Reading Motor Current from a DC1/DE1

on Smartwire via Ethernet/IP

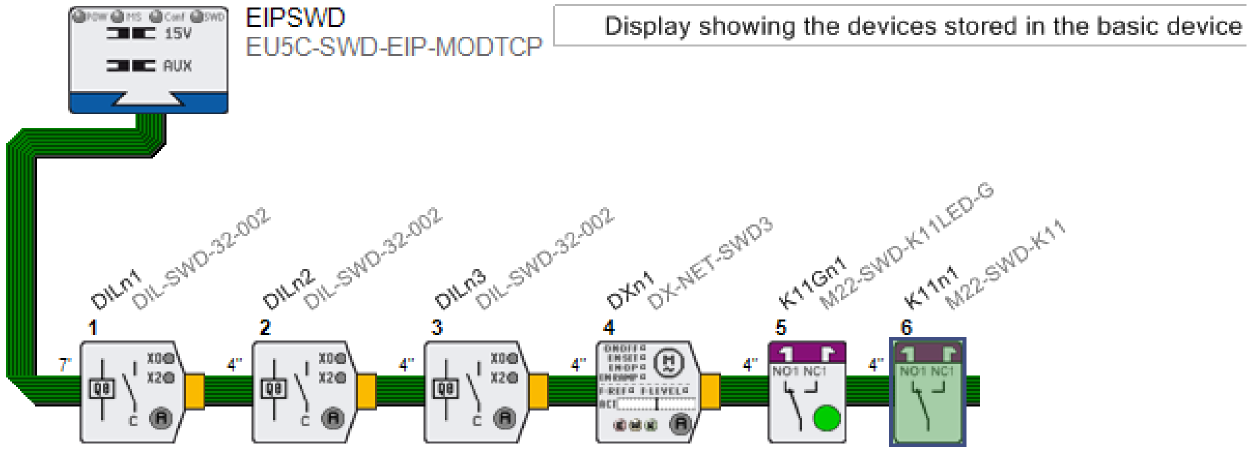
The motor current is not included in the cyclic data from a DC1 Variable Frequency Drive or from the DE1 Variable Speed Starter on Smartwire. This application note will demonstrate how to read the motor current from these devices with explicit messages via Ethernet/IP and a Rockwell PLC with the EU5C-SWD-EIP-MODTCP Ethernet gateway when they are connected to a Smartwire network.

Connecting the DC1/DE1 to a Smartwire Network

First, the DE1/DC1 must be connected to a Smartwire network where the network master is the EU5C-SWD-EIP-MODTCP Ethernet gateway. For this example, a DC1 Variable Frequency Drive is connected to a Smartwire network with 3 XT contactors (DIL-SWD-32-002), an LED (M22-SWD-K11LED-G) and a pushbutton (M22-SWD-K11) as shown below. Reading motor current from a DE1 is identical to reading it from a DC1.

Smartwire Assist software is used to setup and configure the network and the devices. It is also used to determine the I/O assembly numbers and data lengths for each. In addition, it also has the capability of creating a CSV file containing the I/O tags for each Smartwire device that can be imported into RSLogix5000 and aliased to the generic I/O tags created by the programming software for third-party Ethernet/IP devices.

Below is a screen shot of the Smartwire network used for this application example.



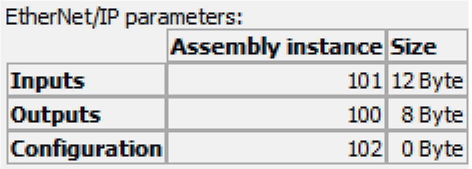
Display showing the devices stored in the basic device

Note that the DC1 Drive (DX-NET-SWD3) is node 4 on the Smartwire network. This information will be needed when using Message instructions in RSLogix5000 to read the motor current via Acyclic/Explicit messages.

The Ethernet Gateway

Select the EIPSWD gateway and view its various tabs.

Under the Device Information tab for the gateway is where the information concerning I/O assembly numbers and I/O data lengths are stored. For this example, the following assemblies and data lengths have been generated.



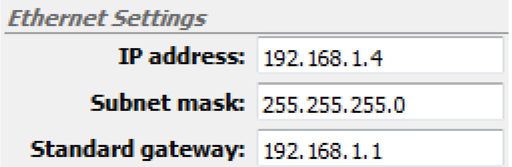
EtherNet/IP parameters:

	Assembly instance	Size
Inputs	101	12 Byte
Outputs	100	8 Byte
Configuration	102	0 Byte

This is the data that must be entered for the Generic Ethernet Device in RSLogix5000 when adding this device to the Ethernet/IP network. Since there are an almost infinite amount of possible Smartwire network combinations, an eds file is not used with the Smartwire Ethernet Gateway.

Under the Device Parameters tab, be sure the Ethernet/IP Mode is selected. This is the default.

Under the Ethernet Parameters tab is where its IP address, subnet mask and gateway address are configured. For this example, the following are used.



Ethernet Settings

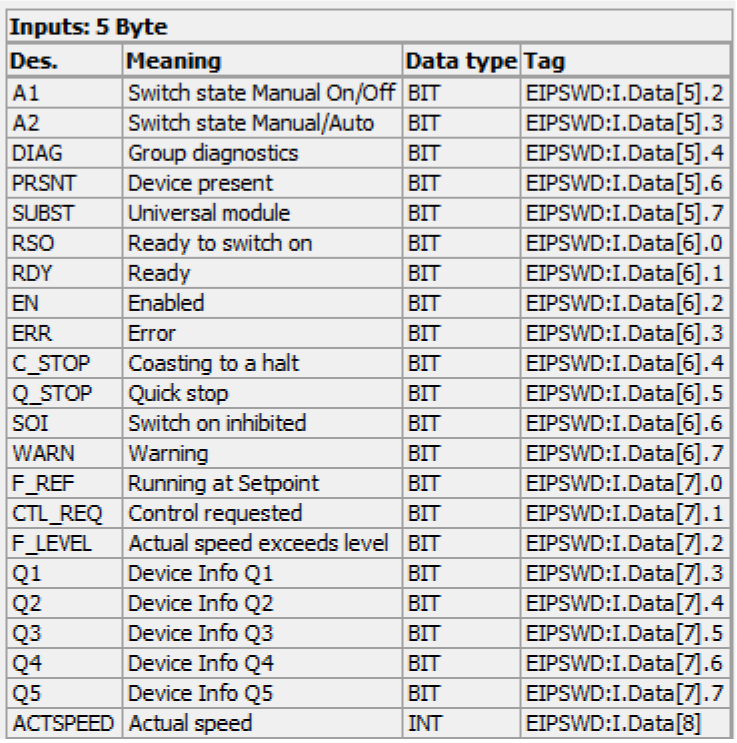
IP address: 192.168.1.4

Subnet mask: 255.255.255.0

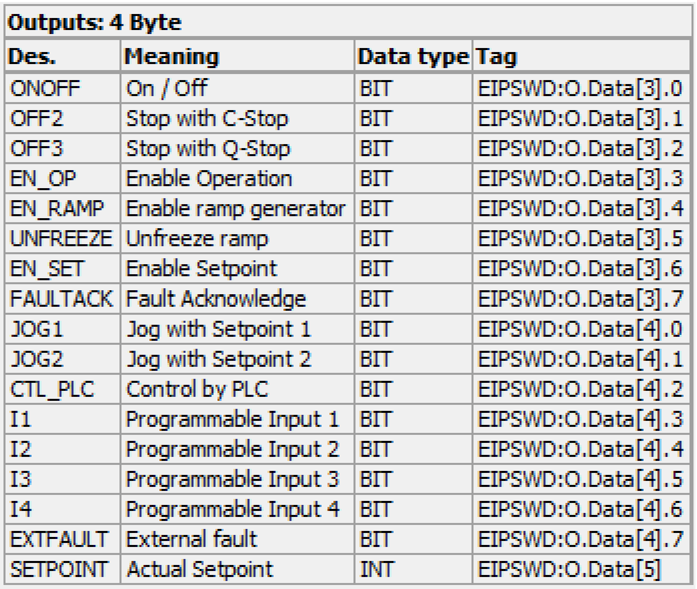
Standard gateway: 192.168.1.1

The IP address for the 1756-EN2T Ethernet/IP scanner is 192.168.1.2.

Select each individual Smartwire device to view the I/O data showing the actual RSLogix5000 generic I/O address for each parameter. “EIPSWD” is used for the name when entering the gateway into RSLogix5000 and is also the name for each generic I/O tag. For example, the I/O tags for the DC1 drive are shown below.

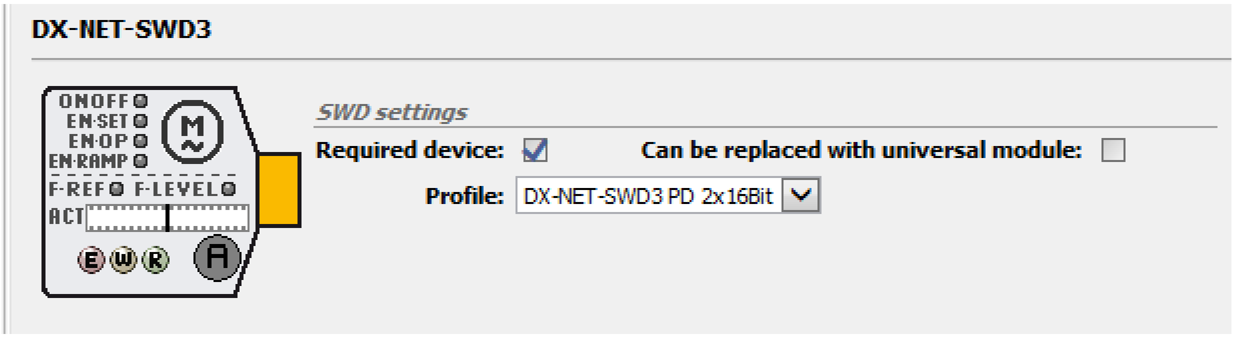


Inputs: 5 Byte			
Des.	**Meaning**	**Data type**	**Tag**
A1	Switch state Manual On/Off	BIT	EIPSWD:I.Data[5].2
A2	Switch state Manual/Auto	BIT	EIPSWD:I.Data[5].3
DIAG	Group diagnostics	BIT	EIPSWD:I.Data[5].4
PRSNT	Device present	BIT	EIPSWD:I.Data[5].6
SUBST	Universal module	BIT	EIPSWD:I.Data[5].7
RSO	Ready to switch on	BIT	EIPSWD:I.Data[6].0
RDY	Ready	BIT	EIPSWD:I.Data[6].1
EN	Enabled	BIT	EIPSWD:I.Data[6].2
ERR	Error	BIT	EIPSWD:I.Data[6].3
C_STOP	Coasting to a halt	BIT	EIPSWD:I.Data[6].4
Q_STOP	Quick stop	BIT	EIPSWD:I.Data[6].5
SOI	Switch on inhibited	BIT	EIPSWD:I.Data[6].6
WARN	Warning	BIT	EIPSWD:I.Data[6].7
F_REF	Running at Setpoint	BIT	EIPSWD:I.Data[7].0
CTL_REQ	Control requested	BIT	EIPSWD:I.Data[7].1
F_LEVEL	Actual speed exceeds level	BIT	EIPSWD:I.Data[7].2
Q1	Device Info Q1	BIT	EIPSWD:I.Data[7].3
Q2	Device Info Q2	BIT	EIPSWD:I.Data[7].4
Q3	Device Info Q3	BIT	EIPSWD:I.Data[7].5
Q4	Device Info Q4	BIT	EIPSWD:I.Data[7].6
Q5	Device Info Q5	BIT	EIPSWD:I.Data[7].7
ACTSPEED	Actual speed	INT	EIPSWD:I.Data[8]



Outputs: 4 Byte			
Des.	**Meaning**	**Data type**	**Tag**
ONOFF	On / Off	BIT	EIPSWD:O.Data[3].0
OFF2	Stop with C-Stop	BIT	EIPSWD:O.Data[3].1
OFF3	Stop with Q-Stop	BIT	EIPSWD:O.Data[3].2
EN_OP	Enable Operation	BIT	EIPSWD:O.Data[3].3
EN_RAMP	Enable ramp generator	BIT	EIPSWD:O.Data[3].4
UNFREEZE	Unfreeze ramp	BIT	EIPSWD:O.Data[3].5
EN_SET	Enable Setpoint	BIT	EIPSWD:O.Data[3].6
FAULTACK	Fault Acknowledge	BIT	EIPSWD:O.Data[3].7
JOG1	Jog with Setpoint 1	BIT	EIPSWD:O.Data[4].0
JOG2	Jog with Setpoint 2	BIT	EIPSWD:O.Data[4].1
CTL_PLC	Control by PLC	BIT	EIPSWD:O.Data[4].2
I1	Programmable Input 1	BIT	EIPSWD:O.Data[4].3
I2	Programmable Input 2	BIT	EIPSWD:O.Data[4].4
I3	Programmable Input 3	BIT	EIPSWD:O.Data[4].5
I4	Programmable Input 4	BIT	EIPSWD:O.Data[4].6
EXTFAULT	External fault	BIT	EIPSWD:O.Data[4].7
SETPOINT	Actual Setpoint	INT	EIPSWD:O.Data[5]

Also note the Profile that has been selected for this application example for the DC1 drive shown below. This is under the Device Parameters tab.



DX-NET-SWD3

SWD settings

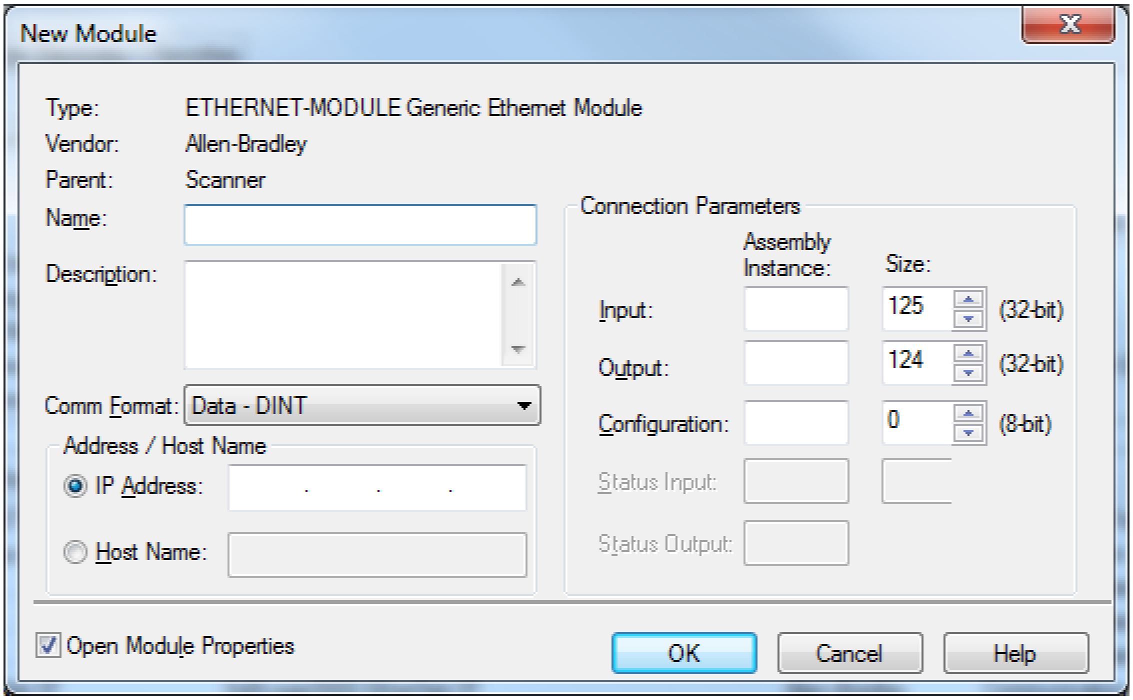
Required device: ☑ Can be replaced with universal module: □

Profile: DX-NET-SWD3 PD 2x16Bit

Adding the EU5C-SWD-EIP-MODTCP Ethernet gateway to the Ethernet/IP Network in RSLogix5000

For this example a 1756-L71 controller and 1756-EN2T Ethernet/IP scanner are used.

In RSLogix5000, after adding the 1756-EN2T to the Controller Organizer under the backplane, right click on Ethernet and select New Module. From the large list of Ethernet/IP slave devices, near the bottom is “Generic Ethernet Device” as shown under the Description column. Select it, then the create button and the following window will open.



New Module

Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: Scanner
Name:
Description:

Comm Format: Data - DINT

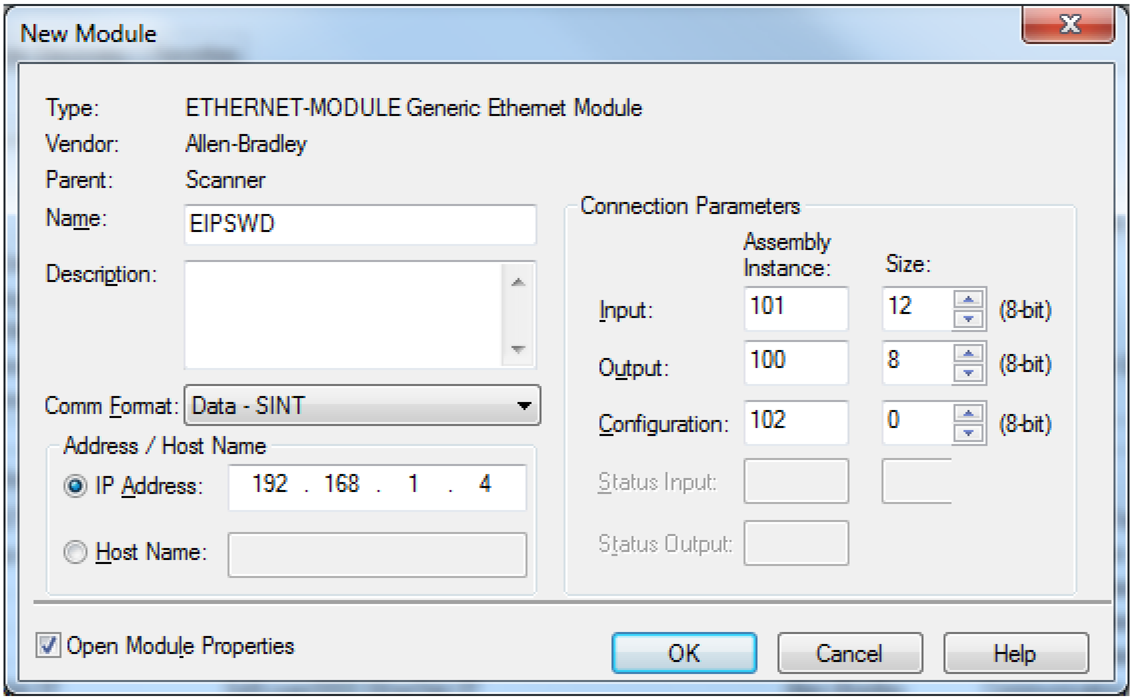
Address / Host Name
● IP Address: . . .
○ Host Name:

Connection Parameters

	Assembly Instance:	Size:	
Input:		125	(32-bit)
Output:		124	(32-bit)
Configuration:		0	(8-bit)
Status Input:			
Status Output:			

☑ Open Module Properties

The information for this window has already been provided as follows. Be sure to select “Data-SINT” (bytes) for the Comm Format.



New Module

Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: Scanner
Name: EIPSWD
Description:
Comm Format: Data - SINT

Address / Host Name
● IP Address: 192 . 168 . 1 . 4
○ Host Name:

Connection Parameters

	Assembly Instance:	Size:	
Input:	101	12	(8-bit)
Output:	100	8	(8-bit)
Configuration:	102	0	(8-bit)
Status Input:			
Status Output:			

☑ Open Module Properties

Click OK, then enter an RPI time on the Connection Tab and click OK again. Generic I/O tags will be created with the name “EIPSWD” in the Controller tag area.

Creating the CSV tag file in SWD-Assist and Importing it into RSLogix5000

In the SWD-Assist software, once the network is finalized and when in the offline mode, select the Ethernet gateway. Then under the Project drop down menu select “Export Ethernet/IP Configuration…”. You will be prompted for a name for the CSV file. Give it a name and save it to your hard drive. This name does not matter. The name for the tags in the file (EIPSWD) and the name given the gateway in RSLogix5000 (EIPSWD) must match for the tags to be properly imported and aliased to the generic I/O tags. Import this tag file into RSLogix5000, under Tools/Import/Tags and Logic Comments… There will now be descriptive I/O tags for each device on the Smartwire network in RSLogix5000 in the Controller Tags area. They may be used to monitor and control each device because they have been aliased to the generic tags.

Monitoring Motor Current and other Parameters

As noted earlier, motor current is not included in the cyclic Input data for the DC1 or DE1 on Smartwire. Only control bits and setpoint/speed reference along with status bits and actual speed are included. So any other data such as motor current must be read using explicit messages with imbedded Smartwire commands. This paper will demonstrate how to accomplish this and provide a sample RSLogix5000 program.

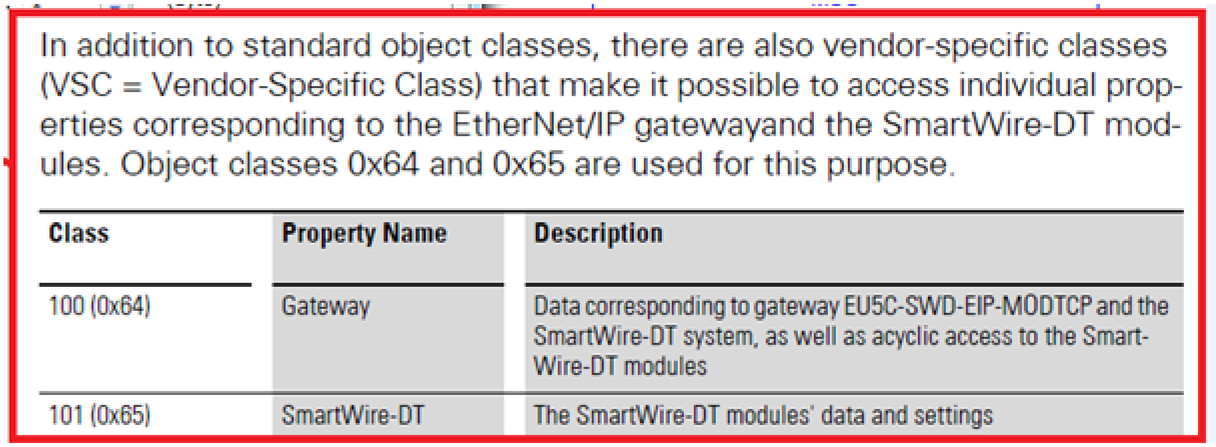
Note: The RSLogix5000 program was developed using version 24. The program may need to be converted when importing into other versions of the software or if a different controller is used. The software will prompt the user for this.

Reading Motor Current or Any Other Supported Parameter

There are 8 total message instructions required in RSLogix5000 to monitor motor current from each DE1 or DC1. They must be executed in the order shown below each time the motor current is to be updated.

The description for the parameters of the message instructions are as follows:

Class: 65



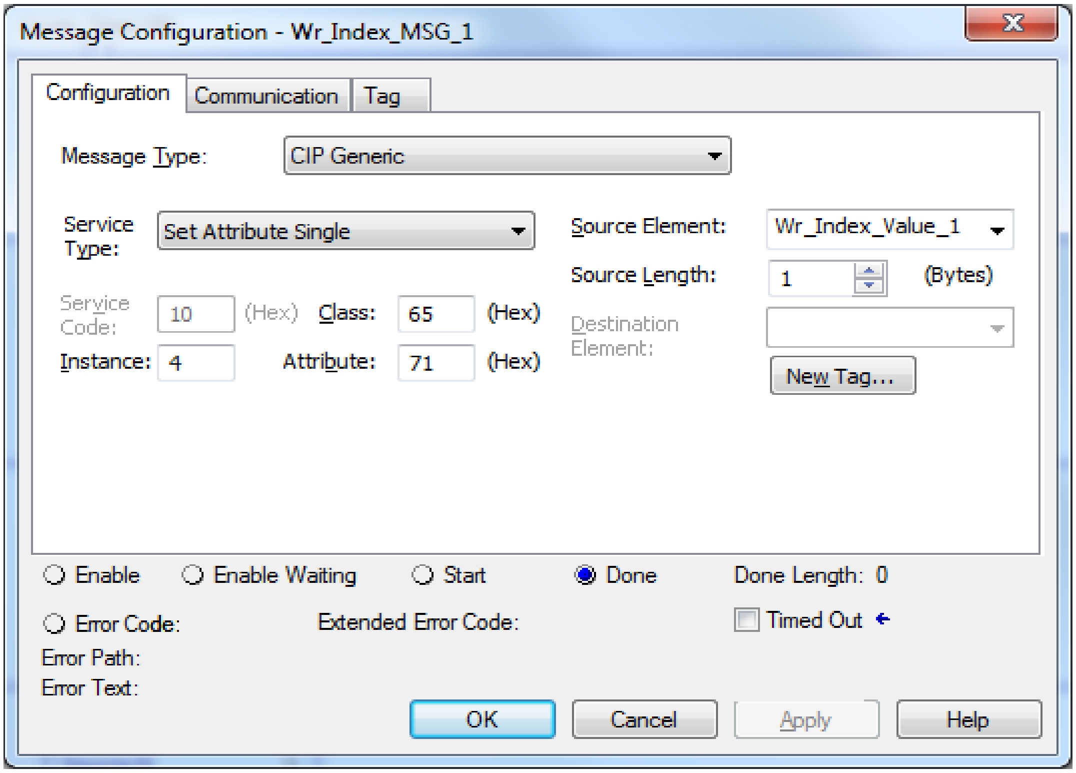
In addition to standard object classes, there are also vendor-specific classes (VSC = Vendor-Specific Class) that make it possible to access individual properties corresponding to the EtherNet/IP gatewayand the SmartWire-DT modules. Object classes 0x64 and 0x65 are used for this purpose.

Class	Property Name	Description
100 (0x64)	Gateway	Data corresponding to gateway EU5C-SWD-EIP-MODTCP and the SmartWire-DT system, as well as acyclic access to the Smart-Wire-DT modules
101 (0x65)	SmartWire-DT	The SmartWire-DT modules' data and settings

Instance: 4 for this example. This is the Smartwire node address of the device (DC1).

Attribute: Node Index Object. Note the values used for each Message instruction are in the sample program and shown below for each Message Instruction.

1. A Set Attribute Single message, writing one byte of data is to be sent first, per the screen shot below. The value of the 1 byte of data is 47 decimal.



Message Configuration - Wr_Index_MSG_1

Message Type: CIP Generic

Service Type: Set Attribute Single

Service Code: 10 (Hex) Class: 65 (Hex)

Instance: 4 Attribute: 71 (Hex)

Source Element: Wr_Index_Value_1

Source Length: 1 (Bytes)

Destination Element:

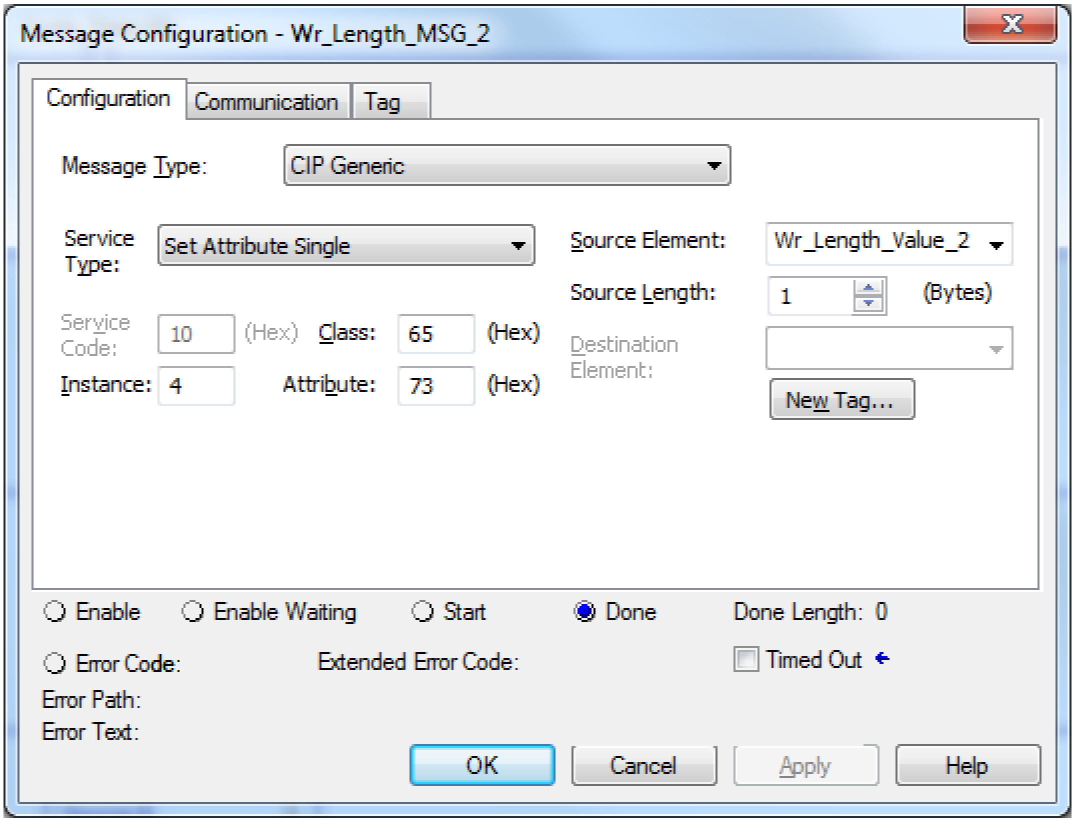
○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out ←

Error Path:

Error Text:

2. Another Set Attribute Single message, sending 1 byte of data containing the length of the subsequent Smartwire command. The length is 10 bytes.



Message Configuration - Wr_Length_MSG_2

Message Type: CIP Generic

Service Type: Set Attribute Single

Service Code: 10 (Hex) Class: 65 (Hex)

Instance: 4 Attribute: 73 (Hex)

Source Element: Wr_Length_Value_2

Source Length: 1 (Bytes)

Destination Element:

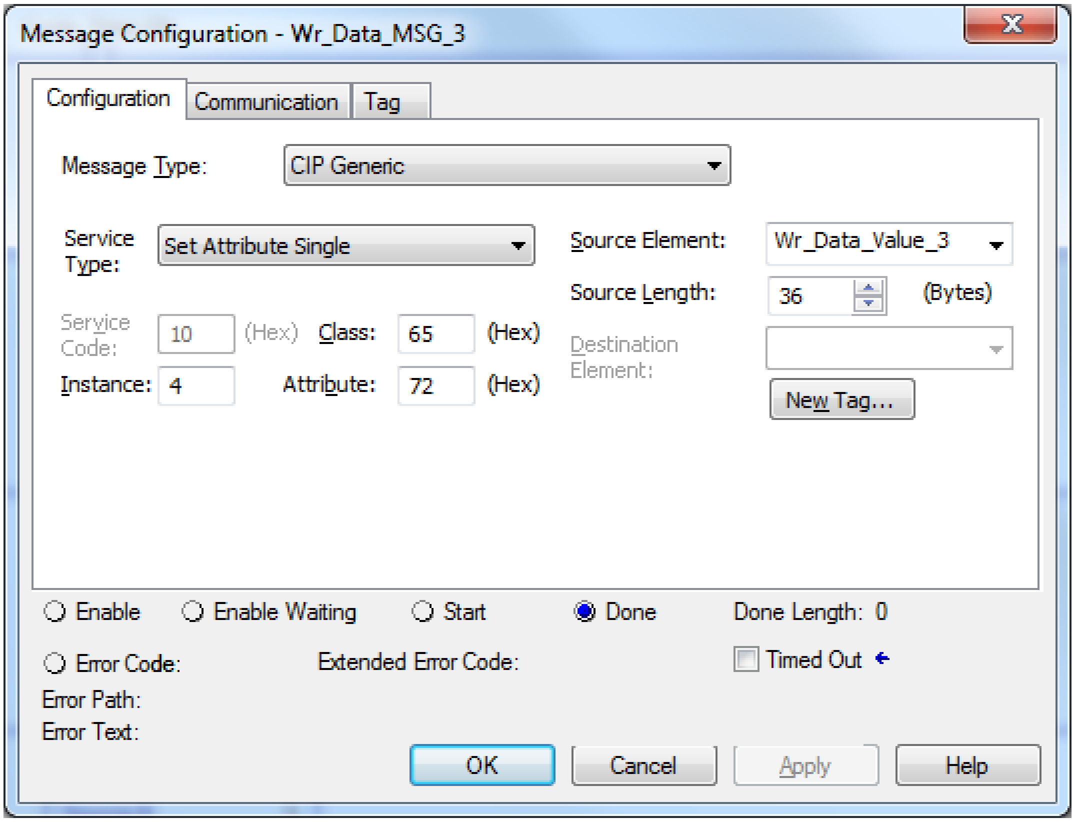
○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

3. Another Set Attribute Single sending 36 bytes, containing a 10 byte Smartwire command. The message is below, followed by the command data.



Message Configuration - Wr_Data_MSG_3

Message Type: CIP Generic

Service Type: Set Attribute Single

Source Element: Wr_Data_Value_3

Source Length: 36 (Bytes)

Service Code: 10 (Hex) Class: 65 (Hex)

Destination Element:

Instance: 4 Attribute: 72 (Hex)

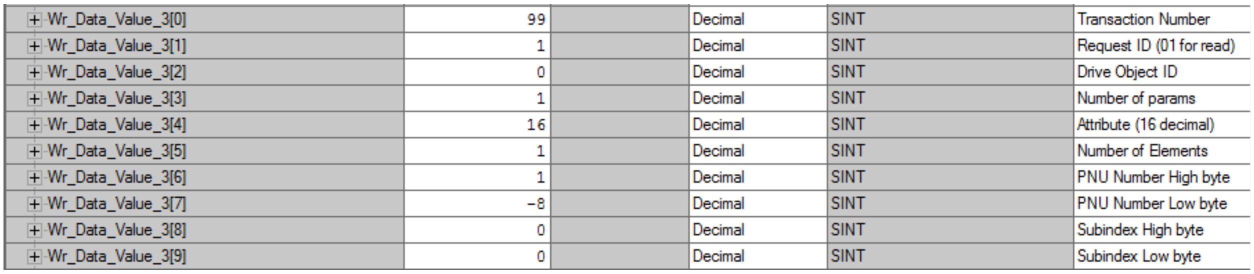
○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

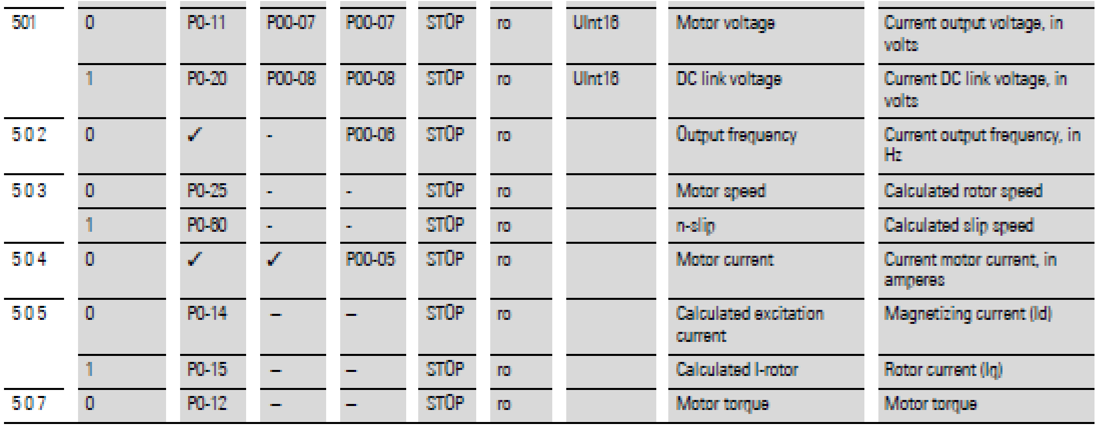
Command written to the DC1 drive for this example, to set it up to read the motor current in a subsequent message read. Refer to page 77 in publication MN04012009Z for a complete description of each value.



+ Wr_Data_Value_3[0]	99		Decimal	SINT	Transaction Number
+ Wr_Data_Value_3[1]	1		Decimal	SINT	Request ID (01 for read)
+ Wr_Data_Value_3[2]	0		Decimal	SINT	Drive Object ID
+ Wr_Data_Value_3[3]	1		Decimal	SINT	Number of params
+ Wr_Data_Value_3[4]	16		Decimal	SINT	Attribute (16 decimal)
+ Wr_Data_Value_3[5]	1		Decimal	SINT	Number of Elements
+ Wr_Data_Value_3[6]	1		Decimal	SINT	PNU Number High byte
+ Wr_Data_Value_3[7]	-8		Decimal	SINT	PNU Number Low byte
+ Wr_Data_Value_3[8]	0		Decimal	SINT	Subindex High byte
+ Wr_Data_Value_3[9]	0		Decimal	SINT	Subindex Low byte

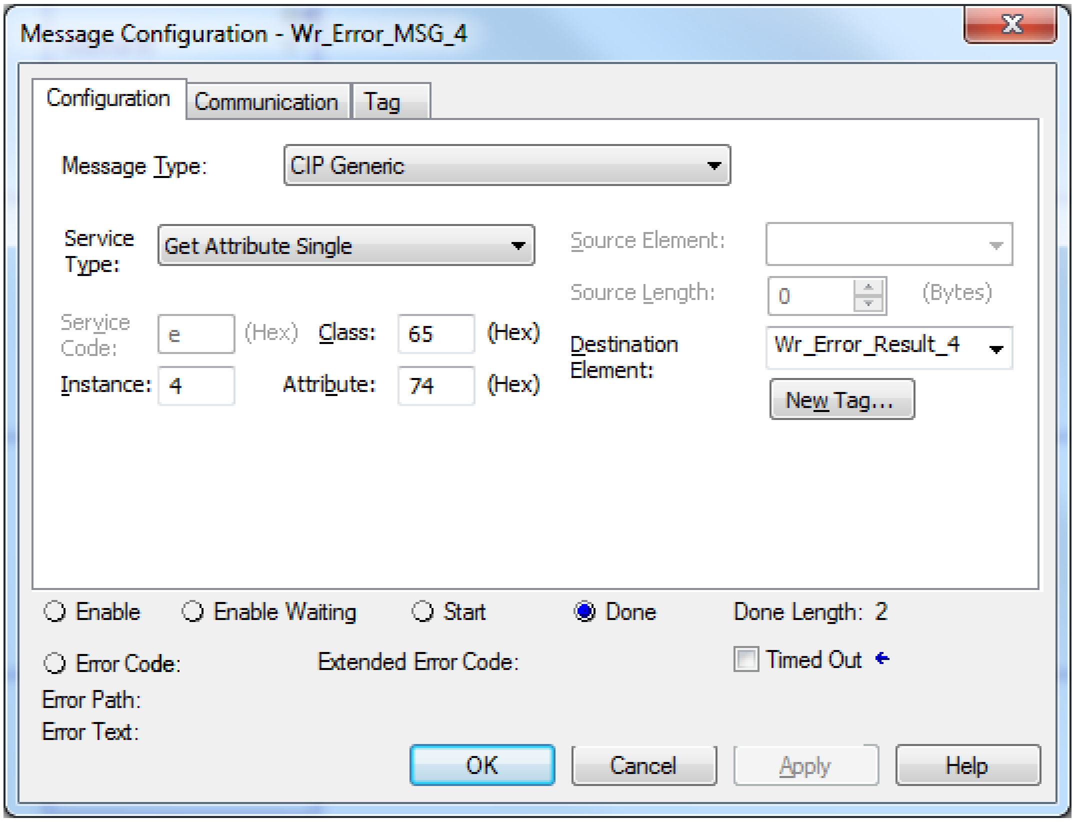
The PNU Index number is from the DE1/DC1 Smartwire manual. Note which is the high byte and which is the low byte. The program provided with this application note uses the SWPB and COP instructions to take care of this byte order, so PNU Index numbers and where necessary, subindex numbers can be entered as an integer (decimal) value. The program also puts the 2 bytes of data returned into an integer value that properly displays the actual value returned. In the case of motor current, the program puts the data into an integer and also takes care of scaling. The motor current is scaled by a factor of 10.

In the command shown above to read motor current, the PNU Index value high byte is 1 and the low byte is -8. When these values are put into an integer word in the correct order, the value is 504 decimal, which is the PNU Index number for Motor Current as shown in the DE1/DC1 Smartwire user manual, page 59 as shown below.



501	0	P0-11	P00-07	P00-07	STOP	ro	UInt16	Motor voltage	Current output voltage, in volts
	1	P0-20	P00-08	P00-08	STOP	ro	UInt16	DC link voltage	Current DC link voltage, in volts
502	0	✓	-	P00-06	STOP	ro		Output frequency	Current output frequency, in Hz
503	0	P0-25	-	-	STOP	ro		Motor speed	Calculated rotor speed
	1	P0-60	-	-	STOP	ro		n-slip	Calculated slip speed
504	0	✓	✓	P00-05	STOP	ro		Motor current	Current motor current, in amperes
505	0	P0-14	–	–	STOP	ro		Calculated excitation current	Magnetizing current (Id)
	1	P0-15	–	–	STOP	ro		Calculated I-rotor	Rotor current (Iq)
507	0	P0-12	–	–	STOP	ro		Motor torque	Motor torque

4. This message reads the Error code. This value will be 0 unless a problem occurs. The message instruction is shown below.



Message Configuration - Wr_Error_MSG_4

Message Type: CIP Generic

Service Type: Get Attribute Single

Service Code: e (Hex) Class: 65 (Hex)

Instance: 4 Attribute: 74 (Hex)

Source Element:

Source Length: 0 (Bytes)

Destination Element: Wr_Error_Result_4

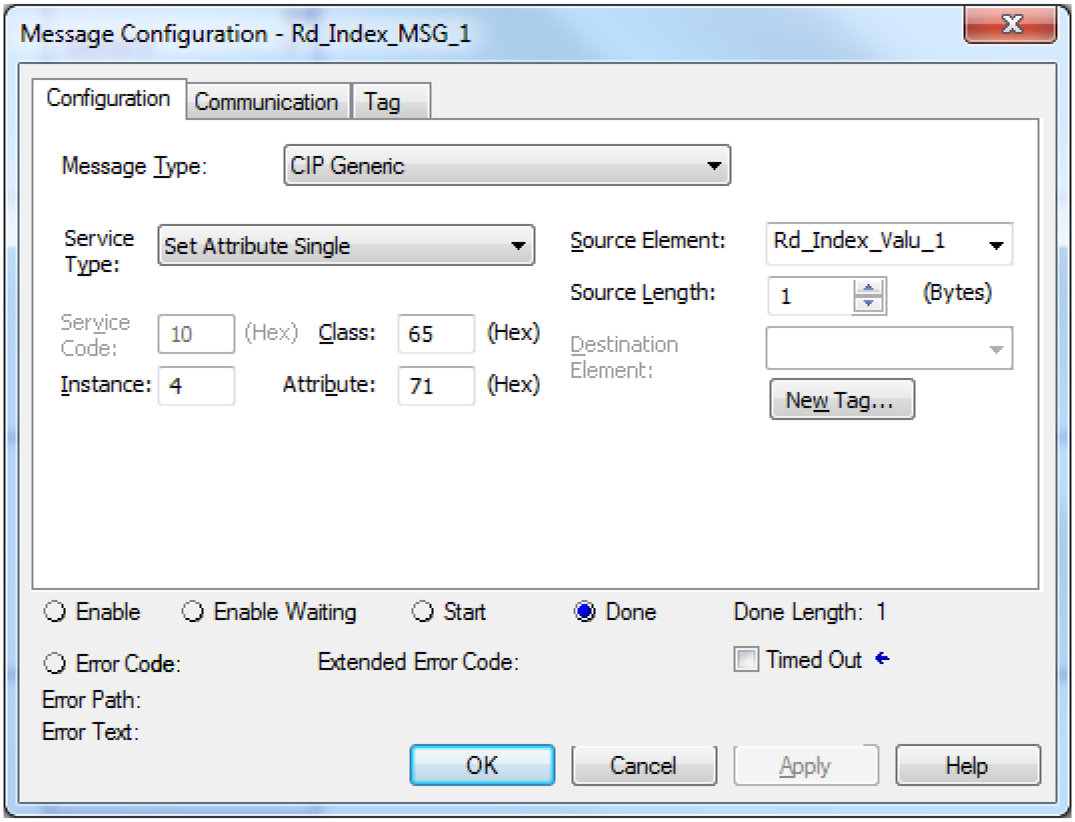
○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 2

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

5. This is a message read or Get Attribute Single to read the Index value to verify it is 47 decimal. The message instruction is shown below.



Message Configuration - Rd_Index_MSG_1

Message Type: CIP Generic

Service Type: Set Attribute Single

Service Code: 10 (Hex) Class: 65 (Hex)

Instance: 4 Attribute: 71 (Hex)

Source Element: Rd_Index_Valu_1

Source Length: 1 (Bytes)

Destination Element:

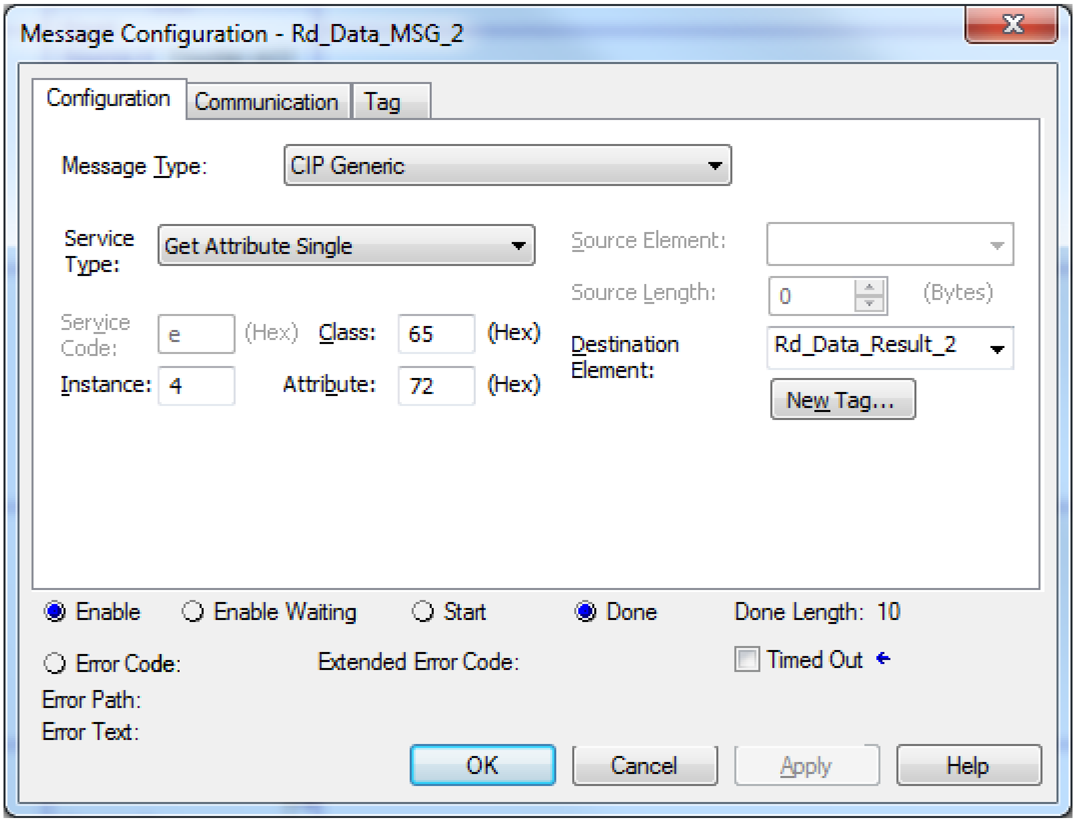
○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 1

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

6. This message read or Get Attribute single reads the message length to verify it is 10 bytes. The message instruction is shown below.



Message Configuration - Rd_Data_MSG_2

Message Type: CIP Generic

Service Type: Get Attribute Single

Service Code: e (Hex) Class: 65 (Hex)

Instance: 4 Attribute: 72 (Hex)

Source Element:

Source Length: 0 (Bytes)

Destination Element: Rd_Data_Result_2

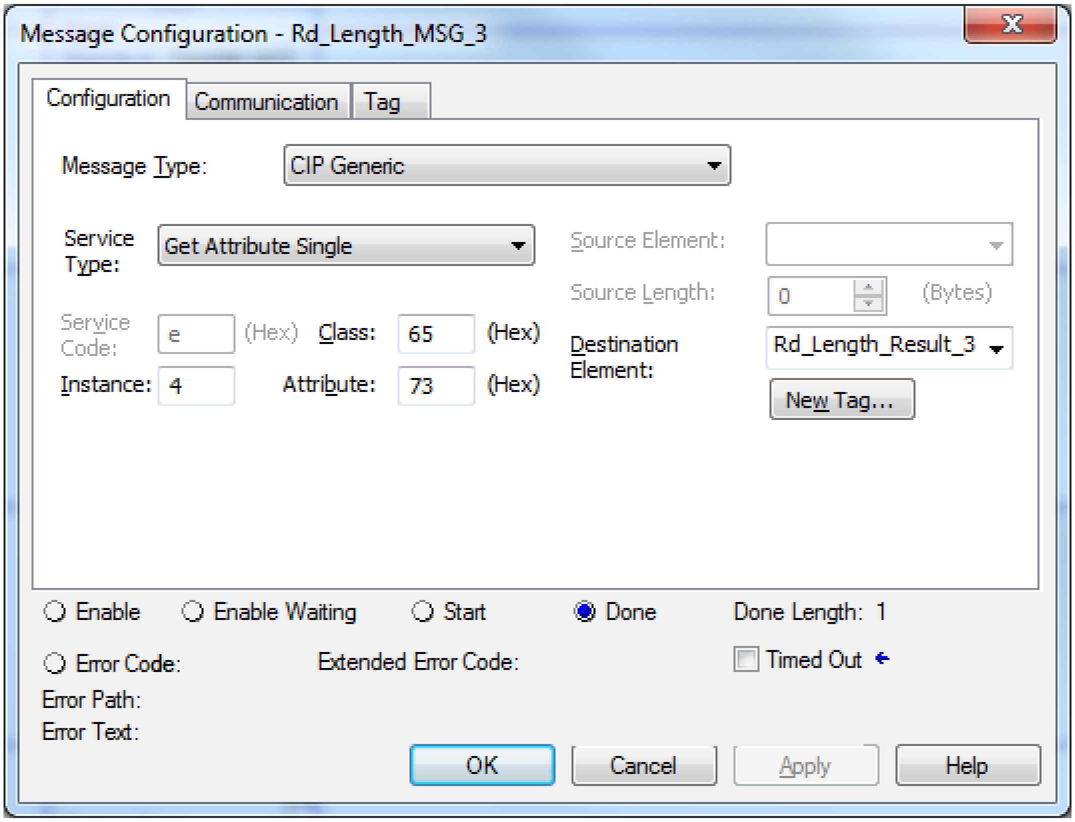
● Enable ○ Enable Waiting ○ Start ● Done Done Length: 10

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

7. This message read or Get Attribute single reads the command reply, which is 10 bytes long as follows. The message instruction is shown below.



Message Configuration - Rd_Length_MSG_3

Message Type: CIP Generic

Service Type: Get Attribute Single

Service Code: e (Hex) Class: 65 (Hex)

Instance: 4 Attribute: 73 (Hex)

Source Element:

Source Length: 0 (Bytes)

Destination Element: Rd_Length_Result_3

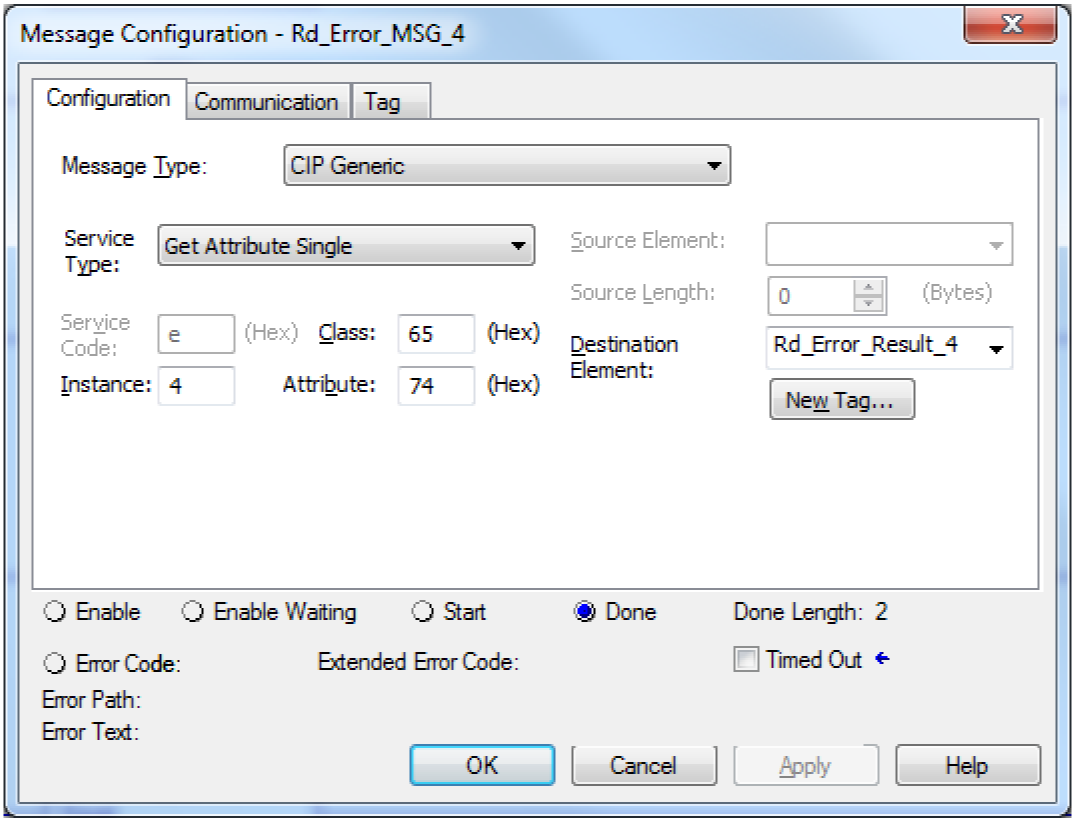
○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 1

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

8. This message reads the Error code. This value will be 0 unless a problem occurs. The message instruction is shown below.



Message Configuration - Rd_Error_MSG_4

Message Type: CIP Generic

Service Type: Get Attribute Single

Service Code: e (Hex) Class: 65 (Hex)

Instance: 4 Attribute: 74 (Hex)

Source Element:

Source Length: 0 (Bytes)

Destination Element: Rd_Error_Result_4

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 2

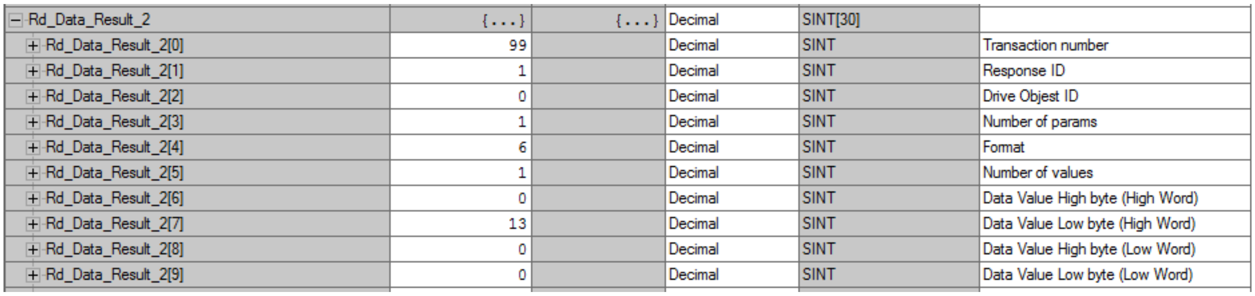
○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

The result of the Message read that is requesting the Motor Current (PNU Index 504) or any other parameter (PNU Index number) is shown in the RD_Data_Result in the attached program. This tag includes the command header information as well as the raw data for the requested PNU Index number.

The 10 byte reply is shown below. Refer to page 78 in publication MN04012009Z for a complete description of each value.



− Rd_Data_Result_2	{...}	{...}	Decimal	SINT[30]	
+ Rd_Data_Result_2[0]	99		Decimal	SINT	Transaction number
+ Rd_Data_Result_2[1]	1		Decimal	SINT	Response ID
+ Rd_Data_Result_2[2]	0		Decimal	SINT	Drive Objest ID
+ Rd_Data_Result_2[3]	1		Decimal	SINT	Number of params
+ Rd_Data_Result_2[4]	6		Decimal	SINT	Format
+ Rd_Data_Result_2[5]	1		Decimal	SINT	Number of values
+ Rd_Data_Result_2[6]	0		Decimal	SINT	Data Value High byte (High Word)
+ Rd_Data_Result_2[7]	13		Decimal	SINT	Data Value Low byte (High Word)
+ Rd_Data_Result_2[8]	0		Decimal	SINT	Data Value High byte (Low Word)
+ Rd_Data_Result_2[9]	0		Decimal	SINT	Data Value Low byte (Low Word)

Motor Current

In the screen shot above, the motor current was requested (PNU Index 504). The motor current is in tags 6 and 7 above, where 7 is the low byte. The value of 13 is scaled by a factor of 10. The actual motor current in this case is 1.3 amps. In the attached program, there is a Motor_Current_Amps tag which is a Real tag that contains the motor current with the 2 bytes put together in the correct order and the scaling applied. The result for this example is 1.3 amps.

Reading 16 bit Integer Values

When reading 16 bit integer values such as Motor Voltage (PNU Index number 501, subindex 0), the data is received in bytes 6 and 7 as shown above. Bytes 8 and 9 are not used. In all cases, for 16 bit integer values, bit 7 is the low byte and bit 8 the high byte. The sample program provides the data responses for 16 bit integer data values in tag: PNU_Actual_Data_INT.

Reading 32 bit Double Integer Values

When reading 32 bit double integer values, such as t-Run (the drives total operating time in hours, minutes, seconds – PNU Index number 821, subindex 0), bytes 6 and 7 comprise the high word and bytes 8 and 9 comprise the low word. Byte 7 is the low byte of the high word and byte 6 the low byte. In addition, byte 9 is the low byte of the low word and byte 8 the high byte. The sample program provides the data responses for 32 bit double integer data values in tag: PNU_Actual_Data_DINT.

The Sample Program Description

The attached sample program does not use the descriptive tag file for the cyclic I/O data. It simply writes to the control word and the speed reference word to operate the DC1 drive and motor. The motor used is an unloaded 1.3 HP motor that draws 1.3 amps at 100% speed (60Hz). The program operates as follows:

1. At power up or when put into the Run mode, it immediately sends a 047E to the drive. This is a Stop command.

2. After a 2 second delay, it then sends a 047F command to instruct the DC1 drive to Run along with a value of 4000 hex or 16384 decimal for the speed reference to instruct the drive to run the motor at 100% speed or 60Hz.

3. The 8 message instructions described above are there to read any PNU Index value supported by the product (DE1 or DC1). The PNU number can be inserted in decimal into the Tag: PNU_Index. The Sub Index number for a specific PNU Index number may be entered into tag: PNU_Subindex. Examples are PNU Index number 504 for Motor Current and 501 for Motor Voltage. The Subindex for these two values are 0.

4. The program executes the 8 message instructions every 10 seconds and during each cycle it executes the 8 message instructions 1 second apart.

5. If Motor Current is being read, the Motor_Current_Amps tag should be used to view the current in amps. If any other PNU Index value is read, the data should be

viewed using tag PNU-Actual_Data_INT for a 16 bit integer value and tag PNU_Actual_Data_DINT for a 32 bit double integer value. The motor current is scaled and the result in the PNU_Actual_Data tag for it is the raw value. The value for the current in tag Motor_Current_Amps has been divided by 10, the scale factor so the result in the Real (floating point) tag is in amps.

References

DC1/DE1 Smartwire User Manual, Publication MN04012009Z-EN
DC1 variable frequency drive: MN04020003Z-EN
DE1 variable speed starter: MN040011EN

Additional Help

In the US or Canada: please contact the Technical Resource Center at 1-877-ETN-CARE or 1-877-326-2273 option 2. Or, email to: www.TRCAutomation.com

All other supporting documentation is located on the Eaton web site at www.eaton.com/drives

Eaton
1000 Eaton Boulevard
Cleveland, OH 44122 USA
Eaton.com

Printed in USA
Publication No. AP040061EN
January 2017

Powering Business Worldwide