# **PowerXL DG1 to XV HMI by Modbus TCP**

#### Introduction

The purpose of this application note is to demonstrate how to operate a PowerXL DG1 drive via Modbus TCP and an XV-102 CoDeSys-3 controller. The PowerXL drive has an onboard Ethernet port that supports both Ethernet/IP and Modbus TCP. The drive needs to be configured with an IP address for Ethernet communications, but it auto senses the protocol. In other words, as long as the Modbus TCP master polls the PowerXL DG1 drive reading and writing valid Modbus data addresses, the drive will respond.

The IP address for the drive can be set via the display on the front of the drive. This will be described later in this document.

While this application example uses an Eaton XV-102 HMI/PLC to control and monitor the PowerXL drive over Modbus TCP, any Modbus TCP master may be used for this purpose. Eaton's XSoft-CoDeSys, version 3.5.4 programming software is used to create the XV-102 project. The XV-102 project is used to configure the controller to poll the PowerXL DG1 drive for control and monitoring purposes.

Modbus data addresses are published for the DG1 drive in the Modbus TCP section of the PowerXL DG1 drive's Communication Manual, publication MN040010EN.

## Configuring the PowerXL DG1 Drive

The IP addresses for the devices used in this example will be as follows:

- PowerXL Drive: 192.168.1.2
- XV102 HMI/PLC: 192.168.1.8
- Computer: 192.168.1.51

Subnet mask: 255.255.255.0

Connect your computer, PLC and the PowerXL drive to an Ethernet switch.

To configure the IP address of the PowerXL DG1 drive via its keypad/display, power the drive and navigate to the Active IP Address to set the IP address for the drive. As noted above, the IP address for the drive in this example is 192.168.1.2.

To access the Ethernet parameters from the drives keypad/display:

- 1. Start by selecting Parameters with the up/down arrow keys.
- 2. Then press the right arrow key which will display Basic Parameters
- 3. Press the up arrow key twice until Communications is displayed
- 4. Press the right arrow key once, then the down arrow key twice so Ethernet/IP/Modbus TCP is displayed
- 5. Press the right arrow key to access the following Ethernet parameters:
  - a. Address Mode, select Static IP



- b. Active IP Address, 192.168.1.2 is entered for this example
- c. Active Subnet Mask, 255.255.255.0 is entered for this example

Power cycle the drive to activate these new Ethernet parameters.

Refer to the Communication manual for this drive for additional details. Publication MN040010EN.

The drive must also be configured to accept remote control and speed via the network.

- Set Remote1 Control Place to: Fieldbus
- Set Remote1 Reference to: Fieldbus Ref

Both of these parameters can be found under: Parameters / Basic Parameters from the drive's key-pad/display.

#### Change the IP Address of your computer

To change the IP address for a computer running Windows 7, follow the procedure below:

- 1. From the Start menu, choose Control Panel. From the Control Panel, choose Network and Sharing Center.
- 2. With the computer connected to an Ethernet network, select the Local Area Connection. Unless the computer is connected to a network, this Local Area Connection will not be present.
- 3. The Local Area Connection Status window will be displayed. Select Properties.
- 4. From the window shown below, select Internet Protocol Version 4 (TCP/IPv4) to highlight it, then select Properties.

Local Area Connection Properties	23
Networking Authentication Sharing	
Connect using:	
Intel(R) 82579LM Gigabit Network Connection	
Configure	
This connection uses the following items:	
<ul> <li>Trend Micro LightWeight Filter Driver</li> <li>QoS Packet Scheduler</li> <li>File and Printer Sharing for Microsoft Networks</li> <li>Internet Protocol Version 6 (TCP/IPv6)</li> <li>Internet Protocol Version 4 (TCP/IPv4)</li> <li>Internet Protocol Version 9 (TCP/IPv4)</li> <li>Link-Layer Topology Discovery Mapper I/O Driver</li> <li>Link-Layer Topology Discovery Responder</li> </ul>	
Install Uninstall Properties	
Description Allows your computer to access resources on a Microsoft network.	
ОК Саг	ncel

5. Per the following window, select Use the following IP Address, then enter an IP address, Subnet mask and a Default gateway if it applies.

nternet Protocol Version 4 (TCP/IPv4) Properties							
General							
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.							
Obtain an IP address automatical	y						
• Use the following IP address:							
IP address:	192.168.1.51						
Subnet mask:	255.255.255.0						
Default gateway:	192.168.1.1						
Obtain DNS server address autom	atically						
<ul> <li>Use the following DNS server add</li> </ul>	resses:						
Preferred DNS server:	· · ·						
Alternate DNS server:	· · ·						
Validate settings upon exit	Advanced						
	OK Cancel						

6. When finished, select OK and close all the windows used along the way. Your computer's Ethernet port will now be actively using the IP address and Subnet mask you just entered.

## Creating a Project in XSoft-CoDeSy 3.5.4

Create a project in XSoft-CoDeSys 3.5.4. Give the project a name and select the controller type and programming language per the following:

You are about within this pro - One program - A program Pl - A cyclic task - A reference	to create a new standard project. This wiz ject: mable device as specified below LC_PRG in the language specified below which calls PLC_PRG every 20 milliseconds to the newest version of the Standard libra	ard will create the following objec	ts
Device: PLC_PRG in:	XV100 (Eaton Automation) Ladder Logic Diagram (LD)	OK Cancel	•

Select OK to create the project.

Note that an XC-152 or XC-202 PLC can also be used. Both of these CoDeSys controllers also have an Ethernet port that supports Modbus TCP and Ethernet/IP. The same project can be used by simply changing the controller type. An XV-102 (XV100) controller was used for this example.

Note also that CoDeSys supports 6 different programming languages and any of them can be used, including: CFC, FBD, IL, LD, SFC and ST. Ladder Logic (LD) is used for this example.

## Creating a Modbus TCP Network in XSoft-CoDeSys 3.5.4

On the left portion of the project screen in XSoft-CoDeSys, right click on "Device (XV100)" and select Add Device. The following screen will open:

Add Devi	ce					23
Name:						
Action:						
Action:	d davisa 🖉 Tasast da	vice 🔿 Dhu	a davica 🔘	Updata device		
Append	device Insert de	vice 🔘 Piug	g device 🔘	Opdate device		
Device:						
Vendor:	<all vendors=""></all>					
Name		Vendor	Version			
🖃 👘 Fi	ieldbusses					
😟 - CA	CANbus					
	Ethernet Adapter					
🕀 🗰	Modbus					
	📅 Profibus					
📄 🖻 · 👔	SmartWire-DT					
Group	by category					
Display	/ all versions (for experi	ts only)				
Display	outdated versions					
-Information	n:					
🔍 (You	can select another targ	jet node in the	a navigator wi	hile this window is	open.)	
					Add Dovico	Class

Select the plus sign to the left of Ethernet Adapter. Then select Ethernet below it and select the Add Device button. Don't close the Add Device Window yet.

Double click Ethernet that is now displayed at the bottom of the tree on the left and the Add Device Window will change as follows:

File Eart View Project Build Online Debug Tools Win		
🗎 🚅 📕   🎒   い つ よ 階 🋍 🗙   🐴 🕼   🖺   🦄	Add Device	21
	Name: Ethernet	
Devices v A X		
ModbusTCO XV DG1	Action:	
Device (XV100)	Append device     Insert device     Plug device     Update device	
DI PLC Logic (D	- Device:	
🖹 💮 Application	Vendor: <a>All vendors&gt;</a>	
👘 Library Manager	Name Vertin	
PLC_PRG (PRG)		
Task Configuration		
🖻 😻 MainTask		
PLC_PRG		
Ethernet (Ethernet)		
	Group by category	
	Display all versions (for experts only)	
	Display outdated versions	
	Information:	
	(You can select another target node in the navigator while this window is open.)	
	Add Device Close	
		-

Select the plus sign to the left of Modbus in the Add Device Window then the plus sign in front of Modbus TCP Master as well. Then select Modbus TCP Master followed by the Add Device button to add it to the tree on the left as shown below. Do not close the Add Device Window yet.

Devices 🗸 🗸 🗙	Ethernet X						
ModbusTCO_XV_DG1	Add Device	23					
Device (XV100)							
	Name: Modbus_TCP_Master_1						
	Action:						
	Annend device     Tosert device     Plug device     Indate device						
AinTask	Device:						
PLC_PRG	Vendor: <all vendors=""></all>	<b>•</b>					
Ethernet (Ethernet)	Name Vendor Version						
Modbus_TCP_Master (Modbus TCP Master)	E- I Fieldbusses						
	🖶 🖨 EtherNet/IP						
	⊟ KIIIS Modbus						
	🖨 - 🎟 Modbus TCP Master						
	Modbus TCP Master 35 - Smart Software Solutions GmbH 3.5.4.0						
	🕀 - 📖 ModbusTCP Slave Device						
	Group by category						
	Display all versions (for experts only)						
	Information:						
	Append selected device as last child of Ethernet						
	(You can select another target node in the navigator while this window is open.)						
	Add Device	Close					

Double click the Modbus TCP Master on the left and select the Modbus TCP Slave under Modbus/Modbus TCP Slave on the Add Device Window as follows to add the Modbus slave to the tree on the left:

Devices 👻 🕂 🗙	Ethernet Modbus_TCP_Master X
ModbusTCO_XV_DG1      Device (XV100)      PLC Logic      Optimized Application      PLC_PRG (PRG)      PLC_PRG (PRG)      Task Configuration      Standard MainTask	Add Device     X       Name:     Modbus_TCP_Slave_1       Action:     Append device       Plug device     Update device
Ethernet (Ethernet)     Modbus_TCP_Master (Modbus TCP Master)     Modbus_TCP_Slave (Modbus TCP Slave)	Vendor: <a>Ivendors&gt;</a> Name     Vendor     Version       Image: Image of the second
	Image: Constraint of the second se

Now close the Add Device Window. The Ethernet port on the XV HMI/PLC will be the master and the PowerXL DG1 Drive will be the Modbus slave.

Double click the Modbus TCP Master in the tree to open its configuration pages on the right. Select the ModbusTCP Master Configuration tab. Select "auto-reconnect". This will allow the Modbus TCP Master to re-connect automatically following a loss of communications, once the communication issue has been resolved.

ModbusTCP Master Configu	ration	ModbusTCPMaster I/O Mapping	ModbusTCPMaster Configuration	Stat
Modbus-TCP			MODRIE	V
Response Timeout (ms)	1000	×		
Socket Timeout (ms)	10	×		
✓ auto-reconnect				

Next double click the Modbus TCP Slave on the tree to open its configuration pages. Select the ModbusTCP Slave tab and configure the following:

ModbusTCP Slave Modbus Slave (	hannel	Mo	odbu	is Sl	ave	Init	M	odbusTCPSlave Configu	ration	ModbusTCPSlave I
Modbus-TCP								M	OD	BUS
Slave IP Address:	192	•	168	•	1	÷	2			
Unit-ID [1247]	1									
Response Timeout (ms)	1000	)								
Port	502									

Then select the Modbus Slave Channel tab. Select the Add Channel button at the bottom of that screen and the following window will be displayed:

ModbusChannel	2X
Channel	
Name	Channel 0
Access Type	Read Holding Registers (Function Code 3)
Trigger	Cyclic   Cycle Time (ms) 100
Comment	
READ Register	
Offset	0x0000 👻
Length	1
Error Handling	Keep last Value
WRITE Register	
Offset	0x0000 👻
Length	1
	OK Cancel

Two Modbus Channels will be added to this project allowing it to control the On/Off state of the drive as well as speed and monitor the Running status and the Actual Speed.

First configure the Read message. Per the PowerXL DG1 Communication Manual, the Status Word, General Status Word and Actual Speed begin at Modbus data address 2100 decimal or 0834 Hexadecimal. The Read Modbus Channel should look like the following:

ModbusChannel		23
Channel		
Name	Channel 0	
Access Type	Read Holding Registers (Function Code 3)	•
Trigger	Cyclic   Cycle Time (ms) 100	
Comment		
READ Register		
Offset	0x0834	-
Length	3	
Error Handling	Keep last Value	
-WRITE Register		
Offset	0x0000 🔹	·
Length	0	
	OK Cano	el

Select the OK button to add it.

Select the Add Channel button again and fill in the following for the Write message of 3 registers in length that includes the Control Word, the General Control Word and the Speed Reference. The starting Modbus data address is 2000 decimal or 07D0 hexadecimal.

ModbusChannel	X
Channel	
Name	Channel 1
Access Type	Write Multiple Registers (Function Code 16)
Trigger	Cyclic   Cycle Time (ms) 100
Comment	
READ Register	
Offset	· · · · · · · · · · · · · · · · · · ·
Length	3
Error Handling	Keep last Value 👻
WRITE Register	
Offset	2000 👻
Length	3
	OK Cancel

The Modbus Slave Channel tab should now look like the following:

ModbusTCP Slave Modbus Slave Channel Modbus Slave Init ModbusTCPSlave Configuration ModbusTCPSlave I/O Mapping Status Information									
Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offset	Length	Comment	
Channel 0	Read Holding Registers (Function Code 03)	CYCLIC, t#100ms	16#0834	3	Keep last Value				
Channel 1	Write Multiple Registers (Function Code 16)	CYCLIC, t#100ms				16#07D0	3		

Select the ModbusTCPSlave I/O Mapping tab. This is where the 3 input and 3 output registers are shown per the following:

ModbusTCP Slave	Modbus Slave Channel	Modbus Slave Init	ModbusTCPSla	ave Configuration	juration ModbusTCPSlave I/O Mapping		Status	Information	
Channels									
Variable	Mappi	ng Channel	Address	Туре		Default Value	Unit	Description	
<b>⊨</b> ¥≱		Channel 0	%IW0	ARRAY [02] OF WORD				Read Holding Registers	
😟 ᡟ		Channel 0[0]	%IW0	WORD				READ 16#0000 (=00000)	
😟 ᡟ		Channel 0[1]	%IW2	WORD				READ 16#0001 (=00001)	
😟 ᡟ		Channel 0[2]	%IW4	WORD				READ 16#0002 (=00002)	
<u> </u>		Channel 1	%QW0	ARRAY [02] 0	F WORD			Write Multiple Registers	
😟 Kø		Channel 1[0]	%QW0	WORD				WRITE 16#07D0 (=020	
<u>ن</u> ۲		Channel 1[1]	%QW2	WORD				WRITE 16#07D1 (=020	
😟 Kø		Channel 1[2]	%QW4	WORD				WRITE 16#07D2 (=020	

Descriptive variable names can now be added to these generic I/O tags. These descriptive variables can then be accessed in the program and from visualization screens. The descriptive variable names can be applied to entire words or individual bits as follows:

ModbusTCP Slave Modbus Slave C		nel Mod	bus Slave Init ModbusTCPSlave Configuration		ModbusTCPSlave I/O Mapping		Status	Information		
Channels										
Variable Mapping		oping	Channel	Address	Туре		Default Value	Unit	Description	
📮 🍫			Channel 0	%IW0	ARRAY [02] OF WORD				Read Holdir	1g Registers
🖨 ᡟ			Channel 0[0]	%IW0	WORD				READ 16#0	834 (=02100)
ᡟ R	eady	*	Bit0	%IX0.0	BOOL		FALSE			
- 🁋 R	unning	*	Bit1	%IX0.1	BOOL		FALSE			
🏷 F	WD_REV	*	Bit2	%IX0.2	BOOL		FALSE			
- 🍽 F	ault	*	Bit3	%IX0.3	BOOL		FALSE			
- 🍽 🗤	/arning	*	Bit4	%IX0.4	BOOL		FALSE			
- 🄟 A	tRef	*	Bit5	%IX0.5	BOOL		FALSE			
			Bit6	%IX0.6	BOOL		FALSE			
			Bit7	%IX0.7	BOOL		FALSE			
			Bit8	%IX1.0	BOOL		FALSE			
			Bit9	%IX1.1	BOOL		FALSE			
🍫			Bit10	%IX1.2	BOOL		FALSE			
- *>			Bit11	%IX1.3	BOOL		FALSE			
<b>*</b> >			Bit12	%IX1.4	BOOL		FALSE			
			Bit13	%IX1.5	BOOL		FALSE			
			Bit14	%IX1.6	BOOL		FALSE			
			Bit15	%IX1.7	BOOL		FALSE			
🗄 🦄 Gener	ral_Drive_Status	*	Channel 0[1]	%IW2	WORD				READ 16#0	835 (=02101)
🖻 👋 DG1_	Actual_Speed	*	Channel 0[2]	%IW4	WORD				READ 16#0	836 (=02102)

Variable names have been added to the 3 input status registers above. The first register contains individual status bits, so variable names have been applied at the bit level. The third input status register contains a 16-bit decimal value for the scaled speed in Hz x 100. So, a variable name has been applied at the word level for the Actual Speed.

Below are descriptive variable names added to the 3 output registers in a similar manner.

É		Channel 1	%QW0	ARRAY [02] OF WORD		Write Multiple Registers
🚔 <b>*</b> ø		Channel 1[0]	%QW0	WORD		WRITE 16#07D0 (=02000)
🐶 Run	***	Bit0	%QX0.0	BOOL	FALSE	
···· 🍫 Direction	*	Bit1	%QX0.1	BOOL	FALSE	
► Fault_Reset	**	Bit2	%QX0.2	BOOL	FALSE	
···· **		Bit3	%QX0.3	BOOL	FALSE	
🍫		Bit4	%QX0.4	BOOL	FALSE	
<b>*</b> ø		Bit5	%QX0.5	BOOL	FALSE	
🍫		Bit6	%QX0.6	BOOL	FALSE	
<b>*</b> ø		Bit7	%QX0.7	BOOL	FALSE	
🐶 Net_Ctrl	*	Bit8	%QX1.0	BOOL	FALSE	
	*	Bit9	%QX1.1	BOOL	FALSE	
🍢		Bit10	%QX1.2	BOOL	FALSE	
<b>*</b> @		Bit11	%QX1.3	BOOL	FALSE	
<b>*</b> @		Bit12	%QX1.4	BOOL	FALSE	
···· **		Bit13	%QX1.5	BOOL	FALSE	
🍢		Bit14	%QX1.6	BOOL	FALSE	
<b>*</b> ø		Bit15	%QX1.7	BOOL	FALSE	
Seneral_Drive_Control	*	Channel 1[1]	%QW2	WORD		WRITE 16#07D1 (=02001)
* Voltage Speed_Reference	***	Channel 1[2]	%QW4	WORD		WRITE 16#07D2 (=02002)

The descriptions for these Status and Control registers can be found in the PowerXL DG1 Drive Communications manual, in the Modbus TCP chapter. These variables can now be used when creating a program to control and monitor the PowerXL DG1 drive. These same variables can also be used to develop visualization screens to control and monitor the drive.

A few things to note:

- 1. Set the Net\_Ctrl (Network Control) and Net\_Ref (Network Reference) output bits prior to operating the drive through the network.
- 2. The Run bit instructs the drive to Run the motor and the Direction bit toggles between FWD (0) and REV (1).
- 3. The DG1\_Speed\_Reference value sent to the drive is in percent x 100. So, the range is 0-10000 decimal (0-100.00%)
- 4. The DG1\_Actual\_Speed value returned from the drive is in frequency x 100. So, the range is 0-6000 decimal (0-60.00Hz).

#### References

- PowerXL DG1 Series VFD Installation Manual, Publication MN040002EN
- PowerXL DG1 Series VFD Application Manual, Publication MN040004EN
- PowerXL DG1 Series Communication User Manual, Publication MN040010EN

## 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, option 6.

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

© 2014 Eaton All Rights Reserved Printed in USA Publication No. AP040086EN September 2014 Eaton is a registered trademark of Eaton Corporation.

All other trademarks are property of their respective owners