

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

**Руководство пользователя**

## Содержание:

<b>1</b>	<b>Установка дополнительных библиотек для проекта.....</b>	<b>3</b>
<b>2</b>	<b>Описание интерфейса библиотеки ФБ для работы с протоколом Modbus (режим Master).....</b>	<b>7</b>
2.1	Введение .....	7
2.2	Инициализация и открытие порта .....	7
2.3	Функциональные блоки для работы с протоколом Modbus .....	8
2.3.1	Чтение статуса дискретных выходов (0x01) Read Coils .....	8
2.3.1.1	Входы: .....	8
2.3.1.2	Входы/выходы:.....	8
2.3.1.3	Выходы: .....	8
2.3.2	Чтение статуса дискретных входов (0x02) Read Discrete Inputs.....	9
2.3.2.1	Входы: .....	9
2.3.2.2	Входы/выходы:.....	9
2.3.2.3	Выходы: .....	9
2.3.3	Чтение содержания регистров (0x03) Read Holding Registers.....	10
2.3.3.1	Входы: .....	10
2.3.3.2	Входы/выходы:.....	10
2.3.3.3	Выходы: .....	10
2.3.4	Чтение содержания входных регистров (0x04) Read Input Registers .....	11
2.3.4.1	Входы: .....	11
2.3.4.2	Входы/выходы:.....	11
2.3.4.3	Выходы: .....	11
2.3.5	Установка единичного выхода (0x05) Write Single Coil.....	12
2.3.5.1	Входы: .....	12
2.3.5.2	Выходы: .....	12
2.3.6	Запись значения единичного регистра (0x06) Write Single Register .....	13
2.3.6.1	Входы: .....	13
2.3.6.2	Выходы: .....	13
2.3.7	Запись дискретных ячеек 15 (0x0F) Write Multiple Coils.....	14
2.3.7.1	Входы: .....	14
2.3.7.2	Входы/выходы:.....	14
2.3.7.3	Выходы: .....	14
2.3.8	Запись значения регистров 16 (0x10) Write Multiple registers .....	15
2.3.8.1	Входы: .....	15
2.3.8.2	Входы/выходы:.....	15
2.3.8.3	Выходы: .....	15
2.3.8.4	Примечания .....	15
	Приложение А .....	16
	Приложение Б .....	20

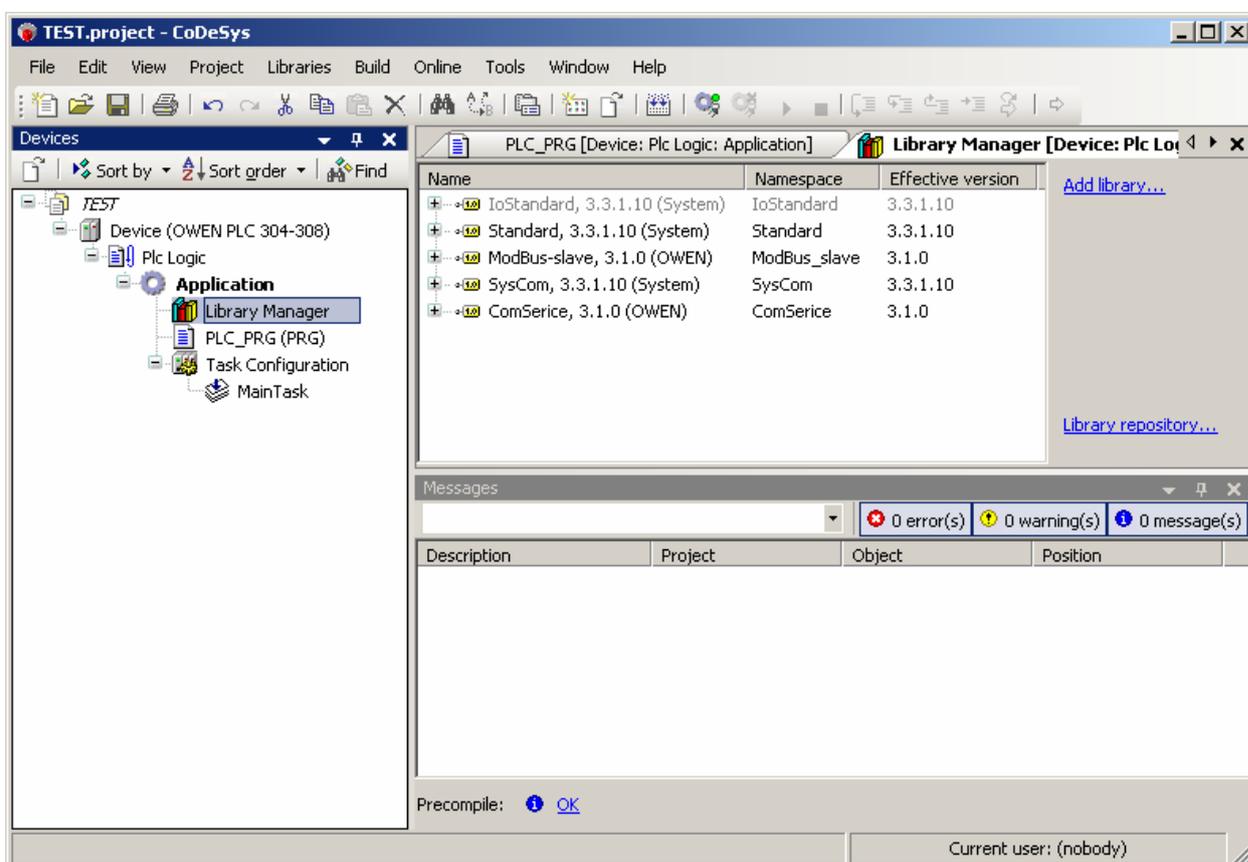
## 1 Установка дополнительных библиотек для проекта

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**)».

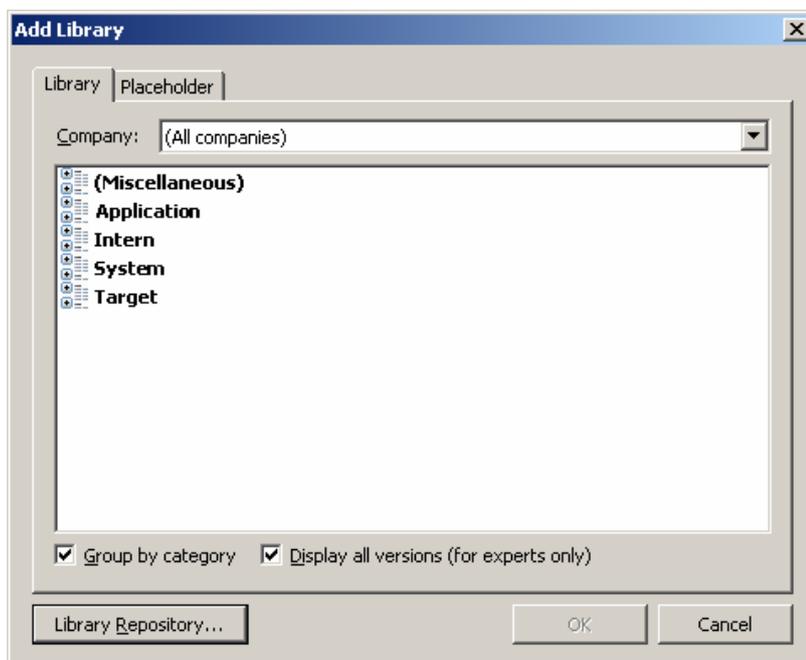
Менеджер библиотек открывают из главного меню CoDeSys командами **Window | Library Manager** или через менеджер проекта (вкладка **«Devices»**) открывают папку **«Library Manager»** (рисунок 1.1). В средней верхней части окна отображается список установленных библиотек.



**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):

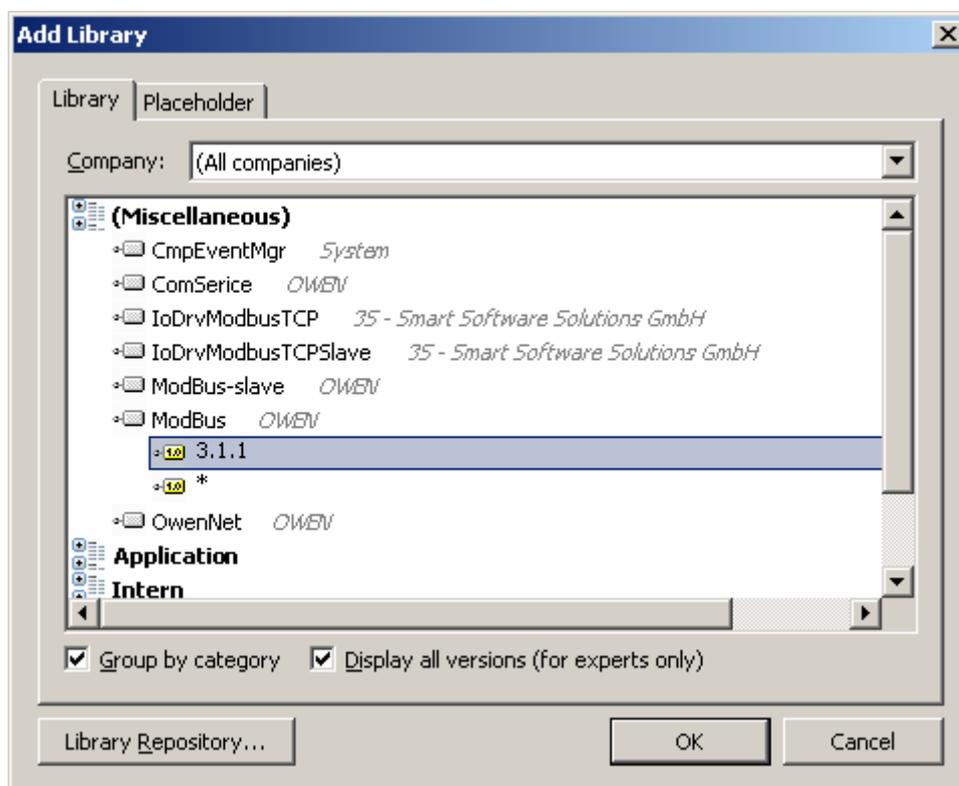


**Рисунок 1.2 – Окно добавления библиотеки.**

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

После закрытия диалога выбора файла открыть раздел Miscellaneous и и проверить, что библиотека добавилась.

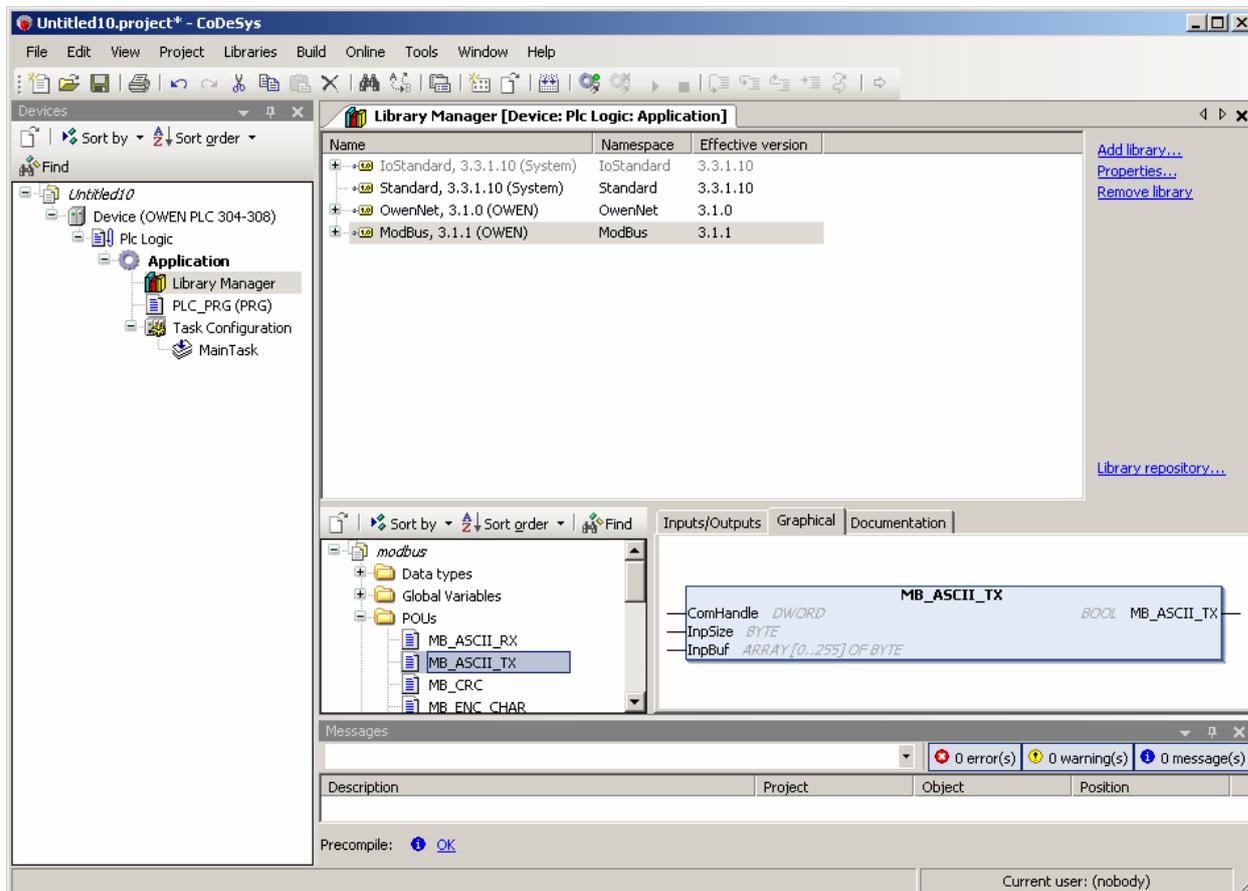
Затем нажать кнопку Close. Вы снова окажетесь в окне добавления библиотеки. Необходимо войти в раздел Miscellaneous, выбрать библиотеку



**Рисунок 1.3 – Окно добавления библиотеки – выбор версии библиотеки.**

ModBus и последнюю версию, а затем нажать кнопку ОК (рисунок 1.3):

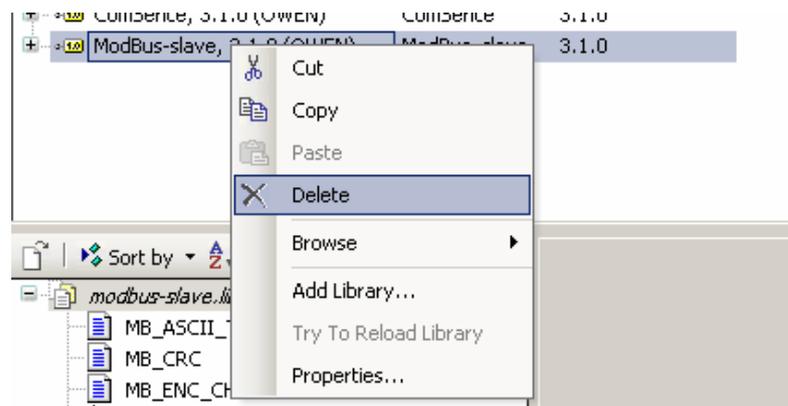
Для просмотра состава и свойств функциональных блоков следует выбрать требуемую библиотеку. Откроется окно с перечнем блоков (рис. 1.4, в нижней части окна) и краткой информацией о них.



**Рисунок 1.4 – Library manager (управление библиотеками) – просмотр функциональных блоков.**

Для удаления дополнительных библиотек следует:

- 1) Выделить требуемую библиотеку в списке библиотек (см. рисунок 1.4) и выбрать ссылку «**Remove library**» справа на вкладке или команду Delete контекстного меню (рисунок 1.5):



**Рисунок 1.5 – Library manager (управление библиотеками) – удаление библиотеки.**

**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 Master)

### 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**.

Библиотека содержит структуру, описывающую режимы работы протокола Modbus:

```
TYPE MB_MODE:      (* Serial Transmission Modes of MODBUS networks *)
(
    MB_RTU := 0,  (*MODBUS RTU *)
    MB_ASCII:= 1  (*MODBUS ASCII*)
);
END_TYPE
```

### 2.2 Инициализация и открытие порта

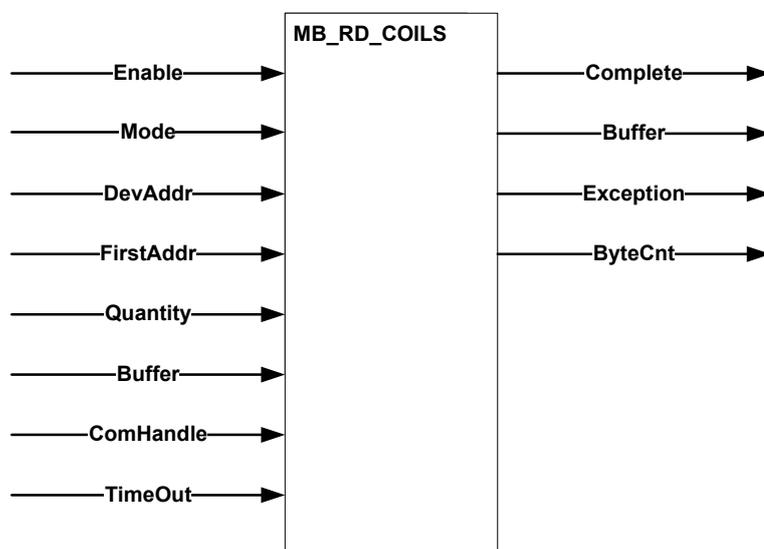
Для работы с библиотеками ModBus и OWENNET на ПЛК требуется сначала проинициализировать и открыть COM-порт. Для открытия порта используется специальная библиотека **ComService.compiled-library**.

```
(*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 передачи
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 Чтение статуса дискретных выходов (0x01) Read Coils



#### 2.3.1.1 Входы:

- **Enable:** BOOL – разрешение работы блока
- **Mode:** MB\_MODE – режим передачи ASCII/RTU.
- **DevAddr:** BYTE (1..247) – адрес ведомого устройства в сети Modbus
- **FirstAddr:** WORD – номер первой ячейки для чтения
- **Quantity:** WORD(1..2000) – количество считываемых ячеек
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom
- **TimeOut:** TIME – тайм-аут [мс] – максимальная задержка ведомого устройства на обработку запроса (интервал времени после получения корректного запроса и отправкой первого символа квитанции). Рекомендуемое значение 10мс.

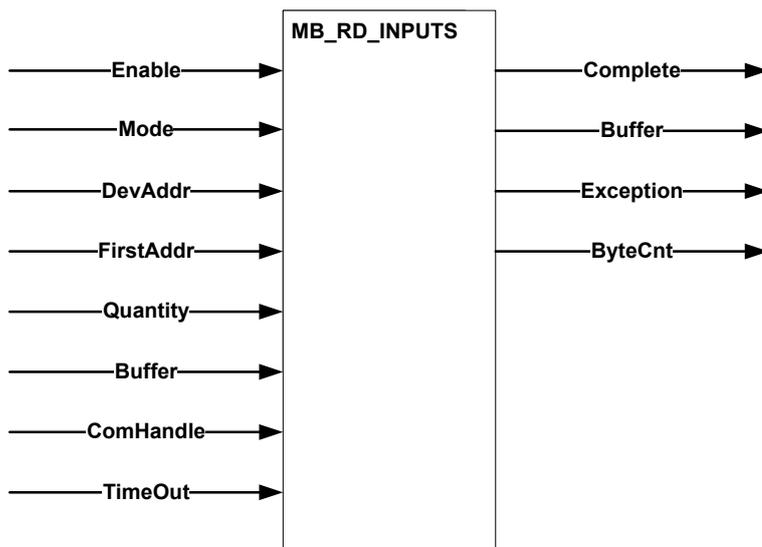
#### 2.3.1.2 Входы/выходы:

- **Buffer:** ARRAY [0..255] OF BYTE – массив для чтения

#### 2.3.1.3 Выходы:

- **Complete:** BOOL – признак завершения операции
- **Exception:** BYTE – исключение протокола MODBUS или ошибка:
  - 0x00 – ошибок или исключений MODBUS не обнаружено,
  - 0xFE – аппаратная ошибка ПЛК,
  - 0xFF – ошибка таймаута.
- **ByteCnt:** BYTE – кол-во считанных байтов (не ячеек)

## 2.3.2 Чтение статуса дискретных входов (0x02) Read Discrete Inputs



### 2.3.2.1 Входы:

- **Enable:** BOOL; – разрешение работы блока
- **Mode:** MB\_MODE; – режим передачи ASCII/ RTU.
- **DevAddr:** BYTE (1..247) – адрес ведомого устройства
- **FirstAddr:** WORD – номер первой ячейки для чтения
- **Quantity:** WORD(1..2000) – количество считываемых ячеек
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom
- **TimeOut:** TIME – тайм-аут [мс] – максимальная задержка ведомого устройства на обработку запроса (интервал времени после получения корректного запроса и отправкой первого символа квитанции).  
Рекомендуемое значение 10мс.

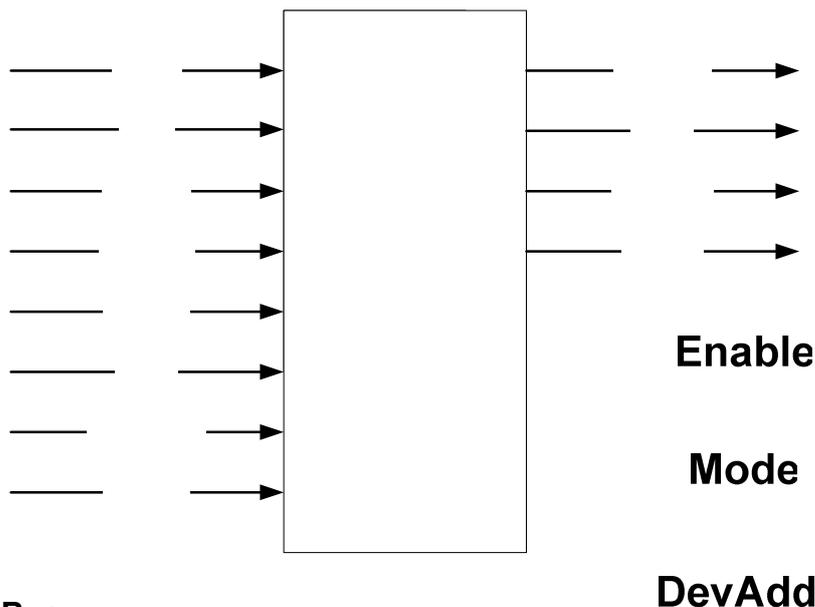
### 2.3.2.2 Входы/выходы:

- **Buffer:** ARRAY [0..255] OF BYTE – массив для чтения

### 2.3.2.3 Выходы:

- **Complete:** BOOL – признак завершения операции
- **Exception:** BYTE – исключения протокола MODBUS или ошибки:
  - 0x00 – ошибок или исключений MODBUS не обнаружено,
  - 0xFE – аппаратная ошибка ПЛК,
  - 0xFF – ошибка таймаута.
- **ByteCnt:** BYTE – кол-во считанных байтов (не ячеек)

### 2.3.3 Чтение содержания регистров (0x03) Read Holding Registers



#### 2.3.3.1 Входы:

- **Enable:** BOOL – разрешение работы блока
- **Mode:** MB\_MODE – режим передачи ASCII/ RTU
- **DevAddr:** BYTE (1..247) – адрес ведомого устройства
- **FirstAddr:** WORD – номер первого регистра для чтения
- **Quantity:** BYTE (1..125) – количество считываемых регистров
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom
- **Timeout:** TIME – тайм-аут [мс] – максимальная задержка ведомого устройства на обработку запроса (интервал времени после получения корректного запроса и отправкой первого символа квитанции). Рекомендуемое значение 10мс.

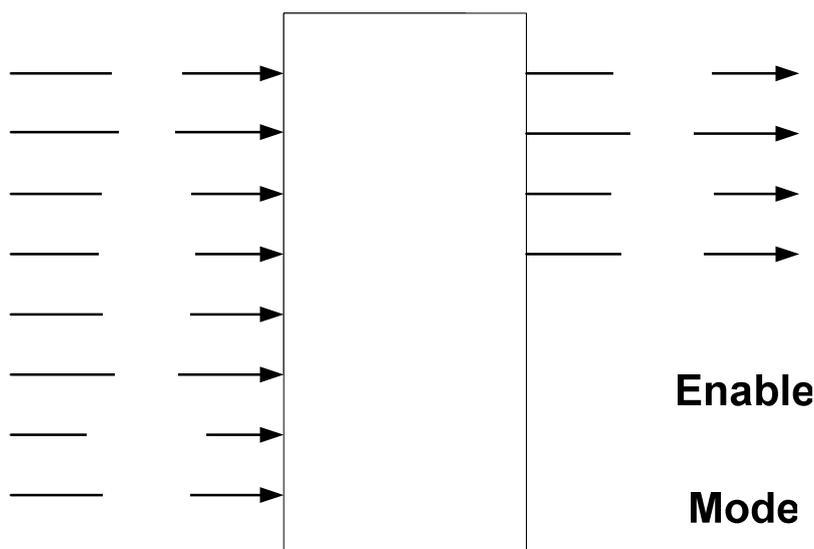
#### 2.3.3.2 Входы/выходы:

- **Buffer:** ARRAY [0..255] OF BYTE – массив для чтения

#### 2.3.3.3 Выходы:

- **Complete:** BOOL – признак завершения операции
- **Exception:** BYTE – исключения протокола MODBUS или ошибки:
  - 0x00 – ошибок или исключений MODBUS не обнаружено,
  - 0xFE – аппаратная ошибка ПЛК,
  - 0xFF – ошибка таймаута.
- **ByteCnt:** BYTE – кол-во считанных байтов (не регистров)

### 2.3.4 Чтение содержания входных регистров (0x04) Read Input Registers



#### 2.3.4.1 Входы:

- **Enable**: BOOL – разрешение работы блока
- **Mode**: MB\_MODE – режим передачи ASCII/RTU.
- **DevAddr**: BYTE (1..247) – адрес ведомого устройства
- **FirstAddr**: WORD – номер первого регистра для чтения
- **Quantity**: BYTE (1..125) – количество считываемых регистров
- **ComHandle**: DWORD – дескриптор последовательного порта библиотеки SysLibCom
- **TimeOut**: TIME – тайм-аут [мс] – максимальная задержка ведомого устройства на обработку запроса (интервал времени после получения корректного запроса и отправкой первого символа квитанции). Рекомендуемое значение 10мс.

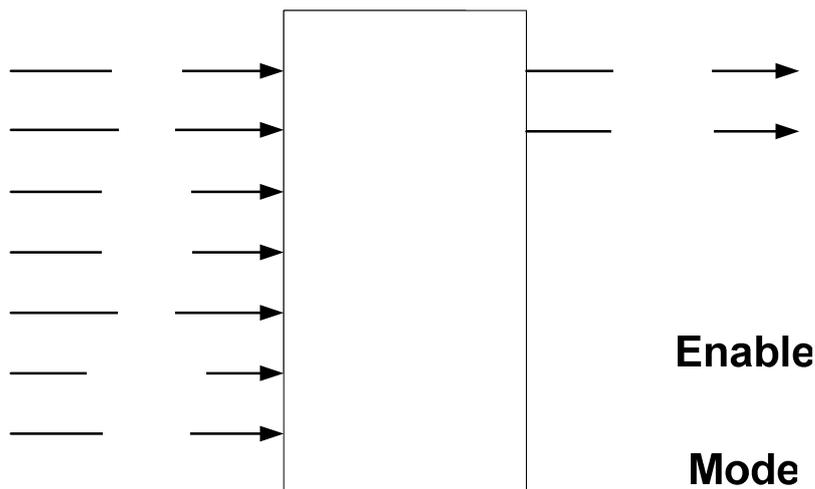
#### 2.3.4.2 Входы/выходы:

- **Buffer**: ARRAY [0..255] OF BYTE – массив для чтения

#### 2.3.4.3 Выходы:

- **Complete**: BOOL – признак завершения операции
- **Exception**: BYTE – исключения протокола MODBUS или ошибки:
  - 0x00 – ошибок или исключений MODBUS не обнаружено,
  - 0xFE – аппаратная ошибка ПЛК,
  - 0xFF – ошибка таймаута.
- **ByteCnt**: BYTE – кол-во считанных байтов (не регистров)

### 2.3.5 Установка единичного выхода (0x05) Write Single Coil



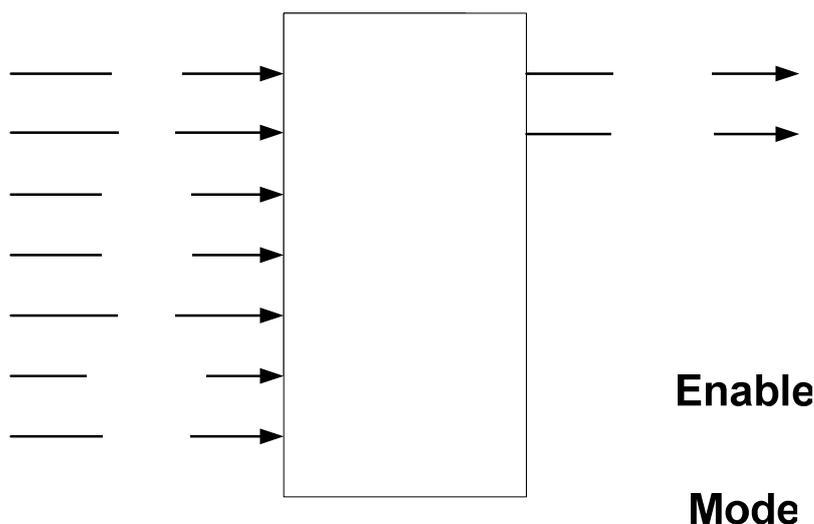
#### 2.3.5.1 Входы:

- **Complete:** BOOL – признак завершения операции
- **Enable:** BOOL – разрешение работы блока
- **Mode:** MB\_MODE – режим передачи ASCII/RTU.
- **DevAddr:** BYTE (0..247) – адрес ведомого устройства
- **CoilAddr:** WORD – номер ячейки для записи
- **Value:** BOOL – значение единичного выхода (ячейки)
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom
- **TimeOut:** TIME – тайм-аут [мс] – максимальная задержка ведомого устройства на обработку запроса (интервал времени от получения корректного запроса и отправкой первого символа квитанции, данный интервал выдерживается так же при работе с широковещательным адресом для гарантированного окончания обработки запроса). Рекомендуемое значение 10мс.

#### 2.3.5.2 Выходы:

- **Complete:** BOOL – признак завершения операции
- **Exception:** BYTE – исключения протокола MODBUS или ошибки:
  - 0x00 – ошибок или исключений MODBUS не обнаружено,
  - 0xFE – аппаратная ошибка ПЛК,
  - 0xFF – ошибка таймаута.

### 2.3.6 Запись значения единичного регистра (0x06) Write Single Register



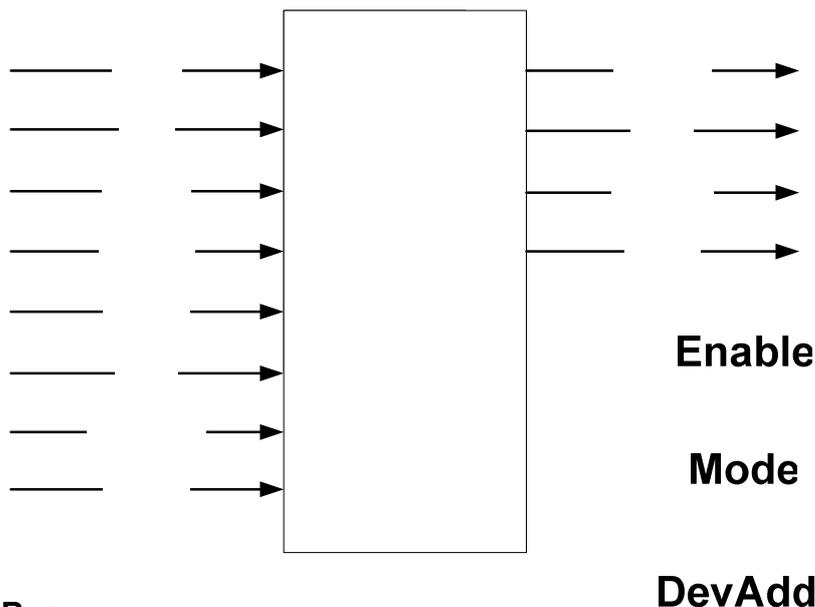
#### 2.3.6.1 Входы:

- **Enable:** BOOL – разрешение работы блока
- **Mode:** MB\_MODE – режим передачи ASCII/RTU.
- **DevAddr:** BYTE (0..247) – адрес ведомого устройства
- **RegAddr:** WORD – номер регистра для записи
- **Value:** WORD – значение регистра
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom
- **TimeOut:** TIME – тайм-аут [мс] – максимальная задержка ведомого устройства на обработку запроса (интервал времени после получения корректного запроса и отправкой первого символа квитанции, данный интервал выдерживается так же при работе с аппаратным адресом для гарантированного окончания обработки запроса). Рекомендуемое значение 10мс.

#### 2.3.6.2 Выходы:

- **Complete:** BOOL – признак завершения операции
- **Exception:** BYTE – исключения протокола MODBUS или ошибки:
  - 0x00 – ошибок или исключений MODBUS не обнаружено,
  - 0xFE – аппаратная ошибка ПЛК,
  - 0xFF – ошибка таймаута.

### 2.3.7 Запись дискретных ячеек 15 (0x0F) Write Multiple Coils



#### 2.3.7.1 Входы:

- **Enable:** BOOL – разрешение работы блока
- **Mode:** MB\_MODE – режим передачи ASCII/RTU
- **DevAddr:** BYTE (0..247) – адрес ведомого устройства
- **FirstAddr:** WORD – номер первой ячейки для записи
- **Quantity:** WORD(1..2000) – количество записываемых ячеек
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom
- **TimeOut:** TIME – тайм-аут [мс] – максимальная задержка ведомого устройства на обработку запроса (интервал времени после получения корректного запроса и отправкой первого символа квитанции, данный интервал выдерживается так же при работе с широковещательным адресом для гарантированного окончания обработки запроса). Рекомендуемое значение 10мс.

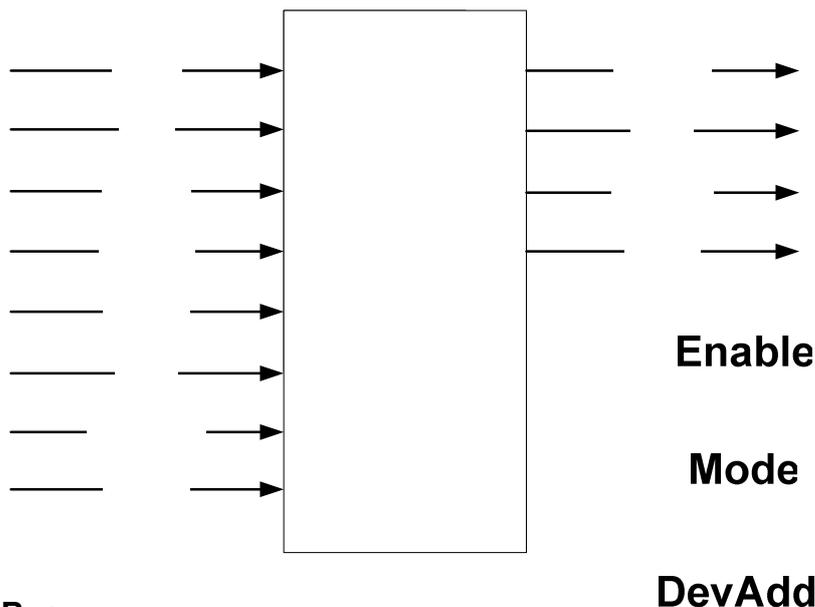
#### 2.3.7.2 Входы/выходы:

- **Buffer:** ARRAY [0..255] OF BYTE – массив для записи

#### 2.3.7.3 Выходы:

- **Complete:** BOOL – признак завершения операции
- **Exception:** BYTE – исключения протокола MODBUS или ошибки:
  - 0x00 – ошибок или исключений MODBUS не обнаружено,
  - 0xFE – аппаратная ошибка ПЛК,
  - 0xFF – ошибка таймаута.
- **CoilCnt:** WORD – кол-во записанных ячеек

### 2.3.8 Запись значения регистров 16 (0x10) Write Multiple registers



#### 2.3.8.1 Входы:

- **Enable:** BOOL – разрешение работы блока
- **Mode:** MB\_MODE – режим передачи ASCII/RTU
- **DevAddr:** BYTE (0..247) – адрес ведомого устройства
- **FirstAddr:** WORD – номер первого регистра для записи
- **Quantity:** WORD(1..125) – количество записываемых регистров
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom
- **TimeOut:** TIME – тайм-аут [мс] – максимальная задержка ведомого устройства на обработку запроса (интервал времени после получения корректного запроса и отправкой первого символа квитанции, данный интервал выдерживается так же при работе с широковещательным адресом для гарантированного окончания обработки запроса). Рекомендуемое значение 10мс.

#### 2.3.8.2 Входы/выходы:

- **Buffer:** ARRAY [0..255] OF BYTE – массив для записи

#### 2.3.8.3 Выходы:

- **Complete:** BOOL – признак завершения операции
- **Exception:** BYTE – исключения протокола MODBUS или ошибки:
  - 0x00 – ошибок или исключений MODBUS не обнаружено,
  - 0xFE – аппаратная ошибка ПЛК,
  - 0xFF – ошибка таймаута.
- **RegCnt:** BYTE – кол-во записанных регистров

#### 2.3.8.4 Примечания

При операциях записи с широковещательным адресом DevAddr = 0 запись осуществляется во все устройства, подключенные к сети MODBUS. Ответ (квитанция) устройствами при этом не возвращается.

## Приложение А

Пример на языке ST реализует работу по интерфейсу RS-485: чтение параметров типа Int и типа Real (Float) (функция Modbus 03 - MB\_RD\_HOLD\_REGS), а также считывания трех параметров типа Int (функция Modbus 04 - MB\_RD\_INP\_REGS). Для работы примера в PLC Configuration должны быть подключены библиотеки SisLibCom, Modbus, ComService.

```

PROGRAM PLC_PRG
VAR
  get1_modbus: MB_RD_HOLD_REGS; (*функция 03 - чтение параметра
    типа INT*)
  get2_modbus: MB_RD_INP_REGS; (*функция 04 - чтение трех
    параметров типа INT*)
  get3_modbus: MB_RD_HOLD_REGS; (*функция 03 - чтение параметра
    типа Float*)

  Buffer: ARRAY[0..255] OF BYTE;      (* байтовый буфер данных
    *)
  smpl: BOOL;
  port_opened: BYTE := 0;
  Init: BOOL; (* признак инициализации пользовательской
    программы *)
  Settings: COMSETTINGS; (* настройки последовательного порта *)
  com_num: PORTS:=0; (*0 - RS-485, 1 - RS-232*)
  enabl: BOOL; (*состояние работы блока*)
  err: INT; (*номер ошибки*)
  TimeOut: TIME:=T#50ms; (*таймаут*)
  Exception: BYTE;
  DataSize: WORD;
  master1: BYTE:= 1;

  t: DWORD; (*переменная для организации
    счетчика*)
  A: WORD := 0; (*счетчик*)
  x:WORD; (*считанное значение*)
  x1: WORD; (*переменная для записи по сети*)
  x2: WORD; (*переменная для записи по сети*)
  x3: WORD; (*переменная для записи по сети*)
  d: REAL; (*считанное значение*)
  ptr_D: POINTER TO BYTE;

  COM_SERVICE1: COM_SERVICE;
END_VAR

(*Организуем счетчик, что бы передавать эти данные по сети*)
t:=t+1;
IF (t MOD 1000)=0 THEN
  A := A + 1;
  IF A > 9999 THEN
    A := 0;
  
```

```

        END_IF
    END_IF

    (*Устанавливаем настройки COM-порта*)
    IF port_opened=0 THEN
        Settings.Port:=com_num;           (*номер COM-порта*)
        Settings.dwBaudRate:=115200;     (*скорость*)
        Settings.byParity:=0;
        Settings.dwTimeout:=0;
        Settings.byStopBits:=0;
        Settings.dwBufferSize:=0;
        Settings.dwScan:=0;
    END_IF

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

    IF port_opened=2 THEN (*Удачно проинициализировали*)

    CASE master1 OF

    0: (* функция 03 инт - ФБ считывает значение параметра типа
        int из прибора с адресом 2 в регистр с номером 8 по
        протоколу Modbus-ASCII*)
    get1_modbus(
        Enable:=enabl ,           (* разрешение работы блока *)
        Mode:=MB_ASCII ,         (*режим передачи*)
        DevAddr:=2 ,             (*адрес*)
        FirstAddr:=8 ,          (*номер регистра*)
        Quantity:=1,            (*количество регистров*)
        ComHandle:=Settings.Port , (*номер COM-порта*)
        TimeOut:=TimeOut ,      (*Таймаут T#50ms*)
        Buffer:=Buffer ,         (* буфер данных *)
        Complete=>smpl ,         (* скопировать признак завершения
        операции *)
        Exception=>err ,         (* скопировать регистр ошибок *)
        ByteCnt=>DataSize );     (*кол-во считанных байтов *)
    (*если установлен признак завершения операции, то *)
    IF smpl THEN
        IF err=0 THEN (*Если нет ошибок, то получаем данные из буфера
            типа INT*)
            x:=BYTE_TO_WORD(BUFFER[1]) OR
                SHL(BYTE_TO_WORD(BUFFER[0]), 8);
        END_IF
        master1:=1; (*переходим к выполнению следующего ФБ*)
    END_IF

    1: (* функция 03 флоат - ФБ считывает значение параметра типа

```

```

        int из прибора с адресом 2 в регистр с номером 10 по
        протоколу Modbus-ASCII *)
get3_modbus(
    Enable:=enabl ,                (* разрешение работы блока *)
    Mode:=MB_ASCII ,              (*режим передачи*)
    DevAddr:=2 ,                  (*адрес*)
    FirstAddr:=10 ,              (*номер регистра*)
    Quantity:=2,                  (*количество регистров*)
    ComHandle:=Settings.Port ,    (*номер СОМ-порта*)
    TimeOut:=TimeOut ,           (*Таймаут Т#50ms*)
    Buffer:=Buffer ,              (* буфер данных *)
    Complete=>cmpl ,              (* скопировать признак завершения
        операции *)
    Exception=>err ,              (* скопировать регистр ошибок *)
    ByteCnt=>DataSize );         (*кол-во считанных байтов *)
(*если установлен признак завершения операции, то *)
IF cmpl THEN
    master1:=2; (*переходим к выполнению следующего ФБ*)
    IF err=0 THEN (*Если нет ошибок, то получаем данные из
        буфера типа FLOAT*)
        ptr_D:=ADR(d);
        ptr_D^:=buffer[1];
        ptr_D:=ptr_D+1;
        ptr_D^:=buffer[0];
        ptr_D:=ptr_D+1;
        ptr_D^:=buffer[3];
        ptr_D:=ptr_D+1;
        ptr_D^:=buffer[2];
    END_IF
END_IF

2: (* функция 04 инт - ФБ считывает значения трех параметров
    типа Int из прибора с адресом 2 начиная с регистра с
    номером 12*)
get2_modbus(
    Enable:= enabl,                (* разрешение работы блока *)
    Mode:=MB_ASCII ,              (*режим передачи*)
    DevAddr:=2 ,                  (*адрес*)
    FirstAddr:=12 ,              (*номер регистра*)
    Quantity:=6 ,                 (*количество регистров*)
    ComHandle:= Settings.Port,    (*номер СОМ-порта*)
    TimeOut:=TimeOut ,           (*Таймаут Т#50ms*)
    Buffer:=Buffer ,              (* буфер данных *)
    Complete=>cmpl ,              (* скопировать признак завершения
        операции *)
    Exception=>err ,              (* скопировать регистр ошибок *)
    ByteCnt=> DataSize);         (*кол-во считанных байтов *)
(*если установлен признак завершения операции, то *)
IF cmpl THEN
    IF err=0 THEN (*Если нет ошибок, то получаем данные из буфера
        типа INT*)
        x1:=BYTE_TO_WORD(BUFFER[1]) OR
        SHL(BYTE_TO_WORD(BUFFER[0]), 8);

```

```
        x2:=BYTE_TO_WORD(BUFFER[5]) OR
        SHL(BYTE_TO_WORD(BUFFER[4]),8);
        x3:=BYTE_TO_WORD(BUFFER[9]) OR
        SHL(BYTE_TO_WORD(BUFFER[8]),8);
    END_IF
    master1:=0;(*переходим к выполнению следующего ФБ*)
END_IF
END_CASE

IF enabl = FALSE THEN
    enabl := TRUE;
END_IF

IF err <> 0 THEN
    enabl := FALSE;
END_IF

END_IF
```

## Приложение Б

Пример на языке ST реализует работу по интерфейсу RS-485: запись параметров типа Int (функция Modbus 06 - **MB\_WR\_SNG\_REG**) и записи параметра типа Float (функция Modbus 16 - **MB\_WR\_REGS**). Для работы примера в PLC Configuration должны быть подключены библиотеки SisLibCom, Modbus, ComService.

```

PROGRAM PLC_PRG
VAR
    send1_modbus:MB_WR_SNG_REG; (*функция 06 - запись параметра*)
    send2_modbus: MB_WR_REGS;    (*функция 16 - запись
    параметров*)
    COM_SERVICE1: COM_SERVICE;    (*ФБ открытия порта*)

    Buffer: ARRAY[0..255] OF BYTE; (* байтовый буфер данных *)
    cmpl: BOOL;                    (* признак завершения операции *)
    port_opened: BYTE := 0;        (*состояние порта*)
    Init: BOOL;                    (* признак инициализации
    пользовательской программы *)
    Settings:COMSETTINGS;          (* настройки последовательного
    порта *)
    com_num: PORTS:=1;             (*0 - RS-485, 1 - RS-232*)
    enabl: BOOL;                  (*состояние работы блока*)
    err: INT;                     (*номер ошибки*)
    TimeOut: TIME:=T#50ms;        (*таймаут*)
    Exception: BYTE;
    DataSize: WORD;
    master1: BYTE;

    t: DWORD;                     (*переменная для организации
    счетчика*)
    A: WORD := 0;                 (*счетчик*)
    f1:DINT:=65;                  (*Записываемый параметр типа DINT*)
    f2:REAL:=12.7;                (*Записываемый параметр типа REAL*)
    ptr_f2:POINTER TO BYTE;

END_VAR

(*Организуем счетчик, что бы передавать эти данные по сети*)
t:=t+1;
IF (t MOD 1000)=0 THEN
    A := A + 1;
    IF A > 9999 THEN
        A := 0;
    END_IF
END_IF

(*Устанавливаем настройки COM-порта*)
IF port_opened=0 THEN
    Settings.Port:=com_num;        (*номер COM-порта*)

```

```

        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 передачи данных
*)
IF COM_SERVICE1.ready THEN
    port_opened:=2;
END_IF

IF port_opened=2 THEN (*Удачно проинициализировали*)

CASE master1 OF

0: (* функция 06 - запись параметра типа Int в прибор с
    адресом 1 в регистр 0*)
send1_modbus(
    Enable:=enabl ,                (* разрешение работы блока *)
    Mode:=MB_ASCII ,              (*режим передачи*)
    DevAddr:=1 ,                  (*адрес*)
    RegAddr:=0 ,                  (*регистр*)
    Value:=A,                      (*значение переменной, которое
    надо передать*)
    ComHandle:= Settings.Port,     (*номер ком-порта*)
    TimeOut:=TimeOut ,            (*таймаут T#50ms*)
    Complete=>cmpl ,               (* скопировать признак
    завершения операции *)
    Exception=>err );              (* скопировать регистр ошибок
    *)
(*если установлен признак завершения операции, то *)
IF cmpl THEN
    master1:=1; (*переходим к выполнению следующего блока*)
END_IF

1: (* функция 16 - запись параметров типа Int (регистр 4) и
    Real (регистр 6) в прибор с адресом 2 *)
(*запись в буффер параметра типа INT*)
Buffer[1] := DINT_TO_BYTE(f1);
Buffer[0] := DINT_TO_BYTE( SHR(f1,8));
Buffer[3] := DINT_TO_BYTE( SHR(f1,16));
Buffer[2] := DINT_TO_BYTE( SHR(f1,24));

(*запись в буффер параметра типа Float*)
ptr_f2:=ADR(f2);
    buffer[5] := ptr_f2^;
    ptr_f2:=ptr_f2+1;

```

```
    buffer[4] := ptr_f2^;  
    ptr_f2:=ptr_f2+1;  
    buffer[7] := ptr_f2^;  
    ptr_f2:=ptr_f2+1;  
    buffer[6] := ptr_f2^;  
  
send2_modbus (  
    Enable:= enabl,           (* разрешение работы блока *)  
    Mode:=MB_ASCII ,        (*режим передачи*)  
    DevAddr:=2 ,            (*адрес*)  
    FirstAddr:= 4,          (*номер регистра*)  
    Quantity:= 4,          (*количество записываемых  
        регистров*)  
    ComHandle:=Settings.Port , (*номер сом-порта*)  
    TimeOut:=TimeOut ,     (*таймаут T#50ms*)  
    Buffer:=Buffer ,        (* буфер данных *)  
    Complete=>cmpl ,        (* скопировать признак завершения  
        операции *)  
    Exception=>err ,        (* скопировать регистр ошибок *)  
    RegCnt=> DataSize);     (*кол-во считанных байтов *)  
(*если установлен признак завершения операции, то *)  
IF cmpl THEN  
    master1:=0; (*переходим к выполнению следующего блока*)  
END_IF  
  
END_CASE  
  
IF enabl = FALSE THEN  
    enabl := TRUE;  
END_IF  
IF err <> 0 THEN  
    enabl := FALSE;  
END_IF  
  
END_IF
```