UdpSend()

This function block is used to transmit datagrams.

Because the function block is executed asynchronously, it returns status ERR_FUB_BUSY until the function block has either completed successfully or an error occurs.

This function block can only be used for **redundancy** in certain circumstances!
Applicable limitations are described in the following sections:

Communication
Handle must be initialized
Hidden pointer usage
Asynchronous execution

For additional information about redundancy, see section "Redundancy".

Parameters

I/O	Parameter	Data type	Description
IN	enable	BOOL	Enables the function block.
IN	ident	UDINT	identifier from UdpOpen.
IN	pHost	UDINT (given as a pointer to STRING)	Pointer to the IP address or name of the receiver. If the port has been connected (UdpConnect), this parameter is ignored!
IN	port	UINT	Port number of the receiver. If the port has been connected (UdpConnect), this parameter is ignored!
IN	pData	UDINT (given as a pointer to STRING)	Pointer to the data to be transmitted (datagram).
IN	datalen	UDINT	Length of data to be transmitted [bytes].
IN	flags	UDINT	Transmit flags.
OUT	status	UINT	Return value of the function (error number (0 = No error))
OUT	sentlen	UDINT	Number of transmitted bytes.

Call syntax (Automation Basic)

UdpSend (enable, ident, adr(Host), port, adr(Data), datalen, flags, status, sentlen)

Function description

This function block is used to transmit datagrams. If a port has been connected (UdpConnect), parameters *pHost* and *port* are ignored. This function block returns ERR_OK as soon as the datagram has been

transferred to the transmit buffer of the UDP socket. However, this does not mean that the remote station has already received the packet.

UDP has **no flow control**: A fast transmitter can flood a slow receiver with data. As a result, the receiving UDP port discards the datagrams.

Supported flags

- **udpMSG_DONTROUTE** - Specifies that the normal routing mechanism should be bypassed. The datagram is routed directly to the local interface (network and subnet part of the destination address). If the local interface cannot be determined, an error is returned (even if a gateway is configured).

Flags are set bit by bit.

Error numbers

Value	Name	Description	Correction
0	ERR_OK	Status OK.	
32500	udpERR_INVALID_IDENT	The specified identifier is not permitted.	Check the specified identifier.
32503	udpERR_PARAMETER	The specified IP address (name) could not be cleared, or the port number is not permitted (0). Data pointer *pData* is not set (0). An invalid flag is set.	Check the specified parameters. DNS enabled?
32506	udpERR_SENTLEN	Not all data could be applied.	Check whether the specified data length exceeds the length of the transmit buffer (udpSO_SNDBUF_GET) and correct it if necessary. If the transmit buffer is large enough to apply the whole packet, a new transmit request should be sent (load). In the event of error, the actual length of data is also entered in output parameter "sentlen".
32507	udpERR_WOULDBLOCK	The transmit request could not be processed "quickly".	Retry.
32508	udpERR_CONNREFUSED	There was a problem with a previous transmit request. This error is only returned with connected UDP ports (UdpConnect). This error is only returned with connected UDP ports (UdpConnect).	Check the connected parameters (UdpConnect: IP address, port number).
32509	udpERR_ACCESS	Transmitting broadcasts is	Check whether this port is

		not permitted.	permitted to send broadcasts (udpSO_BROADCAST_GET) and permit this if necessary (udpSO_ROADCAST_SET).
32510	udpERR_UNREACHABLE	Host/Network not available.	Check the IP address of the receiver. Check the Ethernet configuration (gateway configured?).
32512	udpERR_FIREWALL	The packet was blocked by the firewall.	Check the firewall configuration.
32599	udpERR_SYSTEM	Unexpected transmission error.	Check the system.
65534	ERR_FUB_ENABLE_FALSE	Function block not active.	Enable the function block.
65535	ERR_FUB_BUSY	Function block still working.	Continue calling the function block.