ЦИФРОВЫЕ РЕШЕНИЯ

Библиотека SysLibSocket

Руководство пользователя

11.2026
версия 1.4

Содержание

1 Цель документа 3
2 Установка библиотеки 4
3 Описание библиотеки SysLibSocket 5
3.1 SysSockCreate 5
3.2 SysSockBind 6
3.3 SysSockListen 7
3.4 SysSockAccept 7
3.5 SysSockRecv и SysSockSend. Протокол TCP 8
3.6 SysSockRecvFrom и SysSockSendTo. Протокол UDP 9
3.7 SysSockShutdown 11
3.8 SysSockClose 11
3.9 SysSockSetOption 11
3.10 SysSockGetOption 13
3.11 SysSockHtonl 13
3.12 SysSockHtons 14
3.13 SysSockNtohl 14
3.14 SysSockNtohs 14
3.15 SysSockConnect 15
3.16 SysSockIoctl 15

1 Цель документа

Данный документ описывает программную реализацию передачи данных с помощью сетевых протоколов UDP и TCP для контроллеров ОВЕН, программируемых в среде ALTA IDE, с помощью библиотеки SysLibSocket.

2 Установка библиотеки

Чтобы подключить библиотеку к проекту:

1. Откройте редактор **Менеджер библиотек** одним из способов:

- дважды нажмите ЛКМ на системную папку **Менеджер библиотек** в дереве проекта;
- или нажмите ПКМ на системную папку **Менеджер библиотек** в дереве проекта и выберите в контекстном меню пункт **Открыть**:

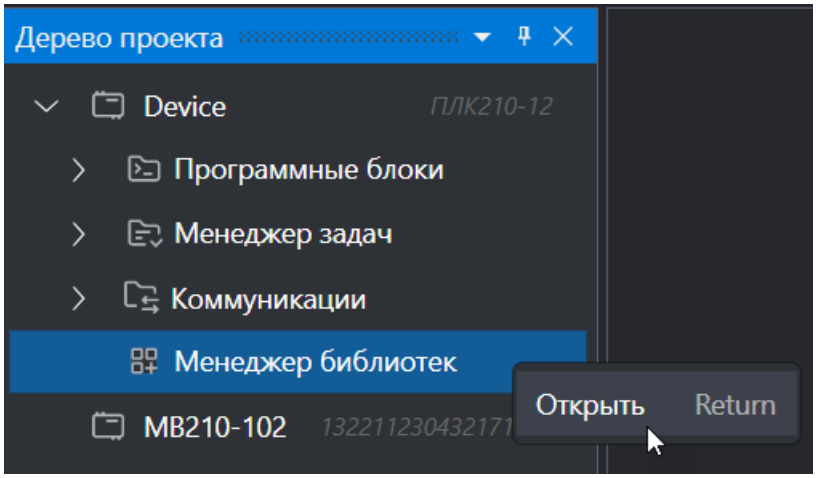


Рисунок 2.1

- В редакторе **Менеджер библиотек** откройте вкладку **Магазин**.
- Выберите библиотеку **SysLibSocket** и нажмите кнопку **Подключить** на карточке библиотеки.
- Библиотека добавится в проект и отобразится на вкладке **Мои библиотеки** в редакторе **Менеджер библиотек**.

3 Описание библиотеки SysLibSocket

Библиотека SysLibSocket поддерживает работу с сокетами по TCP/IP и UDP.

Сокеты могут работать в блокирующем и неблокирующем режиме. В блокирующем режиме все операции, производимые с сокетом, являются синхронными, а в неблокирующем – асинхронными. Например, в блокирующем режиме функция чтения завершает свою работу только после получения данных, что, соответственно, делает время цикла ПЛК непрогнозируемым. В неблокирующем режиме вызов любой функции не останавливает цикл ПЛК.

Библиотека SysLibSocket содержит функции:

- SysSockCreate;
- SysSockBind;
- SysSockListen;
- SysSockAccept;
- SysSockRecv и SysSockSend. Протокол TCP;
- SysSockRecvFrom и SysSockSendTo. Протокол UDP;
- SysSockShutdown;
- SysSockClose;
- SysSockSetOption;
- SysSockGetOption;
- SysSockHtonl;
- SysSockHtons;
- SysSockNtohl;
- SysSockNtohs;
- SysSockConnect.

Часть функций библиотеки ссылаются на структуры:

Таблица 3.1 – Описание структур библиотеки SysLibSocket

Структура	Описание
SOCKADDRESS	Структура для хранения адреса. См. описание в таблице ниже
SOCKET_FD_SET	Структура используется в функции SysSockSelect, которая не поддержана
SOCKET_TIMEVAL	Структура используется в функции SysSockSelect, которая не поддержана

Описание структуры SOCKADDRESS

Таблица 3.2 – Описание структуры SOCKADDRESS

Переменная	Тип	Описание
sin_family	INT	Семейство протоколов. Входные аргументы: • SOCKET_AF_INET (для семейства IPv4)
sin_port	WORD	Порт
sin_addr	DWORD	IP-адрес, к которому будет привязан сокет. Входные аргументы: • SOCKET_INADDR_ANY (все адреса локального хоста) – TCP и UDP сервер (SysSockBind); • 32-х битное значение IP-адреса – TCP клиент (SysSockConnect) и UDP клиент (SysSockRecvFrom/ SysSockSendTo).
sin_zero	ARRAY [0..7] OF SINT	Дополнение до размера структуры SOCKADDRESS. Поле должно содержать массив нулей и служит только для увеличения размера структуры до стандартных 16 байт. Переменную не обязательно использовать в проекте

Далее в документе описаны все функции библиотеки.

3.1 SysSockCreate

Функция типа данных DINT создает новый сокет и возвращает для него идентификатор (handle) сокета.

Имя переменной	Тип данных	Описание
Входные переменные		
diAddressFamily	DINT	Семейство протоколов создаваемого сокета. Используется значение SOCKET_AF_INET для IPv4.
diType	DINT	Тип создаваемого сокета. Для TCP используется SOCKET_STREAM (потоковый сокет), для UDP — SOCKET_DGRAM (датаграммный сокет).
diProtocol	DINT	Протокол сокета. Для TCP используется SOCKET_IPPROTO_TCP, для UDP — SOCKET_IPPROTO_UDP.

Пример

```
VAR
    diSocket: DINT; (* дескриптор сокета. Объявление переменной для записи
идентификатора сокета *)
END_VAR

// Код программы
diSocket := SysLibSocket.SysSockCreate(SysLibSocket.SOCKET_AF_INET,
SysLibSocket.SOCKET_STREAM, SysLibSocket.SOCKET_IPPROTO_TCP);
// Комментарии

END_PROGRAM
```

Семейство IPv4, потоковый сокет, протокол TCP.

При попытке создать сокет в системе, где достигнуто максимальное число сокетов функция SysSockCreate вернёт -1 (SOCKET_INVALID)

3.2 SysSockBind

Функция SysSockBind типа данных BOOL используется только в случае реализации на ПЛК сервера и привязывает сокет к IP адресу и порту. Для привязки функция SysSockBind ссылается на структуру SOCKADDRESS, в которой хранится заданный адрес и порт.

Имя переменной	Тип данных	Описание
Входные переменные		
diSocket	DINT	Дескриптор сокета, полученный от функции SysSockCreate
pSockAddr	REF_TO_ SOCKADDRESS	Указатель на переменную типа SOCKADDRESS
diSockAddrSize	DWORD	Размер структуры SOCKADDRESS в байтах

Пример для TCP/UDP сервера

```
// Объявление
VAR
    diSocket:         DINT;         (* Дескриптор сокета *)
    stServerSettings: SysLibSocket.SOCKADDRESS;  (* Структура для хранения адреса сокета *)
    xBinded:          BOOL;         (* Результат функции SysSockBind *)
    wPort:            WORD;         (* Порт сокета *)
END_VAR

// Код программы
stServerSettings.sin_family := SysLibSocket.SOCKET_AF_INET;
stServerSettings.sin_addr := SysLibSocket. SysSockHtonl(SysLibSocket.SOCKET_INADDR_ANY
stServerSettings.sin_port := SysLibSocket.SysSockHtons (wPort);

xBinded := SysLibSocket.SysSockBind(diSocket,ADR(stServerSettings),
SIZEOF(stServerSettings));
```

Функция SysSockBind после выполнения возвращает TRUE. Для корректного заполнения структуры SOCKADDRESS следует использовать функции SysSockHtonl и SysSockHtons.

3.3 SysSockListen

Функция SysSockListen типа данных BOOL задаёт максимальное количество входящих соединений.

Функция используется при настройке обмена по протоколу TCP.

Имя переменной	Тип данных	Описание
Входные переменные		
diSocket	DINT	Дескриптор сокета
diMaxConnections	DINT	Максимальное количество входящих соединений

Пример

```
VAR
    diSocket:         DINT;       // Дескриптор сокета
    diMaxConnections: DINT := 5;  // Максимальное количество входящих соединений
    xListened:        BOOL;       // Результат функции SysSockListen
END_VAR

xListened := SysLibSocket.SysSockListen(diSocket, diMaxConnections);
```

Функция SysSockListen после выполнения возвращает TRUE.

3.4 SysSockAccept

Функция SysSockAccept типа данных DINT при подключении клиента создает дескриптор клиентского сокета для установленного соединения.

Функция используется при настройке обмена по протоколу TCP.

Имя переменной	Тип данных	Описание
Входные переменные		
diSocket	DINT	Дескриптор серверного сокета
pSockAddr	REF_TO SOCKADDRESS	Указатель на переменную типа SOCKADDRESS
piSocketAddrSize	REF_TO DWORD	Указатель на размер структуры SOCKADDRESS

Пример

```
VAR
    diSocket:       DINT;       // Дескриптор серверного сокета
    diClientSocket: DINT;       // Дескриптор клиентского сокета
    stSettings:     SysLibSocket.SOCKADDRESS;
    // Структура для хранения адреса сокета
END_VAR

diClientSocket := SysLibSocket.SysSockAccept(diSocket, ADR(stSettings),
SIZEOF(stSettings));
```

3.5 SysSockRecv и SysSockSend. Протокол TCP

Функция **SysSockRecv** типа данных DINT выполняет прием данных и возвращает число считанных байт.

Максимальный буфер приема – 1500 байт

Имя переменной	**Тип данных**	**Описание**
Входные переменные		
diSocket	DINT	Дескриптор клиентского сокета
pbyBuffer	REF_TO DWORD	Указатель на буфер принимаемых данных
diBufferSize	DINT	Размер буфера в байтах
diFlags	DINT	Флаг определяет параметры приема данных. Входные аргументы: 0 – не поддерживаются

Пример

```
VAR
    diClientSocket: DINT;                   // дескриптор клиентского сокета
    abyRead:        ARRAY [1..10] OF BYTE; // переменная для приема сообщения
    diRecvBytes:    DINT;                   // количество считанных байт
    c_diFlags:      DINT := 0;              // константа, флаг
END_VAR

diRecvBytes := SysLibSocket.SysSockRecv(diClientSocket,
               REF(abyRead),
               ULINT_TO_DINT(SIZEOF(abyRead)), c_diFlags);

END_PROGRAM
```

Для приема можно использовать любые типы данных, например переменные типа STRING.

Функция **SysSockSend** типа данных DINT выполняет передачу данных и возвращает число переданных байт.

Максимальный буфер передачи – 1500 байт.

Имя переменной	**Тип данных**	**Описание**
Входные переменные		
diSocket	DINT	Дескриптор клиентского сокета
pbyBuffer	REF_TO DWORD	Указатель на буфер для передачи данных
diBufferSize	DINT	Размер буфера
diFlags	DINT	Флаг определяет параметры передачи данных. Входные аргументы: 0 – не поддерживаются

```
VAR    diClientSocket: DINT;                    // дескриптор клиентского сокета
    abySend:        ARRAY [1..10] OF BYTE; // переменная для передачи сообщения
    diSendBytes:    DINT;                  // количество переданных байт
    c_diFlags:      DINT := 0;             // константа, флаг
END_VAR

diSendBytes := SysLibSocket.SysSockSend(diClientSocket,
               REF(abySend), ULINT_TO_DINT(SIZEOF(abySend)), c_diFlags);

END_PROGRAM
```

3.6 SysSockRecvFrom и SysSockSendTo. Протокол UDP

Функция **SysSockRecvFrom** типа данных DINT выполняет прием данных и возвращает число считанных байт.

Максимальный буфер приема – 1500 байт

Имя переменной	**Тип данных**	**Описание**
Входные переменные		
diSocket	DINT	Дескриптор клиентского сокета
pbyBuffer	REF_TO DWORD	Указатель на буфер принимаемых данных
diBufferSize	DINT	Размер буфера
diFlags	DINT	Флаг определяет параметры приема данных. Входные аргументы: 0 – не поддерживаются.
pSockAddr	REF_TO SOCKADDRESS	Указатель на переменную типа SOCKADDRESS
diSockAddrSize	DINT	Размер структуры SOCKADDRESS

Пример

```
VAR
    stClientSettings: SOCKADDRESS;             (*Структура для хранения адреса
сокета*)
    dwIPaddr:         DWORD := 16#0A00060A;    (*IP-адрес сервера: 10.0.6.10*)
    wPort:            WORD;                    (*Порт сокета*)
    diClientSocket:   DINT;                    (*Дескриптор клиентского сокета*)
    abyRead: ARRAY [1..c_iBufferSize] OF BYTE; (*Переменная для приема
сообщения*)
    diRecvBytes:      DINT;                    (*Количество считанных байт*)
    c_diFlags:        DINT := 0;               (*Константа, флаг*)
END_VAR

stClientSettings.sin_family := SysLibSocket.SOCKET_AF_INET;
stClientSettings.sin_addr := SysLibSocket.SysSockHtonl(dwIPaddr);
stClientSettings.sin_port := SysLibSocket.SysSockHtons(wPort);

diRecvBytes := SysLibSocket.SysSockRecvFrom(diClientSocket,
REF(abyRead), ULINT_TO_DINT (
SIZEOF(abyRead)), c_diFlags,
REF(stClientSettings), ULINT_TO_DINT (
SIZEOF(stClientSettings)));

END_PROGRAM
```

Для приема можно использовать любые типы данных, например переменные типа STRING.

Функция **SysSockSendTo** типа данных DINT выполняет передачу данных и возвращает число переданных байт.

Максимальный буфер передачи – 1500 байт.

Имя переменной	Тип данных	Описание
Входные переменные		
diSocket	DINT	Дескриптор клиентского сокета
pbyBuffer	REF_TO DWORD	Указатель на буфер для передачи данных
diBufferSize	DINT	Размер буфера
diFlags	DINT	Флаг определяет параметры передачи данных. Входные аргументы: 0 – не поддерживаются.
pSockAddr	REF_TO SOCKADDRESS	Указатель на переменную типа SOCKADDRESS
diSockAddrSize	DINT	Размер структуры SOCKADDRESS

```
VAR
    stClientSettings: SysLibSocket.SOCKADDRESS; (*структура для хранения
                                                адреса сокета*)
    dwIPaddr:         DWORD := 16#0A00060A;     // IP-адрес сервера: 10.0.6.10
    wPort:            WORD;                     // порт сокета
    diClientSocket:   DINT;                     // дескриптор клиентского сокета
    abySend:          ARRAY [1..10] OF BYTE;    (* переменная для передачи
                                                сообщения*)
    diSendBytes:      DINT;                     // количество переданных байт
    c_diFlags:        DINT := 0;                // константа, флаг

END_VAR

stClientSettings.sin_family := SysLibSocket.SOCKET_AF_INET;
stClientSettings.sin_addr   := SysLibSocket.SysSockHtonl(dwIPaddr);
stClientSettings.sin_port   := SysLibSocket.SysSockHtons(wPort);

diSendBytes := SysLibSocket.SysSockSendTo(diClientSocket,
REF(abySend), ULINT_TO_DINT(SIZEOF(abySend)), c_diFlags,
REF(stClientSettings), ULINT_TO_DINT(SIZEOF(stClientSettings)));

END_PROGRAM
```

Для передачи можно использовать любые типы данных, например переменные типа STRING.

3.7 SysSockShutdown

Функция **SysSockShutdown** типа данных BOOL выполняет запрет передачи и приема данных.

Имя переменной	Тип данных	Описание
Входные переменные		
diSocket	DINT	Дескриптор сокета
diHow	DINT	Аргумент определяет тип запрещаемых действий: • 0 – отключение приема сообщений; • 1 – отключение отправки сообщения; • 2 – отключение приема и отправки сообщения

Пример

```
VAR
    diSocket:    DINT;       // Дескриптор сокета
    xShutdowned: BOOL;       // Результат функции SysSockShutdown
    c_diHow:     DINT := 2; // Константа, определяет тип запрещаемых действий
END_VAR

xShutdowned:= SysLibSocket.SysSockShutdown(diSocket, c_diHow); (*После
выполнения возвращает TRUE*)

END_PROGRAM
```

3.8 SysSockClose

Функция **SysSockClose** типа данных BOOL закрывает сокет.

Имя переменной	Тип данных	Описание
Входные переменные		
diSocket	DINT	Дескриптор сокета

Пример

```
VAR
    diSocket:    DINT;       // Дескриптор сокета
    xSockClosed: BOOL;       // Результат функции SysSockClose
END_VAR

xSockClosed:= SysLibSocket.SysSockClose(diSocket);
(*После выполнения возвращает TRUE*)
```

3.9 SysSockSetOption

Функция **SysSockSetOption** типа данных BOOL устанавливает параметры, связанные с сокетом.

Имя переменной	Тип данных	Описание
Входные переменные		
diSocket	DINT	Дескриптор сокета
diLevel	DINT	Уровень, на котором находится опция. Поддержан уровень для работы с сокетами: SOCKET_SOL

Имя переменной	Тип данных	Описание
diOption	DINT	Опция, которой нужно передать значение. Входные аргументы: • SO_NBIO=0x1014 – установить сокет в неблокирующий режим; • SO_DEBUG=0x00001 – включить запись отладочной информации; • SO_REUSEADDR=0x00004 – разрешить повторное использование локального адреса; • SO_KEEPALIVE=0x00008 – поддерживать соединение; • SO_DONTROUTE=0x00010 – использовать адреса интерфейса; • SO_LINGER=0x00080 – задержка закрытия при наличии данных; • SO_WINSCALE=0x00400 – установить параметр окна масштабирования; • SO_TIMESTAMP=0x00800 – установить опцию отметки времени TCP; • SO_BIGCWND=0x01000 – большое начальное окно перегрузки TCP; • SO_NOSLOWSTART=0x04000 – подавление медленного запуска в этом сокете
diOptionValue	REF_TO STRING	Указатель на переменную, в которую после выполнения функции будет записано значение для запрашиваемой опции
diOptionLength	REF_TO size_t*	Размер в байтах переменной

***size_t** — это беззнаковый целочисленный тип данных в C и C++ (не поддержан в редакторе ALTA IDE). Его основная задача - представлять размер любого объекта в памяти в байтах, включая массивы. Для корректной работы diOptionLength необходимо присвоить результат SIZEOF, как в примере.

Пример

```
VAR
    diSocket:       DINT;           (*Дескриптор сокета*)
    c_diOption:     DINT := 16#1014; (*Константа, опция SO_NBIO переводит
в неблокирующий режим*)
    diOptionValue:  DINT;          (*Значение для запрашиваемой опции*)
    xSetOption:     BOOL;
END_VAR

xSetOption:= SysLibSocket.SysSockSetOption(diSocket, SysLibSocket.SOL, c_diOption,
ADR(diOptionValue), SIZEOF(diOptionValue));
(*После выполнения возвращает TRUE. Детальное описание аргументов дано в
справочных системах соответствующих ОС*)
```

3.10 SysSockGetOption

Функция **SysSockGetOption** типа данных DINT считывает значение параметра, связанного с сокетом.

Имя переменной	Тип данных	Описание
Входные переменные		
diSocket	DINT	Дескриптор сокета
diLevel	DINT	Уровень, на котором находится опция. Входные аргументы: SOCKET_SOL
diOption	DINT	Опция, которой нужно передать значение. Входные аргументы: • SO_ERROR=0x1007 – получить статус ошибки; • SO_RXDATA=0x1011 – получить количество байт recv; • SO_TXDATA=0x1012 – получить количество байт send
diOptionValue	REF_TO VOID*	Указатель на переменную, в которую после выполнения функции будет записано значение для запрашиваемой опции
diOptionLength	REF_TO size_t**	Размер в байтах переменной

***REF_TO VOID** — это указатель на неопределенный тип данных (не поддержан в редакторе ALTA IDE). Для корректной работы diOptionValue необходимо присвоить указатель, как в примере.

****size_t** — это беззнаковый целочисленный тип данных в C и C++ (не поддержан в редакторе ALTA IDE). Его основная задача - представлять размер любого объекта в памяти в байтах, включая массивы. Для корректной работы diOptionLength необходимо присвоить результат SIZEOF, как в примере.

Пример

```
VAR
    diSocket:    DINT;          (*дескриптор сокета*)
    c_diSoError: DINT:=16#1007; (*константа, опция SO_ERROR позволяет
                                 получить статус ошибки*)
    diOption:    DINT;          (*после выполнения функции будет записано
                                 значение для запрашиваемой опции *)
    xGetOption:  DINT;
END_VAR

// Программа
xGetOption := SysLibSocket.SysSockGetOption(diSocket,
              SysLibSocket.SOCKET_SOL , c_diSoError,
              ADR(diOption), SIZEOF(diOption));

END_PROGRAM
```

3.11 SysSockHtonl

Функция **SysSockHtonl**(Host to Network Long) типа данных DWORD конвертирует переменную типа DWORD в соответствии с порядком байт в сетях TCP/IP.

Имя переменной	Тип данных	Описание
Входные переменные		
dwHost	DWORD	Значение для конвертирования

Пример

```
VAR
    dwIPaddr:   DWORD := 16#0A021478;   (* IP-адрес формата DWORD,
                                        например: 16#0A021478*)
    stSettings: SysLibSocket.SOCKADDRESS; // Структура для хранения адреса сокета
END_VAR

stSettings.sin_addr:= SysLibSocket.SysSockHtonl(dwIPaddr);
(*После преобразования в переменную stSettings.sin_addr запишется 16#7814020A*)
```

3.12 SysSockHtons

Функция **SysSockHtons**(Host to Network Short) типа данных WORD конвертирует переменную типа WORD в соответствии с порядком байт в сетях TCP/IP.

Имя переменной	Тип данных	Описание
Входные переменные		
wHost	WORD	Значение для конвертирования

Пример

```
VAR
    wPort:      WORD := 16#01F6;              // Порт сокета, например, 502
    stSettings: SysLibSocket.SOCKADDRESS;     // Структура для хранения адреса сокета
END_VAR

stSettings.sin_port:= SysLibSocket.SysSockHtons(wPort);
(*После преобразования в переменную stSettings.sin_port запишется 16#F601*)
```

3.13 SysSockNtohl

Функция **SysSockNtohl** (Network to Host Long) типа данных DWORD конвертирует из порядка байт в сетях TCP/IP в переменную типа DWORD.

Имя переменной	Тип данных	Описание
Входные переменные		
dwNet	DWORD	Значение для конвертирования

Пример

```
VAR
    dwIPaddr:   DWORD;                    // IP-адрес формата DWORD
    stSettings: SysLibSocket.SOCKADDRESS; // Структура для хранения адреса сокета
END_VAR

dwIPaddr := SysLibSocket.SysSockNtohl(stSettings.sin_addr ); (*Допустим, переменная
равна 16#7814020A*)
(*После преобразования в переменную dwIPaddr запишется 16#0A021478
(10.2.20.120)*)
```

3.14 SysSockNtohs

Функция **SysSockNtohs** (Network to Host Short) типа данных WORD конвертирует из порядка байт в сетях TCP/IP в переменную типа WORD.

Имя переменной	Тип данных	Описание
Входные переменные		
wNet	WORD	Значение для конвертирования

Пример

```
VAR
    wPort:      WORD;                       // Порт сокета
    stSettings: SysLibSocket.SOCKADDRESS;  // Структура для хранения адреса сокета
END_VAR

wPort := SysLibSocket.SysSockNtohs(stSettings.sin_port ); (*Допустим, переменная
равна 16#F601*)
(*После преобразования в переменную wPort запишется 16#01F6 (502)*)
```

3.15 SysSockConnect

Функция **SysSockConnect** типа данных BOOL выполняет подключение к серверному сокету по IP-адресу и порту.

Имя переменной	Тип данных	Описание
Входные переменные		
diSocket	DINT	Дескриптор клиентского сокета
pSockAddr	REF_TO SOCKADDRESS	Указатель на переменную типа SOCKADDRESS
diSockAddrSize	DINT	Размер структуры SOCKADDRESS

Пример

```
VAR
    stClientSettings: SysLibSocket.SOCKADDRESS; (*Структура для хранения адреса
сокета*)
    dwIPaddr:         DWORD := 16#0A00060A; (*IP-адрес сервера: 10.0.6.10*)
    wPort:            WORD;                 (*Порт сокета*)
    diSocket:         DINT;                 (*Дескриптор сокета*)
    xConnected:       BOOL;                 (*Результат функции SysSockConnect*)
END_VAR

stClientSettings.sin_family := SysLibSocket.SOCKET_AF_INET;
stClientSettings.sin_addr := SysLibSocket.SysSockHtonl(dwIPaddr);
stClientSettings.sin_port := SysLibSocket.SysSockHtons(wPort);

xConnected := SysLibSocket.SysSockConnect(diSocket,
REF(stClientSettings),
ULINT_TO_DINT(SIZEOF(stClientSettings)));
(*В неблокирующем режиме факт установки соединения можно определить только
косвенным путем, используя функции SysSockSend и SysSockRecv.
После выполнения функция SysSockConnect всегда возвращает FALSE*)

END_PROGRAM
```

3.16 SysSockIoctl

Функция **SysSockIoctl** типа данных DINT поддерживает только команду SOCKET_FIONREAD для контроллеров ПЛК110 [M03]. Эта команда позволяет собирать входящие сообщения в один буфер. Затем можно вывести весь собранный буфер через функцию приема SysSockRecv/SysSockRecvFrom.

Имя переменной	Тип данных	Описание
Входные переменные		
diSocket	DINT	Дескриптор сокета
diCommand	DINT	Команда, которую нужно применить к сокету. Допустимые команды: • SOCKET_FIONREAD (собирает входящие сообщения в один буфер)
piParameter	REF_TO DWORD	Указатель на параметр, который указывает объем данных в буфере в байтах

Пример

```
VAR
    hSocket:     DINT; //дескриптор сокета
    diParameter: DINT; //переменная содержит размер входящего сообщения в байтах
    diReturn:    DINT; //при поступлении данных возвращает 1
END_VAR

diReturn := SysSockIoctl(hSocket, SOCKET_FIONREAD,  REF(diParameter));
(*Получить объем входных данных можно до функции приема сообщения
SysSockRecv/SysSockRecvFrom. После вызова функции SysSockRecv/SysSockRecvFrom
переменная diParameter, в которой хранится количество накопленных байт,
очищается и в буфер принимаемых данных записывается буфер накопившихся данных*)
```

ЦИФРОВЫЕ
РЕШЕНИЯ

ООО "Овен Цифровые решения"
Россия, г. Москва, пл. Семёновская, д. 1А, помещ. 3/1
support@owendigital.ru
www.owendigital.ru

рег.:1-RU-159225-1.4