

Solution Paper

Using MBP_MSTR function
code15 to perform a Modbus
function

The following example will show how to configure an MBP_MSTR to read four (4) registers from a 140CPU65160 PLC and write three (3) register to the same PLC. This will be performed using Modbus function code 23 from an NOE77111. The function code 23 requests will be sent using an MBP_MSTR configured for function code 15.

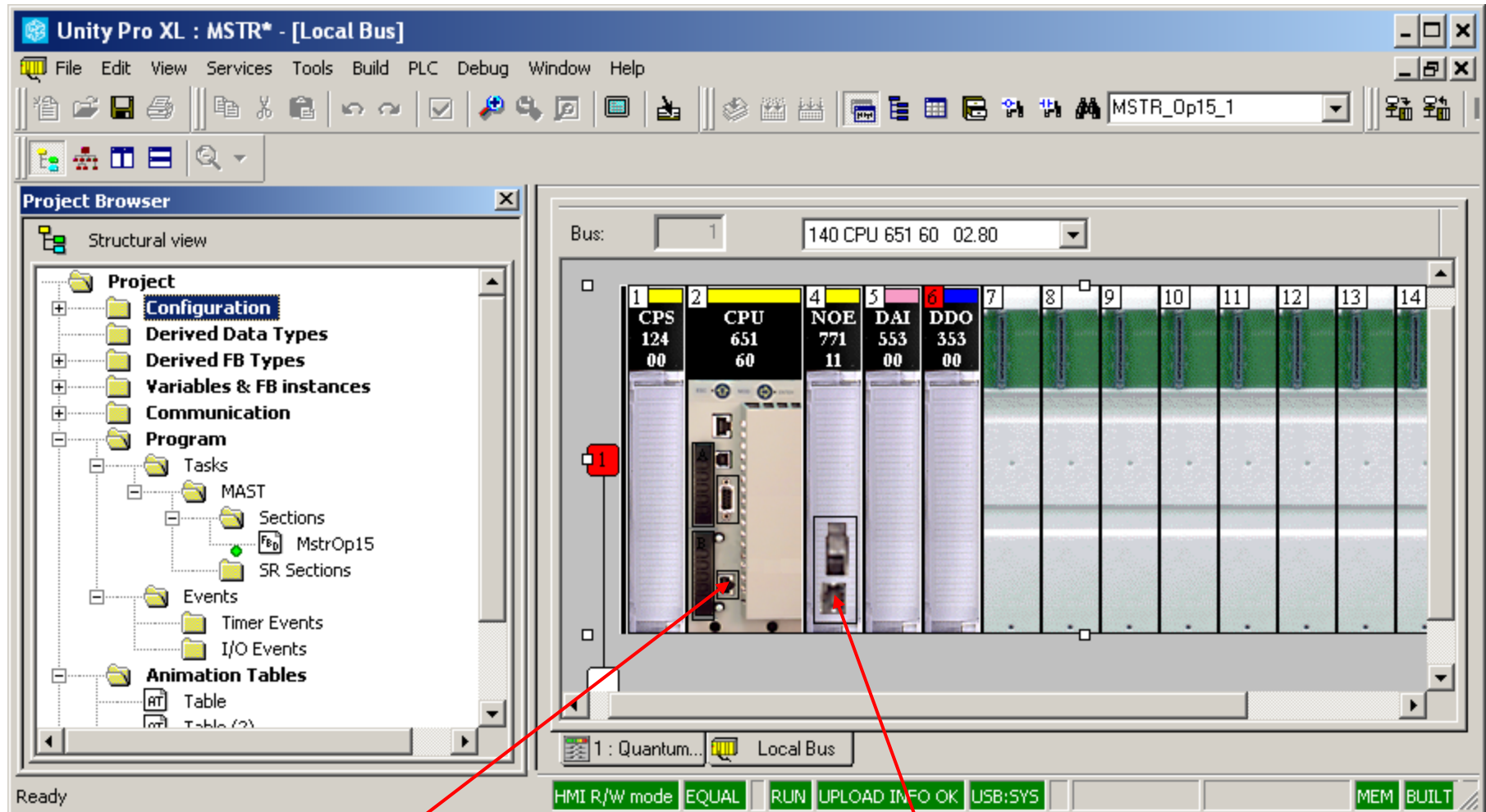
The '140NOE77111' will read four (4) words starting at register %MW16800 and write three (3) words starting at register %MW500.

To configure the function code 23 request in the MBP_MSTR, you will need the information from the Modbus Application Protocol Specification.

The specification can be downloaded from:

http://modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf

IP address assignment



IP address 192.168.1.33

IP address 192.168.1.13

The following description is taken from the Modbus Protocol Specification. It shows how the data buffer request table should be configured.

6.17 23 (0x17) Read/Write Multiple registers

This function code performs a combination of one read operation and one write operation in a single MODBUS transaction. The write operation is performed before the read.

Holding registers are addressed starting at zero. Therefore holding registers 1-16 are addressed in the PDU as 0-15.

The request specifies the starting address and number of holding registers to be read as well as the starting address, number of holding registers, and the data to be written. The byte count specifies the number of bytes to follow in the write data field.

The normal response contains the data from the group of registers that were read. The byte count field specifies the quantity of bytes to follow in the read data field.

Request

Function code	1 Byte	0x17
Read Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity to Read	2 Bytes	0x0001 to 0x007D
Write Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity to Write	2 Bytes	0x0001 to 0x0079
Write Byte Count	1 Byte	2 x N*
Write Registers Value	N*x 2 Bytes	

*N = Quantity to Write

Response

Function code	1 Byte	0x17
Byte Count	1 Byte	2 x N**
Read Registers value	N** x 2 Bytes	

*N' = Quantity to Read

Error

December 28, 2006

<http://www.Modbus-IDA.org>

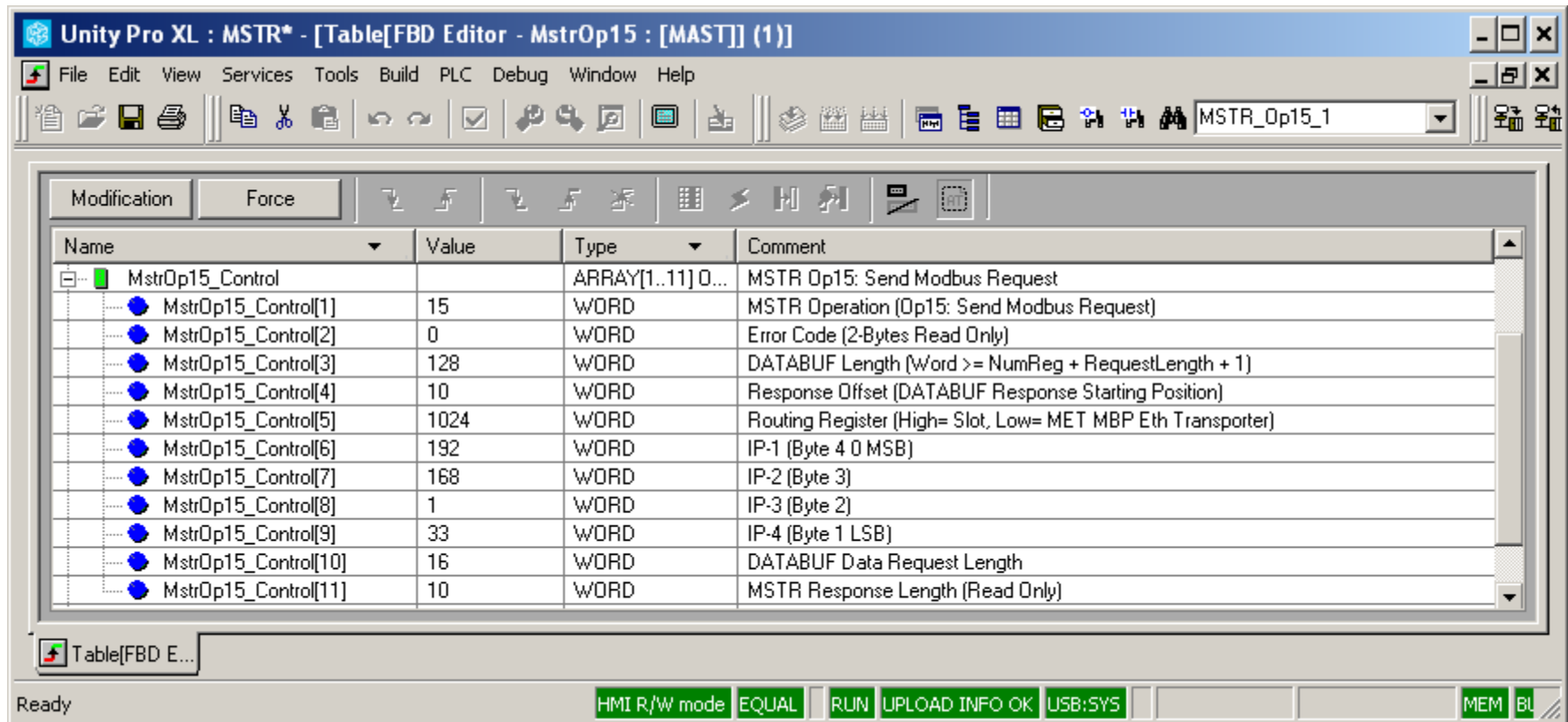
38/51

MODBUS Application Protocol Specification V1.1b

Modbus-IDA

Error code	1 Byte	0x97
Exception code	1 Byte	01 or 02 or 03 or 04

Assign eleven (11) words for the MBP_MSTR Control table



Unity Pro XL : MSTR* - [Table[FBD Editor - MstrOp15 : [MAST]] (1)]

File Edit View Services Tools Build PLC Debug Window Help

MSTR_Op15_1

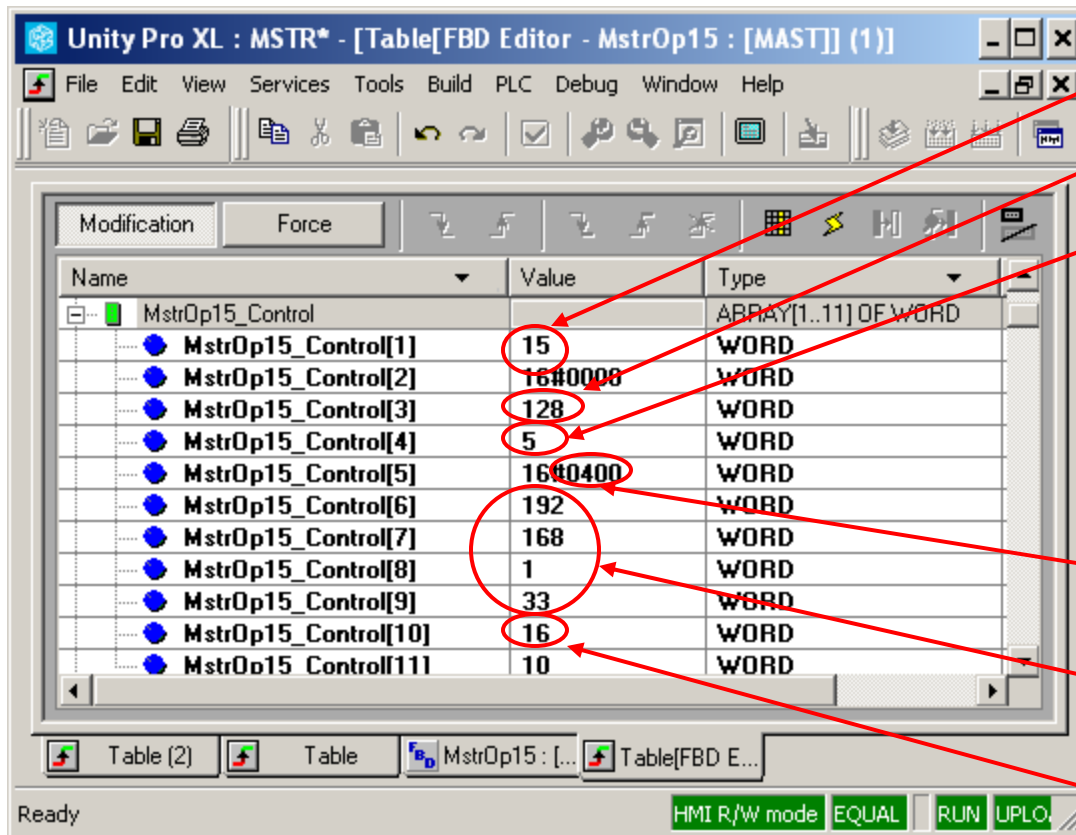
Modification Force

Name	Value	Type	Comment
MstrOp15_Control		ARRAY[1..11] 0...	MSTR Op15: Send Modbus Request
MstrOp15_Control[1]	15	WORD	MSTR Operation (Op15: Send Modbus Request)
MstrOp15_Control[2]	0	WORD	Error Code (2-Bytes Read Only)
MstrOp15_Control[3]	128	WORD	DATABUF Length (Word >= NumReg + RequestLength + 1)
MstrOp15_Control[4]	10	WORD	Response Offset (DATABUF Response Starting Position)
MstrOp15_Control[5]	1024	WORD	Routing Register (High= Slot, Low= MET MBP Eth Transporter)
MstrOp15_Control[6]	192	WORD	IP-1 (Byte 4 0 MSB)
MstrOp15_Control[7]	168	WORD	IP-2 (Byte 3)
MstrOp15_Control[8]	1	WORD	IP-3 (Byte 2)
MstrOp15_Control[9]	33	WORD	IP-4 (Byte 1 LSB)
MstrOp15_Control[10]	16	WORD	DATABUF Data Request Length
MstrOp15_Control[11]	10	WORD	MSTR Response Length (Read Only)

Table[FBD E...

Ready HMI R/W mode EQUAL RUN UPLOAD INFO OK USB:SYS MEM BU

Configure the Control registers



Function code 15

Data buffer size (i.e., 128 words)

Set the offset to a value of 10 registers. The offset is the number of the word in the data buffer where the data begins. The first word of the data contains the function code and byte count

Slot number for the NOE

IP address for the server.
(i.e., 192.168.1.33)

Total number of bytes for the request.
(This value will be different for each Modbus function code)

Request

Function code	1 Byte	0x17
Read Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity to Read	2 Bytes	0x0001 to 0x007D
Write Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity to Write	2 Bytes	0x0001 to 0x0079
Write Byte Count	1 Byte	2 x N*
Write Registers Value	N x 2 Bytes	

*N = Quantity to Write

i.e.,
10 bytes for the request +
6 bytes for the words to be written
equals 16 bytes

Configure the data buffer request information table

23=Hex 16#17

Register 16800 =Hex 16#41A0

4 registers = Hex16#0004

Register 500=Hex 16#01F4

3 words to write

Write byte count for the 3 words to be written

Value of the 3 words to be written

Name	Value	Type
MstrOp15_Databuf[1]	16#1741	WORD
MstrOp15_Databuf[2]	16#A000	WORD
MstrOp15_Databuf[3]	16#0401	WORD
MstrOp15_Databuf[4]	16#F400	WORD
MstrOp15_Databuf[5]	16#0306	WORD
MstrOp15_Databuf[6]	16#0000	WORD
MstrOp15_Databuf[7]	16#15FF	WORD
MstrOp15_Databuf[8]	16#1FFF	WORD
MstrOp15_Databuf[9]	16#0000	WORD
MstrOp15_Databuf[10]	16#0000	WORD
MstrOp15_Databuf[11]	16#1708	WORD
MstrOp15_Databuf[12]	1234	WORD
MstrOp15_Databuf[13]	4321	WORD
MstrOp15_Databuf[14]	5678	WORD
MstrOp15_Databuf[15]	9876	WORD
MstrOp15_Databuf[16]	0	WORD
MstrOp15_Databuf[17]	0	WORD

Results as shown in Unity

The screenshot shows the Unity Pro XL software interface with two data tables. The left table, titled 'Table', lists memory addresses and their values. The right table, titled 'Table[FBD Editor - MstrOp15 : [MAST]] (1)', lists data buffer addresses and their values. Red circles and arrows highlight specific data points and their corresponding function codes and byte counts.

Name	Value	Type	Comment
%MW16801	16#04D2	INT	
%MW16802	16#10E1	INT	
%MW16803	16#162E	INT	
%MW16804	16#2694	INT	
%MW501	16#0000	INT	
%MW502	16#15FF	INT	
%MW503	16#1FFF	INT	

Name	Value	Type
MstrOp15_Databuf[1]	16#1741	WORD
MstrOp15_Databuf[2]	16#A000	WORD
MstrOp15_Databuf[3]	16#0401	WORD
MstrOp15_Databuf[4]	16#F400	WORD
MstrOp15_Databuf[5]	16#0306	WORD
MstrOp15_Databuf[6]	16#0000	WORD
MstrOp15_Databuf[7]	16#15FF	WORD
MstrOp15_Databuf[8]	16#1FFF	WORD
MstrOp15_Databuf[9]	16#0000	WORD
MstrOp15_Databuf[10]	16#0000	WORD
MstrOp15_Databuf[11]	16#1708	WORD
MstrOp15_Databuf[12]	16#04D2	WORD
MstrOp15_Databuf[13]	16#10E1	WORD
MstrOp15_Databuf[14]	16#162E	WORD
MstrOp15_Databuf[15]	16#2694	WORD
MstrOp15_Databuf[16]	0	WORD

Written data

Function code (byte 1) and byte count (byte 2)

Returned data for Read request

Note: In this example, the register are offset by one because the NOE is a Modicon device that uses a data format of 4xxxx and the Unity PLC uses a data format of %MW starting at %MW0.

Wireshark trace showing results for the request

The image shows a Wireshark packet capture window titled "(Untitled) - Wireshark". The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Help) and a toolbar with various icons. A filter bar is set to "Expression... Clear Apply".

The packet list table shows the following entries:

No.	Time	Source	Destination	Port	Protocol	Info
207	4.803	192.168.1.13	192.168.1.33	2729	Modbus/TCP	query [1 p
208	0.000	192.168.1.33	192.168.1.13	502	TCP	asa-appl-protc
209	0.002	192.168.1.33	192.168.1.13	502	Modbus/TCP	response [1 p
210	0.047	192.168.1.13	192.168.1.33	2729	TCP	tcim-control >
211	5.152	RichardH_74:e6:de	Ieee8021_00:00:20		GMRP	GMRP

The packet details pane for the selected packet (No. 207) shows the following structure:

- transaction identifier: 53547
- protocol identifier: 0
- length: 17
- unit identifier: 0
- ☒ Modbus
 - function 23: Read Write Register
 - read reference number: 16800
 - read word count: 4
 - write reference number: 500
 - write word count: 3
 - byte count: 6
 - Data

The packet bytes pane displays the raw data in hexadecimal and ASCII:

```
0000  00 00 54 18 03 d9 00 00 54 10 8a d1 08 00 45 00  ..T..... T.....E.
0010  00 3f 1c 41 00 00 40 06 da f9 c0 a8 01 0d c0 a8  .?.A..@. ....
0020  01 21 0a a9 01 f6 2e a6 1a 3d 2c db f7 3c 50 18  .!..... .,.,<P.
0030  10 00 89 77 00 00 d1 2b 00 00 00 11 00 17 41 a0  ...w...+ .....A.
0040  00 04 01 f4 00 03 06 00 00 15 ff 1f ff          ..... .....
```

The status bar at the bottom indicates: "Packets: 211 Displayed: 211 Marked: 0 Dropped: 0".

Wireshark trace for the response

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Port	Protocol	Info
1	0.000	RichardH_74:e6:de	Ieee8021_00:00:20		GMRP	GMRP
2	0.959	192.168.1.13	192.168.1.33	2729	Modbus/TCP	query [1 p
3	0.000	192.168.1.33	192.168.1.13	502	TCP	asa-appl-protc
4	0.002	192.168.1.33	192.168.1.13	502	Modbus/TCP	response [1 p
5	0.084	192.168.1.13	192.168.1.33	2729	TCP	tcim-control >

Frame 4 (71 bytes on wire, 71 bytes captured)

Ethernet II, Src: Modicon_18:03:d9 (00:00:54:18:03:d9), Dst: Modicon_10:8a:d1 (00:00:54:10:8a:d1)

Internet Protocol, Src: 192.168.1.33 (192.168.1.33), Dst: 192.168.1.13 (192.168.1.13)

Transmission Control Protocol, Src Port: asa-appl-protc (502), Dst Port: tcim-control (2729), Seq: 1, Ack: 24, Len: 17

Modbus/TCP

- transaction identifier: 52464
- protocol identifier: 0
- length: 11
- unit identifier: 0

Modbus

- function 23: Read Write Register
- byte count: 8

Data

```

0000  00 00 54 10 8a d1 00 00 54 18 03 d9 08 00 45 00  ..T.... T....E.
0010  00 39 a6 ee 00 00 40 06 50 52 c0 a8 01 21 c0 a8  .9....@. PR...!..
0020  01 0d 01 f6 0a a9 2c db af 51 2e a5 b9 07 50 18  ...., .Q....P.
0030  10 00 01 5f 00 00 cc f0 00 00 00 0b 00 17 08 04  ....>.....>
0040  d2 10 e1 16 2e 26 94  ....&.
  
```

Packets: 5 Displayed: 5 Marked: 0 Dropped: 0

Requirements

In order to support the MBP_MSTR function code 15 the following exec firmware is required

- NOE77101 version 5.0
- NOE7711 version 5.0
- Unity Quantum Co-Pro CPU Exec firmware version 3.0

Summary

The same procedure can be used for any supported Modbus function code that is listed in the 'Modbus Application Protocol Specification'.

Before using the 'MBP_MSTR function code 15', the user needs to determine the response byte arrangement of the attached device

The data buffer request information table size will vary depending upon the Modbus function code.

The 'MBP_MSTR function code 15' is only applicable on the '140NOE77101' and '140NOE77111' Ethernet modules.

This feature is not supported on the '140NOC771xx' or Unity Quantum Co-Pro Ethernet port.