

Behavior of MBP_MSTR Command 15 Enhancement:

Provides the ability to send a generic Modbus command and provides the response to the command via the MBP_MSTR block.

The functionality mimics the Data Exchange block (DATA_EXCH) supported on the Premium and M340 line.

The MBP_MSTR block can send requests and receive responses up to 255 bytes in length.

The following is the expected **Control Block** structure for Option 15 – Generic Modbus command

Word	Function	Description
CONTROL[1]	Operation	15 = Generic Modbus command
CONTROL[2]	Error status	Holds event code (read-only)
CONTROL[3]	Data buffer length	Length of Data buffer (command dependent)
CONTROL[4]	Response offset	Offset to the beginning of the response in the Data buffer (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
CONTROL[6] ¹	IP Address	Byte 4 of the IP address (MSB)
CONTROL[7] ¹		Byte 3 of the IP address
CONTROL[8] ¹		Byte 2 of the IP address
CONTROL[9] ¹		Byte 1 of the IP address (LSB)
CONTROL[10]	Data buffer request length	Length of the Data buffer request data (bytes)
CONTROL[11]	Data buffer response length	Length of the Data buffer response received (bytes) NOTE: This word in read ONLY, it is set by the module after the operation (command) is complete.
1. For example, the Control parameter handles the IP address 192.168.1.7 in the following order Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 7.		

The following is the expected **Data Buffer** (DATABUF) structure for Option 15 – Generic Modbus command:

Data Buffer Data Buffer Length is set in the CONTROL[3] word.	Modbus Request The length of the request is found in the CONTROL[10] word.
	Modbus Response The location of the response in the Data buffer is found in CONTROL[4] (Response offset). CONTROL[4] needs to be at least 2 times the Data Request length to avoid the response data from overwriting the Modbus Request.

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

The following is the structure of the **Modbus Request** that is first portion of the Data Buffer (DATABUF) structure for Option 15 – Generic Modbus command:

Byte offset	Field	Data type	Description
1	Function code	BYTE	Modbus function code
2	Request data	BYTE	Request Data size is Modbus function code dependent
3	Request data	BYTE	
4	Request data	BYTE	
n...	Request data	BYTE	End of Modbus Request

The following is the structure of the **Modbus Response** that is the send part of the Data Buffer (DATABUF) structure for Option 15 – Generic Modbus command:

Byte offset	Field	Data type	Description
1	Function code	BYTE	Modbus function code
2	Response data	BYTE	Response Data size is Modbus function code dependent
3	Response data	BYTE	
4	Response data	BYTE	
n...	Response data	BYTE	End of Modbus Response

Example:

To read three contiguous holding registers, using Modbus function code 03, starting at register 100 in a remote device, the following Modbus Request is sent and Modbus Response is received via the Data Buffer (DATABUF).

The Modbus read request in **byte** format

Request	
Field Name	Value
Function code	03
Starting Address Hi	00
Starting Address Lo	64
Number of Registers Hi	00
Number of Registers Lo	03

The Modbus read response in **byte** format

Response	
Field Name	Value
Function code	03
Byte Count	06
Register Value Hi (100)	00
Register Value Lo (100)	01
Register Value Hi (101)	00
Register Value Lo (101)	02
Register Value Hi (102)	00
Register Value Lo (102)	03

Data encoding in Data buffer (DATABUF) in **WORDS** is as follows:

	Field	Value	Description
Request	DATABUF[1]	16#0300	Request START - Function code + Starting Address Hi
	DATABUF[2]	16#6400	Starting Address Lo + Number of Registers Hi
	DATABUF[3]	16#0300	Number of Registers Lo + Null
	DATABUF[4]	16#0000	Null – reserved to separate request from response
	DATABUF[5]	16#0000	Null – reserved to separate request from response
Answer	DATABUF[6]	16#0306	Response START - Function code + Byte Count
	DATABUF[7]	16#0001	Register Value Hi + Register Value Lo
	DATABUF[8]	16#0002	Register Value Hi + Register Value Lo
	DATABUF[9]	16#0003	Register Value Hi + Register Value Lo
	DATABUF[10]	16#0000	Null