

**Описание интерфейса библиотеки ФБ
для работы с протоколом ОВЕН**

Руководство пользователя

Содержание:

1	Установка дополнительных библиотек для проекта.....	3
2	Описание интерфейса библиотеки ФБ для работы с протоколом Овен	5
2.1	Введение	5
2.1.1	Структура задает длину адреса для протокола ОВЕН.....	5
2.1.2	Структура описывает форматы целого числа	5
2.1.3	Структура описывает форматы данных протокола ОВЕН	5
2.1.4	Структура описывает прием/передачу.....	6
2.2	Инициализация и открытие порта	6
2.3	Чтение параметра типа Real (числа с плавающей точкой).....	7
2.3.1	Входы	7
2.3.2	Выходы	8
2.3.3	Чтение параметра типа DINT (целого числа)	8
2.3.3.1	Входы	8
2.3.3.2	Выходы	8
2.3.4	Запись параметра типа Real (числа с плавающей точкой)	9
2.3.4.1	Входы	9
2.3.4.2	Выходы	9
2.3.5	Запись параметра типа DINT (целого числа)	10
2.3.5.1	Входы	10
2.3.5.2	Выходы	10
2.3.6	Универсальная функция чтения-записи нетипизированных данных	11
2.3.6.1	Входы	11
2.3.6.2	Выходы	11
2.3.6.3	Вход-Выход (VAR_IN_OUT)	11
2.3.7	Функция прослушивания сети.....	12
2.3.7.1	Входы	12
2.3.7.2	Выходы	12
2.3.7.3	Вход-Выход (VAR_IN_OUT)	12
2.3.8	Опрос пула команд по протоколу Овен.....	12
	Приложение А.....	14
	Приложение Б.....	17

1 Установка дополнительных библиотек для проекта

В системе CoDeSys все файлы библиотек функциональных блоков имеют расширение ***.lib (Library)** и находятся в папке **«Library»**, расположенной по месту размещения основной программы на диске компьютера (как правило, это **«C:\Program Files\3S Software\CoDeSys V2.3\Library»**).

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 **«Resources»**) открывают папку **«Library Manager»** (рисунок 1.1). В средней верхней части окна отображается список установленных библиотек.

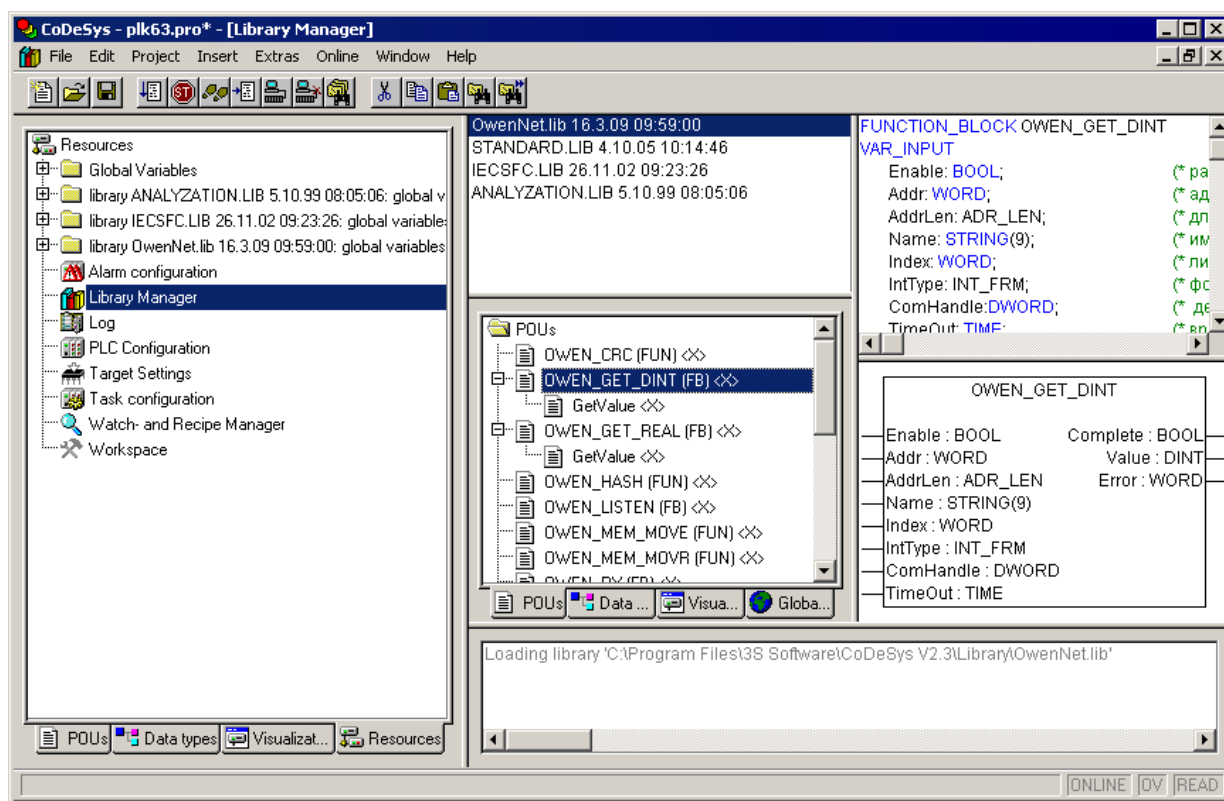


Рисунок 1.1 – Окно вкладки «Ресурсы (Resources)» с открытой папкой «Library Manager»

Для установки дополнительных библиотек следует:

- 1) Выбрать команду **«Insert | Additional Library»** главного меню.
- 2) В открывшемся окне выбора файла (по умолчанию открывается папка **«Library»**) – выбрать файл с именем нужной библиотеки (рисунок 1.2) и нажать кнопку **Открыть**. Дополнительная библиотека установлена.

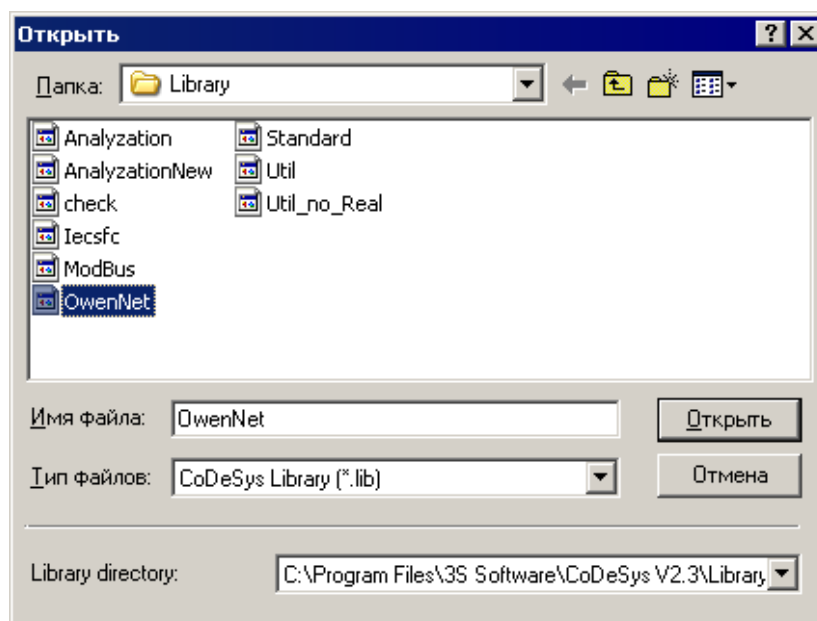


Рисунок 1.2 – Окно выбора подключаемой к проекту дополнительной библиотеки

Для просмотра состава и свойств функциональных блоков следует выбрать требуемую библиотеку. Откроется окно с перечнем блоков (рис. 1.1, в правой части окна) и краткой информацией о них.

Для удаления дополнительных библиотек следует:

- 1) Выделить требуемую библиотеку в списке библиотек (см. рисунок 1.3) и выбрать команду «**Delete**» контекстного меню или команду «**Edit | Delete**» главного меню.

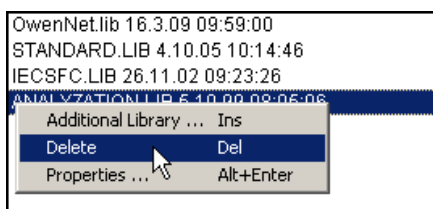


Рисунок 1.3 – Удаление дополнительной библиотеки

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 Овен

2.1 Введение

Библиотека предназначена для работы с параметрами (чтение/запись/прослушивание сети) устройств через последовательный интерфейс ПЛК 63 стандартного протокола Овен.

Возможно так же применение данной библиотеки для других приборов на основе CoDeSys.

При подключении библиотеки **OwenNet.lib** автоматически подключается библиотека **SisLibCom.lib**.

Библиотека содержит следующие структуры:

2.1.1 Структура задает длину адреса для протокола ОВЕН

```
TYPE ADR_LEN: (* длина адреса протокола OWEN, бит *)
(
    A8BIT:= 8, (* длина адреса 8 бит *)
    A11BIT:= 11 (* длина адреса 11 бит *)
);
END_TYPE
```

2.1.2 Структура описывает форматы целого числа

```
TYPE INT_FRM : (* формат целого числа *)
(
    UINT_FRM := 0, (* целое без знака, размерность 1...4 байт *)
    SINT1_FRM := 1, (* целое со знаком, размерность signed char -
    1 байт *)
    SINT2_FRM := 2, (* целое со знаком, размерность signed short -
    2 байта *)
    SINT4_FRM := 3, (* целое со знаком, размерность signed long -
    4 байта *)
);
END_TYPE
```

2.1.3 Структура описывает форматы данных протокола ОВЕН

```
TYPE OWEN_FRM: (* форматы данных протокола ОВЕН *)
(
    OW_FL32T := 0, (* float IEEE754 + time *)
    OW_FL32 := 1, (* float IEEE754 *)
    OW_FL24 := 2, (* укороченное float IEEE754 *)
    OW_FX_BIN := 3, (* знаковое число с фикс. точкой в двоичном
    виде*)
    OW_FX_BCD := 4, (* знаковое число с фикс. точкой в двоично-
    десятичном виде *)
);
```

```

OW_SINT1 := 5, (* знаковое целое signed char - 1 байт*)
OW_SINT2 := 6, (* знаковое целое signed short - 2 байта *)
OW_SINT4 := 7, (* знаковое целое signed long - 4 байта *)
OW_UINT := 8, (* беззнаковое целое *)
OW_NTYPE := 9 (* нетипизированные данные *)
);
END_TYPE

```

2.1.4 Структура описывает прием/передачу

```

TYPE OwenPool:
STRUCT
    Every: TIME; (* период опроса *)
    IsRead: BOOL; (* true=ЧТЕНИЕ, false=ЗАПИСЬ *)
    EncodeType: OWEN_FRM;(* формат данных *)
    Addr: WORD; (* адрес устройства в сети *)
    AddrLen: ADR_LEN; (* длина адреса, 8 или 11 бит *)
    Name: STRING(9); (* имя параметра - например "read" *)
    Index: WORD; (* линейный индекс параметра,
0xFFFF - индекс не используется *)
    Complete: BOOL; (* если = TRUE, то данные получены и
актуальны *)
    TimeOut: TIME; (* время тайм-аута [мс] - макс.
задержка на обработку запроса *)
    ValReal: REAL; (* значение параметра с плавающей
точкой *)
    MeasTime: WORD; (* время измерителя *)
    ValDint: DINT; (* значение целого параметра *)
    Error: WORD; (* значение ошибки согласно
протокола OVEN *)
(* или 0xFFFF- ошибка таймаута *)

END_STRUCT
END_TYPE

```

2.2 Инициализация и открытие порта

Для работы с библиотеками ModBus и OWENNET на ПЛК63 требуется сначала проинициализировать и открыть COM-порт. Для открытия порта используется специальная библиотека **ComService.lib**.

```

(*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;

```

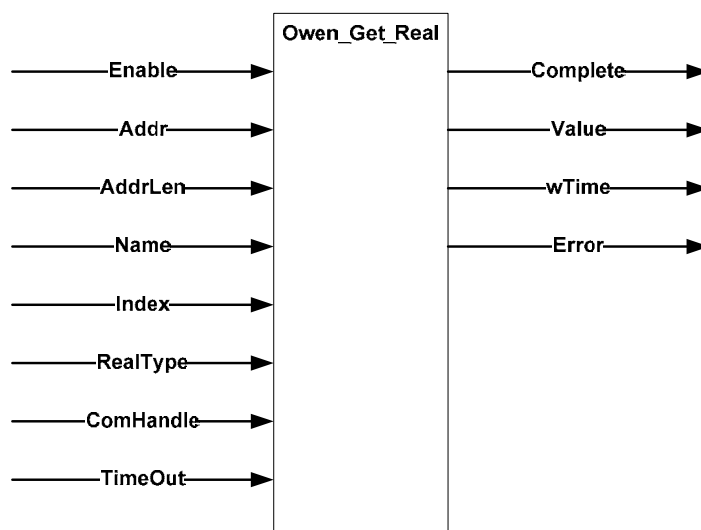
```

        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

```

Библиотека содержит функциональные блоки, перечисленные ниже:

2.3 Чтение параметра типа Real (числа с плавающей точкой)



2.3.1 Входы

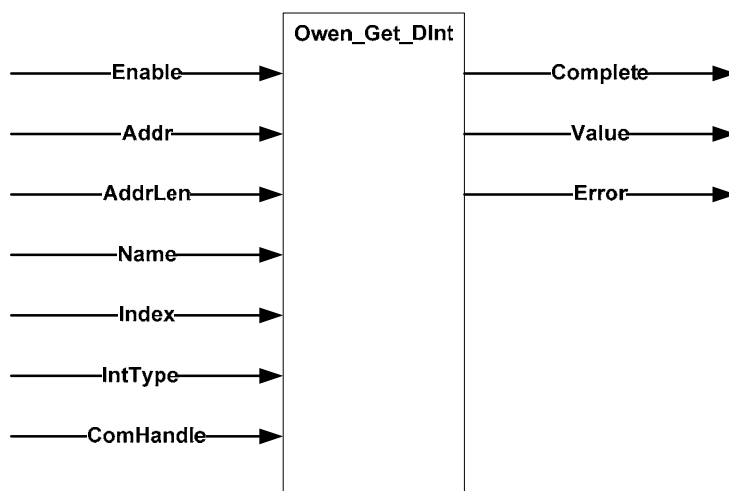
- **Enable:** BOOL – разрешение работы блока – по переднему фронту начинает обмен. По заднему фронту прекращает. Примечание: после остановки обмена необходимо выждать таймаут, достаточный для приема ответа от прибора и только после этого вновь можно разрешать повторную работу ФБ. В худшем случае для протокола ОВЕН задержка составляет 50мсек.
- **Addr:** WORD – адрес устройства в сети
- **AddrLen:** ADR_LEN – длина адреса
- **Name:** String(9) – имя параметра – например “read”
- **Index:** WORD – линейный индекс параметра. Если задан 0xFFFF, то индекс в послылке не используется.
- **RealType:** REAL_FRM – формат рационального числа
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom

- **TimeOut:** TIME – время тайм-аута [мс] – макс. задержка на обработку запроса. Рекомендуемое значение 50мс.

2.3.2 Выходы

- **Complete:** BOOL – если =TRUE, то данные получены и актуальны
- **Value:** REAL – полученное значение
- **wTime:** WORD – значение времени для измерителя
- **Error:** WORD – значение ошибки (исключительной ситуации) согласно протоколу OVEN или 0xffff – ошибка таймаута.

2.3.3 Чтение параметра типа DINT (целого числа)



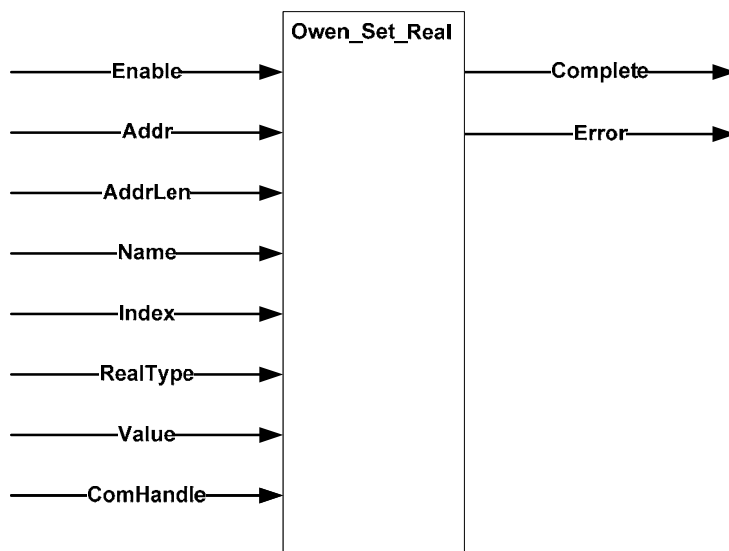
2.3.3.1 Входы

- **Enable:** BOOL – разрешение работы блока – по переднему фронту начинает обмен.
По заднему фронту прекращает. Примечание: после остановки обмена необходимо выждать таймаут, достаточный для приема ответа от прибора и только после этого вновь можно разрешать повторную работу ФБ. В худшем случае для протокола OVEN задержка составляет 50мсек.
- **Addr:** WORD – адрес устройства в сети
- **AddrLen:** ADR_LEN – длина адреса, 8 – 8 бит, 11 – 11 бит
- **Name:** String(9) – имя параметра – например “SP”
- **Index:** WORD – линейный индекс параметра. Если задан 0xFFFF, то индекс в посылке не используется.
- **IntType:** INT_FRM – формат целого числа
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom
- **TimeOut:** TIME – время тайм-аута [мс] – макс. задержка на обработку запроса. Рекомендуемое значение 50мс.

2.3.3.2 Выходы

- **Value:** Dint; -- прочитанное значение
- **Complete:** BOOL – если =TRUE, то данные получены и актуальны
- **Error:** WORD – Значение ошибки (исключительной ситуации) согласно протоколу OVEN или 0xffff – ошибка таймаута.

2.3.4 Запись параметра типа Real (числа с плавающей точкой)



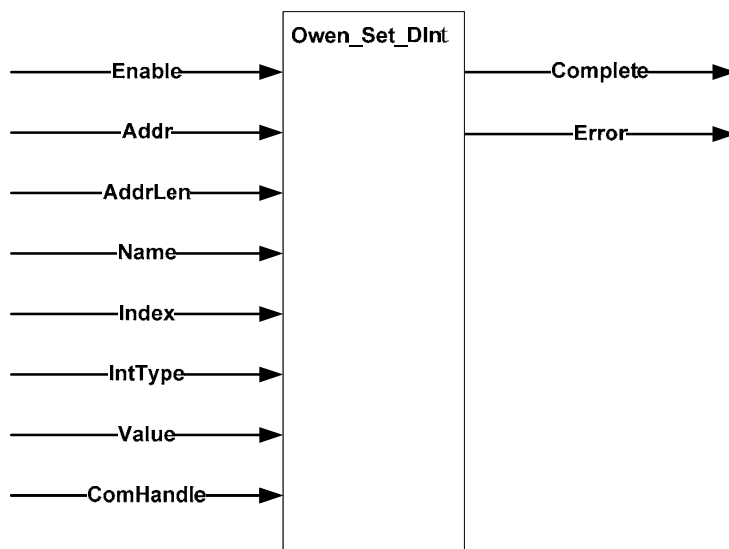
2.3.4.1 Входы

- **Enable:** BOOL – разрешение работы блока – по переднему фронту начинает обмен. По заднему фронту прекращает. Примечание: после остановки обмена необходимо выждать таймаут, достаточный для приема ответа от прибора и только после этого вновь можно разрешать повторную работу ФБ. В худшем случае для протокола ОВЕН задержка составляет 50мсек.
- **Addr:** WORD – адрес устройства в сети
- **AddrLen:** ADR_LEN – длина адреса, 8 – 8 бит, 11 – 11 бит
- **Name:** String(9) – имя параметра – например “read”
- **Index:** WORD – линейный индекс параметра. Если задан 0xFFFF, то индекс в посылке не используется.
- **RealType:** REAL_FRM – формат рационального числа
- **Value:** REAL – значение для записи
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom
- **TimeOut:** TIME – время тайм-аута [мс] – макс. задержка на обработку запроса. Рекомендуемое значение 50мс.

2.3.4.2 Выходы

- **Complete:** BOOL – если =TRUE, то данные получены и актуальны
- **Error:** WORD – Значение ошибки (исключительной ситуации) согласно протоколу ОВЕН или 0xffff – ошибка таймаута.

2.3.5 Запись параметра типа DINT (целого числа)



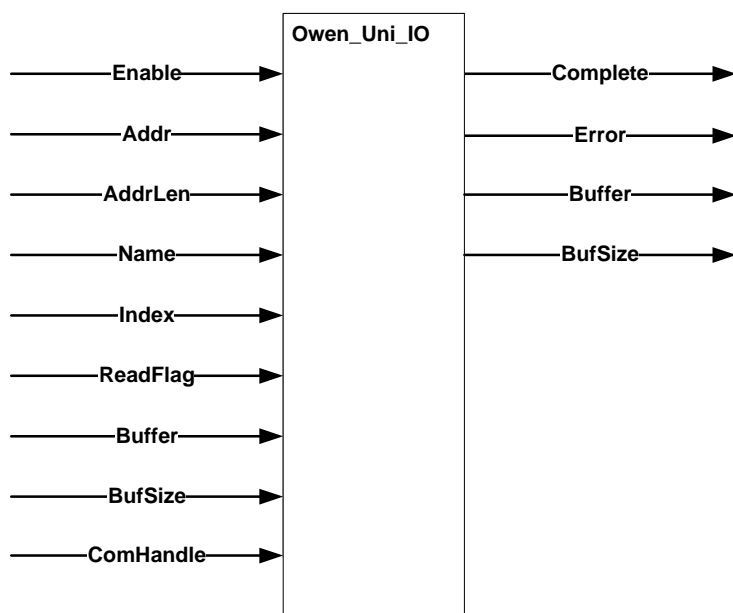
2.3.5.1 Входы

- **Enable:** BOOL – разрешение работы блока – по переднему фронту начинает обмен. По заднему фронту прекращает. Примечание: после остановки обмена необходимо выждать таймаут, достаточный для приема квитанции от прибора и только после этого вновь можно разрешать повторную работу ФБ. В худшем случае для протокола OVEN задержка составляет 50мсек.
- **Addr:** WORD – адрес устройства в сети
- **AddrLen:** ADR_LEN – длина адреса
- **Name:** String(9) – имя параметра – например “SP”
- **Index:** WORD – линейный индекс параметра. Если задан 0xFFFF, то индекс в посылке не используется.
- **IntType:** INT_FRM – формат целого числа
- **Value:** DINT – значение целого числа
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom
- **TimeOut:** TIME – время тайм-аута [мс] – макс. задержка на обработку запроса. Рекомендуемое значение 50мс.

2.3.5.2 Выходы

- **Complete:** BOOL – если =TRUE, то данные записаны
- **Error:** WORD – Значение ошибки (исключительной ситуации) согласно протоколу OVEN или 0xffff – ошибка таймаута.

2.3.6 Универсальная функция чтения-записи нетипизированных данных



2.3.6.1 Входы

- **Enable:** BOOL – разрешение работы блока – по переднему фронту начинает обмен. По заднему фронту прекращает. Примечание: после остановки обмена необходимо выждать таймаут, достаточный для приема квитанции от прибора и только после этого вновь можно разрешать повторную работу ФБ. В худшем случае для протокола ОВЕН задержка составляет 50мсек.
- **Addr:** WORD – адрес устройства в сети
- **AddrLen:** ADR_LEN – длина адреса, 8 – 8 бит, 11 – 11 бит
- **Name:** String(9) – имя параметра – например “read”
- **Index:** WORD – линейный индекс параметра. Если задан 0xFFFF, то индекс в посылке не используется.
- **ReadFlag:** BOOL – признак запроса данных. TRUE означает, что нужно осуществлять чтение данных в буфер, иначе записываем в буфер
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom
- **TimeOut:** TIME – время тайм-аута [мс] – макс. задержка на обработку запроса. Рекомендуемое значение 50мс.

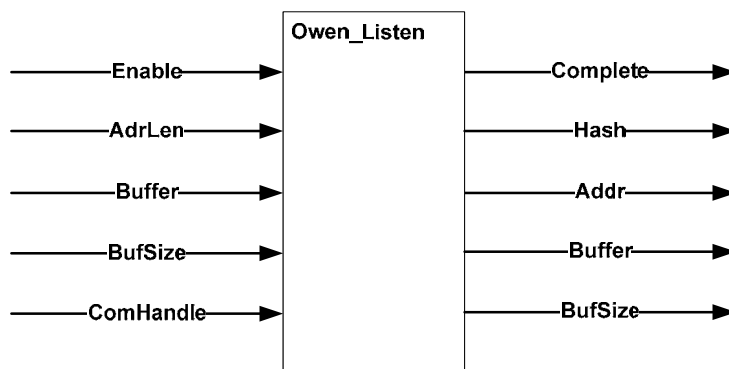
2.3.6.2 Выходы

- **Complete:** BOOL – если = TRUE, то данные получены и актуальны
- **Error:** WORD – Значение ошибки (исключительной ситуации) согласно протоколу ОВЕН или 0xFFFF – ошибка таймаута.

2.3.6.3 Вход-Выход (VAR_IN_OUT)

- **Buffer:** ARRAY [0..15] of byte – данные для чтения/записи по сети.
- **BufSize:** BYTE – Количество полученных или передаваемых байт.

2.3.7 Функция прослушивания сети



2.3.7.1 Входы

- **Enable:** BOOL – разрешение работы блока – по переднему фронту начинает прослушивание По заднему фронту прекращает
- **Adrlen:** ADR_LEN – длина адреса
- **ComHandle:** DWORD – дескриптор последовательного порта SysLibCom

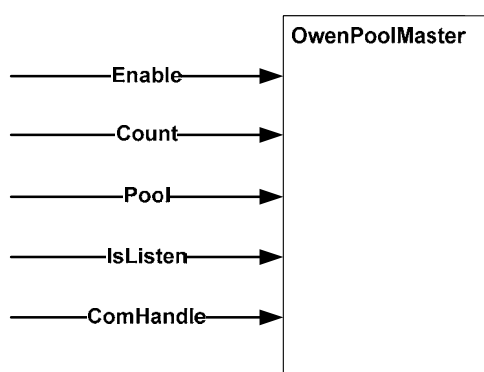
2.3.7.2 Выходы

- **Complete:** BOOL – если =TRUE, то данные прочитаны – по заднему фронту можно определить когда забирать данные из ФБ.
- **Addr:** WORD – адрес устройства в сети
- **Hash:** WORD – Хэш параметра.
- **Error:** WORD – Значение ошибки (исключительной ситуации) согласно протоколу ОВЕН.

2.3.7.3 Вход-Выход (VAR_IN_OUT)

- **Buffer:** ARRAY [0..15] of byte – данные для чтения по сети.
- **BufSize:** BYTE – Количество полученных байт.

2.3.8 Опрос пула команд по протоколу Овен



- **Enable:** BOOL – разрешение работы блока – по переднему фронту начинает обмен.
- **Count:** BYTE – количество заданий – структур типа OwenPool
- **Pool:** POINTER TO OwenPool – указатель на массив структур типа OwenPool
- **IsListen:** BOOL – TRUE означает, что необходимо прослушивать сеть
- **ComHandle:** DWORD – дескриптор последовательного порта библиотеки SysLibCom

TYPE OwenPool:

STRUCT

every: TIME; (* Период опроса*)
IsRead: BOOL; (*true=ЧТЕНИЕ, false=ЗАПИСЬ *)
EncodeType: OWEN_FRM; (* формат данных *)
ADDR: WORD; (*Адрес устройства в сети *)
AddrLen: ADR_LEN; (* Длина адреса, 8 или 11 бит *)
Name: STRING(9); (*Имя параметра- например "read" *)
Index: WORD; (*Индекс, если 0xFFFF, то индекс не используется*)
Complete: BOOL; (* если = TRUE, то данные получены и актуальны *)
TimeOut: TIME; (* время тайм-аута [мс] – макс. задержка на обработку запроса *)
ValReal: REAL; (*Значение параметра с плавающей точкой*)
MeasTime: WORD; (* время измерителя *)
ValDint: DINT; (*Значение целого параметра*)
Error:DWORD; (*значение ошибки согласно протокола OWEN или 0xFFFF- ошибка таймаута *)

END_STRUCT

Приложение А

Пример на языке ST реализует работу по интерфейсу RS-485: Чтение и запись двух параметров с одного прибора TPM201. Для работы примера в PLC Configuration должны быть подключены библиотеки SisLibCom, OwenNet, ComService.

Пример находится на диске в папке \Рекомендации и примеры\OWEN_ПЛК63 запись и чтение REAL из TPM201(ST).pro

```

PROGRAM PLC_PRG
VAR
    get_fl_owen:OWEN_GET_REAL;      (*Чтение параметра типа
Real*)
    send_fl_owen:OWEN_SET_REAL;     (*Запись параметра типа
Real*)
    COM_SERVICE1: COM_SERVICE;      (*открытие COM-порта*)

    cmpl: BOOL;                     (* признак завершения
операции *)
    err: INT;                        (*номер ошибки*)
    port_opened: BYTE := 0;          (*состояние порта*)
    Settings: COMSETTINGS;          (* настройки последова-
тельного порта *)
    com_num: PORTS:=0;               (*0 - RS-485, 1 - RS-232*)
    Timeout: TIME:=T#50ms;          (*таймаут*)
    Enabl: BOOL:=TRUE;              (*состояние работы блока*)
    wTime: WORD;                    (*значение времени для измери-
теля*)
    master1: BYTE := 0;

    A: REAL;                         (*считанное значение из
TPM201*)
    b: REAL:=30;

END_VAR

(*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
    (*чтение параметра типа Real из прибора TRM201 с адресом 8
из параметр с именем PV и индексом 0*)
    0: get_fl_owen(
        Enable:=Enabl ,          (* разрешение работы блока *)
        Addr:=8 ,                (*адрес прибора*)
        AddrLen:=A8BIT ,         (*длина адреса*)
        Name:='pv' ,             (*имя параметра, значение которого
считываем*)
        Index:=65535 ,          (*индекс параметра*)
        RealType:=FLOAT24 ,     (*тип параметра, значение которого
считываем*)
        ComHandle:= Settings.Port, (*номер COM-порта*)
        Timeout:=Timeout ,      (*Таймаут T#50ms*)
        Complete=>smpl ,        (* скопировать признак готовности
результата *)
        Value=>a ,              (*считанное значение*)
        wTime=>wTime ,          (*значение времени для измерителя*)
        Error=> err );          (* скопировать регистр ошибок *)
    (*если установлен признак завершения операции, то *)
    IF smpl THEN
        master1:=1;(*переходим к выполнению следующего блока*)
    END_IF

    1:(*Запись параметра типа Real в прибор TRM201 с адресом 8
в параметр с именем SP и индексом 0*)
    send_fl_owen(
        Enable:=Enabl ,          (* разрешение работы блока *)
        Addr:= 8,                (*адрес прибора*)
        AddrLen:=A8BIT,         (*длина адреса*)
        Name:= 'sp',            (*имя параметра, значение которого
записываем в TRM201*)

```

```
        Index:= 0,                (*индекс параметра*)
        RealType:= FLOAT24, (*тип параметра, значение которого
записываем*)
        Value:=b ,                (*параметр, значение значение
которого записываем в TPM201*)
        ComHandle:=Settings.Port, (*номер COM-порта*)
        Timeout:=Timeout , (* время тайм-аута [мс] - макс. за-
держка на обработку запроса *)
        Complete=>cmpl , (* скопировать признак готовности
результата *)
        Error=> err);            (* скопировать регистр ошибок *)
(*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
```


Приложение Б

Пример на языке ST реализует работу по интерфейсу RS-485: чтение 8 входов из прибора MBA8 с адресом 16. Для работы примера в PLC Configuration должны быть подключены библиотеки SisLibCom, OwenNet, ComService.

Пример находится на диске в папке \Рекомендации и примеры\OWEN_ПЛК63 чтение 8 входов MBA8(ST).pro

```

PROGRAM PLC_PRG
VAR
    get_fl_owen:OWEN_GET_REAL;    (*Чтение параметра типа
Real*)
    COM_SERVICE1: COM_SERVICE;

    cmpl: BOOL;                  (* признак завершения
операции *)
    err: INT;                    (*номер ошибки*)
    port_opened: BYTE := 0;      (*состояние порта*)
    Settings: COMSETTINGS;(* настройки последовательного
порта *)
    com_num: PORTS:=0    (*0 - RS-485, 1 - RS-232*);
    Timeout: TIME:=T#50ms;(*таймаут*)
    Enabl: BOOL:=TRUE;          (*состояние работы блока*)
    wTime: WORD;                (*значение времени для измери-
теля*)

    A: REAL;                    (*считанное значение*)
    adres: INT:=16;              (*адрес MBA8*)
    B: ARRAY [0..7] OF REAL; (*массив для хранения значений
входов MBA8*)
    i: INT;                      (*номер входа MBA**)
END_VAR

(*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*)

(*чтение 8 входов из прибора MBA8 с адресом 16 *)
get_fl_owen(
    Enable:=Enabl ,          (* разрешение работы блока *)
    Addr:=adres ,           (*адрес прибора*)
    AddrLen:=A8BIT ,        (*длина адреса*)
    Name:='read' ,          (*имя параметра, значение которого
считываем*)
    Index:=65535 ,          (*индекс параметра*)
    RealType:=FLOAT32T ,    (*тип параметра, значение которого
считываем*)
    ComHandle:= Settings.Port, (*номер COM-порта*)
    Timeout:=Timeout ,     (*Таймаут T#50ms*)
    Complete=>сmpl ,        (* скопировать признак завершения
операции *)
    Value=>a ,              (*считанное значение*)
    wTime=>wTime ,          (*значение времени для измерителя*)
    Error=> err );          (* скопировать регистр ошибок *)
(*если установлен признак завершения операции, то *)
IF Enabl = FALSE THEN
    Enabl := TRUE;
END_IF
(*Если завершен обмен и нет ошибок*)
IF смpl THEN
    IF (err=0) THEN
        B[i]:=a;            (*присваиваем массиву значение на
входе*)
    ELSE
        Enabl := FALSE;
    END_IF;
    i:=i+1;                (*увеличиваем номер входа*)
    adres:=16+i;           (*увеличиваем адрес*)
    IF (i=8) THEN          (*если номер входа меньше 8*)
        adres:=16;

```

```
        i:=0;  
    END_IF  
  
END_IF  
  
END_IF
```