBIND**

Название	Тип данных	Описание
sOid	STRING(255)	Идентификатор параметра
eType	<a href="#">VALUE_TYPE</a>	Тип параметра
pData	CAA.PVOID	Указатель на переменную
szSize	CAA.SIZE	Размер переменной в байтах
xWritable	BOOL	Атрибут доступа к записи переменной. <b>TRUE</b> – запись разрешена (используется только для ФБ <a href="#">Agent</a> )
eError	ERROR	Код ошибки связанной переменной, возвращаемый ФБ

### 2.3.6 Список параметров Settings

Список параметров **Settings** содержит список глобальных констант библиотеки. Они задаются в менеджере библиотек и не могут быть изменены из кода программы.

**Таблица 2.3.6 – Описание переменных списка параметров OSNMP.Settings**

Название	Тип данных	Описание
gc_usiMaxCntVarbindTrap	USINT(3..32)	Максимальное число связанных переменных в трапе
gc_usiMaxCntVarbindRequest	USINT(1..32)	Максимальное число связанных переменных в запросе

## 2.4 Функциональные блоки

### 2.4.1 ФБ Manager

Функциональный блок **Manager** используется для отправки GET/SET-запросов. По переднему фронту на входе **xExecute** происходит отправка запроса типа **eTypeRequest** на IP-адрес **slpAddr** и порт **uiAgentPort**. На стороне контроллера для отправки используется порт **uiLocalPort** и IP-адрес **0.0.0.0** (т. е. отправка запроса осуществляется по всем доступным интерфейсам). Для авторизации используются строки сообщества (пароли) **sCommunityRead** (пароль на чтение) и **sCommunityWrite** (пароль на запись).

Данные связанных переменных запроса располагаются по указателю **pArrayVarBind**. По указателю должен быть размещен массив структур типа [VARBIND](#). Для каждого экземпляра структуры должны быть заполнены поля **sOid** (OID параметра), **eType** (тип), **pData** (указатель на буфер считываемого или записываемого значения) и **szSize** (размер буфера в байтах). На входе **usiCntVarBind** указывается число переменных массива, которые будут добавлены в запрос. Максимальное число переменных в запросе определяется глобальной константой **gc\_usiMaxCntVarbindRequest**, расположенной в списке параметров [Settings](#).

Ответ от агента ожидается в течение времени **tTimeout**. В случае отсутствия ответа ФБ повторяет запрос. Число переповторов определяется входом **usiRetry** (значение **0** соответствует отсутствию переповторов). Если ни на один из запросов не был получен ответ, то выход **xError** принимает значение **TRUE**, а выход **eError = TIME\_OUT**.

В случае получения корректного ответа выход **xDone** принимает значение **TRUE**.

В случае ошибки выход **xError** принимает значение **TRUE**, а на выходе **eError** отображается код ошибки из перечисления [ERROR](#). Выход **usiErrorIndex** содержит номер элемента массива, расположенного по указателю **pArrayVarBind**, с которым связана произошедшая ошибка (см. поле **eError** в структуре [VARBIND](#)).

Для отправки нового запроса следует создать передний фронт на входе **xExecute**.

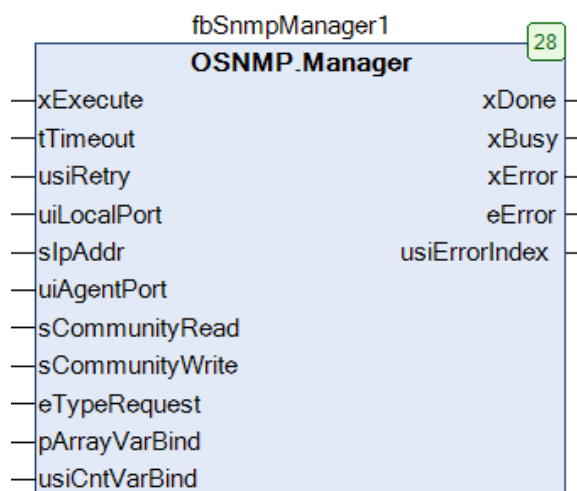


Рисунок 2.4.1 – Внешний вид ФБ Manager на языке CFC

Таблица 2.4.1 – Описание входов и выходов ФБ Manager

Название	Тип	Описание
<b>Входы</b>		
xExecute	BOOL	По переднему фронту происходит однократная (с возможностью переповторов в случае отсутствия ответа) отправка запроса
tTimeout	TIME	Таймаут ожидания ответа от агента ( <b>T#0ms</b> – время ожидания не ограничено)
usiRetry	USINT	Число переповторов в случае отсутствия ответа
uiLocalPort	UINT	Порт контроллера, через который отправляется запрос
slpAddr	STRING(15)	IP-адрес агента в формате <a href="#">IPv4</a> ('xxx.xxx.xxx.xxx')
uiAgentPort	UINT	Порт агента
sCommunityRead	STRING(31)	Сообщество (пароль) для запроса чтения
sCommunityWrite	STRING(31)	Сообщество (пароль) для запроса записи
eTypeRequest	<a href="#">REQUEST_PDU_TYPE</a>	Тип запроса (GetRequest или SetRequest)
pArrayVarBind	POINTER TO VARBIND	Указатель на массив связанных переменных (ARRAY [...] OF <a href="#">VARBIND</a> )
usiCntVarBind	USINT(1.. <a href="#">gc_usiMaxCntVarbindRequest</a> )	Число связанных переменных (число элементов массива)
<b>Выходы</b>		
xDone	BOOL	<b>TRUE</b> – получен корректный ответ от slave-устройства
xBusy	BOOL	<b>TRUE</b> – ФБ находится в работе
xError	BOOL	Принимает значение <b>TRUE</b> в случае возникновения ошибки
eError	<a href="#">ERROR</a>	Статус работы ФБ (или код ошибки)
usiErrorIndex	USINT(0.. <a href="#">gc_usiMaxCntVarbindRequest</a> )	Индекс связанной переменной (в массиве структур <a href="#">VARBIND</a> ), к которой относится ошибка

### 2.4.2 ФБ TrapReceiver

Функциональный блок **TrapReceiver** используется для приема трапов. Поддерживается прием трапов протокола SNMPv2 и InformRequest'ов (с отправкой подтверждения о получении).

Пока вход **xEnable** имеет значение **TRUE**, то блок находится в работе и ожидает приема трапов на порт **uiTrapPort**. Выход **xActive** в этом случае имеет значение **TRUE**.

При получении нового трапа значение выхода **udiCounterTrap** увеличивается на единицу, отображая число трапов, полученных за время работы блока с момента загрузки ПЛК, а выход **xNewTrap** принимает значение **TRUE** на один цикл ПЛК. В течение этого цикла выходы **arrVarBind** (массив параметров трапа), **usiCntVarBind** (число параметров трапа) и **sHost** (IP-адрес агента) имеют актуальные значения.

В случае ошибки выход **xError** принимает значение **TRUE**, а на выходе **eError** отображается код ошибки из перечисления [ERROR](#).

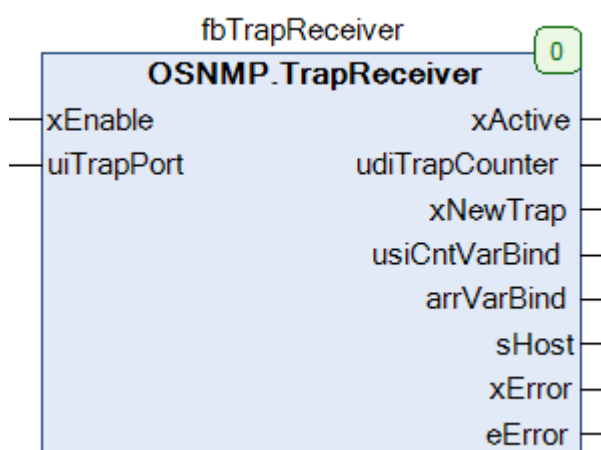


Рисунок 2.4.2 – Внешний вид ФБ TrapReceiver на языке CFC

Таблица 2.4.2 – Описание входов и выходов ФБ TrapReceiver

Название	Тип	Описание
<b>Входы</b>		
xEnable	BOOL	<b>TRUE</b> – блок ожидает приема трапов
uiTrapPort	UINT	Порт для приема трапов
<b>Выходы</b>		
xActive	BOOL	<b>TRUE</b> – ФБ активен
udiTrapCounter	UDINT	Счетчик трапов, принятых за время работы блока после старта ПЛК
xNewTrap	BOOL	<b>TRUE</b> – получен новый трап

## 2 Описание библиотеки OwenSnmp

---

usiCntVarBind	USINT(0.. <a href="#">Settings.gc_usiMaxCntVarbindTrap</a> )	Количество принятых связанных переменных массиве <b>arrVarBind</b> , которые были получены в трапе
arrVarBind	ARRAY [...] OF <a href="#">VARBIND</a>	Массив для принятых связанных переменных, которые были получены в трапе
sHost	STRING(15)	IP-адрес агента, передавшего трап, в формате <a href="#">IPv4</a> ('xxx.xxx.xxx.xxx')
xError	BOOL	Принимает значение <b>TRUE</b> в случае возникновения ошибки
eError	<a href="#">ERROR</a>	Статус работы ФБ (или код ошибки)

### 2.4.3 ФБ SendTrap

Функциональный блок **SendTrap** используется для отправки трапов.

По переднему фронту на входе **xExecute** происходит отправка трапа типа **eTrapType** на IP-адрес **slpAddr** и порт **uiManagerPort**. На стороне контроллера для отправки используется порт **uiLocalPort** и IP-адрес **0.0.0.0** (т. е. отправка трапа осуществляется по всем доступным интерфейсам).

Данные связанных переменных трапа располагаются по указателю **pArrayVarBind**. По указателю должен быть размещен массив структур типа [VARBIND](#). Для каждого экземпляра структуры должны быть заполнены поля **sOid** (OID параметра), **eType** (тип), **pData** (указатель на буфер значения) и **szSize** (размер буфера в байтах). На входе **usiCntVarBind** указывается число переменных массива, которые будут добавлены в трап. Максимальное число переменных в трапе определяется глобальной константой **gc\_usiMaxCntVarbindTrap**, расположенной в списке параметров [Settings](#).

При этом спецификация SNMP определяет, что в трапе должны быть как минимум два параметра:

- время наработки устройства (OID 1.3.6.1.2.1.1.3.0, тип TimeTicks);
- OID события (OID 1.3.6.1.6.3.1.1.4.1.0, тип OID).

Эти параметры должны быть самостоятельно сформированы пользователем и добавлены в трап в качестве первого и второго параметра соответственно.

Трапы типа **InformRequest** требуют подтверждения о получении. Ответ от менеджера ожидается в течение времени **tTimeout**. В случае отсутствия подтверждения ФБ повторяет отставку InformRequest'a. Число переповторов определяется входом **usiRetry** (значение **0** соответствует отсутствию переповторов). Если ни на один из InformRequest'ов не был получен ответ, то выход **xError** принимает значение **TRUE**, а выход **eError = TIME\_OUT**.

В случае получения успешного ответа выход **xDone** принимает значение **TRUE**.

В случае ошибки выход **xError** принимает значение **TRUE**, а на выходе **eError** отображается код ошибки из перечисления [ERROR](#). Выход **usiErrorIndex** содержит номер элемента массива, расположенного по указателю **pArrayVarBind**, с которым связана произошедшая ошибка (см. поле **eError** в структуре [VARBIND](#)).

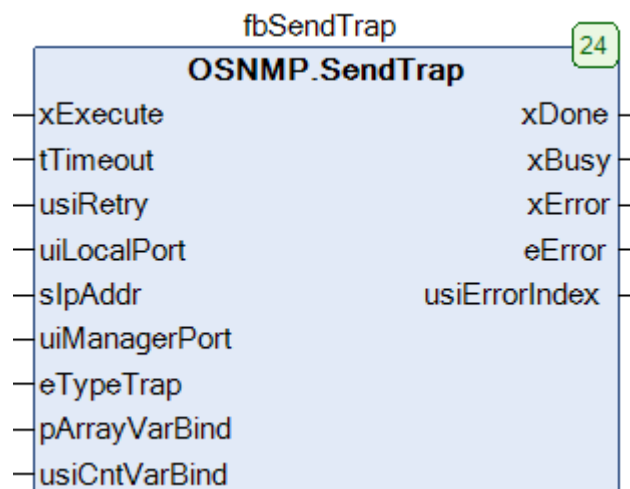


Рисунок 2.4.3 – Внешний вид ФБ TrapReceiver на языке CFC

Таблица 2.4.3 – Описание входов и выходов ФБ SendTrap

Название	Тип	Описание
<b>Входы</b>		
xExecute	BOOL	По переднему фронту происходит однократная (с возможностью повторов в случае отсутствия ответа) отправка трапа
tTimeout	TIME	Таймаут ожидания подтверждения от менеджера ( <b>T#0ms</b> – время ожидания не ограничено). Только для трапов типа InformRequest
usiRetry	USINT	Число повторов в случае отсутствия ответа
uiLocalPort	UINT	Порт контроллера, через который отправляется запрос
slpAddr	STRING(15)	IP-адрес менеджера в формате <a href="#">IPv4</a> ('xxx.xxx.xxx.xxx')
uiManagerPort	UINT	Порт менеджера
eTrapType	<a href="#">TRAP_PDU_TYPE</a>	Тип трапа (трап или InformRequest)
pArrayVarBind	POINTER TO <a href="#">VARBIND</a>	Указатель на массив связанных переменных (ARRAY [...] OF <a href="#">VARBIND</a> )
usiCntVarBind	USINT(3.. <a href="#">Settings.gc_usiMaxCntVarbindTrap</a> )	Число связанных переменных (число элементов массива)
<b>Выходы</b>		
xDone	BOOL	<b>TRUE</b> – получено подтверждение о доставке (для InformRequest) или отправка успешно завершена (для трапа)
xBusy	BOOL	<b>TRUE</b> – ФБ находится в работе
xError	BOOL	Принимает значение <b>TRUE</b> в случае возникновения ошибки
eError	<a href="#">ERROR</a>	Статус работы ФБ (или код ошибки)
usiErrorIndex	USINT(0.. <a href="#">Settings.gc_usiMaxCntVarbindTrap</a> )	Индекс связанной переменной (в массиве структур <a href="#">VARBIND</a> ), к которой относится ошибка

#### 2.4.4 ФБ Agent

Функциональный блок **Agent** представляет собой реализацию SNMP-агента. Поддерживается обработка запросов GET и SET. Блок может хранить до 255 параметров.

Пока вход **xEnable** имеет значение **TRUE**, то блок находится в работе и ожидает приема запросов на порт **uiAgentPort**. Выход **xActive** в этом случае имеет значение **TRUE**. Для авторизации используются строки сообщества (пароли) **sCommunityRead** (пароль на чтение) и **sCommunityWrite** (пароль на запись).

Данные параметров располагаются по указателю **pArrayVarBind**. По указателю должен быть размещен массив структур типа [VARBIND](#). Для каждого экземпляра структуры должны быть заполнены поля **eType** (тип), **pData** (указатель на буфер считываемого или записываемого значения), **szSize** (размер буфера в байтах) и **xWritable** (тип доступа, **FALSE** – только чтение, **TRUE** – чтение и запись). На входе **usiCntVarBind** указывается число параметров агента.

OID параметров в структуре не должны задаваться пользователем. Они формируются автоматически по следующему принципу:

##### 1.3.6.1.4.1.51014.9.[usiTypeDevice].[IndexVarbind].0

(без учета квадратных скобок, которые использованы в документе для повышения читабельности) где **usiTypeDevice** – значение соответствующего входа ФБ, **IndexVarbind** – номер элемента массива структур (нумерация с 1).

При получении нового запроса SET значение выхода **udiSetRequestCounter** увеличивается на единицу, отображая число SET-запросов, полученных за время работы блока с момента загрузки ПЛК, а выход **xNewSetRequest** принимает значение **TRUE** на один цикл ПЛК. В течение этого цикла выходы **usiCntVarBind** (число записанных параметров) и **arrIndexSetVarBind** (массив индексов записанных параметров) имеют актуальные значения. При получении любого запроса на выходе **sHost** отображается IP-адрес менеджера, приславшего запрос.

В случае ошибки выход **xError** принимает значение **TRUE**, а на выходе **eError** отображается код ошибки из перечисления [ERROR](#). Выход **usiErrorIndex** содержит номер элемента массива, расположенного по указателю **pArrayVarBind**, с которым связана произошедшая ошибка (см. поле **eError** в структуре [VARBIND](#)).

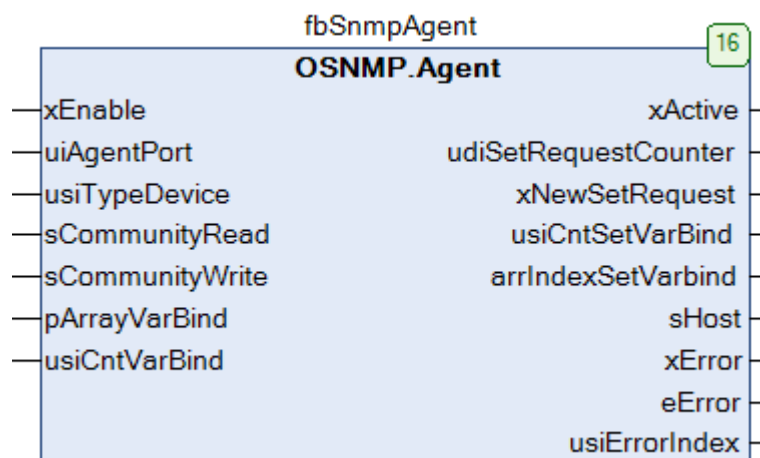


Рисунок 2.4.4 – Внешний вид ФБ Agent на языке CFC

Таблица 2.4.4 – Описание входов и выходов ФБ Agent

Название	Тип	Описание
<b>Входы</b>		
xEnable	BOOL	<b>TRUE</b> – блок находится в работе
uiAgentPort	UINT	Порт для приема запросов
usiTypeDevice	USINT	Тип устройства. Используется при формировании OID параметров
sCommunityRead	STRING(31)	Сообщество (пароль) для запроса чтения
sCommunityWrite	STRING(31)	Сообщество (пароль) для запроса записи
pArrayVarBind	POINTER TO <a href="#">VARBIND</a>	Указатель на параметры агента (ARRAY [...] OF <a href="#">VARBIND</a> )
usiCntVarBind	USINT(1..255)	Число параметров агента
<b>Выходы</b>		
xActive	BOOL	<b>TRUE</b> – ФБ активен
udiSetRequestCounter	UDINT	Счетчик SET-запросов, принятых за время работы блока после старта ПЛК
xNewSetRequest	BOOL	<b>TRUE</b> – получен новый SET-запрос
usiCnatVarBind	USINT(0.. <a href="#">gc_usiMaxCntVarbindRequest</a> )	Число записанных параметров
arrIndexSetVarbind	ARRAY [1.. <a href="#">gc_usiMaxCntVarbindRequest</a> ] OF USINT	Индексы записанных переменных в массиве структур <a href="#">VARBIND</a>
sHost	STRING(15)	IP-адрес менеджера, приславшего запрос, в формате <a href="#">IPv4</a> ('xxx.xxx.xxx.xxx')
xError	BOOL	Принимает значение <b>TRUE</b> в случае возникновения ошибки
eError	<a href="#">ERROR</a>	Статус работы ФБ (или код ошибки)
usiIndexError	USINT	Индекс связанной переменной (в массиве структур <a href="#">VARBIND</a> ), к которой относится ошибка

## 2.5 Функции

### 2.5.1 Функция REAL\_TO\_OPAQUE

Функция преобразует значение типа **REAL** в значение типа **OPAQUE** в соответствии с [ТЕКСТОВЫМ СОГЛАШЕНИЕМ](#) David T. Perkins.

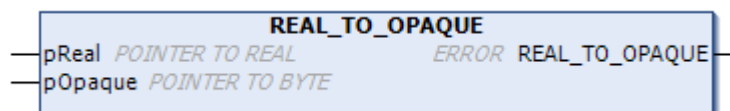


Рисунок 2.5.1 – Внешний вид функции REAL\_TO\_OPAQUE на языке CFC

Таблица 2.5.1 – Описание входов и выходов функции REAL\_TO\_OPAQUE

Название	Тип	Описание
<b>Входы</b>		
pReal	POINTER TO REAL	Указатель на значение типа REAL
pOpaque	POINTER TO BYTE	Указатель на значение типа OPAQUE (ARRAY [1..7] OF BYTE)
<b>Выходы</b>		
REAL_TO_OPAQUE	<a href="#">ERROR</a>	Код ошибки

## 2.5.2 Функция LREAL\_TO\_OPAQUE

Функция преобразует значение типа **LREAL** в значение типа **OPAQUE** в соответствии с [ТЕКСТОВЫМ СОГЛАШЕНИЕМ](#) David T. Perkins.

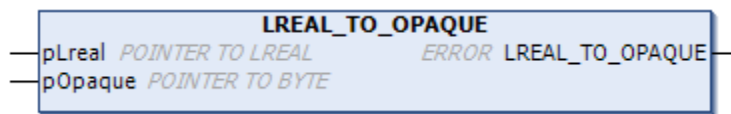


Рисунок 2.5.2 – Внешний вид функции LREAL\_TO\_OPAQUE на языке CFC

Таблица 2.5.2 – Описание входов и выходов функции LREAL\_TO\_OPAQUE

Название	Тип	Описание
<b>Входы</b>		
pLreal	POINTER TO LREAL	Указатель на значение типа LREAL
pOpaque	POINTER TO BYTE	Указатель на значение типа OPAQUE (ARRAY [1..11] OF BYTE)
<b>Выходы</b>		
LREAL_TO_OPAQUE	<a href="#">ERROR</a>	Код ошибки

### 2.5.3 Функция OPAQUE\_TO\_REAL

Функция преобразует значение типа **OPAQUE** в значение типа **REAL** в соответствии с [ТЕКСТОВЫМ СОГЛАШЕНИЕМ](#) David T. Perkins.

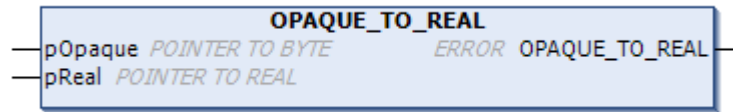


Рисунок 2.5.3 – Внешний вид функции OPAQUE\_TO\_REAL на языке CFC

Таблица 2.5.3 – Описание входов и выходов функции OPAQUE\_TO\_REAL

Название	Тип	Описание
<b>Входы</b>		
pOpaque	POINTER TO BYTE	Указатель на значение типа OPAQUE (ARRAY [1..7] OF BYTE)
pReal	POINTER TO REAL	Указатель на значение типа REAL
<b>Выходы</b>		
OPAQUE_TO_REAL	<a href="#">ERROR</a>	Код ошибки

## 2.5.4 Функция OPAQUE\_TO\_LREAL

Функция преобразует значение типа **OPAQUE** в значение типа **LREAL** в соответствии с [ТЕКСТОВЫМ СОГЛАШЕНИЕМ](#) David T. Perkins.

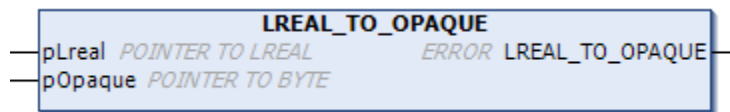


Рисунок 2.5.4 – Внешний вид функции OPAQUE\_TO\_LREAL на языке CFC

Таблица 2.5.4 – Описание входов и выходов функции OPAQUE\_TO\_LREAL

Название	Тип	Описание
<b>Входы</b>		
pLreal	POINTER TO LREAL	Указатель на значение типа LREAL
pOpaque	POINTER TO BYTE	Указатель на значение типа OPAQUE (ARRAY [1..11] OF BYTE)
<b>Выходы</b>		
OPAQUE_TO_LREAL	<a href="#">ERROR</a>	Код ошибки

### 2.5.5 Функция SWAP\_DATA

Функция **SWAP\_DATA** используется для копирования данных из одного буфера в другой с перестановкой байт и регистров.

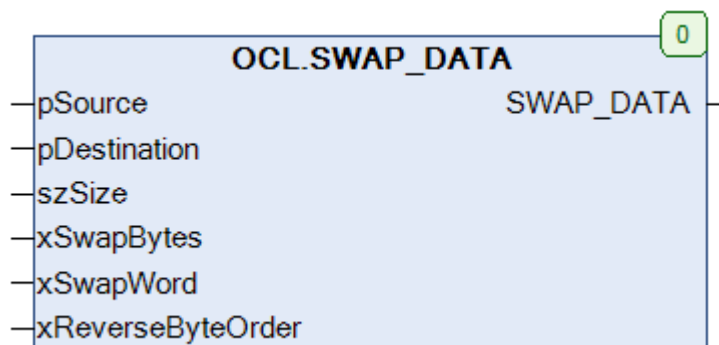


Рисунок 2.5.5 – Внешний вид функции SWAP\_DATA на языке CFC

Таблица 2.5.5 – Описание входов и выходов функции SWAP\_DATA

Название	Тип	Описание
<b>Входы</b>		
pSource	CAA.PVOID	Указатель на буфер исходных данных
pDestination	CAA.PVOID	Указатель на буфер, в который будут скопированы данные. Может совпадать с <b>pSource</b> – тогда после изменения данные будут помещены в тот же буфер
szSize	CAA.SIZE	Размер копируемых данных в байтах (должен не превышать размеры буферов)
xSwapBytes	BOOL	<b>TRUE</b> – выполнить перестановку байт (A1 B2 C3 D4 ---> B2 A1 D4 C3)
xSwapWord	BOOL	<b>TRUE</b> – выполнить перестановку регистров (A1 B2 C3 D4 ---> C3 D4 A1 B2)
xReverseByteOrder	BOOL	<b>TRUE</b> – изменить порядок байт на противоположный (A1 B2 C3 D4 ---> D4 C3 B2 A1). Если данный вход имеет значение <b>TRUE</b> , то входы <b>xSwapBytes</b> и <b>xSwapWord</b> не обрабатываются
<b>Выходы</b>		
SWAP_DATA	BOOL	<b>TRUE</b> – операция выполнена

### 3 Примеры

Ниже приведены ссылки на примеры использования библиотеки.

Примеры разработаны в **CODESYS V3.5 SP14 Patch 3** для контроллера **СПК1xx [M01]** с целевым файлом версии **3.5.14.34**. В случае необходимости запуска примеров на другом устройстве следует нажать **ПКМ** на узел **Device** и выбрать команду **Обновить устройство**.

**Таблица 3.1 – Список примеров**

<b>Пример</b>	<b>Ссылка</b>
ФБ <a href="#">Manager</a>	<a href="#">скачать</a>
ФБ <a href="#">TrapReceiver</a> , ФБ <a href="#">SendTrap</a>	<a href="#">скачать</a>
ФБ <a href="#">Agent</a>	<a href="#">скачать</a>