CIP Modbus Object Read Example
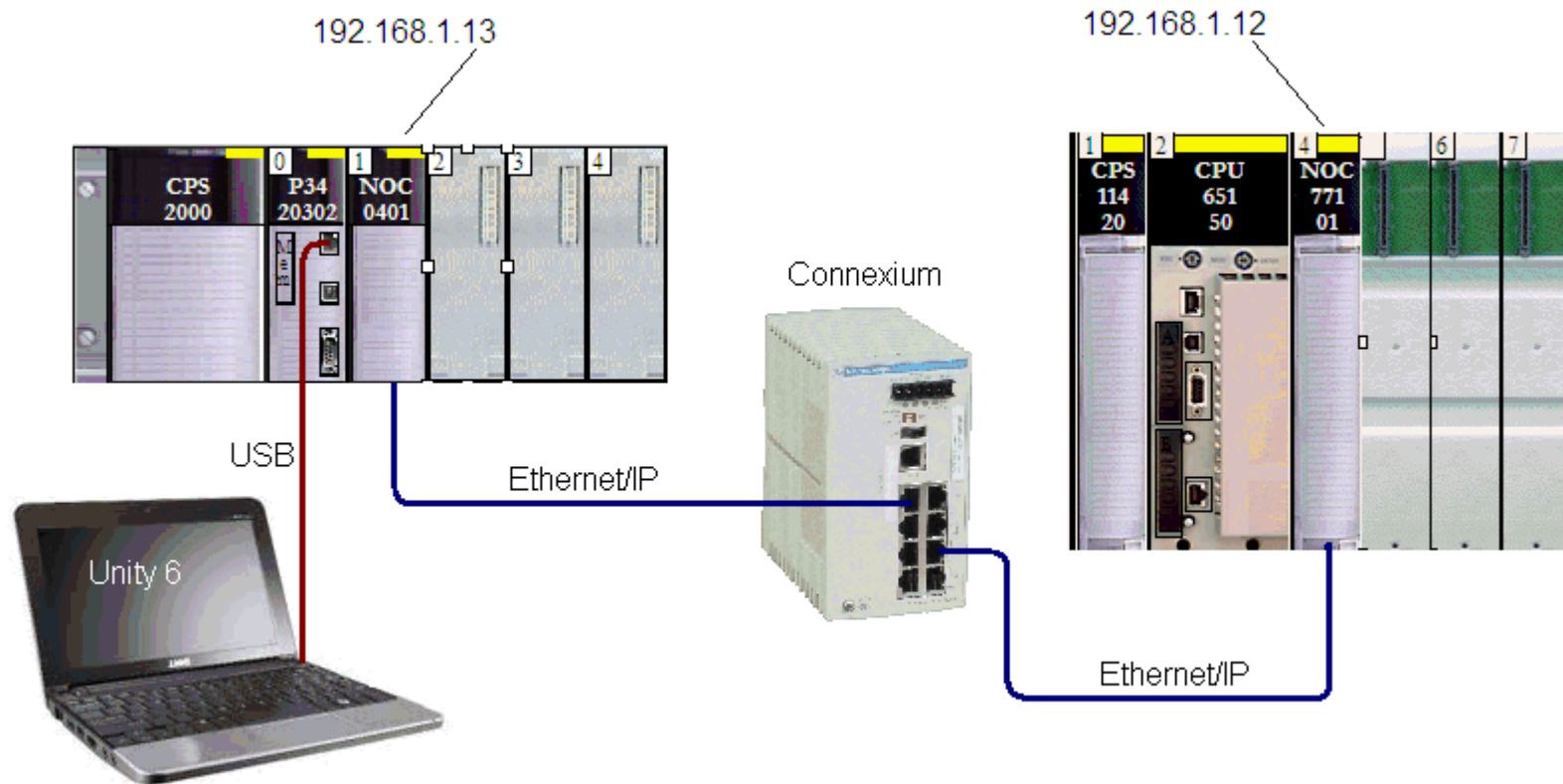
BMXNOC0401 using
Explicit Messaging via DATA_EXCH

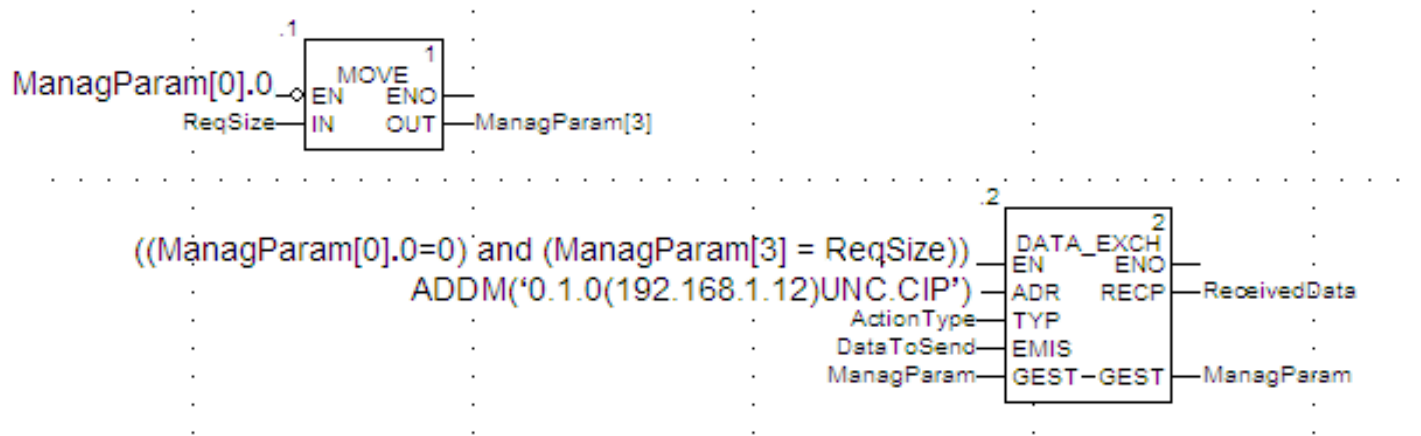Dec 15, 2012

Version 1.0

Schneider Electric

PLC Hardware Configuration

• BMXNOC0401 (192.168.1.13) to query NOC77101 (192.168.1.12) with Explicit Messaging CIP Modbus Object Read_Holding_Register.

• The USB connection is for Unity to M340 PLC communications.

Unity Program

- Note to add the 'Pin negation' on the AND_BOOL IN1 input.

• Configuring the DATA_EXCH ADR Input:

ADDM('0.1.0{192.168.1.12}UNC.CIP')

        Rack = 0
        Module (Slot Number) = 1
        Channel = 0
        Remote Device IP address = 192.168.1.12
        Message Type = UNConnected
        Protocol = CIP

Declaring Variables

Name	Value	Type	Comment
● ReqSize	10	INT	
● ActionType	1	INT	Transmission, followed by await reception
⊟ ■ DataToSend		ARRAY[0..4] OF...	
● DataToSend[0]	16#024E	INT	HiByte=02 (Path Size); LowByte=4E (ServiceCode= Read Holding Reg)
● DataToSend[1]	16#4420	INT	HiByte= 44 (Class); LowByte=20 (Class Segment)
● DataToSend[2]	16#0124	INT	HiByte=01 (Instance); LowByte=24 (Instance Segment)
● DataToSend[3]	49	INT	First word to be Read
● DataToSend[4]	1	INT	Number of Words to Read
⊟ ■ ManagParam		ARRAY[0..3] OF...	
● ManagParam[0]	16#5501	INT	MSB:Exchange #; LSB:bit 1=activity, bit 2=cancel
● ManagParam[1]	0	INT	Operation Report; Communication Report
● ManagParam[2]	2	INT	Function Block Timeout = 2 (200ms)
● ManagParam[3]	10	INT	Length of DataToSend parameter (in Bytes)
⊟ ■ ReceivedData		ARRAY[0..49] O...	
● ReceivedData[0]	16#00CE	INT	Service Response
● ReceivedData[1]	16#0000	INT	Service Response 0 = success
● ReceivedData[2]	-26574	INT	Data Response

EIP_DataBuf
CIP Request

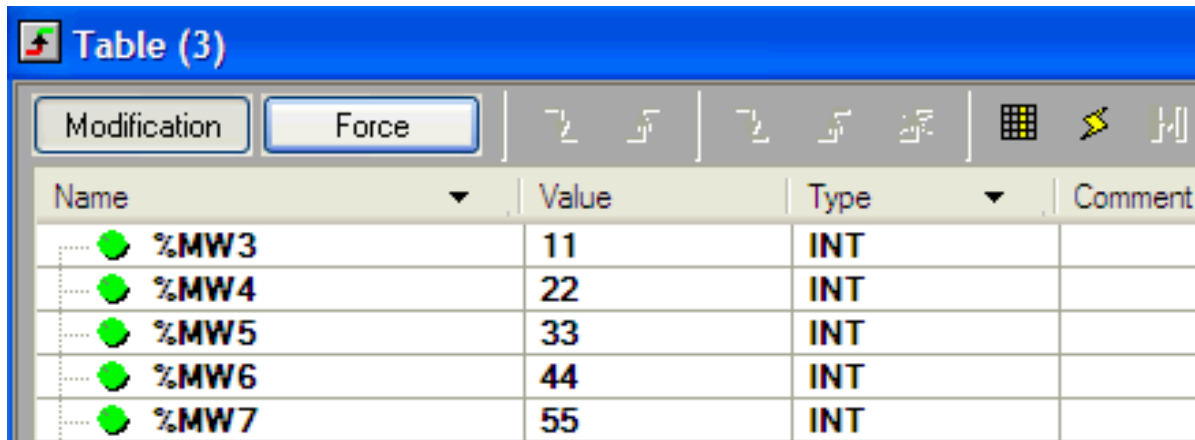| Name	Value	Type	Comment
⊟ EIP_DataBuf		ARRAY[0..100] ...	
● EIP_DataBuf[0]	16#024E	INT	HiByte=02 (Path Size); LoByte=4E (Service Code-Read Holding Reg)
● EIP_DataBuf[1]	16#4420	INT	Hi Bye=44 (Class Assembly Object); LoByte=20 (Logical Class Segment)
● EIP_DataBuf[2]	16#0124	INT	HiByte=01 (Instance); LoByte=24 (Logical Instance Segment)
● EIP_DataBuf[3]	16#0002	INT	First Word to be Read (value + %MW1 = First Word)
● EIP_DataBuf[4]	16#0005	INT	Number of Words to Read
● EIP_DataBuf[5]	16#0000	INT	[Service Code + Response Bit [MSB]) Response = CE (Read Only)
● EIP_DataBuf[6]	16#0000	INT	[Service Response=0, Success] (Read Only)
● EIP_DataBuf[7]	0	INT	Response Word 1
● EIP_DataBuf[8]	0	INT	Response Word 2
● EIP_DataBuf[9]	0	INT	Response Word 3
● EIP_DataBuf[10]	0	INT	Response Word 4
● EIP_DataBuf[11]	0	INT	Response Word 5

Control and Status Variables

- Set Enable to a value of 1 to start the messaging.
- In a successful implementation, EMActive, EMStart, and EMSuccess will flash between 0 and 1.

Data in PLC to be Read

- This data can be changed manually in the PLC to observe the data changing in the response.

EIP_DataBuf
CIP Response

- The message response is located in the EIP_DataBuf array beginning at EIP_DataBuf(5) as indicated in the area highlighted in red.
- The area highlighted in blue in the EIP_DataBuf array contains part of the query message previously entered.

Name	Value	Type	Comment
EIP_DataBuf		ARRAY[0..100] ...	
EIP_DataBuf[0]	16#024E	INT	HiByte=02 (Path Size); LoByte=4E (Service Code-Read Holding Reg)
EIP_DataBuf[1]	16#4420	INT	Hi Bye=44 (Class Assembly Object); LoByte=20 (Logical Class Segment)
EIP_DataBuf[2]	16#0124	INT	HiByte=01 (Instance); LoByte=24 (Logical Instance Segment)
EIP_DataBuf[3]	16#0002	INT	First Word to be Read (value + %MW1 = First Word)
EIP_DataBuf[4]	16#0005	INT	Number of Words to Read
EIP_DataBuf[5]	16#00CE	INT	[Service Code + Response Bit [MSB]) Response = CE (Read Only)
EIP_DataBuf[6]	16#0000	INT	[Service Response=0, Success] (Read Only)
EIP_DataBuf[7]	11	INT	Response Word 1
EIP_DataBuf[8]	22	INT	Response Word 2
EIP_DataBuf[9]	33	INT	Response Word 3
EIP_DataBuf[10]	44	INT	Response Word 4
EIP_DataBuf[11]	55	INT	Response Word 5