Udploctl()

This function block is used to change or retrieve settings.

Because it is executed asynchronously, the function block returns ERR_FUB_BUSY until it 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 more 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 or udpINVALID_IDENT.
IN	ioctl	UDINT	I/O control
IN	pData	UDINT (given as a pointer to STRING)	Pointer to input/output data buffer
IN	datalen	UDINT	Length of the input data or maximum length of the output data
OUT	status	UINT	Return value of function (error number (0 = no error))
OUT	outlen	UDINT	Output data length

Call syntax (Automation Basic)

UdpIoctl (enable, ident, ioctr, adr(Data), datalen, status, outlen)

Function description

This function block allows special settings to be made. Care should be taken when modifying certain parameters, however!

I/O controls independent of UPD port

udpINVALID_IDENT must be specified as the ident for these controls.

Ident	Description
udpID_LIST_GET	Outputs a list of open idents. Parameter *pData* points to an array of type UDINT in which the open identifiers are written. The *datalen* parameter contains the length of the array. After a successful call, the actual number of open identifiers can be taken from parameter *outlen*. If the number of open identifiers exceeds the length of the array (*datalen*), determination is terminated: Error udpERR_PARAMETER, the number of actually open identifiers is in parameter *outlen*.
udpID_MAX_GET	Returns the number of maximum available idents. Parameter *pData* points to a variable of type UDINT, and parameter *datalen* is set to the length of type UDINT (4). The variable pointed to by *pData* contains the maximum number of idents after the function block has executed successfully.

I/O controls specific to UDP port

A valid ident must be specified as the ident for these controls.

Ident	Description

udpSO_ADDRESS_GET	Gets the IP address and port of the specified ident (socket).
	
	Parameter *pData* points to a structure of type *udpSO_ADDRESS_typ*, and parameter *datalen* is set to the length of the structure.
	
	`udpSO_ADDRESS_typ : STRUCT`
	`  pPort  : UDINT;`
	`  pIpAddr  : UDINT;`
	`END_STRUCT`
	
	Structure parameter *pPort* is a pointer to a variable of type UINT where the port number of the UDP port is written.
	Structure parameter *pIpAddr* is a pointer to a string (min. 16 characters) where the IP address of the UDP port is written.
	
	These parameters are optional. Simply set them to 0 and they will be ignored.

udpSO_BROADCAST_SET	Enables the sending of broadcasts. This option can also be set directly when a UDP port is opened (UdpOpen). Parameter *pData* points to a variable of type UDINT, and parameter *datalen* is set to the length of type UDINT (4). The option can be enabled/disabled by setting (1) or resetting (0) the variable.
udpSO_BROADCAST_GET	Reads the BROADCAST option. Parameter *pData* points to a variable of type UDINT where the status of the option is written (0 - no broadcasts possible). The *datalen* parameter is set to the length of the UDINT type (4).
udpSO_SNDBUF_SET	Adapts the size of the socket transmit buffer. Parameter *pData* points to a variable of type UDINT that contains the desired size of the transmit buffer. The *datalen* parameter is set to the length of the UDINT type (4). The default size should only be changed in certain applications, e.g. to optimize performance.
udpSO_SNDBUF_GET	Reads the size of the socket transmit buffer. Parameter *pData* points to a variable of type UDINT where the size of the transmit buffer is entered. The *datalen* parameter is set to the length of the UDINT type (4).
udpSO_RCVBUF_SET	Adapts the size of the socket receive buffer. Parameter *pData* points to a variable of type UDINT that contains the desired size of the receive buffer. Parameter *datalen* is set to the length of type UDINT (4). 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. The default size should only be modified in certain applications, e.g. to optimize performance.
udpSO_RCVBUF_GET	Reads the size of the socket receive buffer. Parameter *pData* points to a variable of type UDINT where the size of the receive buffer is entered. The *datalen* parameter is set to the length of the UDINT type (4).
udpIP_TOS_SET	Sets the "Type of service" array for outgoing packets from this socket. Valid values: • udpIP_TOS_LOWDELAY • udpIP_TOS_THROUGHPUT • udpIP_TOS_RELIABILITY • udpIP_TOS_MINCOST Parameter *pData* points to a variable of type UDINT that contains the value to be set. The *datalen* parameter is set to the length of the UDINT type (4).
udpIP_TOS_GET	Reads the currently set value of the "Type of service" array. Parameter "pData" points to a variable of type UDINT where the result is entered. Parameter "datalen" is set to the length of type UDINT (4).
udpIP_TTL_SET	Sets the "Time to live" array for outgoing packets from this socket. Parameter "pData" points to a variable of type UDINT where the result is entered. Parameter "datalen" is set to the length of type UDINT (4).

<table>
<tr><td>udpIP_TTL_GET</td><td>Reads the currently set value of the "Time to live" array.

Parameter "pData" points to a variable of type UDINT where the result is entered. Parameter "datalen" is set to the length of type UDINT (4).</td></tr>
<tr><td>udpIP_ADD_MEMBERSHIP</td><td>Connects a multicast group.

Parameter pData points to a structure of type udpIP_MREQ_typ, and parameter datalen is set to the length of the structure.

<pre>udpIP_MREQ_typ   : STRUCT
        pMcastAddr   : UDINT;
        pIfAddr   : UDINT;
END_STRUCT</pre>
Structure parameter pMcastAddr is a pointer to the multicast address (string format).
Structure parameter plfAddr is a pointer to the IP address of the interface (string format).

A multicast interface must be defined in order to connect a multicast group.

If a UDP socket is connected to an interface, then it cannot receive any multicasts.</td></tr>
<tr><td>udpIP_DROP_MEMBERSHIP</td><td>Leaves a multicast group.

Parameter pData points to a structure of type udpIP_MREQ_typ, and parameter datalen is set to the length of the structure.

<pre>udpIP_MREQ_typ   : STRUCT
        pMcastAddr   : UDINT;
        pIfAddr   : UDINT;
END_STRUCT</pre>
Structure parameter pMcastAddr is a pointer to the multicast address (string format).
Structure parameter plfAddr is a pointer to the IP address of the interface (string format).</td></tr>
<tr><td>udpIP_MULTICAST_IF_SET</td><td>Specifies the interface for outgoing multicasts.

Parameter pData points to the IP address (string format) of the interface used to send multicasts.</td></tr>
<tr><td>udpIP_MULTICAST_IF_GET</td><td>Reads the interface for outgoing multicasts.

Parameter pData points to a string (min. 16 characters) that is written with the IP address of the interface being used to send multicasts.</td></tr>
<tr><td>udpIP_MULTICAST_TTL_SET</td><td>Specifies the "time to life" for outgoing multicasts.

Parameter "pData" points to a variable of type USINT that contains the desired size of the TTL. Parameter "datalen" is set to the length of type USINT (1).

<table>
<tr><th>TTL</th><th>Application</th><th>Scope</th></tr>
<tr><td>0</td><td></td><td>Same interface</td></tr>
<tr><td>1</td><td></td><td>Same subnet</td></tr>
<tr><td>31</td><td>Local event video</td><td></td></tr>
<tr><td>32</td><td></td><td>Same site</td></tr>
<tr><td>63</td><td>Local event audio</td><td></td></tr>
<tr><td>64</td><td></td><td>Same region</td></tr>
<tr><td>95</td><td>IETF channel 2 video</td><td></td></tr>
<tr><td>127</td><td>IETF channel 1 video</td><td></td></tr>
<tr><td>128</td><td></td><td>Same continent</td></tr>
<tr><td>159</td><td>IETF channel 2 audio</td><td></td></tr>
<tr><td>191</td><td>IETF channel 1 audio</td><td></td></tr>
<tr><td>223</td><td>IETF channel 2 low-rate audio</td><td></td></tr>
<tr><td>255</td><td>IETF channel 1 low-rate audio</td><td></td></tr>
<tr><td></td><td>Unrestricted in scope</td><td></td></tr>
</table></td></tr>
<tr><td>udpIP_MULTICAST_TTL_GET</td><td>Reads the "time to life" for outgoing multicasts.

Parameter "pData" points to a variable of type USINT where the size of the TTL is entered. Parameter "datalen" is set to the length of type USINT (1).</td></tr>
<tr><td>udpIP_MULTICAST_LOOP_SET</td><td>Turns multicast loopback on/off.

Parameter "pData" points to a variable of type USINT, and parameter "datalen" is set to the length of type USINT (1). The option can be enabled/disabled by setting (1) or resetting (0) the variable.</td></tr>
<tr><td>udpIP_MULTICAST_LOOP_GET</td><td>Reads the LOOPBACK option.</td></tr>
</table>

	Parameter "pData" points to a variable of type USINT where the status of the option is written (0 - inactive). Parameter "datalen" is set to the length of type USINT (1).

udpSO_BINDTODEVICE	Bind a socket to a device (not available on ARsim).
	
	When a socket is bound to a physical Ethernet interface, only packets arriving via that physical Ethernet device will be received via that socket.
	If pIfAddr=0 is specified for UdpOpen and udpSO_BINDTODEVICE is then applied to this socket, for example, it is possible to detect which device has received broadcast messages.
	In addition, the messages to be transmitted are always transferred on this physical Ethernet interface. Thus, a "limited broadcast" (IP=255.255.255.255) can be transferred via the specified interface, for example!
	
	`pData ..... Pointer auf einen String mit dem Namen des Ethernet Devices (z.B. 'IF2')`
	`datalen ..... Länge des Strings`

Error numbers

Value	Description	Description	Fix
0	ERR_OK	Status OK	
32500	udpERR_INVALID_IDENT	The specified identifier is not permitted.	Check the specified identifier.
32503	udpERR_PARAMETER	Illegal parameter specification (null pointer, data length).	Check the specified parameters.
32504	udpERR_INVALID_IOCTL	Illegal I/O control.	Check the specified I/O controls. This error is also returned if an identifier is specified but an I/O control should be carried out independent of the identifier!
32599	udpERR_SYSTEM	Unexpected error when setting or retrieving the desired option (IOCTL).	Check the system.
65534	ERR_FUB_ENABLE_FALSE	Function block not enabled.	Enable the function block.
65535	ERR_FUB_BUSY	Function block still working.	Continue calling the function block.