

Using the AnyBus[®]-X Gateway to Communicate between a DVT camera and a Profibus Master

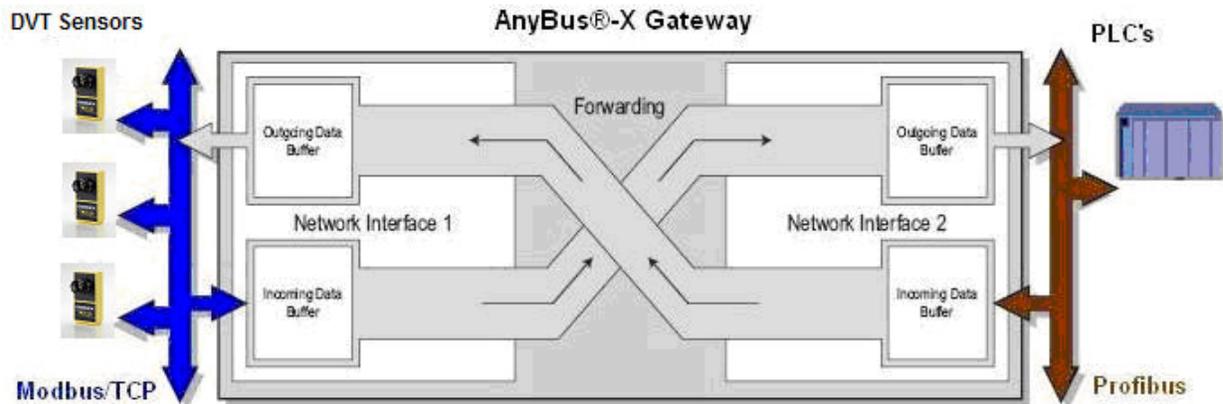
Table of Contents

1 OVERVIEW	3
2 INSTALLING AND CONFIGURING THE ANYBUS[®]-X GENERIC GATEWAY AND ETHERNET INTERFACE.....	3
2.1 Configuration Options.....	3
2.2 Configuring the AnyBus [®] Ethernet Interface using IPconfig	4
3 ANYBUS[®] X-GATEWAY COMMUNICATIONS WITH DVT CAMERAS USING MODBUS TCP	5
3.1 X-Gateway and Modbus TCP	5
3.1.1 Input Register Map.....	5
3.1.2 Holding Register Map	6
3.2 Modbus TCP Communications with DVT Cameras.....	6
3.2.1 Operation.....	6
3.2.2 Transfer Setup	7
3.2.3 Example	8
4 CONFIGURING THE ANYBUS[®] X-GATEWAY PROFIBUS SLAVE	10
4.1 Profibus Introduction	10
4.2 AnyBus [®] Profibus Support.....	11
4.2.1 Input Buffer.....	11
4.2.2 Output Buffer	11
4.3 Profibus Slave Configuration.....	12
5 CONCLUSION	12

1 Overview

This document explains how to establish communication between a DVT camera and a Profibus master device through an AnyBus[®] X-Gateway device.

The AnyBus[®] X-Gateway has two network interfaces and the device exchanges data via two pairs of exchange buffers.



Up to 512 bytes of data can be forwarded between network interfaces. The input buffer on one interface is routed through the output buffer on the other.

In the figure shown above, the gateway is configured with an Ethernet interface on one side and a Profibus interface on the other. The gateway device can act as Modbus TCP slave or EtherNet/IP slave on its Ethernet Interface and Profibus slave on its Profibus interface. This document explains how to set up communications with a DVT camera via Modbus TCP. The X-Gateway device also supports access via HTTP, FTP, and Telnet.

This document describes the following:

- How to configure the Ethernet interface of the X-Gateway device.
- How to establish communications between a DVT sensor and the gateway via Modbus TCP.
- How to configure the Profibus slave interface on the AnyBus[®] X-Gateway.

2 Installing and Configuring the AnyBus[®]-X Generic Gateway and Ethernet Interface

2.1 Configuration Options

DVT cameras use Modbus TCP to communicate with the AnyBus[®] X-Gateway device over Ethernet. The X-Gateway, in turn, communicates through Profibus with the Master device.

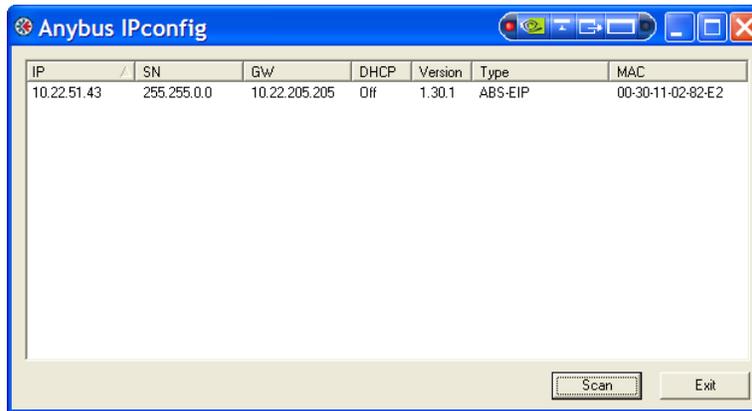
In order to communicate with the DVT sensor over Modbus TCP, the X-Gateway's Ethernet interface needs to be configured, which is described in this section. You can refer to the *AnyBus[®]-X Generic Gateway User Manual* and the *AnyBus[®]-X Ethernet Network Interface Addendum* for more information. These can be downloaded from the HMS website (www.hms-networks.com).

There are several methods available to configure the Ethernet interface network settings.

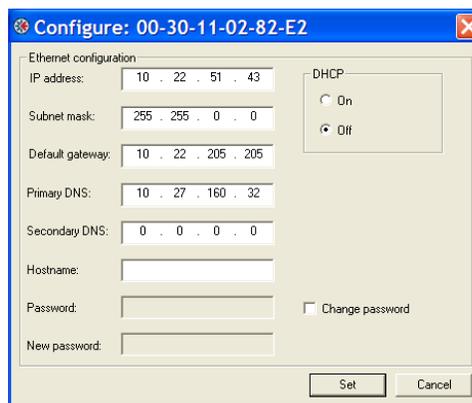
- **Dip Switches**—If an address of type 192.168.0.x is desired, the configuration dip switches on the Ethernet interface can be used to specify the last octet. The switches would be used to create a binary representation (8 bits for 0-255) of the last octet. The subnet mask in this case is fixed at 255.255.255.0 and a gateway address is not set.
- **IPconfig Utility**—A utility program called IPconfig is provided by HMS to configure the AnyBus[®]-X Ethernet interface network parameters. This utility implements the HMS IP Configuration Protocol (HICP) and can be downloaded from the HMS website (www.hms-networks.com). The next section discusses how to configure network settings with AnyBus[®] IPconfig.
- **Other Configuration Methods**—There are other methods of configuring the Ethernet interface network settings such as using ARP, loading configuration files and through the X-Gateway's web interface. These are not discussed in this integration note but are covered in the *AnyBus[®]-X Ethernet Network Interface Addendum* document.

2.2 Configuring the AnyBus[®] Ethernet Interface using IPconfig

Download the AnyBus IPconfig utility from the HMS website; install it and start it up. Upon startup the program scans the network for AnyBus devices with Ethernet interfaces.



Once the device has been identified and listed, you can edit its network settings by double-clicking it in the list.



In the new window that appears, you can set the IP address, subnet mask, default gateway address and primary and secondary DNS addresses. You can also specify a host name and a password to lock the network settings. Finally, click the "Set" button to configure the device with the new parameters.

3 AnyBus® X-Gateway Communications with DVT cameras using Modbus TCP

3.1 X-Gateway and Modbus TCP

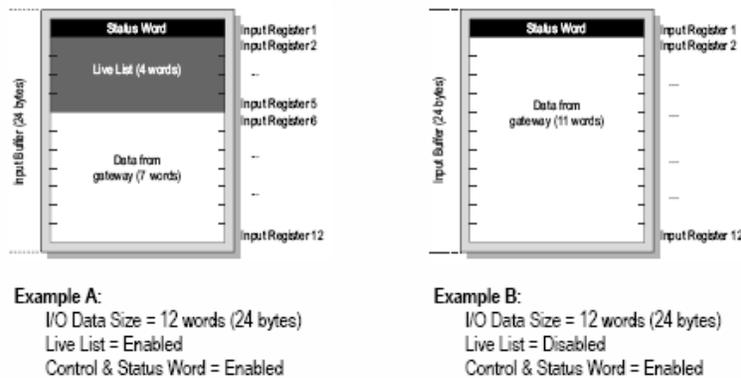
The AnyBus®-X Gateway Ethernet interface supports Modbus TCP. The following table lists the Modbus function codes supported by the Ethernet interface (from the *AnyBus®-X Ethernet Network Interface Addendum* document):

Function Code	Function Name	Mapped to Buffer...
3	Read multiple registers (Read Holding Registers)	Input
4	Read input registers	Input
6	Write single register	Output
7	Read exception status	-
8	Diagnostics	-
16	Force multiple registers (Write Holding Registers)	Output
22	Mask write register	Output
23	Read/Write registers	Output

The Input and Output Buffers in the AnyBus®-X Gateway Ethernet interface are mapped to Input and Holding Registers respectively.

3.1.1 Input Register Map

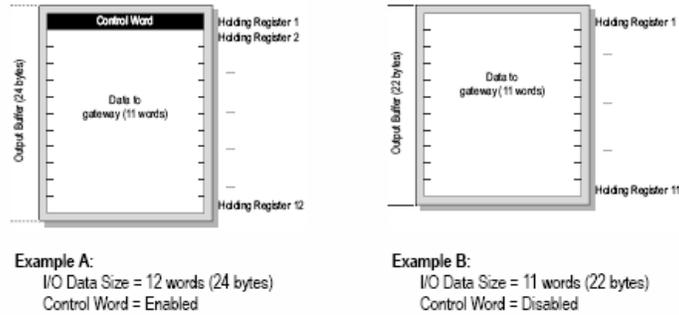
The Input Buffer (data from Gateway) is mapped to Input Register 1 and on. Depending on the type of gateway and how it has been set up to operate, up to 5 words (registers 1 to 5) may be occupied by the Status Word and the Live List, as shown in the following figure from the *AnyBus®-X Ethernet Network Interface Addendum* document:



Note: From the *AnyBus®-X Ethernet Network Interface Addendum* document: “The Live List is only available on master-slave gateway versions. Consult the AnyBus®-X Generic Gateway user manual for further information”.

3.1.2 Holding Register Map

The Output Buffer (data from Ethernet network) is mapped to holding register 1 and on. Depending on how the gateway is set to operate, holding register 1 may be occupied by the Control Word, as shown in the following figure from the *AnyBus®-X Ethernet Network Interface Addendum* document:



Refer to the *AnyBus®-X Generic Gateway User Manual* and *AnyBus®-X Ethernet Network Interface Addendum* documents for further details.

Note that the Input Buffer (data coming from the Profibus network) is mapped to Modbus Input Register 1 and the Output Buffer (data forwarded to the Profibus network) is mapped to Modbus Holding Register 1. Input registers are accessed at Modbus address 0 if function 4 (Read Input Registers) is used and address 1024 if function 3 (Read Multiple Registers) is used. Holding Registers are accessed at address 0 using function 16 (Write Multiple Registers) or function 6 (Write Single Register). That is, if you want to use Modbus class 0 function codes, you can use Read Holding Registers (function code 3) and read the X-Gateway input registers (to read from the Profibus network) with an offset of 1024 and you can use Write Holding Registers (function code 16) with an offset of 0 to write to the Profibus network.

In order to configure the input and output buffer size for the Modbus slave interface you need to access the Gateway Configuration Interface. This is done by connecting to the RS-232 programming port on the AnyBus®-X Gateway using any ASCII-capable terminal program such as Microsoft HyperTerminal. For more information refer to *Chapter 5 – Gateway Config Interface (RS232)* in the *AnyBus®-X Generic Gateway User Manual*. You would also use this interface to configure the Profibus interface parameters.

3.2 Modbus TCP Communications with DVT Cameras

3.2.1 Operation

Modbus Communications is one of the most widely used industrial protocols for communication between industrial devices. It is a class-structured protocol. DVT vision sensors implement all Class 0 function codes (Read and Write Multiple Registers) and most Class 1 function codes (Read Status, Read Input Discretes, Read Multiple Coils, Write Coil, Read Input Registers, Write a single Register).

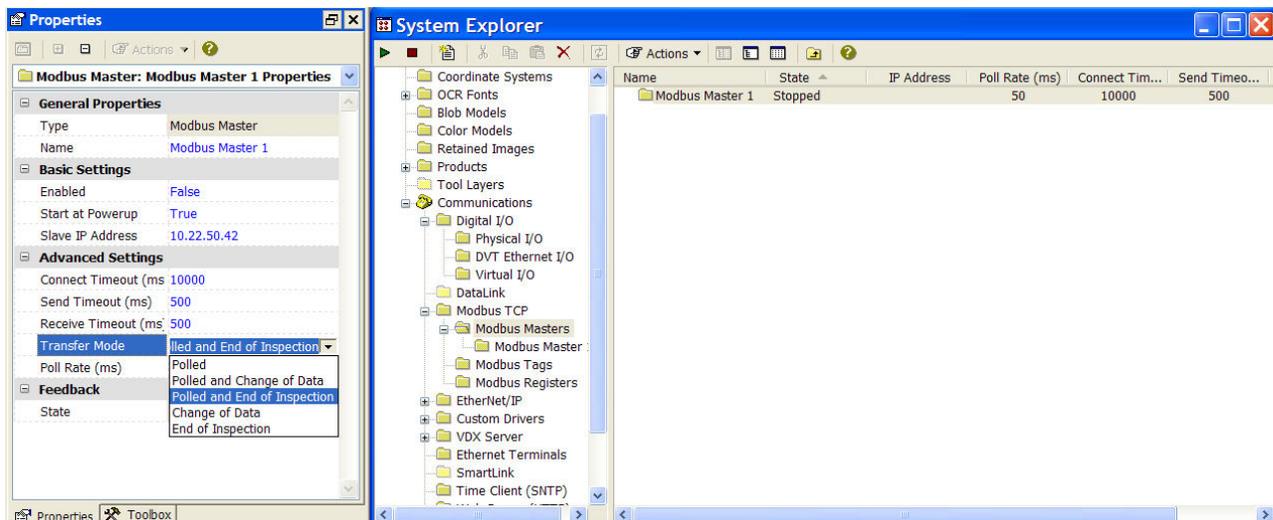
The DVT vision sensor can be configured as either a Modbus master or as a slave. To have the DVT vision sensor act as a Modbus slave, you would set the Enabled property to true. However, in this case, the AnyBus®-X Ethernet interface acts as the Modbus slave; so you need to configure the DVT vision sensor as a Modbus master.

There are two methods to perform Modbus communications as a master on a DVT vision sensor. One method is to use an instance of the MBTransfer class through a script; in this case, you could execute any class 0 or class 1 function codes supported by the DVT camera. The second method is to configure Modbus master transfers through the Communications Explorer in Intellect; in this case, only class 0 function codes (Read and Write Holding Registers) are supported. This document covers the second method.

3.2.2 Transfer Setup

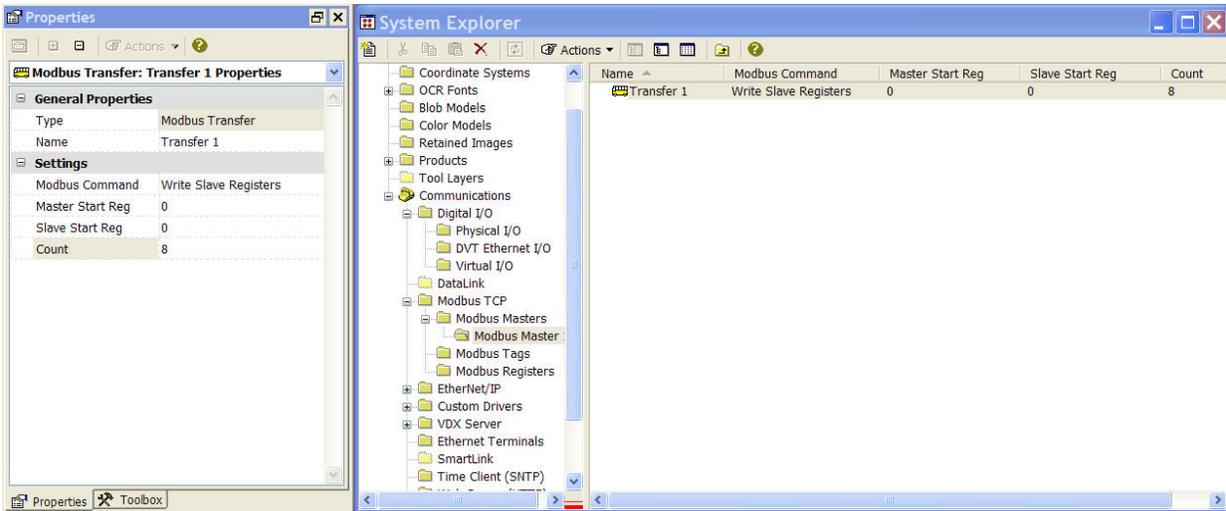
This section covers the graphical method of establishing Modbus communications as a Modbus master from a DVT vision sensor.

To configure the camera, first connect to it using Intellect. Once you have established the connection and presuming you have already programmed the inspection, select the desired product on the camera and proceed to the *Communications* section of the *System Explorer* (*System* → *Communication Settings*). Select *Modbus Masters* under the *Modbus TCP* subsection. Double click on the blank area to the right of the system settings tree on the System Explorer. Select the new *Modbus Master* entry and check the *Properties* window. At this point you would enter the appropriate values to establish a Modbus connection with the AnyBus[®]-X Modbus slave, the most important being the IP address of the AnyBus[®]-X Ethernet interface, the transfer mode for the Modbus data and the poll rate (if the transfer mode includes a polling mechanism).

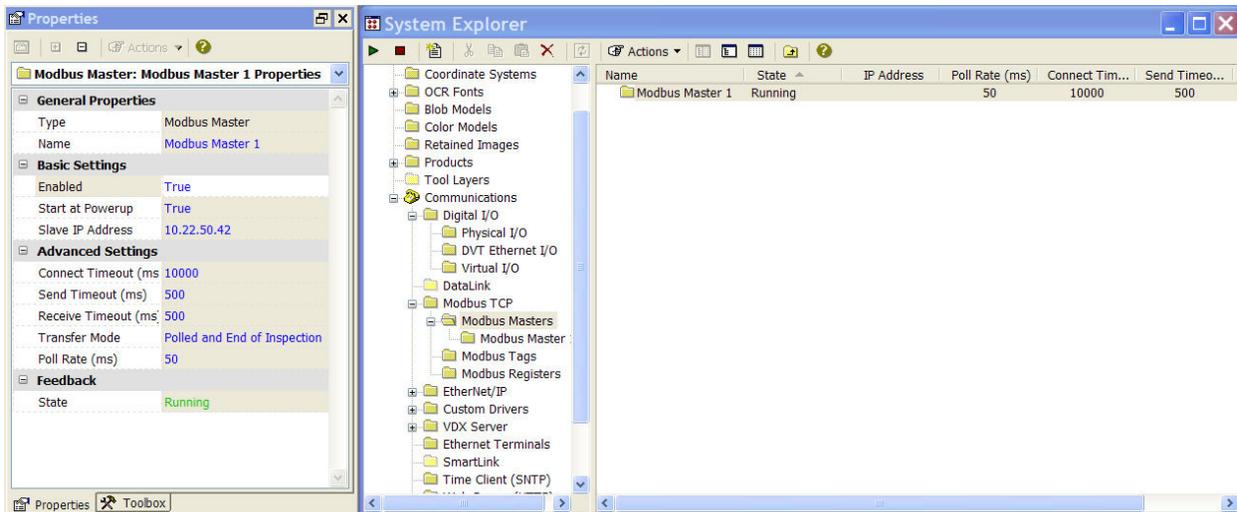


Once the Modbus Master entry has been configured, double-click on that entry on the right pane of the System Explorer. Now, you can double-click on the empty space on the right pane to add individual Modbus transfers. Here you might want to configure both read and write Modbus operations. Set the correct properties for that Modbus transfer: Modbus command (Read or Write Holding Registers), master start register, slave start register and number of Modbus registers to transfer). Note that you can configure more than one read or write operation. This would be the case if you are transferring output data that does not reside on contiguous Modbus registers on the camera.

Remember that the Read Input Registers Modbus function (code 4) is not directly accessible from the Modbus master configuration in the System Explorer. Instead this configuration uses Read Multiple Registers function (code 3). So a user must specify the slave start register property starting from address 1024, not 0 when reading from the input buffer on the X-Gateway.



Finally, go back to the Modbus Master list and set the *Enabled* property to *True* to connect to the Modbus slave and start the transfer(s). The *State* feedback property should display “*Running*”.



The number of registers transferred must not exceed the number of registers specified via Telnet on the X-Gateway side.

Several cameras can be configured to communicate with a single AnyBus®-X Gateway. You would just need to allocate consecutive blocks of data on the input and output buffers of the X-Gateway to each camera. From the Modbus standpoint, you would just add offsets to the slave start registers being accessed by each camera according to their allotted block. All the allotted blocks, as a single package would then be propagated through the Profibus network. The Profibus master would need to know the offsets for each block to distinguish between each camera. Note that the sizes for the input and output buffers must match for both networks (Modbus and Profibus) on the AnyBus®-X Gateway.

3.2.3 Example

For this example, you need to have access to the camera's virtual I/O functions and also exchange some general purpose input and output values (two 16-bit integer values as input to the camera and two 16-bit integer values to output inspection results from the camera).

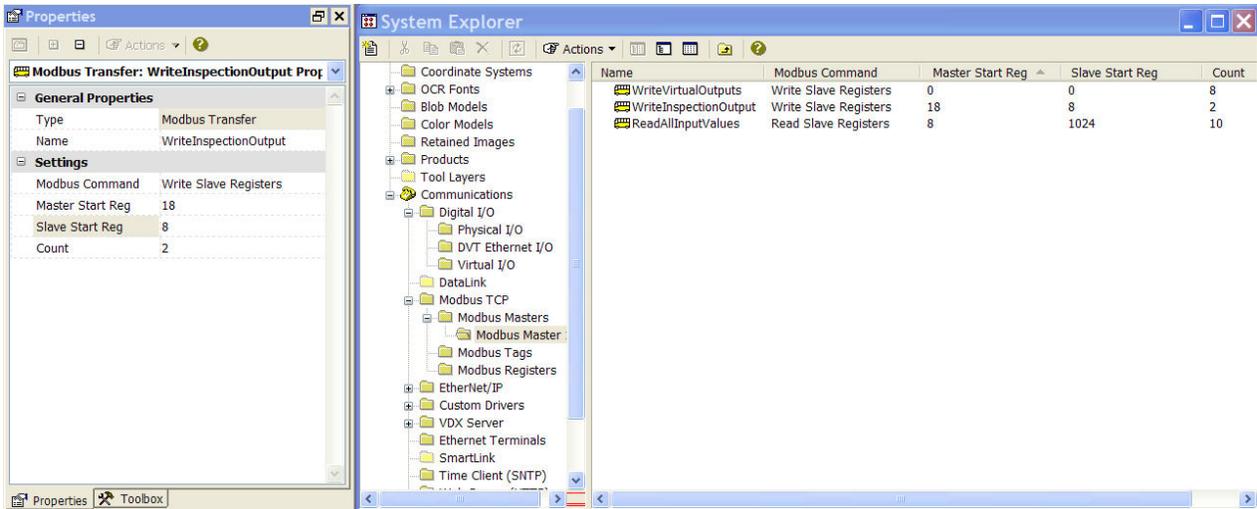
The following figures are the Modbus maps from the Intellect Help files:

Modbus Coil and Virtual Output Map									Modbus Coil and Virtual Inputs Map										
Modbus Coil	Bit Position								Type	Modbus Coil	Bit Position								Type
	7	6	5	4	3	2	1	0			7	6	5	4	3	2	1	0	
0	Script 16	Script 15	Script 14	Script 13	Script 12	Script 11	Script 10	Script 9	Virtual Output Signals	8	User 32	User 31	User 30	User 29	User 28	User 27	User 26	User 25	Virtual Input Signals
	Coil 7	Coil 6	Coil 5	Coil 4	Coil 3	Coil 2	Coil 1	Coil 0			Coil 135	Coil 134	133	Coil 132	Coil 131	Coil 130	Coil 129	Coil 128	
	Script 8	Script 7	Script 6	Script 5	Script 4	Script 3	Script 2	Script 1			User 24	User 23	User 22	User 21	User 20	User 19	User 18	User 17	
1	Coil 15	Coil 14	Coil 13	Coil 12	Coil 11	Coil 10	Coil 9	Coil 8	Virtual Output Signals	9	User 16	User 15	User 14	User 13	User 12	User 11	User 10	User 9	Virtual Input Signals
	Coil 23	Coil 22	Coil 21	Coil 20	Coil 19	Coil 18	Coil 17	Coil 16			Coil 151	Coil 150	Coil 149	Coil 148	Coil 147	Coil 146	Coil 145	Coil 144	
	User 8	User 7	User 6	User 5	User 4	User 3	User 2	User 1			User 8	User 7	User 6	User 5	User 4	User 3	User 2	User 1	
2	Coil 31	Coil 30	Coil 29	Coil 28	Coil 27	Coil 26	Coil 25	Coil 24	Virtual Output Signals	10	Coil 159	Coil 158	Coil 157	Coil 156	Coil 155	Coil 154	Coil 153	Coil 152	Virtual Input Signals
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved			Reserved	Product ID Bit 14	Product ID Bit 13	Product ID Bit 12	Product ID Bit 11	Product ID Bit 10	Product ID Bit 9	Product ID Bit 8	
	Coil 39	Coil 38	Coil 37	Coil 36	Coil 35	Coil 34	Coil 33	Coil 32			Coil 167	Coil 166	Coil 165	Coil 164	Coil 163	Coil 162	Coil 161	Coil 160	
3	Reserved	Reserved	Reserved	Wrong Code	Inspection Toggle	Inspecting	Acquiring	Busy	Virtual Output Signals	11	Product ID Bit 7	Product ID Bit 6	Product ID Bit 5	Product ID Bit 4	Product ID Bit 3	Product ID Bit 2	Product ID Bit 1	Product ID Bit 0	Virtual Input Signals
	Coil 47	Coil 46	Coil 45	Coil 44	Coil 43	Coil 42	Coil 41	Coil 40			Coil 175	Coil 174	Coil 173	Coil 172	Coil 171	Coil 170	Coil 169	Coil 168	
	Coil 55	Coil 54	Coil 53	Coil 52	Coil 51	Coil 50	Coil 49	Coil 48			Reserved	Reserved	Reserved	Reserved	Digital Release 4	Digital Release 3	Digital Release 2	Digital Release 1	
4	Trigger Mode	Run Mode	Resource Conflict	Fail	Pass	Product Select Fail	Product Select Pass	Trigger Ack	Output Values	12	Coil 183	Coil 182	Coil 181	Coil 180	Coil 179	Coil 178	Coil 177	Coil 176	Input Values
	Coil 63	Coil 62	Coil 61	Coil 60	Coil 59	Coil 58	Coil 57	Coil 56			External Trigger Mode	Disable Run Mode	Quadrature Input	Event Counter	Results Ack	Reserved	Product Select	Trigger	
	Reserved										Coil 191	Coil 190	Coil 189	Coil 188	Coil 187	Coil 186	Coil 185	Coil 184	
5	Reserved								Output Values	13	Reserved								Input Values
	Reserved										Reserved								
	Reserved										Reserved								
6	Reserved								Output Values	14	Reserved								Input Values
	Reserved										Reserved								
	Reserved										Reserved								
7	Reserved								Output Values	15	Reserved								Input Values
	Reserved										Reserved								
	Reserved										Reserved								
Total Inspection Count									Output Values	16	Reserved								Input Values
Current Inspection Product Digital ID											Reserved								
Reserved											Reserved								

If you exchange these registers with the AnyBus[®]-X Gateway accordingly (write registers 0 through 7 and read into registers 8 through 15) you can have access to each individual I/O point on the camera from the Profibus device.

In this case, it will more convenient to have the two 16-bit integer input values read by the camera going into Modbus registers 16 and 17 respectively. This will allow us to transfer the virtual input data (registers 8 through 15) and these two integer values (registers 16 through 17) in one Read Multiple Registers operation (spanning 10 registers: 8 through 17). Two Write Multiple Registers operations will be necessary in order to transfer the virtual output data (registers 0 through 7) and the two 16-bit integer output values (written into registers 18 and 19).

The final configuration is shown in the following figure and table:



Modbus Operation	Modbus Registers on DVT Vision Sensor (Master)	Modbus Registers on AnyBus [®] -X Gateway (Slave)
Write Holding Registers	0 – 7 (Camera's Virtual Output)	0 - 7
Write Holding Registers	18 – 19 (Two integer inspection result values)	8 - 9
Read Holding Registers	8 – 17 (Camera's Virtual Input + Two integer input values)	1024 – 1033 (offset of 1024 applied to Holding Registers)

4 Configuring the AnyBus[®] X-Gateway Profibus Slave

4.1 Profibus Introduction

Profibus stands for Process Field Bus. Profibus is an international, vendor-independent, open fieldbus standard. European standards EN 50170 and EN 50254 ensure vendor-independence and openness. The standard allows devices from multiple vendors to communicate without special interface adjustments. Profibus can be used for both complex communication tasks and high-speed time critical applications.

Profibus offers functionally graduated communication protocols (Communication Profiles): DP and FMS. Depending on the application, the transmission technologies (Physical Profiles) RS-485, IEC 1158-2 or fiber optics are available. Profibus Communication Profiles define how users transmit their data serially via the common transmission medium. DP is the most frequently used communication profile. It is optimized for speed, efficiency and low connection costs and is designed especially for communication between automation systems and distributed peripherals. DP is suitable as a replacement for conventional, parallel signal transmission with 24 volts in manufacturing automation as well as for analog signal transmission with 4 ... 20 mA or Hart in process automation.

The media of this Fieldbus is a shielded copper cable composed of a twisted pair. The baud rate for the bus is between 9.6k baud to a maximum of 12M baud. The Profibus network can consist of 126 nodes and the total amount of data for Profibus-DP is 244 Byte out and 244 Byte in per module (node 126 should not be used to exchange user data. This node is used for commissioning purposes).

4.2 AnyBus[®] Profibus Support

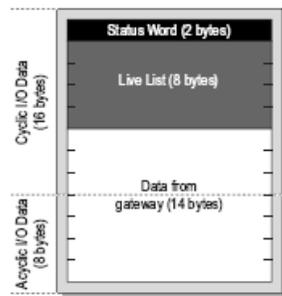
In order to communicate with the master device over Profibus, the X-Gateway's Profibus slave interface needs to be configured. This step is described in this section. The user can refer to the *AnyBus[®]-X Generic Gateway User Manual*, the *AnyBus[®]-X Profibus Slave Network Interface Addendum* and the *AnyBus[®]-X Profibus Slave Installation Sheet* for more information. These can be downloaded from the HMS website (www.hms-networks.com).

From the *AnyBus[®]-X Profibus Slave Network Interface Addendum*: "On Profibus, data from the gateway is represented as cyclical I/O data (DP) and Acyclical I/O data (DPV). The amount of data that is represented as Cyclical I/O data vs. the amount that is represented as acyclical I/O data can be configured via the Gateway Config interface. The Slave interface can handle up to 244 bytes of I/O data in each direction. However, the total data size (input + output) cannot exceed 416 bytes."

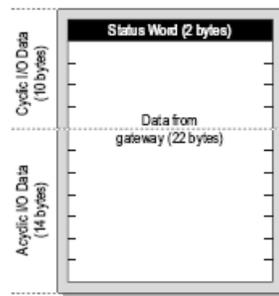
The AnyBus-X Profibus Slave interface offers complete slave functionality according to extensions of EN 50170 (DPV1). It supports Class 1 and Class 2 services as well as the standard DP functionality. The original version, designated DPV0, has been expanded to include version DPV1, offering acyclic data exchange between master and slave.

4.2.1 Input Buffer

From the *AnyBus[®]-X Profibus Slave Network Interface Addendum*: "Depending on the type of gateway and how it has been configured, up to 10 bytes of the data produced by the gateway may be occupied with status information".



Example A:
 Cyclic I/O Data Size = 16
 Acyclic I/O Data Size = 8
 Live List = Enabled
 Control & Status Word = Enabled

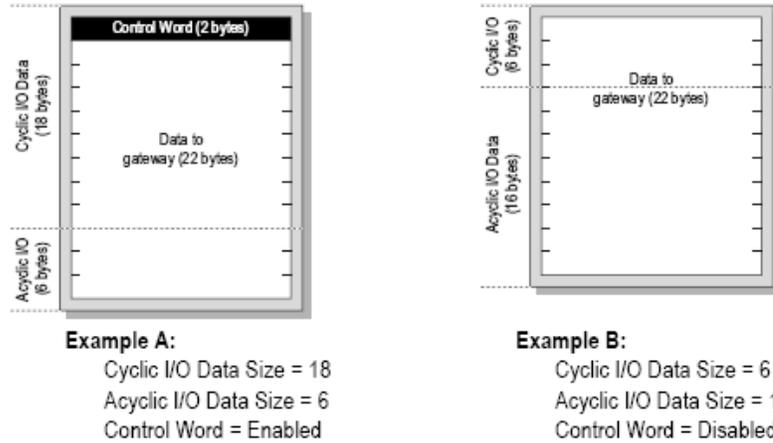


Example B:
 Cyclic I/O Data Size = 10
 Acyclic I/O Data Size = 14
 Live List = Disabled
 Control & Status Word = Enabled

Note: The Live List is only available on master-slave gateway versions.

4.2.2 Output Buffer

From the *AnyBus[®]-X Profibus Slave Network Interface Addendum*: "Depending on how the gateway is configured, the first two bytes consumed by the gateway may be interpreted as control information".



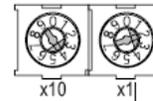
Note: The Live List is only available on master-slave gateway versions.

4.3 Profibus Slave Configuration

Several parameters need to be configured to establish communications with the AnyBus®-X Profibus Slave interface. First, the node address should be configured by using the on board rotary switches. From the *AnyBus®-X Profibus Slave Network Interface Addendum*:

Example:

In this example, the profibus node address will be 42 (4 x 10)+(2 x 1).



Note: Depending on if the Slave interface is top or bottom mounted, the orientation of the switches are different.

You would also need to install the .GSD file, which you can download from the HMS website (www.hms-networks.com). This file is used to add the AnyBus®-X Gateway Profibus interface as a Profibus slave device on the Profibus network.

In order to configure the input and output buffer size for the Profibus slave interface you need to access the Gateway Configuration Interface. This is done by connecting to the RS-232 programming port on the AnyBus®-X Gateway using any ASCII-capable terminal program such as Microsoft HyperTerminal. For more information refer to *Chapter 5 – Gateway Config Interface (RS232)* in the *AnyBus®-X Generic Gateway User Manual*. You would also use this interface to configure the Modbus interface parameters.

5 Conclusion

This document has covered the use of the AnyBus®-X Gateway, configured with a Modbus TCP slave interface on one side and a Profibus interface on the other, as an intermediary device between a DVT camera and a Profibus network.

The DVT vision sensor is configured as a Modbus TCP master through a simple graphical setup to communicate with the Modbus slave interface of the AnyBus®-X Gateway. Several cameras can be configured to communicate with a single X-Gateway device. You would just need to allocate consecutive blocks of data on the input and output buffers of the X-Gateway to each camera. From the Modbus standpoint, you would just add offsets to the registers being accessed by each camera according to their allotted block.

For further reading and more detailed coverage of the configuration steps and specification of the AnyBus®-X Gateway interfaces, please refer to the following documents accessible through the

HMS website (www.hms-networks.com): *AnyBus[®]-X Generic Gateway User Manual, AnyBus[®]-X Ethernet Network Interface Addendum, AnyBus[®]-X Ethernet Interface Network Installation Sheet, AnyBus[®]-X Profibus Slave Network Interface Addendum and the AnyBus[®]-X Profibus Slave Installation Sheet.*