

The Library SysLibSockets.lib

This library supports the access on sockets for the communication via TCP_IP and UDP.

If the target system supports the functionality then the following functions are available, each calling the corresponding function of the operating system.

The execution is synchronous.

See the help system of the operating system for detailed information on the operating system functions.

- | | |
|------------------------|----------------------------------|
| • SysSockAccept | • SysSockIoctl |
| • SysSockBindSys | • SysSockListen |
| • SysSockClose | • SysSockNtohl |
| • SysSockConnect | • SysSockNtohs |
| • SysSockCreate | • SysSockSelect |
| • SysSockGetHostByName | • SysSockSetIPAddress |
| • SysSockGetHostName | • SysSockSetOption |
| • SysSockGetOption | • SysSockShutdown |
| • SysSockHtonl | TCP specific: |
| • SysSockHtons | • SysSockRecv, SysSockSend |
| • SysSockInetAddr | UDP specific: |
| • SysSockInetNtoa | • SysSockRecvFrom, SysSockSendTo |

SysSockAccept

This function of type DINT calls the function **accept** of the operating system, which can accept a connection request to the socket. A new descriptor (handle) for the socket is returned. The original socket will be reset to the "listening" status (see SysSockListen).

For a description of the **accept** function please see the help system of the operating system.

Variable	Data type	Description
diSocket	DINT	A descriptor identifying a socket that has been placed in a listening state with the listen function. The connection will actually be made with the socket that is returned by the SysSockListen function. The requested connection then will be made with that socket, for which the SysSockAccept function returned a handle. (corresponding parameter, e.g. in Win32: s)
pSockAddr	DWORD	Pointer on a variable of type SOCKADDR; will be filled with the address of the caller. (corresponding parameter, e.g. in Win32: addr)
piSocketAddrSize	DWORD	Pointer on a variable of type DINT. This variable has got assigned the length of the structure SockAddr (can be retrieved with the aid of the SIZEOF operator) (corresponding parameter, e.g. in Win32: addrlen)

Structure SOCKADDR:

```

sin_family :    INT;                (* Address-family, defines address format *)
sin_port :     UINT;               (* Port of the connection requesting unit *)
sin_addr :     UDINT;              (* IP-address of the requesting unit *)
sin_zero :     ARRAY [0..7] OF SINT; (* buffer *)

```

SysSockBind

This function of type BOOL calls the function **bind** of the operating system. This function will allocate a local address to the socket which was assigned before just to an address range by SysSockCreate. Usually the "binding" will be done before functions like SysSockListen or SysSockAccept are called for a socket.

In case of successful operation the function will return TRUE, otherwise FALSE.

For a description of the **bind** function please see the Online Help resp. documentation for the operating system.

Variable	Data type	Description
diSocket	DINT	Descriptor of the socket, returned by SysSockCreate (corresponding parameter e.g. in Win32: s)
pSockAddr	DWORD	Pointer on a variable of type SOCKADDR; (see SysSockAccept)
diSockAddrSize	DINT	Length of the structure SOCKADDR (can be retrieved with the aid of the SIZEOF operator)

SysSockClose

This function of type BOOL calls the function **closesocket** of the operating system, in order to close a socket.

In case of successful operation the function will return TRUE, otherwise FALSE.

Variable	Data type	Description
diSocket	DINT	Descriptor of the socket, returned by SysSockCreate (corresponding parameter e.g. in Win32: s)

SysSockConnect

This function of type BOOL calls the function connect of the operating system. In case the socket has not yet been "bound" by the SysSockBind function, now automatically a local address will be assigned to it. Afterwards the socket will be ready to send and /or receive data.

In case of successful operation the function will return TRUE, otherwise FALSE.

Variable	Data type	Description
diSocket	DINT	Descriptor of the socket, returned by SysSockCreate (corresponding parameter e.g. in Win32: s)
pSockAddr	DWORD	Pointer auf eine Variable vom Typ SOCKADDR; (siehe SysSockAccept)
diSockAddrSize	DINT	Length of the structure SOCKADDR (can be retrieved with the aid of the SIZEOF operator)

For a description of the **connect** Function please see the Online Help resp. documentation on the operating system.

SysSockCreate

This function of type DINT calls the function **socket** of the operating system. A new socket will be created and assigned to a Service Provider.

The function returns the descriptor of the new socket, which is used as input parameter in other functions of the library, e.g. SysSockBind, SysSockConnect.

Variable	Data type	Description
diAddressFamily	DINT	Address family (corresponding parameter e.g. in Win32: af)
diType	DINT	One of the following two types can be used e.g. for Windows Sockets 1.1: SOCK_STREAM, SOCK_DGRAM (corresponding parameter e.g. in Win32: type)
diProtocol	DINT	Protocol, depending on the chosen address family (corresponding parameter e.g. in Win32: protocol)

For a description of the **socket** Function please see the Online Help resp. documentation on the operating system.

SysSockGetHostByName

This function of type DWORD calls function **hostGetByName (VxWorks)** bzw. **gethostbyname (win32)** of the operating system.

In case of successful operation the function will return the host address, otherwise SOCKET_INADDR_NONE (defined in the library as a global constant).

Variable	Data type	Description
stHostName	POINTER TO STRING	Name of the host (corresponding parameter e.g. in Win32: name)

For a description of the **hostGetByName**- resp. **gethostbyname** function please see the Online Help resp. documentation on the operating system.

SysSockGetHostName

This function of type BOOL calls the function **gethostname** of the operating system auf and returns the host name.

In case of successful operation the function will return TRUE, otherwise FALSE.

Variable	Data type	Description
stHostName	STRING	Host name (corresponding parameter e.g. in Win32: name)
diNameLength	DINT	Length of the host name (corresponding parameter e.g. in Win32: buflen)

For a description of the **gethostname** function please see the Online Help resp. documentation on the operating system.

SysSockGetOption

This function of type BOOL calls the function **getsockopt** of the operating system, in order to get the value of a particular socket option. In case of successful operation the function will return TRUE, otherwise FALSE.

Variable	Data type	Description
diSocket	DINT	Descriptor of the socket, returned by SysSockCreate (corresponding parameter e.g. in Win32: s)
diLevel	DINT	protocol specific level; possible values: SOL_SOCKET, IPPROTO_TCP (corresponding parameter e.g. in Win32: level)
diOption	DINT	Name of the option, for which you want to get the current value; see function SysSockSetOption for a list of the options (corresponding parameter e.g. in Win32: optname)
diOptionValue	DWORD	Pointer to the variable, to which the current value of the option should be written (corresponding parameter e.g. in Win32: optval)
diOptionLength	DWORD	Pointer to the size of the variable, to which the current value of the option should be written (corresponding parameter e.g. in Win32: optlen)

For a description of the **getsockopt** Function please see the Online Help resp. documentation on the operating system.

SysSockHtonl

This function of type DWORD calls the function **htonl** of the operating system, which will convert a u_long value from host byte order to TCP/IP network order.

The function returns the converted value.

For a description of this function please see the Online Help resp. documentation on the operating system.

Variable	Data type	Description
dwHost	DWORD	Value to be converted.

SysSockHtons

This function of type WORD calls the function **htons** of the operating system, which converts a short value from host byte order to TCP/IP network order.

The function returns the converted value.

For a description of this function please see the Online Help resp. documentation on the operating system.

Variable	Data type	Description
wHost	WORD	Value to be converted.

SysSockInetAddr

This function of type DWORD calls the function **inet_addr** of the operating system, which converts a string, containing an internet address, in an Yaddress which can be used in the IN_ADDR structure.

The function returns the converted address.

For a description of this function please see the Online Help resp. documentation on the operating system.

Variable	Data type	Description
stIPAddr	STRING	IP address (dotted notation) (corresponding parameter e.g. in Win32: cp)

SysSockInetNtoa

This function of type BOOL calls the function **inet_ntoa (Win32) bzw. inet_ntoa_b (VxWorks)**, which converts an Internet network address in a string in Internet standard format.

In case of successful operation the function will return TRUE, otherwise FALSE.

Variable	Data type	Description
pInAddr	INADDR	Pointer to structure INADDR, which contains the Internet address, see below (corresponding parameter e.g. in Win32: in)
stIPAddr	STRING	IP address
dilIPAddrSize	DINT	Size of the IP address

Structure INADDR:

S_addr : DWORD; (* Internet address as DWORD *)

For a description of the **inet_ntoa** Function please see the Online Help resp. documentation on the operating system.

SysSockIoctl

This function of type DINT calls the function `ioctl` of the operating system in order to control the I/O mode of the socket.

In case of successful operation the function will return TRUE, otherwise FALSE.

Variable	Data type	Description
diSocket	DINT	Descriptor of the socket, returned by SysSockCreate (corresponding parameter e.g. in Win32: s)
diCommand	DINT	Command which you want to apply on the socket. (corresponding parameter e.g. in Win32: cmd). Valid commands are: SOCKET_FIONBIO, SOCKET_FIONREAD.
piParameter	DWORD	Pointer to the command parameter (corresponding parameter e.g. in Win32: argp)

For a description of the `ioctl`-Function please see the Online Help resp. documentation on the operating system.

SysSockListen

This function of type BOOL calls the function **listen** of the operating system. This function will cause the socket to listen to connection requests and to queue them until they can be accepted by the SysSocketAccept function.

In case of successful operation the function will return TRUE.

As soon as the maximum number of connection requests in the queue is exceeded the function will return FALSE.

Variable	Data type	Description
diSocket	DINT	Descriptor of the socket, returned by SysSockCreate (corresponding parameter e.g. in Win32: s)
diMaxConnections	DINT	Maximum number of connection requests, which can be put in the input queue of the socket. (corresponding parameter e.g. in Win32: backlog)

For a description of the **listen** function please see the Online Help resp. documentation on the operating system.

SysSockNtohl

This function of type DWORD calls the function **ntohl** of the operating system, which converts a `u_long` value of the TCP/IP network order to the host byte order.

The function will return the value in host byte order.

Variable	Data type	Description
dwNet	DWORD	<code>u_long</code> value to be converted (corresponding parameter e.g. in Win32: netlong)

For a description of the function **ntohl** please see the Online Help resp. documentation on the operating system.

SysSockNtohs

This function of type WORD calls the function `ntohs` of the operating system, which converts a `u_short` value from the TCP/IP network order to the host byte order.
The function will return the value in host byte order.

Variable	Data type	Description
wNet	WORD	<code>u_short</code> value to be converted (corresponding parameter e.g. in Win32: <code>netshort</code>)

For a description of the **ntohs** Function please see the Online Help resp. documentation on the operating system.

SysSockSelect

This function of type DINT calls the function **select** of the operating system to check whether one or several sockets are ready for certain communication actions. The group of sockets, to which this request should be applied, can be defined via the structure `SOCKET_FD_SET`.

For a description of the `select` function please see the Online Help resp. documentation on the operating system.

The function will return the result of the `select` function.

Variable	Data type	Description
diWidth	DINT	Size of structure <code>SOCKET_FD_SET</code> .
fdRead	DWORD	Optionally a pointer to the structure defining the socket set for which the status of the read actions should be checked. You also can pass 0. Structure <code>SOCKET_FD_SET</code> see below (corresponding parameter e.g. in Win32: <code>readfds</code>)
fdWrite	DWORD	Optionally a pointer to the structure, defining the socket for which the status of the write actions should be checked. You also can pass 0. Structure <code>SOCKET_FD_SET</code> see below (corresponding parameter e.g. in Win32: <code>writfds</code>)
fdExcept	DWORD	Optionally a pointer to the structure, defining the socket for which the error status should be checked. You also can pass 0. Structure <code>SOCKET_FD_SET</code> see below (corresponding parameter e.g. in Win32: <code>exceptfds</code>)
ptvTimeout	DWORD	Maximum time which the <code>SysSockSelect</code> function will wait for an answer; Structure <code>SOCKET_TIMEVAL</code> , see below (corresponding parameter e.g. in Win32: <code>timeout</code>)

Structure SOCKET_FD_SET

`fd_count:` UDINT; (* Number of sockets *)
`fd_array:` ARRAY [0..63] OF DINT; (* Field with socket descriptors *)

Structure SOCKET_TIMEVAL:

`tv_sec:` DINT; (* seconds *)
`tv_usec:` DINT; (* microseconds *)

SysSockSetIPAddress

This function of type BOOL is only implemented for **VxWin** targets. It sets the IP address of the given card.

In case of successful operation the function will return TRUE, otherwise FALSE.

For other operating systems the function always will return FALSE.

Variable	Data type	Description
stCardName	STRING	Name of the network card
stIPAddress	STRING	IP address to be set

SysSockSetOption

This function of type BOOL calls the function **getsockopt** of the operating system in order to set particular socket options.

For a description of the **getsockopt** function please see the Online Help resp. documentation on the operating system.

In case of successful operation the function will return TRUE, otherwise FALSE.

Variable	Data type	Description
diSocket	DINT	Descriptor of the socket, returned by SysSockCreate (corresponding parameter e.g. in Win32: s)
diLevel	DINT	Protocol specific level (corresponding parameter e.g. in Win32: level)
diOption	DINT	Name of the option: (corresponding parameter e.g. in Win32: optname) depends on the operating system
diOptionValue	DWORD	Option value; Deactivate boolean option values by setting a "0"; otherwise set the value (corresponding parameter e.g. in Win32: optval)
diOptionLength	DWORD	Length of the buffer for the option value (corresponding parameter e.g. in Win32: optlen)

SysSockShutdown

This function of type BOOL calls the function **shutdown** of the operating system in order to inhibit further send or receive actions. The function does not close the socket ! This must be done via SysSockClose.

In case of successful operation the function will return TRUE, otherwise FALSE.

Variable	Data type	Description
diSocket	DINT	Descriptor of the socket, returned by SysSockCreate (corresponding parameter e.g. in Win32: s)
diHow	DINT	Here you define, which type of communication actions should be inhibited (corresponding parameter e.g. in Win32: how)

For a description of the **shutdown** function please see the Online Help resp. documentation on the operating system.

TCP specific functions

The following functions are only relevant for TCP/IP communication: SysSockRecv , SysSockSend

SysSockRecv

This TCP/IP specific function of type DINT calls the function **read** (VxWorks) resp. **recv** (Win32) of the operating system in order to receive data which have been sent to the socket.
The function will return the number of read bits.

Variable	Data type	Description
diSocket	DINT	Descriptor of the socket, returned by SysSockCreate (corresponding parameter e.g. in Win32: s)
pbyBuffer	DWORD	Address of the buffer from which the data should be read (corresponding parameter e.g. in Win32: buf)
diBufferSize	DINT	Size of the buffer from which the data should be read (corresponding parameter e.g. in Win32: len)
diFlags	DINT	Defines in which way the function should be called; depending on the socket options. (corresponding parameter e.g. in Win32: flags)

For a description of the **recv** function please see the Online Help resp. documentation on the operating system. If the socket has been "gracefully closed" , 0 will be returned, otherwise 1.

SysSockSend

This TCP/IP specific function of type DINT calls the function **send** of the operating system in order to send the data which are buffered at the socket.
The function will return the number of sent bits. If the socket has been "gracefully closed" , 0 will be returned, otherwise 1.

Variable	Data type	Description
diSocket	DINT	Descriptor of the socket, returned by SysSockCreate (corresponding parameter e.g. in Win32: s)
pbyBuffer	DWORD	Address of the buffer from which the data should be send (corresponding parameter e.g. in Win32: buf)
diBufferSize	DINT	Size of the buffer from which the data should be send (corresponding parameter e.g. in Win32: len)
diFlags	DINT	Defines in which way the function should be called; depending on the socket options. (corresponding parameter e.g. in Win32: flags)

For a description of the **send**-Function please see the Online Help resp. documentation on the operating system.

UDP specific functions

The following functions are only relevant for the UDP communication: SysSockRecvFrom, SysSockSendTo

SysSockRecvFrom

This UDP specific function of type DINT calls the function **recvfrom** of the operating system, in order to read the data which have been sent to the socket.

The function will return the number of read bits. If the socket has been "gracefully closed" , 0 will be returned, otherwise 1.

Variable	Data type	Description
diSocket	DINT	Descriptor of the socket, returned by SysSockCreate (corresponding parameter e.g. in Win32: s)
pbyBuffer	DWORD	Address of the buffer from which the data should be received (corresponding parameter e.g. in Win32: buf)
diBufferSize	DINT	Size of the buffer from which the data should be received (corresponding parameter e.g. in Win32: len)
diFlags	DINT	Defines in which way the function should be called; depending on the socket options. (corresponding parameter e.g. in Win32: flags)
pSockAddr	DWORD	Pointer to a variable of type SOCKADDR; (see SysSockAccept)
diSockAddrSize	DINT	Length of structure SockAddr (can be retrieved via the SIZEOF operator) (corresponding parameter e.g. in Win32: iSockAddrSize)

For a description of the **recvfrom**-function please see the Online Help resp. documentation on the operating system.

SysSockSendTo

This UDP specific function of type DINT calls the function **send** of the operating system in order to send the data which are stored at the socket. The function will return the number of read bits. If the socket has been "gracefully closed" , 0 will be returned, otherwise 1.

Variable	Data type	Description
diSocket	DINT	Descriptor of the socket, returned by SysSockCreate (corresponding parameter e.g. in Win32: s)
pbyBuffer	DWORD	Address of the buffer from which the data should be sent (corresponding parameter e.g. in Win32: buf)
diBufferSize	DINT	Size of the buffer from which the data should be sent (corresponding parameter e.g. in Win32: en)
diFlags	DINT	Defines in which way the function should be called; depending on the socket options. (corresponding parameter e.g. in Win32: flags)
pSockAddr	DWORD	Pointer to a variable or type SOCKADDR; (see SysSockAccept)
diSockAddrSize	DINT	Length of the structure SockAddr (can be retrieved via the SIZEOF operator) (corresponding parameter e.g. in Win32: iSockAddrSize)

For a description of the **send** Function please see the Online Help resp. documentation on the operating system.