

**Описание интерфейса библиотеки ФБ
для работы с протоколом Modbus
(Slave)**

Руководство пользователя

Содержание:

1	Установка дополнительных библиотек для проекта.....	3
2	Описание интерфейса библиотеки ФБ для работы с протоколом Modbus (режим Slave).....	7
2.1	Введение	7
2.2	Инициализация и открытие порта	7
2.3	Функциональные блоки для работы с протоколом Modbus	7
2.3.1	Modbus Slave	8
2.3.1.1	Входы:	8
2.3.1.2	Выходы:	8
2.3.1.3	Примечания	8
	Приложение А	10

1 Установка дополнительных библиотек для проекта

В системе CoDeSys все файлы библиотек функциональных блоков имеют расширение ***.compiled-library** и находятся в папке **«Library»**, расположенной по месту размещения основной программы на диске компьютера.

По умолчанию доступен только стандартный набор библиотек. Дополнительные библиотеки добавляются пользователем по мере необходимости, в папку к уже имеющимся библиотекам.

Новые библиотеки подключают с помощью «Менеджера библиотек (**Library Manager**)».

Менеджер библиотек открывают из главного меню CoDeSys командами **Window | Library Manager** или через менеджер проекта (вкладка **«Devices»**) открывают папку **«Library Manager»** (рисунок 1.1). В средней верхней части окна отображается список установленных библиотек.

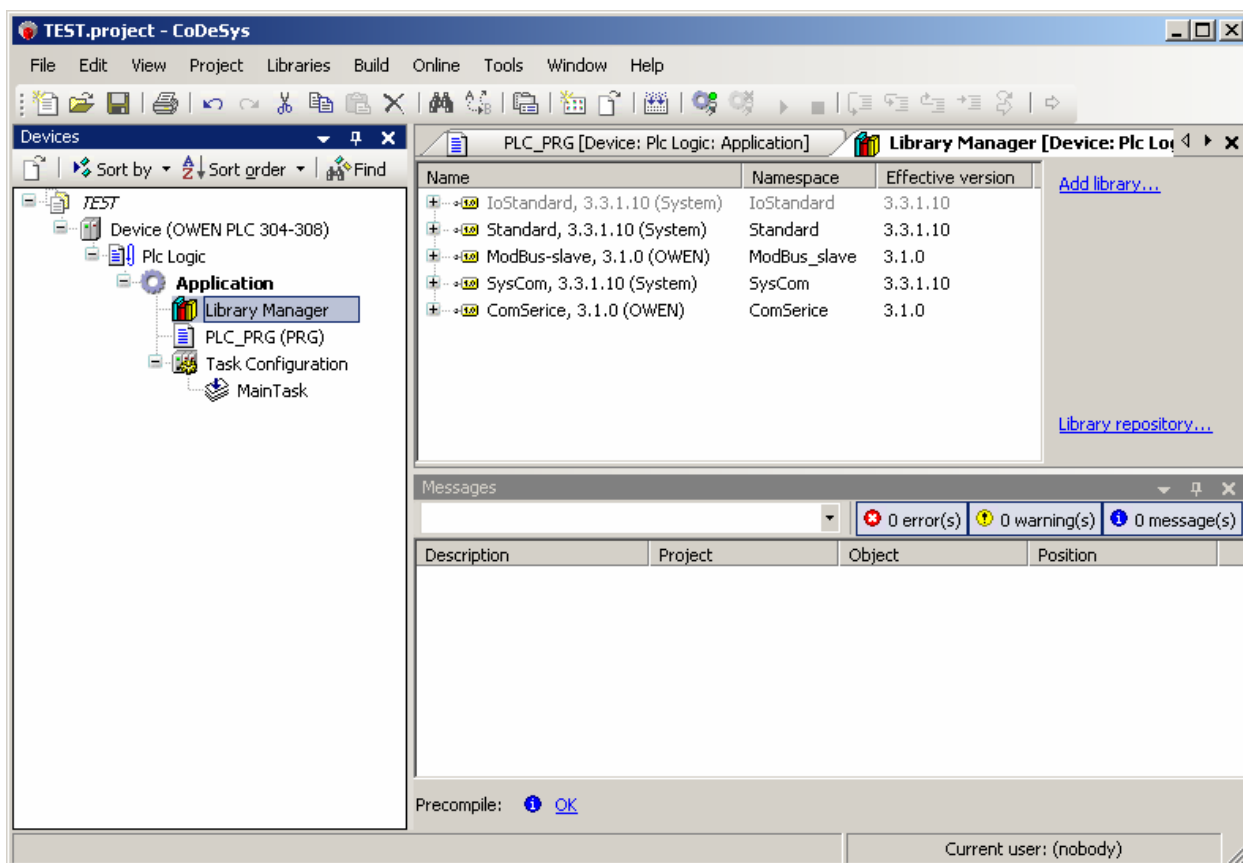


Рисунок 1.1 – Окно вкладки «Устройства (Devices)» с открытой папкой «Library Manager»

Для установки дополнительных библиотек следует:

- 1) Нажать на ссылку **«Add library»** справа на вкладке Library Manager.
- 2) В открывшемся окне нажать Library repository (рисунок 1.2):

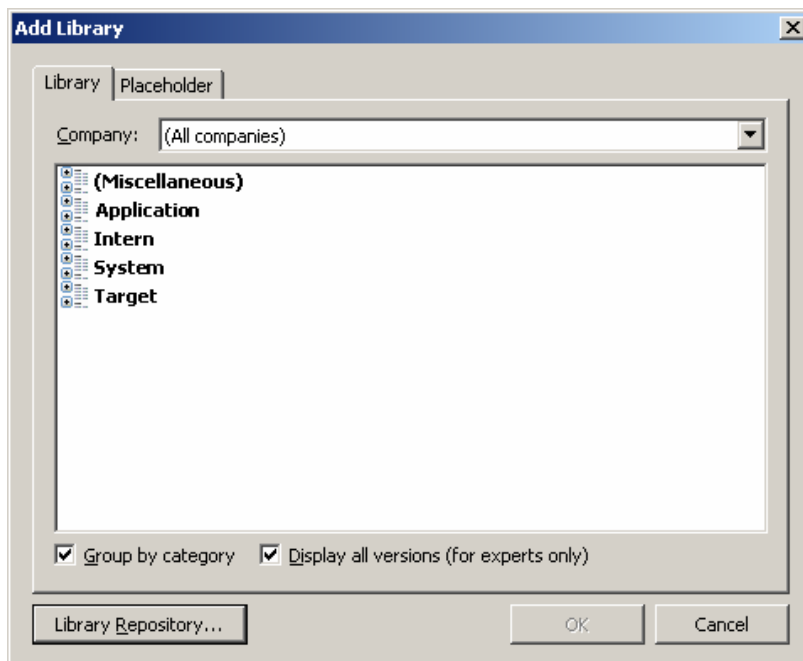


Рисунок 1.2 – Окно добавления библиотеки.

В открывшемся диалоговом окне нажать кнопку Install и выбрать файл с библиотекой: Modbas-slave.compiled-library.

После закрытия диалога выбора файла открыть раздел Miscelaneous и и проверить, что библиотека добавилась (рисунок 1.3):

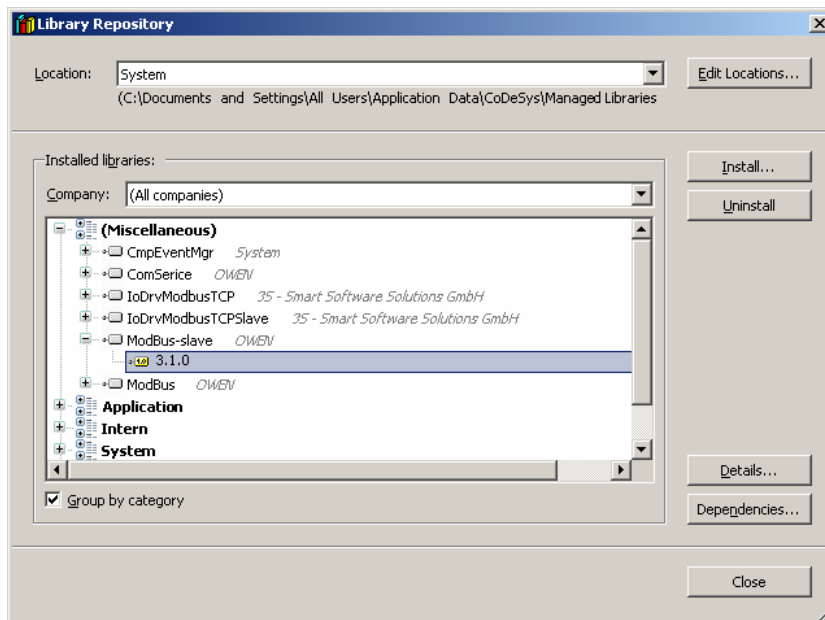


Рисунок 1.3 – Окно Library repository, пример выбора библиотеки.

Затем нажать кнопку Close.

Вы снова окажетесь в окне добавления библиотеки. Необходимо войти в раздел Miscelaneous, выбрать библиотеку ModBus-Slave и последнюю версию, а затем нажать кнопку OK (рисунок 1.4):

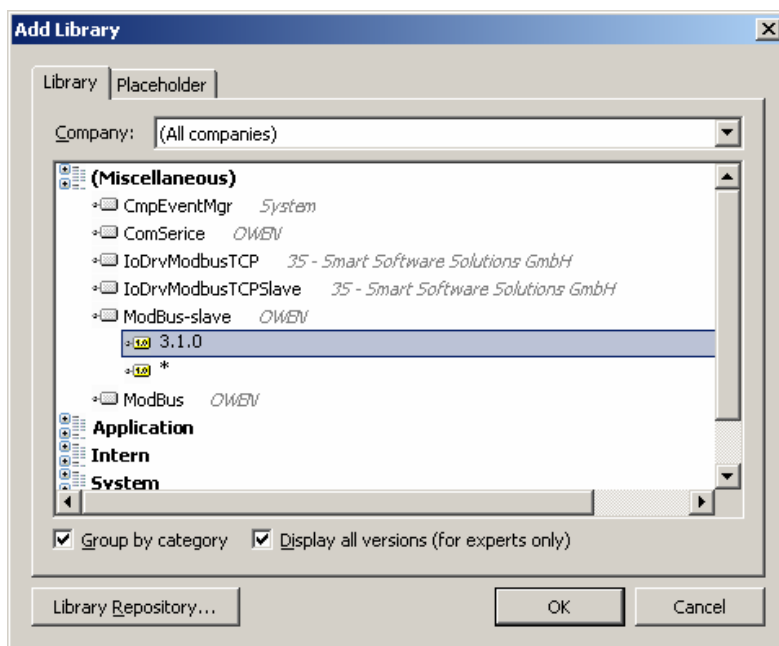


Рисунок 1.4 – Окно добавления библиотеки – выбор версии установленной библиотеки.

Для просмотра состава и свойств функциональных блоков следует выбрать требуемую библиотеку. Откроется окно с перечнем блоков (рис. 1.5, в нижней части окна) и краткой информацией о них.

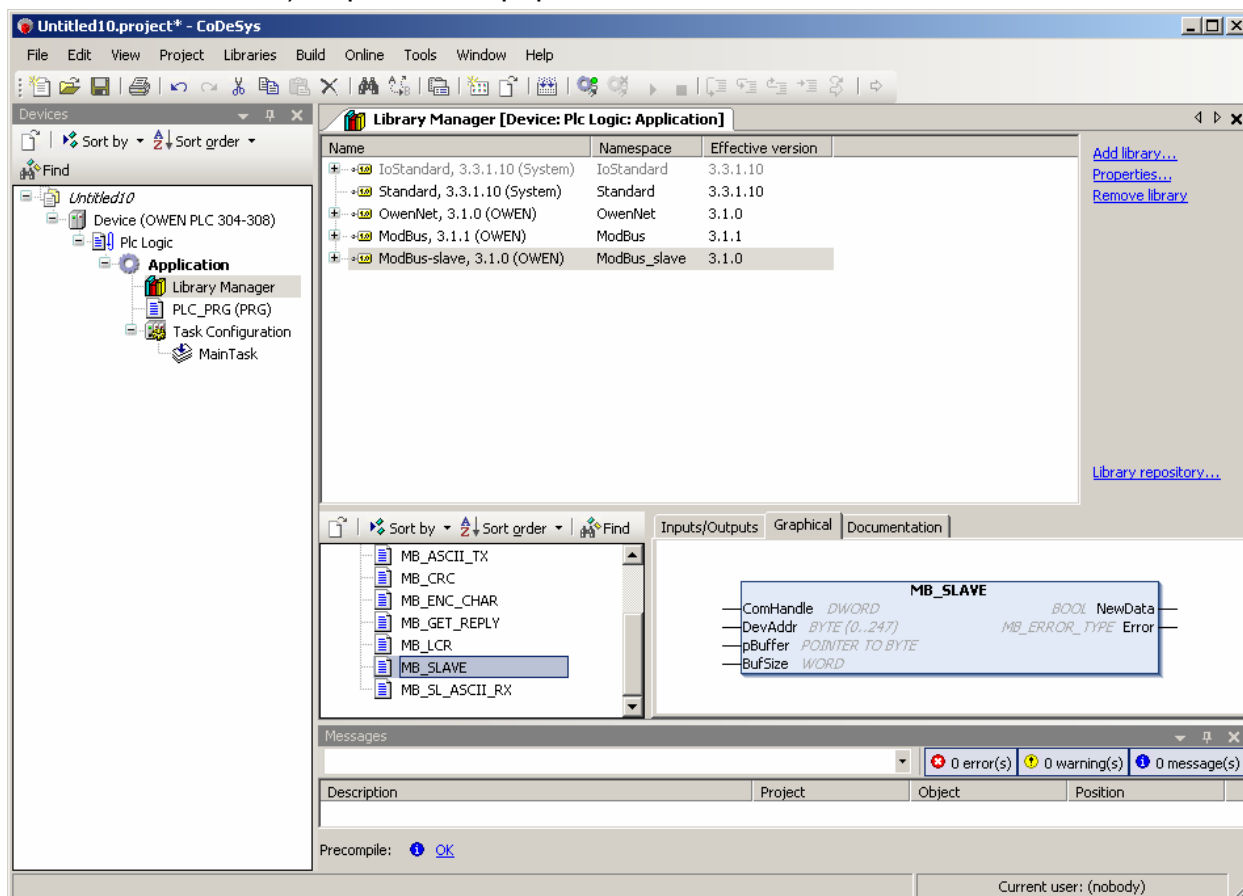


Рисунок 1.5 – Library manager (управление библиотеками).

Для удаления дополнительных библиотек следует:

- 1) Выделить требуемую библиотеку в списке библиотек (см. рисунок 1.3) и выбрать ссылку «**Remove library**» справа на вкладке или команду Delete контекстного меню (рисунок 1.6):

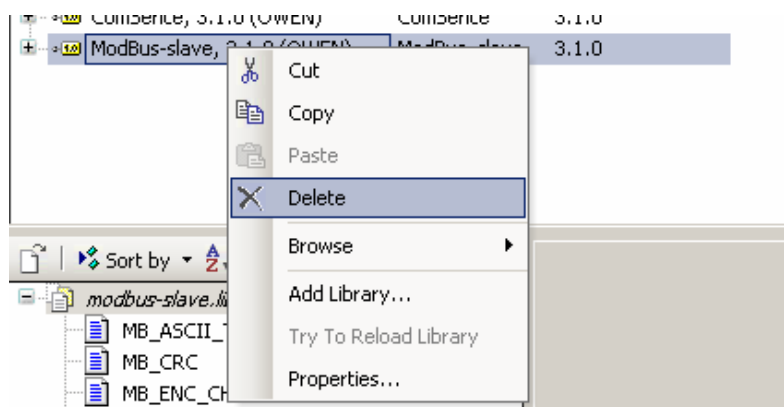


Рисунок 1.6 – Library manager (управление библиотеками).

Примечание. В функциональных блоках, разработанных для контроллеров фирмы ОВЕН, режим симуляции (Simulation Mode) не предусмотрен. Отладка программы в режиме визуализации должна производиться при подключенном контроллере, функциональные блоки при этом работают только в самом контроллере.

2 Описание интерфейса библиотеки ФБ для работы с протоколом Modbus (режим Slave)

2.1 Введение

Библиотека предназначена для работы (чтение/запись) с параметрами устройств через последовательный интерфейс ПЛК стандартного протокола Modbus. Возможно так же применение данной библиотеки для других приборов на основе CoDeSys. Контроллер выступает в качестве Мастера, т.е. ведет обмен с другими подчиненными приборами в сети.

При подключении библиотеки **ModBus.compiled-library** автоматически подключается библиотека **SisLibCom.compiled-library**.

2.2 Инициализация и открытие порта

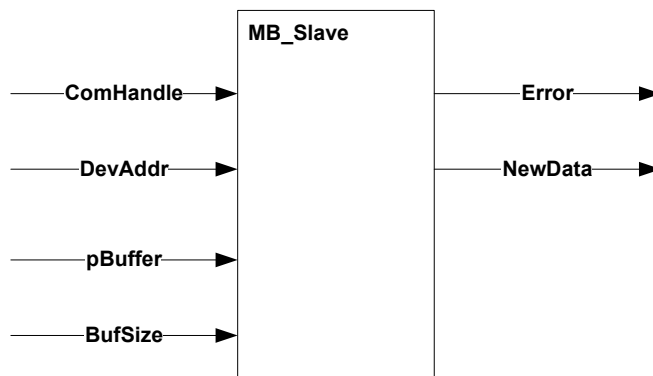
Для работы с библиотеками ModBus и OWENNET на ПЛК требуется сначала проинициализировать и открыть COM-порт. Для открытия порта используется специальная библиотека **ComService.compiled-library**.

```
(*Устанавливаем настройки COM-порта*)
IF port_opened=0 THEN
    Settings.Port:=com_num;    (*номер COM-порта 0 - RS-485,
1 - RS-232*)
    Settings.dwBaudRate:=115200;    (*скорость*)
    Settings.byParity:=0;
    Settings.dwTimeout:=0;
    Settings.byStopBits:=0;
    Settings.dwBufferSize:=0;
    Settings.dwScan:=0;
END_IF
(*Открываем COM-порт*)
COM_SERVICE1(Enable:=(port_opened=0), Settings:=Settings ,
Task:=OPEN_TSK );
(*Если COM-порт открыт, то переходим к приему и передаче
данных *)
IF COM_SERVICE1.ready THEN
    port_opened:=2;
END_IF
```

2.3 Функциональные блоки для работы с протоколом Modbus

Библиотека содержит функциональные блоки, перечисленные ниже.

2.3.1 Modbus Slave



2.3.1.1 Входы:

- **ComHandle**: DWORD – дескриптор последовательного порта;
- **DevAddr**: BYTE (0..247) – адрес в сети Modbus;
- **BufSize**: WORD – размер массива, содержащего отображаемые в сеть данные, в байтах.
- **pBuffer**: ARRAY [0..4096] OF BYTE – ссылка на массив, содержащий отображаемые в сеть данные;

2.3.1.2 Выходы:

- **Exception**: BYTE – исключения протокола MODBUS или ошибки:
 0x00 – ошибок или исключений MODBUS не обнаружено,
 0xFE – аппаратная ошибка ПЛК,
- **NewData**: BOOL – флаг наличия изменённых данных

2.3.1.3 Примечания

Поддерживаются следующие функции:

- Чтение статуса дискретных выходов (0x01) Read Coils;
- Чтение статуса дискретных входов (0x02) Read Discrete Inputs;
- Чтение содержания регистров (0x03) Read Holding Registers;
- Чтение содержания входных регистров (0x04) Read Input Registers;
- Установка единичного выхода (0x05) Write Single Coil;
- Запись значения единичного регистра (0x06) Write Single Register;
- Запись дискретных ячеек 15 (0x0F) Write Multiple Coils;
- Запись значения регистров 16 (0x10) Write Multiple registers;

Для всех функций используется один и тот же буфер. Адресация всегда начинается с 0 и в качестве единицы адресации используется минимальная величина передаваемых/получаемых данных (см. пример на рис. 2.1). Т.е., если мы пользуемся функцией (0x01) Read Coils, адресация будет в битах и адрес 2 будет указывать на 2й бит первого байта. Если используется функция (0x03) Read Holding Registers, то адрес 2 будет указывать на 4й байт (второе слово). В приведённом примере будут считаны, соответственно, значения 0 (для функции (0x01) Read Coils) и 0x9ABC, для функции (0x03) Read Holding Registers, если считывалось одно слово.

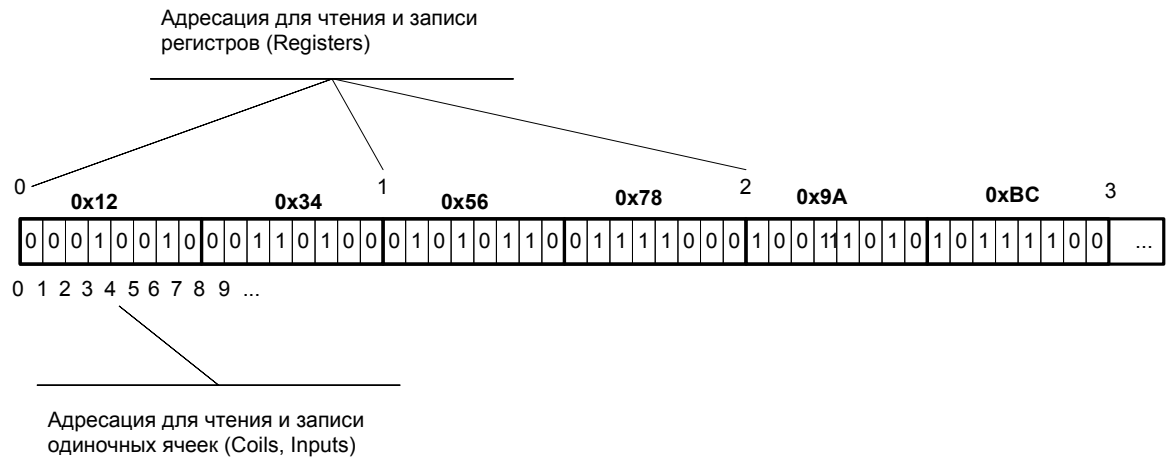


Рисунок 2.1 – Примеры адресации для разных типов команд ModBus.

Приложение А

Пример на языке ST реализует работу по интерфейсу RS-485: настройку модуля ModBus-Slave для трансляции в сеть массива переменных MB_Buffer.

Объявление переменных:

```
PROGRAM PLC_PRG
VAR
    MB_func: MB_SLAVE;
    MB_Buffer: ARRAY[0..3] OF WORD; (* буфер данных для обмена по ModBus *)
    MB_NewData: BOOL; (* данные обновилась *)
    MB_Error: MB_ERROR_TYPE; (* ошибка обработки запроса *)

    port_opened: BYTE := 0;
    Settings: ComSerice.SysCom.COM_Settings; (*настройки последовательного
порта *)
    SettingsEx: ComSerice.SysCom.COM_SettingsEx;
    COM_SERVICE1: COM_SERVICE;
END_VAR
```

Тело программы:

```
IF port_opened=0 THEN
    Settings.sPort:=ComSerice.SysCom.SYS_COMPORT2; (*номер COM-
порта*)
    Settings.ulBaudrate:=9600; (*скорость*)
    Settings.byParity:=0;
    Settings.ulTimeout:=0;
    Settings.byStopBits:=1;
    Settings.ulBufferSize:=0;
    SettingsEx.byByteSize:=8;
    SettingsEx.bBinary:=TRUE;

    COM_SERVICE1(Enable:=(port_opened=0)
Settings:=Settings,sets_ex:=SettingsEx , Task:=OPEN_TSK );
    (*Если COM-порт открыт, то переходим к приему и передачи данных *)
    IF COM_SERVICE1.ready THEN
        port_opened:=2;
    END_IF

    MB_Buffer[0] := 16#1234;
    MB_Buffer[1] := 16#5678;
    MB_Buffer[2] := 16#ABCD;
    MB_Buffer[3] := 16#5555;
END_IF

IF port_opened=2 THEN (*Удачно проинициализировали*)
    MB_func(
        ComHandle:=COM_SERVICE1.handle, (*номер COM-порта*)
        DevAddr := 16,
        pBuffer := ADR(MB_Buffer),
        BufSize := SIZEOF(MB_Buffer),
        NewData => MB_NewData,
        Error => MB_Error
    );
    IF MB_Error <> MB_OK THEN
        MB_NewData := FALSE;
    END_IF
    IF MB_NewData = TRUE THEN
        MB_Error := MB_OK;
    END_IF
END_IF
```