


WAGO-I/O-PRO 32 Library

Ethernet.lib

Contents

ETHERNET_CLIENT_CLOSE	3
ETHERNET_CLIENT_OPEN.....	5
ETHERNET_GET_NETWORK_CONFIG.....	8
ETHERNET_GETSOURCEPORT	10
ETHERNET_GET_VARIABLES	12
ETHERNET_READ	14
ETHERNET_READ_PT.....	16
ETHERNET_SERVER_CLOSE	18
ETHERNET_SERVER_OPEN	20
ETHERNET_SET_NETWORK_CONFIG	23
ETHERNET_SETSOURCEPORT	25
ETHERNET_SET_VARIABLES.....	27
ETHERNET_VERSION.....	29
ETHERNET_WRITE.....	31
ETHERNET_WRITE_PT	33
ETH_ERROR (Data type).....	35
SEL_PROTOCOL (Data type).....	38
SEL_TYPE (Data type).....	40

ETHERNET_CLIENT_CLOSE

WAGO-I/O-PRO 32 Library elements			
Category:	ETHERNET Fieldbus Communication		
Name:	ETHERNET_CLIENT_CLOSE		
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/>	Program <input type="checkbox"/>
Library name:	Ethernet.lib		
Applicable to:	Ethernet controller 750-842		
Input parameter:			
Data type:	Comments:		
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.	
SOCKET	WORD	Socket handle. Used to identify the socket to be closed.	
Output parameter:			
Data type:	Comments:		
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.	
ERROR	ETH_ERROR	Error argument 0: no errors found <>0: error in operation (see ETH_ERROR data type for details)	
Header:			
<pre> FUNCTIONBLOCK ETHERNET_CLIENT_CLOSE VAR_INPUT EN: BOOL; SOCKET: WORD; END_VAR VAR_OUTPUT ENO: BOOL; ERROR: ETH_ERROR; END_VAR </pre>			
Graphic display:			
			

Time behavior:

Function description:

This function block is used to close a client socket that was created by ETHERNET_SERVER_OPEN or ETHERNET_CLIENT_OPEN.

The function's **behavior** depends on the type of socket. If the socket is a UDP or RAW socket, this function can only be used when the socket was created by ETHERNET_CLIENT_OPEN. With TCP sockets the function can be used either with sockets created by ETHERNET_CLIENT_OPEN or with sockets created by ETHERNET_SERVER_OPEN, but there are subtle differences.

If the TCP socket was created by ETHERNET_SERVER_OPEN, this function doesn't close the server socket, but just the underlying socket client that holds the TCP connection. In this case, the socket can be still used to accept data from another nodes. The server remains open.

If the TCP socket was created by ETHERNET_CLIENT_OPEN, this function closes the client socket. Another call to ETHERNET_CLIENT_OPEN must be done before using the socket descriptor for reading or writing data.

After a socket is closed, it is possible to create a new one with different parameters. Since the number of sockets is limited to 2, it is recommended to close a socket that is no longer used.

Closing starts on a rising slope of EN. The connection is closed when the function block call returns.

See also: ETHERNET_CLIENT_OPEN, ETHERNET_SERVER_OPEN, ETHERNET_READ, ETHERNET_WRITE

ETHERNET_CLIENT_OPEN

WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_CLIENT_OPEN	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib	
Applicable to:	Ethernet controller 750-842	
Input parameter:	Data type:	Comments:
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
TYP	SEL_TYPE	Semantics of communication 1: SOCK_STREAM 2: SOCK_DGRAM 3: SOCK_RAW
PROTO	SEL_PROTOCOL	Protocol to be used 1: IPPROTO_ICMP 2: IPPROTO_IGMP 6: IPPROTO_TCP 17: IPPROTO_UDP 255: IPPROTO_RAW
IP_ADR	DWORD	Target IP adress (use 0 if you want a not connected socket) E.g. use 16#861C2A0D for 134.28.42.13 (134 = 16#86 ...)
PORT	WORD	Port address on target system
Output parameter:	Data type:	Comments:
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
SOCKET	WORD	Socket handle. Used in other Ethernet functions.
ERROR	ETH_ERROR	Error argument. 0: no errors found <>0: error in operation (see ETH_ERROR date type for details)

Header:

```

FUNCTIONBLOCK ETHERNET_CLIENT_OPEN
VAR_INPUT
  EN: BOOL;
  TYPE: SEL_TYPE;
  PROTO: SEL_PROTOCOL;
  IP_ADR: DWORD;
  PORT: WORD;
END_VAR
VAR_OUTPUT
  ENO: BOOL;
  SOCKET: WORD;
  ERROR: ETH_ERROR;
END_VAR

```

Graphic display:

Time behavior:

Function description:

This function block is used to create a socket. It is the base for a communication to a remote system. The function block ETHERNET_CLIENT_OPEN combines the function calls **socket** and **connect** known from UNIX or the function calls **CSocket::Create** and **CSocket::Connect** known from MS Windows Winsocket. The function call **connect** is only done if a remote IP address is specified (input IP_ADR). This function block must be successfully called before any communication (client-side) can be done. Only two sockets can be simultaneously used.

The TYP input specifies the semantics of communication. Currently supported semantics are SOCK_STREAM (1), SOCK_DGRAM(2) and SOCK_RAW(3). A SOCK_STREAM TYP input provides sequenced, reliable, two-way connection based byte streams. A SOCK_DGRAM TYP input supports datagrams (connectionless, unreliable messages of a fixed (typically small) maximum length). SOCK_RAW sockets provide access to internal network protocols.

Sockets of type SOCK_STREAM are full-duplex byte streams, similar to pipes. A stream socket must be in a connected state before any data may be sent or received on it. A connection to another socket is created if the input IP_ADR is not zero. Once connected, data may be transferred using ETHERNET_READ and ETHERNET_WRITE function blocks. When a session has been completed a ETHERNET_CLOSE may be performed.

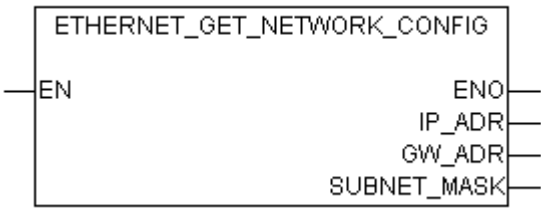
Sockets of type SOCK_DGRAM allow sending of datagrams to correspondents named in ETHERNET_WRITE calls. Therefore a datagram socket doesn't need to be in a connected state before any data may be sent or received. In this case the input IP_ADR can be zero. If a non-zero IP_ADR input is specified, this address is that to which datagrams are to be sent, and the only address from which datagrams are to be received.

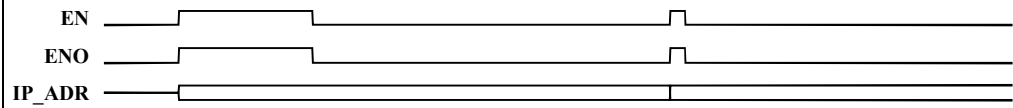
The PROTO input specifies a particular protocol to be used with the socket. Normally only a single protocol exists to support a particular socket type. However, it is possible that many protocols may exist (for example with SOCK_RAW semantics), in which case a particular protocol must be specified in this manner.

On a rising slope of EN this function block creates a socket and, if a remote IP address is specified, opens a communication channel to a remote server as a client. The input values are sampled on a positive slope. The Output ENO goes to high when the communication is established. The output SOCKET is set to a valid value. As long as ENO is low it is possible to abort the communication setup by setting the input EN to low.

See also: ETHERNET_CLOSE, ETHERNET_READ, ETHERNET_WRITE

ETHERNET_GET_NETWORK_CONFIG

WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_GET_NETWORK_CONFIG	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib	
Applicable to:	Ethernet controller 750-842	
Input parameter:	Data type:	Comments:
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
Output parameter:	Data type:	Comments:
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
IP_ADR	DWORD	Current IP address for this node 0 if no IP address configured
GW_ADR	DWORD	Current gateway address for this node 0 if no gateway address configured
SUBNET_MASK	DWORD	Current subnet mask for this node 0 if no subnet mask configured
Header:		
<pre> FUNCTIONBLOCK ETHERNET_GET_NETWORK_CONFIG VAR_INPUT EN: BOOL; END_VAR VAR_OUTPUT ENO: BOOL; IP_ADR: DWORD; GW_ADR: DWORD; SUBNET_MASK: DWORD; END_VAR </pre>		
Graphic display:		
		

Time behavior:**Function description:**

This function is used to get the node's network configuration.

On a rising slope of EN this function block gets the IP address, the gateway address and the subnet mask for this node.

See also: `ETHERNET_SET_NETWORK_CONFIG`

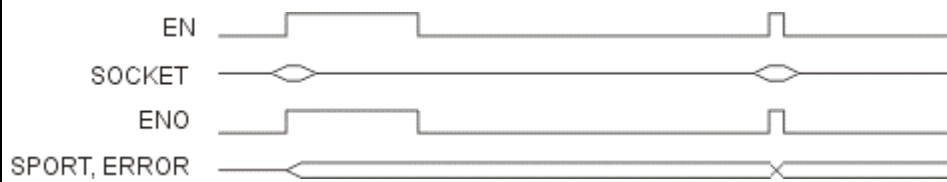
ETHERNET_GETSOURCEPORT

WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_GETSOURCEPORT	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib (as from version 1.1)	
Applicable to:	Ethernet controller 750-842	
Input parameter:		
	Data type:	Comments:
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
SOCKET	WORD	Socket descriptor. This parameter is used for identification of the socket.
Output parameter:		
	Data type:	Comments:
SPORT	WORD	SPORT returns the local TCP/IP or UDP port number of the socket which sends telegrams.
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
ERROR	ETH_ERROR	Error argument. 0: no errors found <>0: error in operation (see ETH_ERROR data type for details)
Header:		
<pre> FUNCTION_BLOCK ETHERNET_GETSOURCEPORT VAR_INPUT EN: BOOL; SOCKET: WORD; END_VAR VAR_OUTPUT SPORT: WORD; ENO: BOOL; ERROR: ETH_ERROR; END_VAR VAR END_VAR </pre>		

Graphic display:



Time behavior:



Function description:

This function is used for querying the local port number from which protocols (data frames) will send.

The function can be used in communication protocols, for example, for which the data application is specified by the port number.

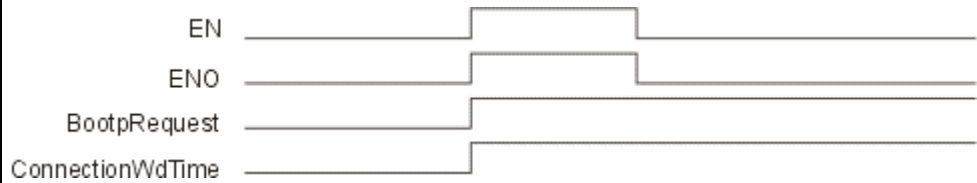
ETHERNET_GET_VARIABLES

WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_GET_VARIABLES	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib	
Applicable to:	ETHERNET controller 750-842	
Input parameter:	Data type:	Comments:
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
Output parameter:	Data type:	Comments:
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
BootpRequest	WORD	Switching the BootP-Request on and off when starting the controller. 1 : BootP-Request is executed 0 : BootP-Request is not executed
ConnectionWdTime	WORD	Connection monitoring Modbus/TCP 0 : No connection monitoring 10 – 65535 Watchdog time in ms
ERROR	ETH_ERROR	Error argument. 0: no errors found <>0: error in operation (see ETH_ERROR data type for details)
Header:		
<pre> FUNCTIONBLOCK ETHERNET_GET_VARIABLES VAR_INPUT EN: BOOL; END_VAR VAR_OUTPUT ENO: BOOL; BootpRequest: WORD; ConnectionWdTime: WORD; ERROR: ETH_ERROR; END_VAR </pre>		

Graphic display:



Time behavior:



Function description:

This function serves the selection for connecting the watch dog timer and the BootP configuration.

See also: ETHERNET_SET_VARIABLES

ETHERNET_READ

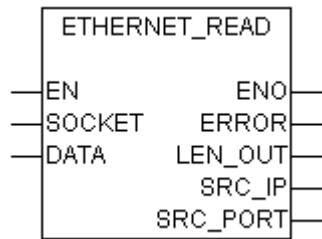
WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_READ	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib	
Applicable to:	Ethernet controller 750-842	
Input parameter:		
Data type:	Comments:	
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
SOCKET	WORD	Socket handle. Used to identify the socket to read from.
Output parameter:		
Data type:	Comments:	
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
LEN_OUT	WORD	Number of bytes read.
ERROR	ETH_ERROR	Error argument. 0: no errors found <>0: error in operation (see ETH_ERROR data type for details)
SRC_IP	DWORD	Sender's IP address (only valid if not connected)
SRC_PORT	WORD	Sender's port number (only valid if not connected)
In/Output parameter:		
Data type:	Comments:	
DATA	ARRAY [1 .. 1500] OF BYTE	Array for the received data

Header:

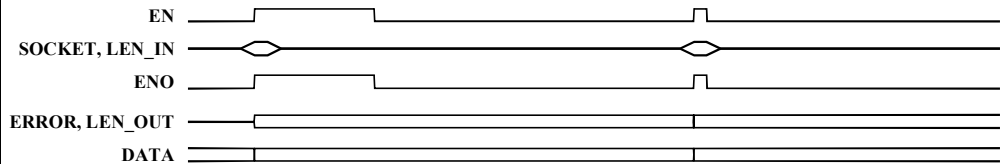
```

FUNCTIONBLOCK ETHERNET_READ
VAR_INPUT
  EN: BOOL;
  SOCKET: WORD;
END_VAR
VAR_OUTPUT
  ENO: BOOL;
  LEN_OUT: WORD;
  ERROR: ETH_ERROR;
  SRC_IP: DWORD;
  SRC_PORT: WORD;
END_VAR
VAR_IN_OUT
  DATA: ARRAY[1 .. 1500] OF BYTE
END_VAR
    
```

Graphic display:



Time behavior:



Function description:

This function block reads data received from a remote system. If no data was received since the last call it returns the data length LEN_OUT 0. If an error occurred, the output variable ERROR shows the error value.

If the socket is not connection-oriented, the source address and source port of the message are filled in (in SRC_IP and SRC_PORT). This values can be used to answer the incoming messages.

See also: ETHERNET_CLIENT_OPEN, ETHERNET_SERVER_OPEN, ETHERNET_CLOSE, ETHERNET_WRITE

ETHERNET_READ_PT

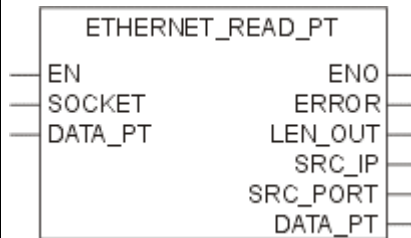
WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_READ_PT	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib (as from version 1.1)	
Applicable to:	Ethernet controller 750-842	
Input parameter:		
Data type:	Comments:	
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
SOCKET	WORD	Socket descriptor. Used for identification of the socket, which is to be read from.
Output parameter:		
Data type:	Comments:	
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
LEN_OUT	WORD	Number of bytes read.
ERROR	ETH_ERROR	Error argument. 0: no errors found <>0: error in operation (see ETH_ERROR data type for details)
SRC_IP	DWORD	Sender's IP address (only valid in a connected state)
SRC_PORT	WORD	Sender's port number (only valid in a connected state)
In/Output parameter:		
Data type:	Comments:	
DATA_PT	POINTER TO ARRAY [1 .. 1500] OF BYTE	Pointer to array of received data

Header:

```

FUNCTIONBLOCK ETHERNET_READ_PT
VAR_INPUT
    EN: BOOL;
    SOCKET:WORD;
END_VAR
VAR_OUTPUT
    ENO: BOOL;
    ERROR:ETH_ERROR;
    LEN_OUT:WORD;
    SRC_IP:DWORD;
    SRC_PORT:WORD;
END_VAR
VAR_IN_OUT
    DATA_PT: POINTER TO ARRAY [1 .. 1500] OF BYTE;
END_VAR
    
```

Graphic display:



Time behavior:

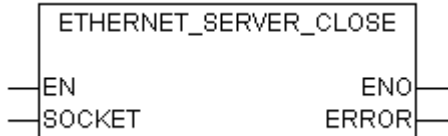


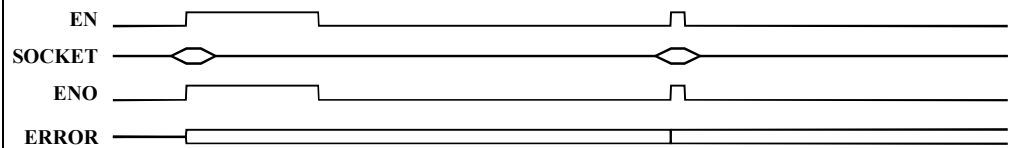
Function description:

This function block reads data received from a decentralized system. If no data has been received since the last call, then it sets the data length LEN_OUT to 0. If an error occurs, the output variable ERROR displays the error value. If the socket is not connection oriented, then the source address and the source port of the message are used (in SRC_IP and SRC_PORT). This value can be used for responding to incoming messages. The function block ETHERNET_READ_PT only differs with regards to the data transfer from the FB ETHERNET_READ. The data to be received must not be copied due to the use of pointers. This therefore leads to an advantage in the processing speed. The general directions on the use of pointers in SPS programming are to be complied with.

See also: ETHERNET_CLIENT_OPEN, ETHERNET_SERVER_OPEN, ETHERNET_CLOSE, ETHERNET_WRITE, ETHERNET_READ, ETHERNET_WRITE_PT

ETHERNET_SERVER_CLOSE

WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_SERVER_CLOSE	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib	
Applicable to:	Ethernet controller 750-842	
Input parameter:		
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
SOCKET	WORD	Socket handle. Used to identify the socket to be closed.
Output parameter:		
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
ERROR	ETH_ERROR	Errors argument. 0: no errors found <>0: error in operation (see ETH_ERROR data type for details)
Header:		
<pre>FUNCTIONBLOCK ETHERNET_SERVER_CLOSE VAR_INPUT EN: BOOL; Socket: WORD; END_VAR VAR_OUTPUT ENO: BOOL; ERROR: ETH_ERROR; END_VAR</pre>		
Graphic display:		
		

Time behavior:**Function description:**

This function block is used to close a socket that was created by `ETHERNET_SERVER_OPEN`. With TCP sockets, the server and all underlying connections (clients connected to the server) are closed.

After a socket is closed, it is possible to create a new one with different parameters. Since the number of sockets is limited to 2, it is recommended to close a socket that is no longer used.

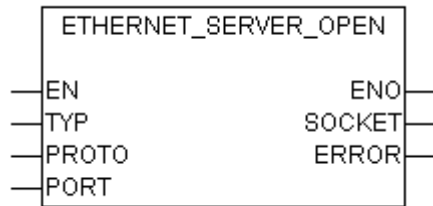
Closing starts on a rising slope of EN. The connection is closed when the function block call returns.

See also: `ETHERNET_CLIENT_OPEN`, `ETHERNET_SERVER_OPEN`, `ETHERNET_READ`, `ETHERNET_WRITE`

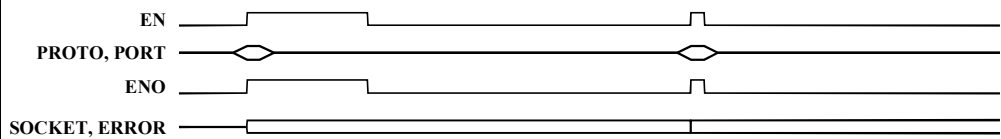
ETHERNET_SERVER_OPEN

WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_SERVER_OPEN	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib	
Applicable to:	Ethernet controller 750-842	
Input parameter:	Data type:	Comments:
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
TYP	SEL_TYPE	Semantics of communication 1: SOCK_STREAM 2: SOCK_DGRAM 3: SOCK_RAW
PROTO	SEL_PROTOCOL	Protocol to be used 1: IPPROTO_ICMP 2: IPPROTO_IGMP 6: IPPROTO_TCP 17: IPPROTO_UDP 255: IPPROTO_RAW
PORT	WORD	Port number for the server on this system
Output parameter:	Data type:	Comments:
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
SOCKET	WORD	Socket handle. Used in other Ethernet functions.
ERROR	ETH_ERROR	Error argument. 0: no errors found <>0: error in operation (see ETH_ERROR data type for details)
Header:		
<pre> FUNCTIONBLOCK ETHERNET_SERVER_OPEN VAR_INPUT EN: BOOL; TYP: SEL_TYPE; PROTO: SEL_PROTOCOL; PORT: WORD; END_VAR VAR_OUTPUT ENO: BOOL; SOCKET: WORD; ERROR: ETH_ERROR; END_VAR </pre>		

Graphic display:



Time behavior:



Function description:

This function block is used to create a server socket. It is the base for a communication to a remote system. The function block ETHERNET_SERVER_OPEN combines the function calls **socket**, **bind**, **listen** and **accept** known from UNIX or the function calls **CSocket::Create**, **CSocket::Listen** und **CSocket::Accept** known from MS Windows Winsocket. This function block must be successfully called before any server-side communication can be done.

The TYP input specifies the semantics of communication. Currently supported semantics are SOCK_STREAM (1), SOCK_DGRAM(2).and SOCK_RAW(3). A SOCK_STREAM TYP input provides sequenced, reliable, two-way connection based byte streams. A SOCK_DGRAM TYP input supports datagrams (connectionless, unreliable messages of a fixed (typically small) maximum length). SOCK_RAW sockets provide access to internal network protocols.

Sockets of type SOCK_STREAM are full-duplex byte streams, similar to pipes. A stream socket must be in a connected state before any data may be sent or received on it. The connection is done in a transparent way for the PFC user. The PFC user reads messages from the socket through the call to ETHERNET_READ. If no data have come, the function returns a 0 byte length. If data have come the PFC user can send an answer by calling the ETHERNET_WRITE function on the socket. There is no need to know the source IP address or port number, because the communication is connected. When a session has been completed an ETHERNET_CLOSE may be performed.

Sockets of type SOCK_DGRAM doesn't need to be in a connected state before any data may be sent or received. The PFC user reads messages from the socket through the call to ETHERNET_READ. If no data have come, the function returns a 0 byte length. If data have come, the PFC user can read the source IP address and source port number from the ETHERNET_READ function call and send the answer to this IP address and port number through ETHERNET_WRITE. When a session has been completed an ETHERNET_CLOSE may be performed.

ETHERNET_SERVER_OPEN

The PROTO input specifies a particular protocol to be used with the socket. Normally only a single protocol exists to support a particular socket type. However, it is possible that many protocols may exist (for example with SOCK_RAW semantics), in which case a particular protocol must be specified in this manner.

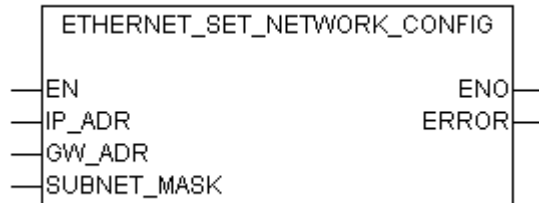
On a rising slope of EN this function block creates a socket and starts listening to port number PORT. The input values are sampled on a positive slope.

See also: ETHERNET_CLOSE, ETHERNET_READ, ETHERNET_WRITE

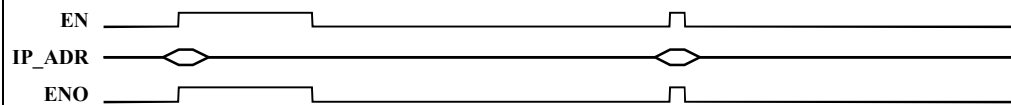
ETHERNET_SET_NETWORK_CONFIG

WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_SET_NETWORK_CONFIG	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib	
Applicable to:	Ethernet controller 750-842	
Input parameter:	Data type:	Comments:
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
IP_ADR	DWORD	New IP address for this node. E.g. use 16#861C2A0D for 134.28.42.13 (134 = 16#86 ...)
GW_ADR	DWORD	New gateway address for this node. E. g. use 16#861C2A0D for 134.28.42.13 (134 = 16#86 ...) Use 0 if no need to configure a gateway
SUBNET_MASK	DWORD	New subnet mask for this node. E. g. use 16#FFFF0000 for 255.0.0.0 Use 0 if no need to configure subnetting
Output parameter:	Data type:	Comments:
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
ERROR	ETH_ERROR	Error argument. 0: no errors found <>0: error in operation (see ETH_ERROR data type for details)
Header:		
<pre> FUNCTIONBLOCK ETHERNET_SET_NETWORK_CONFIG VAR_INPUT EN: BOOL; IP_ADR: DWORD; GW_ADR: DWORD; SUBNET_MASK: DWORD; END_VAR VAR_OUTPUT ENO: BOOL; ERROR: ETH_ERROR; END_VAR </pre>		

Graphic display:



Time behavior:



Function description:

This function block is used to set the node's network configuration.

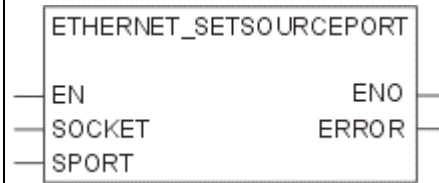
On a rising slope of EN this function block sets the new IP address, the new gateway address and the new subnet mask. In case of wrong configuration, the output variable ERROR contains the error code.

See also: ETHERNET_GET_NETWORK_CONFIG

ETHERNET_SETSOURCEPORT

WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET FieldbusCommunication	
Name:	ETHERNET_SETSOURCEPORT	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib (as from version 1.1)	
Applicable to:	Ethernet controller 750-842	
Input parameter:	Data type:	Comments:
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
SOCKET	WORD	Socket descriptor. This parameter is used for identification of the socket.
SPORT	WORD	SPORT specifies the TCP/IP or UDP port number, which should be used for the next telegram to be sent.
Output parameter:	Data type:	Comments:
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
ERROR	ETH_ERROR	Error argument. 0: no errors found <>0: error in operation (see ETH_ERROR data type for details)
Header:		
<pre> FUNCTION_BLOCK ETHERNET_SETSOURCEPORT VAR_INPUT EN: BOOL; SOCKET: WORD; SPORT: WORD; END_VAR VAR_OUTPUT ENO: BOOL; ERROR: ETH_ERROR; END_VAR VAR END_VAR </pre>		

Graphic display:



Time behavior:



Function description:

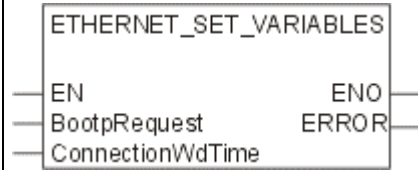
This function sets the port number. All subsequent telegrams are sent from this port.

The function can be used in communication protocols, for example, for which the data use is specified by the port number.

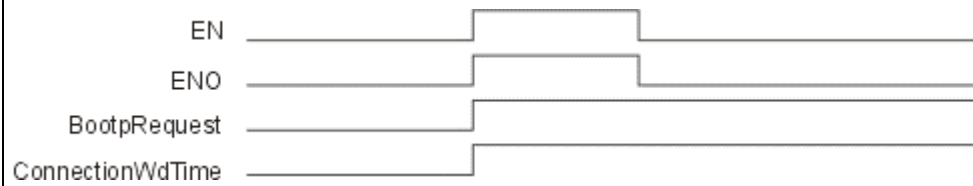
ETHERNET_SET_VARIABLES

WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_SET_VARIABLES	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib	
Applicable to:	ETHERNET controller 750-842	
Input parameter:	Data type:	Comments:
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
BootpRequest	WORD	Switching the BootP-Request on and off when starting the controller. 1 : BootP-Request is executed 0 : BootP-Request is not executed
ConnectionWdTime	WORD	Connection monitoring Modbus/TCP 0 : No connection monitoring 10 – 65535 Watchdog time in ms
Output parameter:	Data type:	Comments:
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
ERROR	ETH_ERROR	Error argument 0: no errors found <>0: error in operation (see ETH_ERROR data type for details)
Header:		
<pre> FUNCTIONBLOCK ETHERNET_SET_VARIABLES VAR_INPUT EN: BOOL; BootpRequest: WORD; ConnectionWdTime: WORD; END_VAR VAR_OUTPUT ENO: BOOL; ERROR: ETH_ERROR; END_VAR </pre>		

Graphic display:



Time behavior:


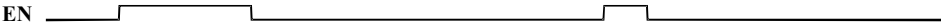


Function description:

The function block sets the connection control time for the Modbus/TCP protocol. The setup range is between 10 and 65535 ms. Assigning a 0 to the connection control deactivates the function. Configuration of the controller can be selected on and/or off over the BootP.

See also: ETHERNET_GET_VARIABLES

ETHERNET_VERSION

WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_VERSION	
Type:	Function <input checked="" type="checkbox"/>	Function block <input type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib	
Applicable to:	Ethernet controller 750-842	
Input parameter:	Data type:	Comments:
EN	BOOL	FB execution control. The function is executed on a rising edge at EN.
Return value:	Data type:	Comments:
ETHERNET_VERSION	WORD	Library version
Header:		
<pre>FUNCTION ETHERNET_VERSION: WORD VAR_INPUT EN: BOOL; END_VAR</pre>		
Graphic display:		
		
Time behavior:		
		
Function description:		
This function is used to get the library's current version		
Version:	Description:	
1.0	First version (16#100)	

WAGO-I/O-PRO 32 Library elements	
1.1	(16#101) Additional function blocks ETHERNET_GETSOURCEPORT, ETHERNET_SETSOURCEPORT, ETHERNET_READ_PT, ETHERNET_WRITE_PT
1.2	(16#102) Additional function blocks ETHERNET_GET_VARIABLES ETHERNET_SET_VARIABLES
1.3	(16#103) The function blocks ETHERNET_GET_VARIABLES and ETHERNET_SET_VARIABLES are improved.
5.0	Compatible with the WAGO-I/O-PRO 32 version 2.2 SR5. The library works with the onlinechange feature.

ETHERNET_WRITE

WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_WRITE	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib	
Applicable to:	Ethernet controller 750-842	
Input parameter:	Data type:	Comments:
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
SOCKET	WORD	Socket handle. Used to identify the socket to write to.
LEN_IN	WORD	Number of bytes to write.
DST_IP	DWORD	Destination IP address (only used if not connected)
DST_PORT	WORD	Destination port (only used if not connected)
Output parameter:	Data type:	Comments:
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
ERROR	ETH_ERROR	Error argument. 0: no errors found <>0: error in operation (see ETH_ERROR data type for details)
In/Output parameter:	Data type:	Comments:
DATA	ARRAY (1 .. 1500] OF BYTE	Array for the data to send

Header:

```

FUNCTIONBLOCK ETHERNET_WRITE
VAR_INPUT
  EN: BOOL;
  SOCKET: WORD;
  LEN_IN: WORD;
  DST_IP: DWORD;
  DST_PORT: WORD;
END_VAR
VAR_OUTPUT
  ENO: BOOL;
  ERROR: ETH_ERROR;
END_VAR
VAR_IN_OUT
  DATA: ARRAY [1 .. 1500] OF BYTE;
END_VAR

```

Graphic display:

Time behavior:

Function description:

This function block writes data to a remote system. If an error occurred, the output variable ERROR shows the error value.

If the socket is not connection-oriented, the destination address and destination port of the message are required (inputs DST_IP and DST_PORT).

See also: ETHERNET_CLIENT_OPEN, ETHERNET_SERVER_OPEN, ETHERNET_CLOSE, ETHERNET_READ

ETHERNET_WRITE_PT

WAGO-I/O-PRO 32 Library elements		
Category:	ETHERNET Fieldbus Communication	
Name:	ETHERNET_WRITE_PT	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Ethernet.lib (as from version 1.1)	
Applicable to:	Ethernet controller 750-842	
Input parameter:		
Data type:	Comments:	
EN	BOOL	FB execution control. The function block is executed on a rising edge at EN.
SOCKET	WORD	Socket descriptor. Used for identification of the socket
LEN_IN	WORD	Number of bytes to be written.
DST_IP	DWORD	Destination of the IP address. (Only used if not connected)
DST_PORT	WORD	Destination port. (Only used if not connected)
Output parameter:		
Data type:	Comments:	
ENO	BOOL	FB execution result. ENO is TRUE as long as EN is TRUE.
ERROR	ETH_ERROR	Error argument. 0: no errors found <>0: error in operation (see ETH_ERROR data type for details)
In/Output parameter:		
Data type:	Comments:	
DATA_PT	POINTER TO ARRAY (1 .. 1500] OF BYTE	Pointer to Array of sending Data.

Header:

```

FUNCTIONBLOCK ETHERNET_WRITE_PT
VAR_INPUT
    EN: BOOL;
    SOCKET: WORD;
    LEN_IN: WORD;
    DST_IP: DWORD;
    DST_PORT: WORD;
END_VAR
VAR_OUTPUT
    ENO: BOOL;
    ERROR: ETH_ERROR;
END_VAR
VAR_IN_OUT
    DATA_PT: POINTER TO ARRAY [1 .. 1500] OF BYTE;
END_VAR
    
```

Graphic display:

Time behavior:

Function description:

This data block writes data into a decentralized system. If an error occurs, the output variable ERROR displays the error value.

If the socket is not connection oriented, then the destination address and the destination port of the message are required (inputs DST_IP and DST_PORT).

The function block ETHERNET_WRITE_PT only differs with regards to the data transfer from the FB ETHERNET_WRITE. The data to be sent must not be copied due to the use of pointers. This therefore leads to an advantage in the processing speed. The general directions on the use of pointers in SPS programming are to be complied with.

See also: ETHERNET_CLIENT_OPEN, ETHERNET_SERVER_OPEN, ETHERNET_CLOSE, ETHERNET_READ, ETHERNET_READ_PT, ETHERNET_WRITE

ETH_ERROR (Data type)

WAGO-I/O-PRO 32 Library elements		
Category:	TCP/IP Sockets Communication	
Name:	ETH_ERROR	
Type:	Data type <input checked="" type="checkbox"/>	
Library name:	Ethernet.lib	
Applicable to:	All Ethernet function blocks	
Structure:	Enumeration	
Elements:	Value:	Comments:
NO_ERROR	0x0000	No error in operation
FNC_NOT_SUPPORTED	0x0001	Function block not supported by the underlying firmware
EIO	0x8001	Input/output error
ENXIO	0x8002	Device not configured
EINVAL	0x8003	Invalid argument
EALREADY	0x8004	Operation already in progress
EDESTADDRREQ	0x8005	Destination address required
EMSGSIZE	0x8006	Message too long
EPROTOTYPE	0x8007	Protocol wrong type for socket
ENOPROTOPT	0x8008	Protocol not available
EPROTONOSUPPORT	0x8009	Protocol not supported
ESOCKTNOSUPPORT	0x800A	Socket type not supported
EOPNOTSUPP	0x800B	Operation not supported
EAFNOSUPPORT	0x800C	Address family not supported by protocol family
EADDRINUSE	0x800D	Address already in use
EADDRNOTAVAIL	0x800E	Can't assign requested address
ENETDOWN	0x800F	Network is down
ENETUNREACH	0x8010	Network is unreachable
ENETRESET	0x8011	Network dropped connection on reset
ECONNABORTED	0x8012	Software caused connection to abort
ECONNRESET	0x8013	Connection reset by peer
ENOBUFS	0x8014	No buffer space available
EISCONN	0x8015	Socket is already connected
ENOTCONN	0x8016	Socket is not connected
ESHUTDOWN	0x8017	Can't send after socket shutdown
ETIMEDOUT	0x8018	Operation timed out
ECONNREFUSED	0x8019	Connection refused
EHOSTDOWN	0x801A	Host is down
EHOSTUNREACH	0x801B	No route to host
EACCES	0x801C	Access permission
ESRCH	0x801D	Search failed
EEXIST	0x801E	Route exists previously
EPIPE	0x801F	Broken communication channel
EWOULDBLOCK	0x8020	Communication should be blocking

ETH_ERROR (Data type)

ETOOMANYREFS	0x8021	Number of maximal multicast members reached
EDOM	0x8022	Value exceeds domain
ENOREF	0x8023	No reference to socket
SOCKAPI_NO_DESCRIPTOR	0xC001	No free docket descriptor for this open call
SOCKAPI_INVALID_IP	0xC002	Trying to configure an invalid IP address

Header:

```

TYPE ETH_ERROR :(
  NO_ERROR:=16#0000, (* no error *)
  FNC_NOT_SUPPORTED:=16#0001,(*the function is not supported by the firmware *)
  EIO:=16#8001,(* Input/output error *)
  ENXIO:=16#8002,(* Device not configured *)
  EINVAL:=16#8003,(* Invalid argument *)
  EALREADY:=16#8004,(* Operation already in progress *)
  EDESTADDRREQ:=16#8005,(* Destination address required *)
  EMSGSIZE:=16#8006,(* Message too long *)
  EPROTOTYPE:=16#8007,(* Protocol wrong type for socket *)
  ENOPROTOOPT:=16#8008,(* Protocol not available *)
  EPROTONOSUPPORT:=16#8009,(* Protocol not supported *)
  ESOCKTNOSUPPORT:=16#800A,(* Socket type not supported *)
  EOPNOTSUPP:=16#800B,(* Operation not supported *)
  EAFNOSUPPORT:=16#800C,(* Address family not supported by protocol family *)
  EADDRINUSE:=16#800D,(* Address already in use *)
  EADDRNOTAVAIL:=16#800E,(* Can't assign requested address *)
  ENETDOWN:=16#800F,(* Network is down *)
  ENETUNREACH:=16#8010,(* Network is unreachable *)
  ENETRESET:=16#8011,(* Network dropped connection on reset *)
  ECONNABORTED:=16#8012,(* Software caused connection abort *)
  ECONNRESET:=16#8013,(* Connection reset by peer *)
  ENOBUFS:=16#8014,(* No buffer space available *)
  EISCONN:=16#8015,(* Socket is already connected *)
  ENOTCONN:=16#8016,(* Socket is not connected *)
  ESHUTDOWN:=16#8017,(* Can't send after socket shutdown *)
  ETIMEDOUT:=16#8018,(* Operation timed out *)
  ECONNREFUSED:=16#8019,(* Connection refused *)
  EHOSTDOWN:=16#801A,(* Host is down *)
  EHOSTUNREACH:=16#801B,(* No route to host *)
  EACCES:=16#801C,(* access permission *)
  ESRCH:=16#801D,(* search failed *)
  EEXIST:=16#801E,(* route exists previously *)
  EPIPE:=16#801F,(* broken communication channel *)
  EWOULDBLOCK:=16#8020, (* communication should be blocking *)
  ETOOMANYREFS:=16#8021,(* number of maximal multicast members reached *)
  EDOM:=16#8022,(* value exceeds domain *)
  ENOREF:=16#8023,(* no reference to socket *)
  SOCKAPI_NODESCRIPTOR:=16#C001, (* no free socket descriptor for this open
  call *)
  SOCKAPI_INVALID_IP:=16#C002 (* Trying to configure an invalid IP address *));
END_TYPE

```

Data type description:

This data type is used by all ETHERNET function blocks except ETHERNET_GET_NETWORK_CONFIG and defines the error codes returned by this function blocks. This definitions are compatible to those in UNIX or Windows.

SEL_PROTOCOL (Data type)

WAGO-I/O-PRO 32 Library elements		
Category:	TCP/IP Sockets Communication	
Name:	SEL_PROTOCOL	
Type:	Data type <input checked="" type="checkbox"/>	
Library name:	Ethernet.lib	
Applicable to:	ETHERNET_CLIENT_OPEN and ETHERNET_SERVER_OPEN	
Structure:	Enumeration	
Elements:	Value:	Comments:
IPPROTO_IP	0	Together with a SOCK_RAW type specifies the default raw protocol, and it's the protocol used when no other match is found.
IPPROTO_ICMP	1	Internet Control Message Protocol. Together with a SOCK_RAW type allows to send ICMP messages to other devices.
IPPROTO_IGMP	2	Internet Group Management Protocol. Currently not supported.
IPPROTO_TCP	6	Transport Control Protocol. The semantics of communication must be SOCK_STREAM. The underlying TCP protocol is used.
IPPROTO_UDP	17	User Datagram Protocol. The semantics of communication must be SOCK_DGRAM. The underlying UDP protocol is used.
IPPROTO_RAW	255	Raw interface to IP. The semantics of communication must be SOCK_RAW. Similar to IPPROTO_IP but with a different protocol number.
Header:		
<pre> TYPE SEL_TYPE: (IPPROTO_IP: =0, IPPROTO_ICMP: =1, IPPROTO_IGMP: =2, IPPROTO_TCP: =6, IPPROTO_UDP: =17, IPPROTO_RAW: =255); END_TYPE </pre>		

Data type description:

This data type is used by ETHERNET_CLIENT_OPEN and ETHERNET_SERVER_OPEN function blocks and defines the transport protocol to use. This definitions are compatible to those in UNIX or Windows.

SEL_TYPE (Data type)

WAGO-I/O-PRO 32 Library elements		
Category:	TCP/IP Sockets Communication	
Name:	SEL_TYPE	
Type:	Data type <input checked="" type="checkbox"/>	
Library name:	Ethernet.lib	
Applicable to:	ETHERNET_CLIENT_OPEN and ETHERNET_SERVER_OPEN	
Structure:	Enumeration	
Elements:	Value:	Comments:
SOCK_STREAM	1	With this semantics for the communication sockets behave like full-duplex byte streams, similar to pipes.
SOCK_DGRAM	2	The socket is used to transport unreliable messages in a connectionless fashion (datagrams).
SOCK_RAW	3	With this type of sockets it is possible to access the internal network protocols.
Header:		
<pre> TYPE SEL_TYPE: (SOCK_STREAM: =1, SOCK_DGRAM: =2, SOCK_RAW: =3); END_TYPE </pre>		
Data type description:		
<p>This data type is used by ETHERNET_CLIENT_OPEN and ETHERNET_SERVER_OPEN function blocks and defines the semantics of communication. This definitions are compatible to those in UNIX or Windows.</p>		

