UdpOpen()

This FBK opens a UDP port.

Asynchronous handling causes the function block to return ERR_FUB_BUSY until the FBK is either finished successfully or an error occurs.

This function block can only be used conditionally for **redundancy**.
The applicable limitations are described in the following sections:

Communication
Handle must be initialized
Hidden pointer usage
Asynchronous execution

Additional information can be found in the "Redundancy" section.

Parameters

I/O	Parameter	Data type	Description
IN	enable	BOOL	Enables the FBK.
IN	pIfAddr	UDINT (transmitted as a pointer to STRING)	Pointer to the IP address of the Ethernet interface where the UDP socket should be connected (e.g. 192.168.0.1). This parameter is optional. If "0" is specified, then "listening in" takes place on all interfaces.
IN	port	UINT	Port number that the UDP socket should connect to. This parameter is optional. If "0" is specified, then a short-lived port number is assigned automatically.
IN	options	UDINT	Options that can be set during opening (options are set by bit; several options can be set).
OUT	status	UINT	Function return value (error numbers, 0 = no error)
OUT	ident	UDINT	Ident used for all other UDP FBKs as an input parameter.

Call syntax (Automation Basic)

UdpOpen (enable, adr(IfAddr), port, options, status, ident)

Function description

This FBK opens a UDP port.

If automatic port assignment (*port* = 0) is used or the **udpOPT_REUSEADDR** option is set, then a new UDP port is always created. If it's not these two cases and an attempt is made to open a UDP port that already exists (port, interface (IP address), then the **udpERR_ALREADY_EXIST** status is reported and the ident of the existing UDP port is returned (ident).

pIfAddr

This parameter specifies the IP address of the Ethernet interface (zero-terminated string, e.g. 192.168.0.1) where the UDP socket should be connected. This means that this interface only receives packages. This parameter is optional. If "0" is specified, then "listening in" takes place on all interfaces.

port

This parameter specifies the port number where the UDP socket should be connected. This parameter is optional. If "0" is specified, then a short-lived port number is assigned automatically.

Use of the port number is application-dependent: If the UDP port is used in a way that it responds to queries (UDP server), then it should have a known port number. This port number is needed by the clients to send a datagram.
On the other hand, a UDP client that queries the UDP server doesn't need to have a known port number since the server receives the port number being used for the query and has access to the correct address for the response.

Options

The following options can be set when opening a UDP port:

- **udpOPT_REUSEADDR** - Allows duplicate linking (port) for a UDP port to a local IP address.
- **udpOPT_REUSEPORT** - Allows a complete duplicate linking (allows a link() to an IP address and a port if they are already connected to another UDP port (socket)). This option must also be specified for the first UDP port!
- **udpOPT_BROADCAST** - Allows broadcasts to be sent.

Options during opening are set bit-by-bit.

Additional options can also be changed with the UdpIoctl FBK!

If several UDP ports are connected to the same port and IP address, then received data packets are routed to all UDP ports!

Task overload management

For a task overload, it's important that already existing UDP ports remain present. If UdpOpen() is then used with either automatic port assignment or the udpOPT_REUSEADDR option, a new UDP port will always be created. In this case, UdpIoctl() can be used to determine the already existing UDP ports.

Connecting a UDP socket to an interface (IP address) is as well possible as a port number. If a UDP socket is connected to an interface, it can only receive data that arrives at this interface. Several UDP sockets with the same port number can be connected to different interfaces (IP addresses).

If a UDP socket is connected to an interface, then it cannot receive any multicasts!

Error numbers

Value	Name	Description	Correction
0	ERR_OK	Status OK.	
32501	udpERR_NOMORE_IDENTS	Could not reserve additional ident.	Free up any existing (and unnecessary) idents.
32502	udpERR_ALREADY_EXIST	UDP port already connected to this port number (device):	If this status occurs, the ident of the opened socket is returned. Close the socket.
32550	udpERR_SOCKET_CREATE	Resource problem in system. Could not create new sockets.	Check whether sockets are still available. Free up resources if necessary.
32551	udpERR_SOCKET_BIND	Problem connecting to the port number or IP address (*pDevice*).	Check the specified port number or device.
32511	udpERR_INTERFACE	The specified interface address is invalid.	Check the specified interface.
32599	udpERR_SYSTEM	Problem setting options on the open socket.	System check.
65534	ERR_FUB_ENABLE_FALSE	FBK not enabled.	Enable the FBK.
65535	ERR_FUB_BUSY	FBK still working.	Call it again.