MBP_MSTR: Modbus Plus Master

12

Introduction

This chapter describes the MBP_MSTR block.

What's in this Chapter?

This chapter contains the following topics:

Block Description

Function Description

You can select one of 14 available network communication operations *(see page 100)* using the MBP_MSTR function block.

EN and ENO can be configured as additional parameters.

NOTE: You must be familiar with the routing procedures of your network when programming an MBP_MSTR function block. Modbus Plus routing path structures are described in detail in the *Modbus Plus Network Planning and Installation Guide*. If TCP/IP or SY/MAX Ethernet routing is implemented, standard Ethernet IP router products must be used. A full description of the TCP/IP routing is provided in the *Quantum with Unity Pro TCP/IP Configuration User Guide*.

CREAD_REG, CWRITE_REG, READ_REG, WRITE_REG and MBP_MSTR function blocks use 1 data transaction path and require multiple cycles to complete an operation. Number of transaction path available is dependent on the communication port used:

- Modbus Plus embedded port or NOM modules support up to 4 blocks at the same time
- Modbus Plus embedded port or NOM modules support up to 4 blocks at the same time
- Modbus Plus embedded port or NOM modules support up to 4 blocks at the same time

More communication function blocks may be programmed on the same communication port. However communication block exceeding the maximun number on that port will not be serviced until one of the transaction paths is freed up. When the transaction path resources become free the next block on the same port will become active and begin using freed path.

NOTE: In FBD and LD sections, this function block can be used on the program level and with derived function blocks (DFBs). When using DFBs, the parameters CONTROL and DATABUF must be directly connected to the I/O pins of the DFB.

NOTE: A TCP/IP communication between a Quantum PLC and a Momentum PLC is possible only when only one read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

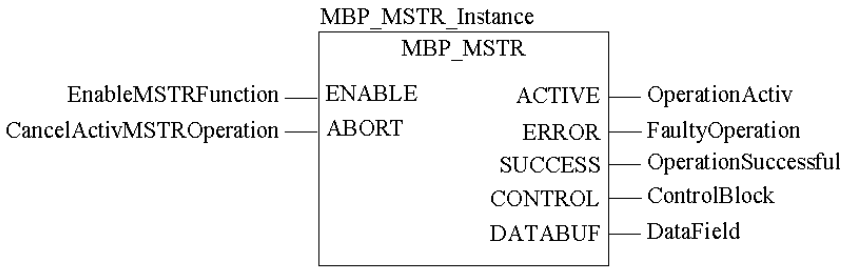
Example:

- you can send MBP_MSTR.Enable:=(HSBY_NOEPLCMSTR_ON) AND (%SW61.1) AND NOT (%SW61.0)
  or
- you can create a boolean variable, primary_state:=(%SW61.1) AND NOT (%SW61.0), and insert it for executing the section

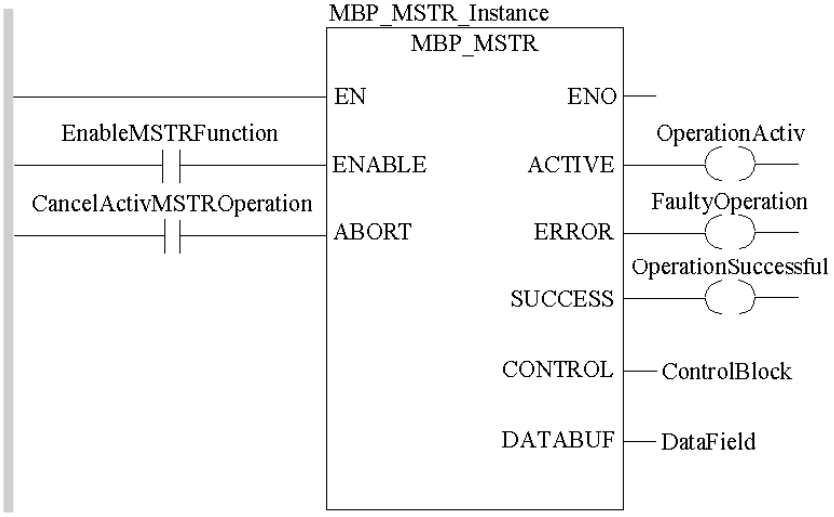
NOTE: To prevent the former standby CPU, which has switched its state for RUN offline from executing communication functions, you must add a condition on the status bits to disable the function, if the CPU is offline.

NOTE: Several copies of this function block can be used in the program. However, multiple instancing of these copies is not possible.

Representation in FBD



Representation in LD



Input Parameters

Parameter	Data Type	Description
ENABLE	BOOL	When ON, the operation specified in the first element of the CONTROL register is enabled.
ABORT	BOOL	When ON, the currently active operation *(see page 100)* is aborted.

Output Parameters

Parameter	Data Type	Description
ACTIVE	BOOL	ON when the operation is active.
ERROR	BOOL	ON when the operation is aborted without success.
SUCCESS	BOOL	ON when the operation concludes successfully.
CONTROL	WORD	This field contains the control block. The first element CONTROL[1] contains the number of the operation code of the operation to be performed *(see page 100)*. The content of the sequence register is determined by the operation. The data field must be declared as a located variable. The structure of the control block differs according to the network used *(see page 101)*.
DATABUF	WORD	For operations providing data, e.g. a write operation, the data field is the data source. For operations receiving data, e.g. the read operation, the data field is the data destination. With Ethernet CTE read and write operations, the data field holds the contents of the Ethernet configuration extension table. DATABUF must be defined as an array of at least 10 elements in this case. The data field must be declared as a located variable.

Runtime Error

In the event of an error occurring during an MBP_MSTR operation, a hexadecimal error code is displayed in the CONTROL[2] register of the control block for one cycle.

Function error codes are network-specific:

- Modbus Plus and SY/MAX Ethernet error codes *(see page 138)*
- SY/MAX specific error codes *(see page 142)*
- TCP/IP Ethernet error codes *(see page 144)*
- CTE Error Codes for SY/MAX and TCP/IP Ethernet *(see page 148)*
- Send e-mail error codes *(see page 149)*

NOTE: For a list of all block error codes and values, refer to the tables of error codes for the communication library.

Operational Function Codes

Valid MBP_MSTR Function Codes

Using the MBP_MSTR block, one of 13 available network communication operations can be triggered via the network. Each operation has a function code assigned to it. The availability of specific operations depends on both the type of network, and the type of module you are using.

Function Code	Operation	Modbus Plus	TCP/IP Ethernet	SY/MAX Ethernet	CIP Ethernet
1	Write data	X	X	X	-
2	Read data	X	X	X	-
3	Get local statistics	X	X	-	-
4	Clear local statistics	X	X	-	-
7	Get remote statistics	X	X	-	-
8	Clear remote statistics *(see page 113)*	X	X	-	-
10	Reset optional module	-	X	X	-
11	Read CTE (Config extension)	-	X	X	-
12	Write CTE (Config extension)	-	X	X	-
13	Send E-mail *(see page 120)*	-	X	-	-
14	CIP Explicit message *(see Quantum, 140 NOC 77100 EtherNet/IP Communication Module, User Manual)*	-	-	-	X
15	Send Modbus Request *(see page 122)*	-	X	-	-
16	Close Connection Request *(see page 126)*	-	X	-	-
23	Read / write data *(see page 127)*	-	X	-	-

where:

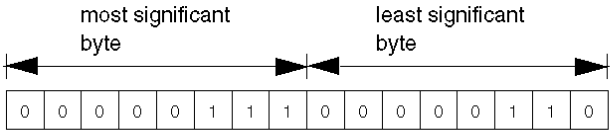
- X indicates *Yes*
- - indicates *No*

Network Control Block Structures

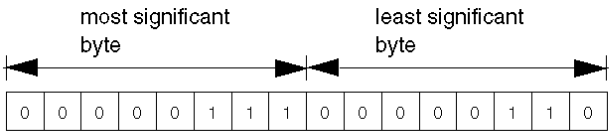
Summary

The structure of the MBP_MSTR control block varies according to the type of network you are using. Structures for Modbus Plus, TCP/IP Ethernet, and SyMax Ethernet are described below.

Control Block for Modbus Plus

Register	Contents
CONTROL[1]	Indicates an operation that is valid for Modbus Plus
CONTROL[2]	Indicates the error status
CONTROL[3]	Indicates the length, i.e., the number of data units transferred (max. 100)
CONTROL[4]	Indicates MSTR operation-dependent information
CONTROL[5]	Routing register 1: used to specify a destination node during network transfer (routing path addresses one of five) Most significant byte: source node address, i.e., the slot for the Modbus Plus Network Options Module (NOM) When using the Modbus Plus Port on the CPU, this byte must be set to 0 (regardless of the CPU slot). Least significant byte: destination node address, i.e., a value that represents a direct or a bridge address. If there is no bridge, this value contains the destination node address. If there is a bridge, this value contains the address of the bridge. If the NOM is inserted in slot 7 on the module rack, the most significant byte of routing register 1 looks as follows (value 0x0706):  **Most significant byte** Slots 1 ... 16 **Least significant byte** Destination address (binary value between 1 and 64 (normal) or 65 to 255 (extended))
CONTROL[6]	Routing register 2, the destination node address (further bridge or Modbus Plus modules). If addressing in the previous routing register has finished, the value is set to 0.
CONTROL[7]	Routing register 3, similar to routing register 2
CONTROL[8]	Routing register 4, similar to routing register 2 (see Routing Register 2)
CONTROL[9]	Routing register 5, similar to routing register 2 (see Routing Register 2)

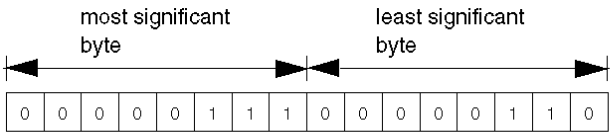
Control Block for TCP/IP Ethernet

Register	Contents
CONTROL[1]	Indicates an operation that is valid for TCP/IP
CONTROL[2]	Indicates the error status
CONTROL[3]	Indicates the length, i.e., the number of data units transferred (max. 100)
CONTROL[4]	Indicates MSTR operation-dependent information
CONTROL[5]	Routing register: used to specify a destination node during network transfer Most significant byte: source node address, i.e., the NOE slot for the NOE module When using an integrated Ethernet on the CPU, this byte must be set to 254 (FE hex) regardless of the CPU slot. Least significant byte: destination node address, i.e, a value that represents a direct or bridge address. If there is no bridge the value in the least significant byte is set to 0. If there is a bridge, this value contains the MBP for the Ethernet mapping index (MET). If the NOE is inserted in slot 7 on the module rack and the Ethernet mapping index (MET) is 6, the routing register looks as follows (value 0x0706):  **Most significant byte** Slots 1 ... 16 **Least significant byte** MBP on Ethernet Transporter (MET) mapping index
CONTROL[6]	Byte 4, MSB of the 32-bit destination IP address
CONTROL[7]	Byte 3 of the 32-bit destination IP address
CONTROL[8]	Byte 2 of the 32-bit destination IP address
CONTROL[9]	Byte 1, LSB of the 32-bit destination IP address
CONTROL[10]	Indicates MSTR operation-dependent information
CONTROL[11]	Indicates MSTR operation-dependent information

NOTE: CONTROL[10] and CONTROL[11] are used when configuring the MBP_MSTR block for a Read/Write Data operation (function code 23).

Control Block for SY/MAX Ethernet

Register	Contents
CONTROL[1]	Indicates an operation that is valid for SY/MAX
CONTROL[2]	Indicates the error status
CONTROL[3]	Indicates the length, i.e., the number of registers transferred (max. 100)

Register	Contents
CONTROL[4]	Indicates MSTR operation-dependent information
CONTROL[5]	Routing register: used to specify a destination node during network transfer Most significant byte: source node address, ie.e, the slot for the NOE module Least significant byte: destination node address, i.e, a value that represents a direct or bridge address. If there is no bridge the value in the least significant byte is set to 0. If there is a bridge, this value contains the MBP for the Ethernet mapping index (MET). If NOM is inserted in slot 7 on the module rack and the Ethernet mapping index (MET) is 6, the routing register looks as follows (value 0x0706):  **Most significant byte** Slots 1 ... 16 **Least significant byte** MBP on Ethernet Transporter (MET) mapping index
CONTROL[6]	Destination drop number (or set to FF hex)
CONTROL[7]	Terminator (set to FF hex)

Read Data

Description

A read operation transfers data from a specified slave source device to a master destination device on the network. It uses a master transaction path and may require several cycles to complete. To program an MBP_MSTR block to perform a write operation, use function code 2 *(see page 100)*.

NOTE: Do not attempt to program an MBP_MSTR to read to its own station address. This action causes the function block to generate an error in the CONTROL[2] register of the control block *(see page 101)*.

You can perform a read operation on a nonexistent slave register. The slave detects the status and logs it. This can last for several cycles.

Network Implementation

The read operation can be performed on Modbus Plus, TCP/IP Ethernet, and SY/MAX Ethernet networks.

Control Block Usage for Modbus Plus

Register	Meaning
CONTROL[1]	2 = Read data
CONTROL[2]	Indicates the error status.

Register	Meaning
CONTROL[3]	Number of registers to be read from the slave
CONTROL[4]	Determines the %MW starting register in the slave from which the data is read, e.g., 1 = %MW1, 49 = %MW49).
CONTROL[5] ... CONTROL[9]	Routing register 1 is used to specify the address (routing path address 1 of 5) of the node during a network transfer. The last byte in the routing path that is not 0 is the destination node.

Control Block Usage for TCP/IP Ethernet

Register	Meaning
CONTROL[1]	2 = read data
CONTROL[2]	Indicates the error status.
CONTROL[3]	Number of addresses to be read from the slave
CONTROL[4]	Determines the %MW starting register in the slave from which the data is read, e.g., 1 = %MW1, 49 = %MW49)
CONTROL[5]	Routing register: Most significant byte: network adapter module slot Least significant byte: MBP on Ethernet Transporter (MET) mapping index
CONTROL[6] ... CONTROL[9]	Each address contains 1 byte of the 32-bit IP address, where the MSB is in CONTROL[6] and the LSB is in CONTROL[9].

Control Block Usage for SY/MAX Ethernet

Register	Meaning
CONTROL[1]	2 = Read data
CONTROL[2]	Indicates the error status.
CONTROL[3]	Number of addresses to be read from the slave
CONTROL[4]	Determines the %MW starting register in the slave to which the data is written, e.g., 1 = %MW1, 49 = %MW49).
CONTROL[5]	Routing register Most significant byte: network adapter module slot Least significant byte: destination drop number
CONTROL[6] ... CONTROL[9]	Terminator: FF hex

Write Data

Description

A write operation transfers data from a master source device to a specified slave destination device on the network. It uses a master transaction path and may require several cycles to complete. To program an MBP_MSTR block to perform a write operation, use function code 1 *(see page 100)*.

NOTE: Do not attempt to program an MBP_MSTR to write to its own drop address. This action causes the function block to generate an error in the CONTROL[2] register of the control block *(see page 101)*.

You can perform a write operation to a nonexistent slave register. The slave detects the status and logs it. This can last for several cycles.

Network Implementation

The write operation can be performed on Modbus Plus, TCP/IP Ethernet, and SY/MAX Ethernet networks.

Control Block Usage for Modbus Plus

Register	Meaning
CONTROL[1]	1 = write data
CONTROL[2]	Indicates the error status
CONTROL[3]	Number of addresses sent to the slave
CONTROL[4]	Determines the %MW starting register in the slave to which the data is written, e.g. 1 = %MW1, 49 = %MW49
CONTROL[5] ... CONTROL[9]	Routing register 1 is used to specify the address (routing path address 1 of 5) of the node during a network transfer. The last byte in the routing path that is not 0 is the destination node.

Control Block Usage for TCP/IP Ethernet

Register	Meaning
CONTROL[1]	1 = write data
CONTROL[2]	Indicates the error status
CONTROL[3]	Number of addresses sent to the slave
CONTROL[4]	Determines the CONTROL[ ] start address of the slave to which the data is written
CONTROL[5]	Routing register Most significant byte: network adapter module slot Least significant byte: MBP on Ethernet transporter (MET) mapping index

Register	Meaning
CONTROL[6] ... CONTROL[9]	Each address contains 1 byte of the 32-bit IP address.

Control Block Usage for SY/MAX Ethernet

Register	Meaning
CONTROL[1]	1 = Write data
CONTROL[2]	Indicates the error status
CONTROL[3]	Number of addresses sent to the slave
CONTROL[4]	Determines the %MW starting register in the slave to which the data is written, e.g., 1 = %MW1, 49 = %MW49)
CONTROL[5]	Routing register Most significant byte: network adapter module slot Least significant byte: destination drop number
CONTROL[6] ... CONTROL[9]	Terminator: FF hex

Get Local Statistics

Description

A `Get Local Statistics` operation reads the data from the local node in one cycle and does not require a master transaction path.To program an `MBP_MSTR` block to `Get Local Statistics`, use function code 3 *(see page 100)*.

Network Implementation

A `Get Local Statistics` operation can be performed on Modbus Plus and TCP/IP Ethernet networks *(see page 135)*.

Control Block Usage for Modbus Plus

Register	Meaning
CONTROL[1]	3 = get local statistics
CONTROL[2]	Indicates the error status
CONTROL[3]	Number of addresses to be read from local statistics (0...37) **Note:** The size of databuf must be at least the size of this entry.
CONTROL[4]	First address from which the statistics table must be read (Reg1=0)
CONTROL[5]	Routing register 1 is used to specify the address (routing path address 1 of 5) of the node during a network transfer. The last byte in the routing path that is not 0 is the destination mode.

Control Block Usage for TCP/IP Ethernet

Register	Meaning
CONTROL[1]	3 = get local statistics
CONTROL[2]	Indicates the error status
CONTROL[3]	Number of addresses to be read from local statistics (0...37) **Note:** The size of databuf must be the size of this entry.
CONTROL[4]	First address from which the statistics table must be read (Reg1=0)
CONTROL[5]	Routing register Most significant byte: Network adapter module slot
CONTROL[6] ... CONTROL[9]	Not used

Clear Local Statistics

Description

A clear local statistics operation clears the values from words 13 ... 22 in the statistics table of the local node. The operation is carried out in one cycle and does not require a master transaction path. To program an `MBP_MSTR` block to clear local statistics, use function code 4 *(see page 100)*.

Network Implementation

A clear local statistics operation can be performed on Modbus Plus and TCP/IP Ethernet networks *(see page 135)*.

Control Block Usage for Modbus Plus

Register	Meaning
CONTROL[1]	4 = clear local statistics
CONTROL[2]	Indicates the error status
CONTROL[3]	Reserved
CONTROL[4]	Reserved
CONTROL[5]	Routing register 1 is used to specify the address (routing path address 1 of 5) of the node during a network transfer. The last byte in the routing path that is not 0 is the destination mode.

Control Block Usage for TCP/IP Ethernet

Register	Meaning
CONTROL[1]	4 = clear local statistics
CONTROL[2]	Indicates the error status
CONTROL[3]	Reserved
CONTROL[4]	Reserved
CONTROL[5]	Routing register Most significant byte: network adapter module slot
CONTROL[6] ... CONTROL[9]	Reserved

Write Global Data

Description

A write global data operation transfers data to the communication processor of the current node; the data can be transmitted on the network as soon as the node receives the token and then read by all nodes connected to the local network *(see page 110)* .

A write global data operation is carried out in one cycle and does not require a master transaction path. To program an `MBP_MSTR` block to write global data, use function code 5 *(see page 100)*.

Network Implementation

A write global data operation can be performed only on Modbus Plus networks. The read and write global data operations comprise a Modbus Plus capability known as *Peer Cop*.

Control Block Usage for Modbus Plus

Register	Meaning
CONTROL[1]	5 = write global data
CONTROL[2]	Indicates the error status
CONTROL[3]	Number of addresses to be written from state RAM into global data memory (comm processor) (1...32)
CONTROL[4]	Reserved
CONTROL[5]	If global data is sent via a NOM, enter the NOM module slot in the most significant byte of this register.

Read Global Data

Description

A read global data operation reads data from the communications processor of a node on the network that has written global data *(see page 109)*. A master transaction path is not required.

A read global data operation can take several cycles if the global data is not currently available with the nodes called. If global data is available, the operation is executed in one cycle. To program an `MBP_MSTR` block to write global data, use function code 6 *(see page 100)*.

Network Implementation

A read global data operation can be performed only on Modbus Plus networks. The read and write global data operations comprise a Modbus Plus capability known as *Peer Cop*.

Control Block Usage for Modbus Plus

Register	Meaning
CONTROL[1]	6 = read global data
CONTROL[2]	Indicates the error status
CONTROL[3]	Number of addresses to be sent from global data memory (comm processor) (1...32)
CONTROL[4]	Displays the addresses available in the scanned node. (This register Is automatically updated.)
CONTROL[5]	The least significant byte contains the address of the node whose global data is to be read. It can be a value in the range 1 ... 64. If global data is received via a NOM, enter the NOM module slot in the most significant byte of this address.

Get Remote Statistics

Description

A get remote statistics operation can be used to read data from remote nodes on the network. With each query, the remote communications processor supplies a complete table of statistics even if the query does not refer to the entire table. It then copies only the words that you queried into identified $MW addresses.

An operation can take several cycles to complete; it does not require a master data transaction path. To program an `MBP_MSTR` block to get remote statistics, use function code 7 *(see page 100)*.

Network Implementation

A get remote statistics operation can be performed on Modbus Plus and TCP/IP Ethernet networks.

Control Block Usage for Modbus Plus

Register	Meaning
CONTROL[1]	7 = get remote statistics
CONTROL[2]	Indicates the error status
CONTROL[3]	Number of addresses to be read from the statistics data field (0 ... 38). **Note:** The size of `databuf` must be at least the size of this entry.
CONTROL[4]	First address from which the node statistics must be read. The number of available statistics registers cannot be exceeded.
CONTROL[5] ... CONTROL[9]	Routing address 1 ... 5 of the node. The last byte in the routing path that is not 0 is the destination node.

Control Block Usage for TCP/IP Ethernet

Register	Meaning
CONTROL[1]	7 = get remote statistics
CONTROL[2]	Indicates the error status
CONTROL[3]	Number of addresses to be read from the statistics data field (0 ... 38). **Note:** The size of `databuf` must be at least the size of this entry.
CONTROL[4]	First address from which the node statistics must be read. The number of available statistics registers cannot be exceeded.
CONTROL[5]	Routing register Most significant byte: network adapter module slot

Register	Meaning
CONTROL[6] ... CONTROL[9]	Each address contains 1 byte of the 32-bit IP address, where the value in CONTROL[6] is the MSB and the value in CONTROL[9] is the LSB.

Clear Remote Statistics

Description

A clear remote statistics operation clears remote-node values from words 13 ... 22 in the statistics table of the local node. It uses a master transaction path and may require several cycles to complete. To program an MBP_MSTR block to perform a clear remote statistics operation, use function code 8 *(see page 100)*.

Network Implementation

A clear remote statistics operation can be performed on Modbus Plus and TCP/IP Ethernet networks *(see page 135)*.

Control Block Usage for Modbus Plus

Register	Meaning
CONTROL[1]	8 = clear remote statistics
CONTROL[2]	Indicates the error status
CONTROL[3]	Reserved
CONTROL[4]	Reserved
CONTROL[5] ... CONTROL[9]	Routing register 1 is used to specify the address (routing path address 1 of 5) of the destination node during a network transfer. The last byte in the routing path that is not 0 is the destination mode.

Control Block Usage for TCP/IP Ethernet

Register	Meaning
CONTROL[1]	8 = clear remote statistics
CONTROL[2]	Indicates the error status
CONTROL[3]	Reserved
CONTROL[4]	Reserved
CONTROL[5]	Routing Register Most significant byte: network adapter module slot
CONTROL[6] ... CONTROL[9]	Each address contains one byte of the 32-bit IP address, where the MSB is in CONTROL[6] and the LSB is in CONTROL[9].

Peer Cop Health

Description

A peer cop health operation reads selected data from the peer cop communications health table and downloads the data to the specified %MW addresses in state RAM. To program an MBP_MSTR block to perform a clear remote statistics operation, use function code 9 *(see page 100)*.

NOTE: Peer cop health is operational only when a peer cop-based I/O scanner has been configured.

The peer cop communications health table is 12 words long; `MBP_MSTR` indexes these words with the numbers 0 ... 11.

Network Implementation

A peer cop health operation can be performed only on Modbus Plus networks.

Control Block Usage for Modbus Plus

Register	Meaning
CONTROL[1]	9 = peer cop health
CONTROL[2]	indicates the error status.
CONTROL[3]	Number of words wanted by the peer cop table (1...12)
CONTROL[4]	First word to be read from the peer cop table, where 0 = the first word and 11 = the last word)
CONTROL[5]	Routing address 1 If this is the second of two local nodes, set the value in the high byte to 1.

Reset Optional Module

Description

A reset optional module operation causes a Quantum NOE Ethernet communications module or the Ethernet port on a 140CPU65150/60 CPU module to enter a cycle that resets its working environment. To program an MBP_MSTR block to perform a reset option module operation, use function code 10 *(see page 100)*.

Network Implementation

A reset optional module operation can be performed on TCP/IP Ethernet *(see page 135)* and SY/MAX Ethernet networks.

Control Block Usage for TCP/IP Ethernet

Register	Meaning
CONTROL[1]	10 = reset optional module
CONTROL[2]	Indicates the error status
CONTROL[3]	No significance
CONTROL[4]	No significance
CONTROL[5]	Routing register The value shown in the high byte in area 1 through 16 indicates the slot in the Quantum backplane where the NOE module is located.
CONTROL[6] ... CONTROL[9]	No significance

Control Block Usage for SY/MAX Ethernet (CONTROL)

Register	Meaning
CONTROL[1]	10 = reset optional module
CONTROL[2]	Indicates the error status
CONTROL[3]	No significance
CONTROL[4]	No significance
CONTROL[5]	Routing register MSB: network adapter module slot
CONTROL[6] ... CONTROL[9]	No significance

Read CTE

Description

A read CTE operation reads a specified number of bytes from the Ethernet configuration extension table in the specified buffer of PLC memory. The bytes to be read start with a byte offset at the start of the CTE table. The contents of the CTE table are displayed in the DATABUF output parameter. *(see page 99)*To program an MBP_MSTR block to perform a clear remote statistics operation, use function code 11 *(see page 100)*.

Network Implementation

A read CTE operation can be performed on TCP/IP Ethernet and SY/MAX Ethernet networks.

Control Block Usage for TCP/IP Ethernet

Register	Meaning
CONTROL[1]	11 = read CTE
CONTROL[2]	Indicates the error status
CONTROL[3]	The length setting: a value from 12 to 37
CONTROL[4]	No significance
CONTROL[5]	Routing register Least significant byte = mapping index Either a value displayed in the byte of the register or is not used. or Most significant byte = network adapter module slot
CONTROL[6] ... CONTROL[9]	No significance

Control Block Usage for SY/MAX Ethernet

Register	Meaning
CONTROL[1]	11 = read CTE
CONTROL[2]	Indicates the error status
CONTROL[3]	Number of words transferred
CONTROL[4]	Byte offset in the PLC register structure, specifying from where the CTE bytes are read

Register	Meaning
CONTROL[5]	Routing register MSB: slot of the NOE module
CONTROL[6] ... CONTROL[9]	Terminator: FF hex

CTE Indicator Implementation (DATABUF)

The values in the CTE table are displayed in the DATABUF output when a CTE read operation is implemented. The registers display the following CTE data:

CTE indicator implementation (DATABUF):

Parameter	Register	Contents
Frame type	DATABUF[0]	1 = 802.3 2 = Ethernet
IP address	DATABUF[1]	First byte of the IP address
	DATABUF[2]	Second byte of the IP address
	DATABUF[3]	Third byte of the IP address
	DATABUF[4]	Fourth byte of the IP address
Lower netmask	DATABUF[5]	Most significant word
	DATABUF[6]	Least significant word
Gateway	DATABUF[7]	First byte of the gateway
	DATABUF[8]	Second byte of the gateway
	DATABUF[9]	Third byte of the gateway
	DATABUF[10]	Fourth byte of the gateway

Write CTE

Description

A write CTE operation writes the CTE configuration table from the specified data (DATABUF) to a specified Ethernet configuration extension table or to a specific slot. To program an MBP_MSTR block to perform a write CTE operation, use function code 12 *(see page 100)*.

Network Implementation

A write CTE operation can be performed on TCP/IP Ethernet *(see page 135)* and SY/MAX Ethernet networks.

Control Block Usage for TCP/IP Ethernet

Register	Meaning
CONTROL[1]	12 = write CTE
CONTROL[2]	Indicates the error status
CONTROL[3]	The length setting: a value from 12 to 37.
CONTROL[4]	No significance
CONTROL[5]	Routing register Least significant byte = mapping index Either a value displayed in the byte of the address or is not used. or Most significant byte = network adapter module slot
CONTROL[6] ... CONTROL[9]	No significance

Control Block Usage for SY/MAX Ethernet

Register	Meaning
CONTROL[1]	12 = Write CTE (Config extension table)
CONTROL[2]	Indicates the error status
CONTROL[3]	Number of words transferred
CONTROL[4]	Byte offset in the PLC address structure specifying where the CTE bytes are written
CONTROL[5]	Routing register Most significant byte = NOE module slot Least significant byte = Destination drop number

Register	Meaning
CONTROL[6]	Terminator: FF hex
CONTROL[7] ... CONTROL[9]	No significance

CTE Indicator Implementation (DATABUF)

The values in the Ethernet configuration extension table are displayed in the DATABUF output field when a write CTE operation is implemented. The registers are used to transfer the following CTE data:

CTE indicator implementation (DATABUF):

Parameter	Register	Contents
Frame type	DATABUF[0]	1 = 802.3 2 = Ethernet
IP address	DATABUF[1]	First byte of the IP address
	DATABUF[2]	Second byte of the IP address
	DATABUF[3]	Third byte of the IP address
	DATABUF[4]	Fourth byte of the IP address
Lower netmask	DATABUF[5]	Most significant word
	DATABUF[6]	Least significant word
Gateway	DATABUF[7]	First byte of the gateway
	DATABUF[8]	Second byte of the gateway
	DATABUF[9]	Third byte of the gateway
	DATABUF[10]	Fourth byte of the gateway

Send Email

Description

The electronic mail notification service allows controller-based projects to report alarms or events. The controller monitors the system and dynamically creates an electronic mail message, which alerts local or remote users.

A user-defined event or condition triggers the MSTR block to create a message. Each message uses one of three user-defined headers. Each message sent from the controller can contain text and variable information (with a maximum of 238 bytes).

The project selects the appropriate header. Each header contains:
- sender name
- list of recipients
- subject

To program an MBP_MSTR block to send email, use function code 13 *(see page 100)*.

Network Implementation

A send email operation can be performed on a TCP/IP Ethernet network.

Control Block Usage for TCP/IP Ethernet

Register	Meaning
CONTROL[1]	13 = send Email
CONTROL[2]	Indicates the email-specific error codes *(see page 149)*
CONTROL[3]	Number of words transferred
CONTROL[4]	Not used
CONTROL[5]	High byte: slot address of the NOE module or 0xFE for the 140 CPU 651 60
	Low byte: always 0
CONTROL[6] ... CONTROL[9]	Not used

DATABUF Parameter Description

Register	Contents
DATABUF 1	The mail header is the least significant byte with a value of 1, 2, or 3.
	The most significant byte contains the number (*n*) of characters in the subject, a value between 0 and 238.
DATABUF 2 ... DATABUF 119	The data (in ASCII format) that will be copied into the Email message. The first *n* characters are added to the configured Email subject. The remaining characters (2 * *N* -2 - *n*) are part of the message body, where *N* is the number of words transferred.

Send Modbus Request

At a Glance

Use MSTR operation 15 to send generic Modbus requests on the network.

NOTE: This operation is not available on Modbus Plus ports (embedded port on CPU or NOM modules) and the embedded Ethernet port on a CPU.

Block Operation

The MBP_MSTR block can send requests and receive responses up to 253 bytes long.

For the operation, refer to Block Operation *(see page 97)*:

- When the ENABLE input pin is turned ON, operation 15 begins.
- If the ABORT input pin is turned ON or if the ENABLE input pin is turned OFF, the operation ends.
- The ACTIVE output pin is ON during the operation.
- The ERROR output pin turns ON if the operation aborts without success.
- The SUCCESS output pin turns ON if the operation completes with success.
- The CONTROL and DATABUF output pins define the operation (refer to the Control Block *(see page 122)* and Data Buffer *(see page 123)*).
- EN and ENO can be configured as additional parameters.

Control Block

The format of the Control block is described in the following table:

Word	Description
CONTROL[1]	15 = Send Modbus Request
CONTROL[2]	Indicates detected error status
CONTROL[3]	DATABUF length (WORDs)
CONTROL[4]	Offset for the beginning of the response in the DATABUF (WORDs). **NOTE:** To avoid overwriting the request, the Response Offset value multiplied by 2 must be greater than the Request Length (CONTROL[10]).
CONTROL[5]	Routing register: High byte = Ethernet communication module slot Low byte = MBP on Ethernet transporter (MET) mapping index (also known as Unit ID)
CONTROL[6]	Byte 4 of the IP address (MSB)
CONTROL[7]	Byte 3 of the IP address
CONTROL[8	Byte 2 of the IP address
CONTROL[9]	Byte 1 of the IP address (LSB)
CONTROL[10]	Length of the DATABUF Request data (bytes)

Word	Description
CONTROL[11]	Length of the DATABUF Response received (bytes) **NOTE:** This is read only, it is set by option module after operation completion.

Data Buffer

The MODBUS protocol defines a simple protocol data unit (PDU) independent of the underlying communication layers.

The data buffer (DATABUF) consists of contiguous registers that include both the Modbus Request PDU and the Modbus Response PDU:

DATABUF *Data Buffer Length* is set in the CONTROL[3] word.	**Modbus Request** PDU: *Data Request Length* is set in the CONTROL[10] word.
	Modbus Response PDU: Data Response Start (*Response offset*) is set in the CONTROL[4] word. **NOTE:** To avoid overwriting the request, the Response Offset value multiplied by 2 must be greater than the Request Length (CONTROL[10]). *Data Response Length* is set in the CONTROL[11] word.

⚠CAUTION

LOSS OF DATA

Verify that the *Response Offset* is greater than the *Data Request Length*.

Failure to follow these instructions can result in injury or equipment damage.

NOTE: Both the Request and Response must be structured in little endian order. Each word of 2 bytes of data in little endian format, where the least significant byte is stored in the smallest memory address.

Modbus Request

The **Modbus Request** PDU format is:

Byte offset	Field	Data type	Description
1	Function code	BYTE	Modbus function code
2	Request data	BYTE area	This field is function code dependent and contains information such as variable references, variable counts, data offsets, sub-function codes and so on.

Modbus Response

The **Modbus Response** PDU format is:

Byte offset	Field	Data type	Description
1	Function code	BYTE	Modbus function code
2	Response data	BYTE area	This field is function code dependent and contains information such as variable references, variable counts, data offsets, sub-function codes and so on.

Modbus Exception Response

The **Modbus Exception Response** PDU format is:

Byte offset	Field	Data type	Description
1	Function code	BYTE	Modbus function code + 80 (hex)
2	Exception code	BYTE	This field is defined in the *MODBUS Application Protocol Specification*.

Modbus Specification

The standard Modbus function codes are defined in the *MODBUS Application Protocol Specification*, V1.1b, which can be downloaded from www.modbus-ida.org.

Example of a Send Modbus Request

To read 4 contiguous Input registers (Modbus Function code 4) starting at register 100 (64 hex) in a remote device. A **Modbus Request** has to be sent on the network with the following **Modbus Response**.

Request Field Name	Value
Function code	04
Starting address, High	00
Starting address, Low	64
Number of registers, High	00
Number of registers, Low	04

Response Field Name	Value
Function code	04
Byte count	08
Register 100 value, High	00
Register 100 value, Low	01

Response Field Name	Value
Register 101 value, High	00
Register 101 value, Low	02
Register 102 value, High	00
Register 102 value, Low	03
Register 103 value, High	00
Register 103 value, Low	04

Using the following CONTROL word values:
- CONTROL[1] (MSTR operation) = 15
- CONTROL[2] (Error Code) = refer to CONTROL block description (read-only)
- CONTROL[3] (DATABUF length) = 11
- CONTROL[4] (Response Offset) = 5
- CONTROL[5] (Routing Register) = refer to CONTROL block description
- CONTROL[6] (IP1) = refer to CONTROL block description
- CONTROL[7] (IP1) = refer to CONTROL block description
- CONTROL[8] (IP1) = refer to CONTROL block description
- CONTROL[9] (IP1) = refer to CONTROL block description
- CONTROL[10] (Request Length) = 5
- CONTROL[11] (Response Length) = 10

the data encoding in the Data Buffer is as follows:

	Field	Value (hex)	Description
Request	DATABUF[1]	0400	Function code + Starting address, High
	DATABUF[2]	6400	Starting address, Low + Number of registers, High
	DATABUF[3]	0400	Number of registers, Low + NULL
	DATABUF[4]	0000	Null
	DATABUF[5]	0000	Null
Response	DATABUF[6]	0408	Function code + Byte count
	DATABUF[7]	0001	Register 100 value, High + Register 100 value, Low
	DATABUF[8]	0002	Register 101 value, High + Register 101 value, Low
	DATABUF[9]	0003	Register 102 value, High + Register 102 value, Low
	DATABUF[10]	0004	Register 103 value, High + Register 103 value, Low
	DATABUF[11]	0000	Null

Close Connection Request

At a Glance

Use MBP_MSTR operation 16 to close a Modbus TCP/IP connection on the network.

NOTE: This operation is not available on Modbus Plus ports (embedded port on CPU or NOM modules).

Block Operation

The MBP_MSTR block can send requests and receive responses up to 253 bytes long. For block operation, refer to the block description *(see page 97)*:

- When the ENABLE input pin is turned ON, operation 15 begins.
- If the ABORT input pin is turned ON or if the ENABLE input pin is turned OFF, the operation ends.
- The ACTIVE output pin is ON during the operation.
- The ERROR output pin turns ON if the operation aborts without success.
- The SUCCESS output pin turns ON if the operation completes with success.
- The CONTROL and DATABUF output pins *(see page 99)* define the operation.
- EN and ENO can be configured as additional parameters.

Control Block

The format of the CONTROL block is described below:

Register	Function	Description
CONTROL[1]	Operation	16 = close connection
CONTROL[2]	Error status	Indicates detected error status *(see page 144)* (read only)
CONTROL[3]	(not used)	–
CONTROL[4]	(not used)	–
CONTROL[5]	Routing Register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index (also known as Unit ID)
CONTROL[6]	IP Address	Byte 4 of the IP address (MSB)
CONTROL[7]		Byte 3 of the IP address
CONTROL[8]		Byte 2 of the IP address
CONTROL[9]		Byte 1 of the IP address (LSB)

Read/Write Data

Introduction

In a single transaction, the MSTR read and write operations can transfer data from a master source device to a specified slave destination device, then transfer data from this specified slave source to the master. It uses a master transaction path and may require several cycles to complete. To program an MBP_MSTR block to perform a combined read/write operation, use function code 23 *(see page 100)*.

The combined read/write operation can be used only with these two Quantum models:

- NOE 771 01 (version 3.0 or later)
- NOE 771 11 (version 3.0 or later)

Control Block Usage

Register	Content
CONTROL[1]	23 = read/write data.
CONTROL[2]	Indicates the error status.
CONTROL[3]	Number of registers to be sent to the slave.
CONTROL[4]	Specifies the %MW starting register in the slave to which the data will be written, e.g. 1 = %MW1, 49 = %MW49.
CONTROL[5]	Routing register: Most significant byte: network adapter module slot. Least significant byte: MBP on Ethernet Transporter (MET) mapping index.
CONTROL[6] ... CONTROL[9]	Each address contains 1 byte of the 32-bit IP address, where the MSB is in CONTROL[6] and the LSB is in CONTROL[9].
CONTROL[10]	Number of registers to be read from slave.
CONTROL[11]	Specifies the %MW starting register in the slave from which the data is read, e.g. 1 = %MW1, 49 = %MW49.

NOTE:

When configuring the MBP_MSTR block for a read/write data operation, note that

- The DATABUF output parameter is used to store, in the following sequence, both:
  1 the data to be written
  2 the data to be read
- The size of the DATABUF output parameter must equal the combined size of the data to be written and the data to be read; if the size is smaller, data will be overwritten and may be lost.
- Both the CONTROL and DATABUF parameters must be stored at located addresses, for example %MW addresses.

Peer Cop Communications Health Status

Peer Cop Communications Health Status

The table containing Peer Cop status information fills 12 contiguous registers indexed with the numbers 0 ... 11 in an `MBP_MSTR` operation. Each individual bit in the table words is used to present one aspect of communications health for a specific node on the Modbus Plus network. To program an MBP_MSTR block to get Peer Cop health status, use function code 9 *(see page 100)*.

Network Implementation

A Peer Cop communications health status operation can be performed only on Modbus Plus networks.

Relation Bit Network Node

The bits of the words 0 to 3 represent the health at the global communications input of nodes 1 to 64. The bits of words 4 ... 7 represent the health of the output of a specific node.

The bits in words 8 to 11 represent the health of the input of a specific node.

Status type	Word index	Relation bit network node
Global receive	0	16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
	1	32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17
	2	48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33
	3	64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49
Send direct	4	16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
	5	32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17
	6	48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33
	7	64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49

Status type	Word index	Relation bit network node
Receive direct	8	16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
	9	32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17
	10	48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33
	11	64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49

Health Bit Status

The status of the Peer Cop health bit indicates the current communications status of its assigned node. A health bit is set when the associated node accepts input for its Peer Cop data block or when it receives a signal that another node has accepted specific output data from its Peer Cop output data block. A health bit is deleted if the associated data block does not accept any communication within the configured Peer Cop health timeout period.

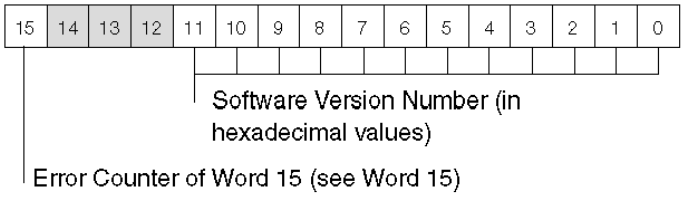
All health bits are deleted when interface command **Put Peer Cop** is executed during PLC startup. The table values become valid when the token is completely bypassed, after the **Put Peer Cop** command has been carried out. The health bit of a specific node is always 0 when the assigned Peer Cop entry is 0.

Modbus Plus Network Statistics

Modbus Plus Network Statistics

The following table shows the statistics available on Modbus Plus. You can obtain this data by running the corresponding `MBP_MSTR` operation (Modbus function code 8).

NOTE: If you edit the clear local statistics *(see page 108)* or clear remote statistics *(see page 113)* operation, only words 13 to 22 in the statistics table are cleared.

Word	Bits	Description
00	Node type ID	
	0	Unknown node type
	1	PLC node
	2	Modbus bridge node
	3	Host computer node
	4	Bridge Plus node
	5	Peer I/O node
	6 ... 15	Reserved
01	0 ... 11	Software version number as hexadecimal value (to read this, isolate bits 12-15 from the word)
	12 ... 14	Reserved
	15	Defines error counters from word 15. The most significant bit defines the use of error counters in word 15. The lower valued half of the most significant byte together with the least significant byte contain the software version. 
02		Network address of this station

Word	Bits	Description
03	MAC status variable:	
	0	Startup status
	1	Offline status indicator signals
	2	Duplicated offline status
	3	Idle status
	4	Token utilization status
	5	Work response status
	6	Token transfer status
	7	Response request status
	8	Status check of transfer
	9	Token request status
	10	Response request status
04	Peer status (LED code); indicates status of this device relative to the network:	
	0	Monitor connect operation
	32	Normal connect operation
	64	Never receives token
	96	Single station
	128	Duplicate station
05	Token transfer counter; increments each time this station receives the token	
06		Token cycle time in ms
07	LOW	Bit representation data master fail during token ownership
	HIGH	Bit representation (bitmap) program master fail during token ownership
08	LOW	Bitmap activity token ownership of the data master
	HIGH	Bitmap activity token ownership of the program master
09	LOW	Bitmap activity token ownership of the data slave
	HIGH	Bitmap activity token ownership of the program slave
10	LOW	
	HIGH	Bitmap transfer request command data slave/slave poll
11	LOW	Bitmap response transfer request program master/master poll
	HIGH	Bitmap transfer request command program slave/slave poll
12	LOW	Bitmap connect status of the program master
	HIGH	Bitmap automatic log-off of program slave
13	LOW	Pretransfer delay error counter
	HIGH	Receive buffer DMA overrun error counter

Word	Bits	Description
14	LOW	Receive counter repeat command
	HIGH	Error counter data block size
15	If bit 15 of word 1 is not set, word 15 has the following significance:	
	LOW	Error counter receiver collision abort
	HIGH	Error counter receiver alignment
	If bit 15 of word 1 is set, word 15 has the following significance:	
	LOW	Data block error on cable A
	HIGH	Data block error on cable B
16	LOW	Error counter CRC receiver
	HIGH	Error counter wrong packet length
17	LOW	Error counter wrong link address
	HIGH	Error counter DMA underflow transfer buffer storage
18	LOW	Error counter wrong internal packet length
	HIGH	Error counter wrong MAC function code
19	LOW	Communication retry counter
	HIGH	Error counter communication failed
20	LOW	Counter package receipt successful
	HIGH	Error counter no response receipt
21	LOW	Error counter unexpected response receipt
	HIGH	Error counter unexpected path
22	LOW	Error counter unexpected response
	HIGH	Error counter skipped transaction
23	LOW	Bitmap active station table, nodes 1 through 8
	HIGH	Bitmap active station table, nodes 9 through 16
24	LOW	Bitmap active station table, nodes 17 through 24
	HIGH	Bitmap active station table, nodes 25 through 32
25	LOW	Bitmap active station table, nodes 33 through 40
	HIGH	Bitmap active station table, nodes 41 through 48
26	LOW	Bitmap active station table, nodes 49 through 56
	HIGH	Bitmap active station table, nodes 57 through 64
27	LOW	Bitmap token station table, nodes 1 through 8
	HIGH	Bitmap token station table, nodes 9 through 16
28	LOW	Bitmap token station table, nodes 17 through 24
	HIGH	Bitmap token station table, nodes 25 through 32

Word	Bits	Description
29	LOW	Bitmap token station table, nodes 33 through 40
	HIGH	Bitmap token station table, nodes 41 through 48
30	LOW	Bitmap token station table, nodes 49 through 56
	HIGH	Bitmap token station table, nodes 57 through 64
31	LOW	Bitmap table regarding existence of global data, nodes 1 through 8
	HIGH	Bitmap table regarding existence of global data, nodes 9 through 16
32	LOW	Bitmap table regarding existence of global data, nodes 17 through 24
	HIGH	Bitmap table regarding existence of global data, nodes 25 through 32
33	LOW	Bitmap table regarding existence of global data, nodes 33 through 40
	HIGH	Bitmap table regarding existence of global data, nodes 41 through 48
34	LOW	Bitmap table regarding existence of global data, nodes 49 through 56
	HIGH	Bitmap table regarding existence of global data, nodes 57 through 64
35	LOW	Bitmap receive buffer used, buffers 1 through 8
	HIGH	Bitmap receive buffer used, buffers 9 through 16
36	LOW	Bitmap receive buffer used, buffers 17 through 24
	HIGH	Bitmap receive buffer used, buffers 25 through 32
37	LOW	Bitmap receive buffer used, buffers 33 through 40
	HIGH	Counter of activated processed commands for station administration
38	LOW	Counter activation command output path 1 of the data master
	HIGH	Counter activation command output path 2 of the data master
39	LOW	Counter activation command output path 3 of the data master
	HIGH	Counter activation command output path 4 of the data master
40	LOW	Counter activation command output path 5 of the data master
	HIGH	Counter activation command output path 6 of the data master
41	LOW	Counter activation command output path 7 of the data master
	HIGH	Counter activation command output path 8 of the data master
42	LOW	Counter command processing input path 41 of the data slave
	HIGH	Counter command processing input path 42 of the data slave
43	LOW	Counter command processing input path 43 of the data slave
	HIGH	Counter command processing input path 44 of the data slave
44	LOW	Counter command processing input path 45 of the data slave
	HIGH	Counter command processing input path 46 of the data slave
45	LOW	Counter command processing input path 47 of the data slave
	HIGH	Counter command processing input path 48 of the data slave

Word	Bits	Description
46	LOW	Counter command activation output path 81 of the program master
	HIGH	Counter command activation output path 82 of the program master
47	LOW	Counter command activation output path 83 of the program master
	HIGH	Counter command activation output path 84 of the program master
48	LOW	Counter command activation output path 85 of the program master
	HIGH	Counter command activation output path 86 of the program master
49	LOW	Counter command activation output path 87 of the program master
	HIGH	Counter command activation output path 88 of the program master
50	LOW	Counter command processing input path C1 of the program slave
	HIGH	Counter command processing input path C2 of the program slave
51	LOW	Counter command processing input path C3 of the program slave
	HIGH	Counter command processing input path C4 of the program slave
52	LOW	Counter command processing input path C5 of the program slave
	HIGH	Counter command processing input path C6 of the program slave
53	LOW	Counter command processing input path C7 of the program slave
	HIGH	Counter command processing input path C8 of the program slave

TCP/IP Ethernet Network Statistics

TCP/IP Ethernet Network Statistics

A TCP/IP Ethernet module replies to local and remote statistics commands from the `MBP_MSTR` block with the the contents of the `databuf` array (see the information in the table below):

Word	**Meaning**		
00 to 02	MAC address e.g. MAC address 00 00 54 00 12 34 is displayed as follows:	**Word** 00 01 02	**Contents** 00 00 54 00 12 34
03	Board status (refer to the following table)		
04 and 05	Number of receiver interrupts		
06 and 07	Number of transfer interrupts		
08 and 09	Transfer timeout error count		
10 and 11	Collision detection error count		
12 and 13	Omitted packets		
14 and 15	Memory error count		
16 and 17	Number of restarts performed by the driver		
18 and 19	Receive framing error count		
20 and 21	Overflow error count receiver		
22 and 23	Receive CRC error counter		
24 and 25	Receive buffer error counter		
26 and 27	Transfer buffer error counter		
28 and 29	Transfer bin underflow counter		
30 and 31	Late collision counter		
32 and 33	Lost carrier counter		
34 and 35	Number of retries		
36 and 37	IP address e.g. the IP address 198.202.137.113 (or C6 CA 89 71) is represented as follows:	**Word** 36 37	**Contents** 89 71 C6 CA

Board Status Word Bit Definition

NOTE: It is best to view the board status word in binary format.

The following table describes the bit definitions of the board status word:
- 140 NOE 771 *x*1, versions 2.0, 3.0, 3.1, 3.3 and 3.6 or higher
- 140 NOE 771 *x*0, versions 3.0, 3.3 and 3.4 or higher

Bit #	Definition
15	0 = Link LED off 1 = Link LED on
14	0 = Appl LED off 1 = Appl LED on
13	0 = twisted pair 1 = fiber
12	0 = 10 Mbit 1 = 100 Mbit
11 ... 8	(Reserved)
7 ... 4	Module type (see table, below)
3	(Reserved)
2	0 = half duplex 1 = full duplex
1	0 = not configured 1 = configured
0	0 = not running 1 = running

NOTE: Bits are numbered from right to left, starting with bit 0 (low bit). For example, **PLC running =** 0000 0000 0000 0001 and **LED connection =** 1000 0000 0000 0000.

The following table describes the word bit definitions for board status for the:
- 140 NOE 771 *x*1, version 3.5
- 140 NOE 771 *x*0, versions 1.02 and 2.0
- 140 CPU 651 *x*0

Bit #	Definition
15 ... 12	Module type (see table below)
11	(Reserved)
10	0 = half duplex 1 = full duplex
9	0 = not configured 1 = configured
8	0 = PLC not running 1 = PLC/NOE running
7	0 = Link LED off 1 = Link LED on
6	0 = Appl LED off 1 = Appl LED on
5	0 = twisted pair 1 = fiber

4	0 = 10 Mbit 1 = 100 Mbit
3 ... 0	(Reserved)

NOTE: Bits are counted from right to left, starting with bit 0 (low bit). For example, **PLC running =** 0x0100, **Application LED =** 0x0040, and **LED Connection =** 0x0080.

Board Status Word Bit Definition by Module Type

The following table describes the values of the module types:

Value of Bits 7...4 or 15...12 **Note: See the previous tables for the bit range that applies to your module's software version.**	**Module Type**
0	NOE 2x1
1	ENT
2	M1E
3	NOE 771 00
4	ETY
5	CIP
6	(reserved)
7	140 CPU 651 x0
8	(reserved)
9	(reserved)
10	NOE 771 10
11	NOE 771 01
12	NOE 771 11
13 ... 15	(reserved)

Modbus Plus, SY/MAX, and Ethernet TCP/IP Error Codes

Form of the Function Error Code

Function error codes for Modbus Plus and SY/MAX Ethernet transactions appear as **Mmss**, where:

- **M** is the high code
- **m** is the low code
- **ss** is a subcode

Modbus Plus and SY/MAX Ethernet Network Errors

Hexadecimal error codes for Modbus Plus and SY/MAX Ethernet:

Hex. Error Code	Description
1001	Abort by user
2001	An operation type that is not supported has been specified in the control block
2002	One or more control block parameters were modified while the `MSTR` element was active (this only applies to operations which require several cycles for completion). Control block parameters my only be modified in inactive `MSTR` components.
2003	Invalid value in the length field of the control block
2004	Invalid value in the offset field of the control block
2005	Invalid value in the length and offset fields of the control block
2006	Unauthorized data field on slave
2007	Unauthorized network field on slave
2008	Unauthorized network routing path on slave
2009	Routing path equivalent to their own address
200A	Attempt to get more global data words than available
200C	Bad pattern for change address request
200D	Bad address for change address request
200E	The control block is not assigned, or parts of the control block are located outside of the %MW (4x) range.
30ss	Exceptional response by Modbus slave *(see page 140)*
4001	Inconsistent response by Modbus slave
5001	Inconsistent response by the network
6007	Invalid slot ID
6mss	Routing path error *(see page 141)* The subfield m shows where the error occurred (a 0 value means local node, 2 means 2nd device in route, etc).

TCP/IP Ethernet Network Errors

Hexadecimal error codes for TCP/IP Ethernet:

Hex. Error Code	Meaning
5004	Interrupted system call
5005	I/O error
5006	No such address
5009	Socket descriptor is invalid
500C	Not enough memory
500D	Permission denied
5011	Entry exists
5016	Argument is invalid
5017	Internal table has run out of space
5020	Connection is broken
5028	Destination address required
5029	Protocol wrong type for socket
502A	Protocol not available
502B	Protocol not supported
502C	Socket type not supported
502D	Operation not supported on a socket
502E	Protocol family not supported
502F	Address family not supported
5030	Address already in use
5031	Cannot assign requested address
5032	Socket operation on a non-socket
5033	Network is unreachable
5034	Network dropped connection on reset
5035	Network caused connection abort
5036	Connection reset by peer
5037	No buffer space available
5038	Socket already connected
5039	Socket not connected
503A	Cannot send after socket shutdown
503B	Too many references, cannot splice
503C	Connection timed out (see note below)
503D	Connection refused

Hex. Error Code	Meaning
503E	Network down
503F	Text file busy
5040	Too many levels of links
5041	No route to host
5042	Block device required
5043	Host is down
5044	Operation now in progress
5045	Operation already in progress
5046	Operation would block
5047	Function not implemented
5048	Hardware length is invalid
5049	Route specified cannot be found
504A	Collision in select call: these conditions have already been selected by another task
504B	Task ID is invalid
5050	No network resource
5051	Length error
5052	Addressing error
5053	Application error
5054	Client in bad state for request
5055	No remote resource -- may indicate no path to remote device (see note below)
5056	Non-operational TCP connection
5057	Incoherent configuration

NOTE:

- Error code 5055 can occur before a 503C error.
- No remote device takes precedence over a timeout.

ss Hexadecimal Value in 30ss Error Code

ss hexadecimal value in 30ss error code:

ss Hex. Value	Description
01	Slave does not support requested operation
02	Non-existing slave registers were requested
03	An unauthorized data value was requested

ss Hex. Value	Description
05	Slave has accepted a lengthy program command
06	Function cannot currently be carried out: lengthy command running
07	Slave has rejected lengthy program command

ss Hexadecimal Value in 6mss Error Code

NOTE: Subfield m in error code 6mss is an `Index` in the routing information that shows where an error has been detected (a 0 value indicates the local node, 2 means the second device in the route, etc.).

The ss subfield in error code 6mss is as follows:

ss Hex. Value	Description
01	No response reception
02	Access to program denied
03	Node out of service and unable to communicate
04	Unusual response received
05	Router-node data path busy
06	Slave out of order
07	Wrong destination address
08	Unauthorized node type in routing path
10	Slave has rejected the command
20	Slave has lost an activated transaction
40	Unexpected master output path received
80	Unexpected response received
F001	Wrong destination node was specified for the `MSTR` operation

SY/MAX-Specific Error Codes

SY/MAX-Specific Error Codes

When utilizing SY/MAX Ethernet, three additional types of errors may appear in the `CONTROL[1]` register of the control block ().

The error codes have the following meaning:

- 71xx Error: Errors found by the SY/MAX remote device
- 72xx Error: Errors found by the server
- 73xx Error: Errors found by the Quantum translator

SY/MAX-Specific Hexadecimal Error Codes

SY/MAX-specific hexadecimal error codes:

Hex. Error Code	Description
7101	Invalid opcode found by the SY/MAX remote device
7103	Invalid address found by the SY/MAX remote device
7109	Attempt to write to a write protected register found by the SY/MAX remote device
F710	Receiver overflow found by the SY/MAX remote device
7110	Invalid length found by the SY/MAX remote device
7111	Remote device not active, no connection (occurs when retry attempts and time-out have been used up), found by the SY/MAX remote device
7113	Invalid parameter in a read operation found by the SY/MAX remote device
711D	Invalid route found by the SY/MAX remote device
7149	Invalid parameter in a write operation found by the SY/MAX remote device
714B	Invalid drop number found by the SY/MAX remote device
7101	Invalid opcode found by the SY/MAX server
7203	Invalid address found by the SY/MAX server
7209	Attempt to write to a write protected register found by the SY/MAX server
F720	Receiver overflow found by the SY/MAX server
7210	Invalid length found by the SY/MAX server
7211	Remote device not active, no connection (occurs when retry attempts and time-out have been used up), found by the SY/MAX server
7213	Invalid parameter in a read operation found by the SY/MAX server
721D	Invalid route found by the SY/MAX server
7249	Invalid parameter in a write operation found by the SY/MAX server
724B	Invalid drop number found by the SY/MAX server

Hex. Error Code	Description
7301	Invalid opcode in an MSTR block request from the Quantum translator
7303	Read/Write QSE module status (200 route address out of range)
7309	Attempt to write to a write protected register when a status write is carried out (200 route)
731D	Invalid route found by the Quantum translator. Valid routes: ● dest_drop, 0xFF ● 200, dest_drop, 0xFF ● 100+drop, dest_drop, 0xFF ● All other routing values produce an error
734B	One of the following errors occurred: ● No CTE (configuration extension table) has been configured ● No CTE table entry has been made for the QSE model slot number ● No valid drop has been specified ● The QSE module has not been reset after the creation of the CTE. Note: After writing and configuring the CTE and downloading to the QSE module, the QSE module must be reset for the modifications to become effective. ● When using an MSTR instruction no valid slot or drop has been specified

TCP/IP Ethernet Error Codes

TCP/IP Ethernet Error Codes

An error in an MSTR routine via TCP/IP Ethernet may produce one of the following errors in the MSTR control block:

The error code appears as **Mmss**, where:

- **M** is the high code
- **m** is the low code
- **ss** is a subcode

Hexadecimal Error Codes TCP/IP Ethernet

Hexadecimal error codes TCP/IP Ethernet:

Hex. Error Code	Meaning
1001	Abort by user
2001	An operation type that is not supported has been specified in the control block
2002	One or more control block parameters were modified while the MSTR element was active (this only applies to operations which require several cycles for completion). Control block parameters my only be modified in inactive MSTR components.
2003	Invalid value in the length field of the control block
2004	Invalid value in the offset field of the control block
2005	Invalid value in the length and offset fields of the control block
2006	Unauthorized data field on slave
2008	Unauthorized network routing path on slave
200E	The control block is not assigned, or parts of the control block are located outside of the %MW (4x) range.
3000	Generic Modbus failure code
30ss	Exceptional response by Modbus slave *(see page 144)*
4001	Inconsistent response by Modbus slave

ss Hexadecimal Value in 30ss Error Code

ss hexadecimal value in 30ss error code:

ss hex. Value	Meaning
01	Slave does not support requested operation
02	Non-existing slave registers were requested

ss hex. Value	Meaning
03	An unauthorized data value was requested
05	Slave has accepted a lengthy program command
06	Function cannot currently be carried out: lengthy command running
07	Slave has rejected lengthy program command

Hexadecimal Error Codes TCP/IP Ethernet Network

An error on the TCP/IP Ethernet network itself may produce one of the following errors in the `CONTROL[1]` register of the control block.

Hexadecimal error codes TCP/IP Ethernet network:

Hex. Error Code	Meaning
5004	Interrupted system invocation
5005	I/O error
5006	No such address
5009	The socket descriptor is not valid
500C	Not enough storage space
500D	Authorization denied
5011	Entry exists
5016	An argument is not valid
5017	An internal table has no more space
5020	There is interference on the connection
5023	This operation was blocked and the socket is non-blocking
5024	The socket is non-blocking and the connection cannot be closed down
5025	The socket is non-blocking and a previous connection attempt has not been concluded
5026	Socket operation on a non-socket
5027	The destination address is not valid
5028	Message too long
5029	Wrong type of protocol for the socket
502A	Protocol not available
502B	Protocol not supported
502C	Socket type not supported
502D	Operation not supported at socket

Hex. Error Code	Meaning
502E	Protocol family not supported
F502	Address family not supported
5030	Address is already in use
5031	Address not available
5032	Network is out of order
5033	Network cannot be reached
5034	Network shut down the connection during reset
5035	The connection was terminated by the peer
5036	The connection was reset by the peer
5037	An internal buffer is required, but cannot be assigned
5038	The socket is already connected
5039	The socket is not connected
503A	Cannot transmit after the socket has been shut off
503B	Too many references; cannot splice
503C	Connection timed out
503D	The connection attempt was denied
5040	Host is out of order
5041	The destination host could not be reached from this node
5042	Directory not empty
5046	NI_INIT returned -1
5047	The MTU is not valid
5048	The hardware length is not valid
5049	The route specified cannot be found
504A	Collision when invoking Select; these conditions have already been selected by another job
504B	The job ID is not valid
5050	No Network Resource
5051	Length Error
5052	Addressing Error
5053	Application Error
5054	Client cannot process request
5055	No Network Resource
5056	Non-Operational TCP connection

Hex. Error Code	Meaning
5057	Incoherent configuration
6003	FIN or RST not expected
F001	In reset mode
F002	Component not fully initialized

CTE Error Codes for SY/MAX and TCP/IP Ethernet

CTE Error Codes for SY/MAX and TCP/IP Ethernet

The following error codes are displayed in the `CONTROL[1]` register of the control block, if there is a problem with the Ethernet configuration extension table (CTE) in your program configuration.

CTE error codes for SY/MAX and TCP/IP Ethernet:

Hex. Error Code	Description
7001	There is no Ethernet configuration extension.
7002	The CTE is not available for access.
7003	The offset is not valid.
7004	Offset + length are not valid.
7005	Bad data field in the CTE.

Mail Service Error Codes

Error Codes

The electronic mail notification service supports the following error codes:

Hex. Error Code	Description
5100	Internal error
5101	SMTP component not operational
5102	Mail Header not configured
5103	Invalid Mail Header value (should be 1, 2 or 3)
5104	Cannot connect to SMTP server
5105	Error in transmitting content of email body to SMTP server
5106	Closing SMTP connection with the server returned an error
5107	SMTP HELO request failed
5108	SMTP MAIL request failed. SMTP server may require authentication
5109	SMTP RCPT request failed
510A	No recipient has been accepted by the SMTP server
510B	SMTP DATA request failed
510C	Send email request contains an invalid length
510D	Authentication failed
510E	A Reset component request has been received while the connection was open