PROFINET IO Installation Quick Start

This document is intended to provide you with a way to locate software and installation documentation for the DeviceMaster UP.

In addition, use this document to quickly configure:

- Read-only devices such as barcode scanners and some RFID readers
- Read/write devices such as printers and some weigh scales

BLUE CAPS link to external documents, which function if reading this document from the web/ftp site or CD. Red, underlined links go to the web site and *blue*, underlined links jump within the document and provide a page number.

Installation Overview

Use the following steps to install the DeviceMaster UP.

1. INSTALL THE HARDWARE.
2. Install PortVision DX from the CD or download and install the latest version.
3. CONFIGURE THE DEVICEMASTER UP network settings using PortVision DX.
4. *Depending on the DeviceMaster UP model*, do the following:
   - *Models without PROFINET IO loaded*, you *must* install the software assembly (.msi) from the CD or download and install the latest file, which contains the PROFINET IO firmware, GSDML, and bitmap for **Step7**. By default, these files are saved in **c:\Comtrol\ProfinetIO** and are need for Step 5
   - *Models with PROFINET IO loaded on the DeviceMaster UP*, you should check to see if a later version of PROFINET IO is available for installation. Check the PROFINET IO version in PortVision DX against the web site to see if a later version is available. Typically, you should download and install the latest .msi file and upload the latest version, which may include updates or enhancements.

   Note: *Models that have PROFINET IO loaded on the DeviceMaster UP are identified in PortVision DX and the DeviceMaster UP is labeled accordingly.*
5. IF NECESSARY, UPLOAD the PROFINET IO firmware into the DeviceMaster UP using PortVision DX.

Note: *Do not perform Step 6 before Steps 1 through 5.*

6. Configure the serial or Ethernet TCP/IP socket port characteristics using the DeviceMaster UP embedded web page (*Server Configuration*).

   If you have *Read-only or read/write* devices, you can use the appropriate procedures for your device, which are located in this *document*:
   - Read-only devices (barcode scanners and some RFID readers), go to *Configuring Read-Only Serial Devices* on Page 2 or *Configuring Read-Only Ethernet TCP/IP Devices* on Page 4.
   - Read/write devices (printers and some weigh scales), first perform the appropriate procedure for a read-only device and then go to *Configuring Read/Write Devices* on Page 6.

Software and Documentation	Web
DeviceMaster UP Hardware Installation and Configuration Guide	
PortVision DX	
PROFINET IO firmware	
DeviceMaster UP Filtering and Data Extraction Reference Guide	

7. Install the GSDML file in **HW Config**. In the **HW Config** window, select **Install GSD** from the *Options* pull-down menu. Choose the directory where you installed the PROFINET IO software in Step 4.
8. CONNECT any serial device or devices.
9. Verify any Ethernet TCP/IP devices are connected as configured in the DeviceMaster UP.

Data Type Definitions

The following data type definitions apply.

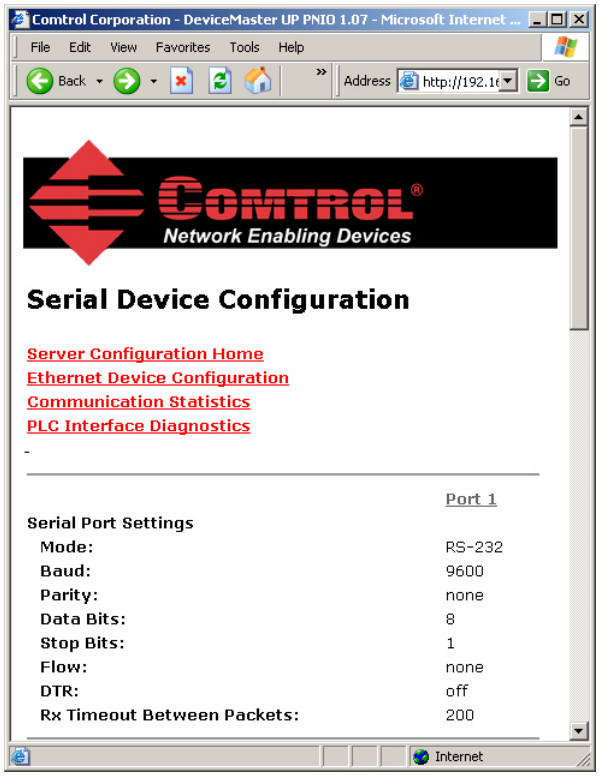
Data Type	Definition
BYTE	An integer 0 - 255 (8-bit) e.g. ASCII strings are a series of bytes
WORD	Unsigned integer (16-bit)
DWORD	Unsigned integer (32-bit)

Configuring Read-Only Serial Devices

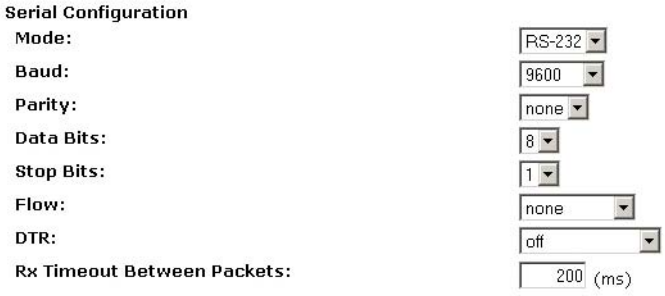
Use the following procedure to configure read-only serial ports.

Note: *Make sure that you have performed Steps 1 through 5 in the Installation Overview on Page 1.*

1. Access the *Server Configuration* web page using one of the following methods:
   - Entering the IP address in your web browser
   - Right-click the DeviceMaster UP in PortVision DX and click **Web Manager**
2. Click *Serial Device Configuration*.
3. Click the **Port #** that you want to configure, which opens the *Edit Serial Port Configuration* page.



4. Set up the serial port configuration. (**Mode**, **Baud rate**, etc.)



Note: *Refer to your serial device's User Manual for these settings.*

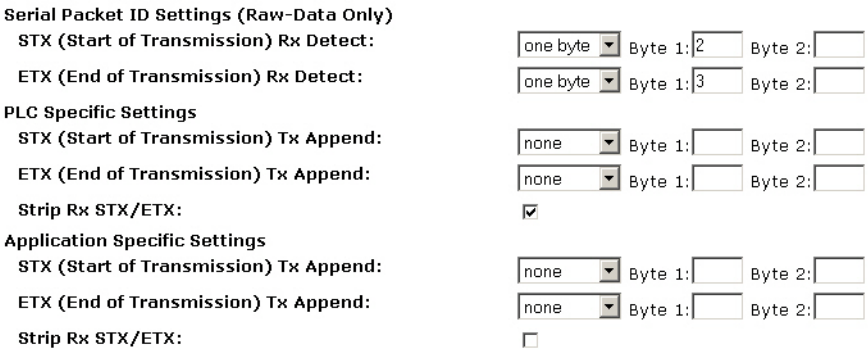
5. If desired, uncheck **Discard Rx Packets with Errors**.

6. Set up the *Serial Packet Identification*.

   a. Set **STX (Start of transmission) Rx Detect** in decimal format.

   b. Set **ETX (End of transmission) Rx Detect** in decimal format.

   c. Set the PLC® specific **Strip Rx STX/ETX Chars** setting if you do not wish to receive the STX/ETX bytes in the received data packet.



Note: *Please refer to your device's User Manual for the Start and End of Transmission byte(s) settings. You may also be able to use the Serial Interface Logs page to determine these settings.*

7. Set the *Filtering/Data Extraction Configuration*:
   - If no filtering/data extraction is required, leave all filtering/data extraction settings to defaults.
   - If filtering/data extraction is required, go to *Filtering/Data Extraction Configuration (Patent Pending)* on Page 11.

8. Set the *Application TCP Connection Configuration.*
   - If no application socket interface is required, leave all application socket interface settings at defaults and the **Application Enable** option unselected.
   - If an application socket interface is required, go to *Application Socket Configuration (Patent Pending)* on Page 16.

9. Verify **Reset Port** and **Save in Flash** are selected and click on **Submit**.

If all is set up correctly, the DeviceMaster UP will place the data packets into the location specified in **Step7**. See the example program in section *Example Programs* on Page 8 or Siemens **Step7** documentation for more information on how this is done. The first WORD is an integer received representing the sequence number. This is incremented with each new data packet. The next WORD indicate the length, which is the number of bytes of data received. The rest is data. For example, the first "packet" received from the DeviceMaster UP might have a sequence number of 1 (0x0001), a length of 7 bytes (0x0007) and seven data bytes (for instance **ABCDEFG**, or {0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47} in ASCII text).

Note: *If the buffer is not large enough to hold the entire packet the data will be truncated (i.e. only the beginning part of the packet will be written to the controller). For example, if we were using a 4 BYTE output buffer only* ***ABCD*** *or {0x41, 0x42, 0x43, 0x44} would appear in the buffer. The other BYTEs would be ignored.*

The format of data sent to a PNIO controller:

Name	Data Type	Data Value(s)
Receive (DeviceMaster to PLC message data. **Structure of:** Produced data sequence Data length (in bytes) Data array	 WORD WORD Array of BYTE	 0-65535 (FFFF Hex) 0-220 0-255

More than 220 bytes of data can not be received by the PNIO controller.

Configuring Read-Only Ethernet TCP/IP Devices

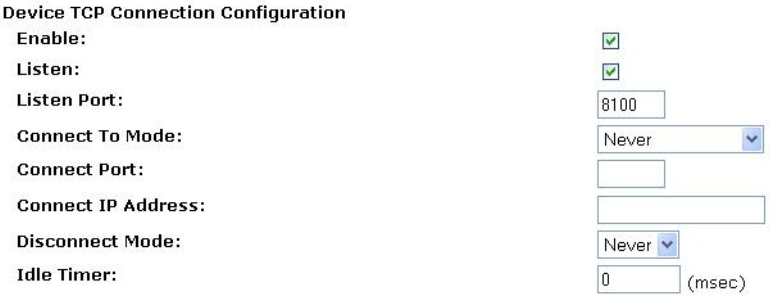
Use the following procedure to configure read-only socket devices.

Note: *Make sure that you have performed Steps 1 through 5 in the Installation Overview on Page 1.*

1. Access the *Server Configuration* web page using one of the following methods:
   - Entering the IP address in your web browser
   - Right-click the DeviceMaster UP in PortVision DX and click **Web Manager**
2. Click on **Ethernet Device Configuration** to open the *Ethernet Device Configuration* page.
3. Click the **Socket #** that you want to configure, which opens the *Edit Socket Port Configuration* page.

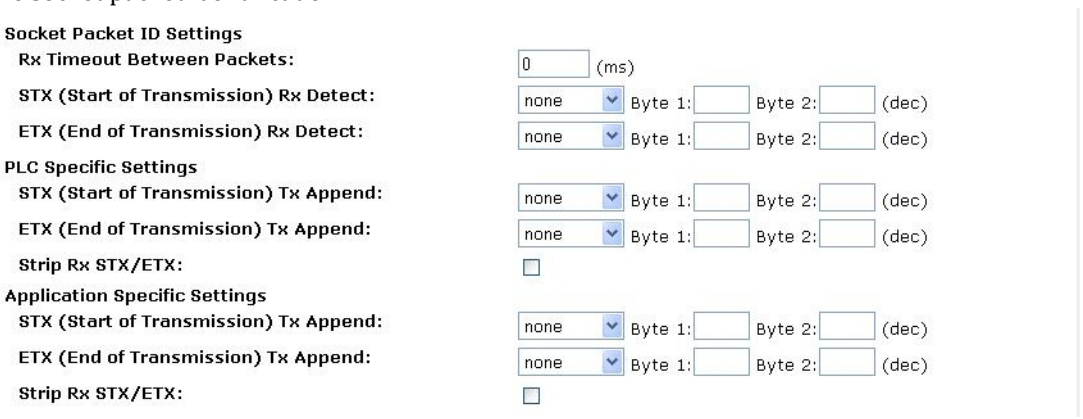
   Note: *Refer to your Ethernet device's User Manual for the following settings.*
4. Under *Device TCP Connection Configuration*, select **Enable**.
   - If your Ethernet TCP/IP device requires another device to connect to it, configure the socket port on the DeviceMaster UP to **Connect** mode:
     - Leave **Listen** unselected.
     - Set **Connect To Mode** to **Connect-Always**.
     - Set the **Connect Port** to the socket port number of your Ethernet device.
     - Set the **Connect IP Address** to the IP address of your Ethernet device.
     - Set **Disconnect Mode** to **Never**.

   - If your Ethernet TCP/IP device is configured to connect to another device, configure the socket port on the DeviceMaster UP to **Listen** mode:
     - Select **Listen**.
     - Use the default **Listen Port** on the DeviceMaster UP of ***8xxx*** or designate your own.
     - Set **Connect To Mode** to **Never**.
     - Set **Disconnect Mode** to **Never**.
     - Configure your Ethernet device to connect to the DeviceMaster UP at the DeviceMaster UP IP address and **Listen Port**.



- If you do not know if your device will connect to another Ethernet device, but do know your device's socket port and IP address, you can do the following to enable both the **Listen** and **Connect** modes:
  - Select **Listen**.
  - Use the default Listen Port on the DeviceMaster UP of *8xxx* or designate your own.
  - Set **Connect To Mode** to **Connect-Always**.
  - Set the **Connect Port** to the port number of your Ethernet device.
  - Set the **Connect IP Address** to the IP address of your Ethernet device.
  - Set **Disconnect Mode** to **Never**.
  - Optionally configure your Ethernet device to connect at the DeviceMaster UP IP address and **Listen Port**.

5. Set up the socket packet identification.



   a. Set the **Rx Timeout Between Packets**. Set to zero to stream data with the **Rx STX/ETX Detect** settings set to none. For normal settings, typical values are 10 to 50 ms.
   b. Set the **STX (Start of transmission) Rx Detect** in decimal format.
   c. Set the **ETX (End of transmission) Rx Detect** in decimal format.
   d. Enable the **Strip Rx STX/ETX** option if you do not want the STX and ETX bytes returned to the PLC or application.

   Note: *Please refer to your device's User Manual for the Start and End of Transmission byte(s) settings. You may also be able to use the Ethernet Device Interface Logs page to determine these settings.*

6. Set the *Filtering/Data Extraction Configuration*:
   - If no filtering/data extraction is required, leave all filtering/data extraction settings to defaults.
   - If filtering/data extraction is required, go to *Filtering/Data Extraction Configuration (Patent Pending)* on Page 11.
7. Set the *Application TCP Connection Configuration*:
   - If no application socket interface is required, leave all application socket interface settings at defaults and the **Enable** option unselected.
   - If an application socket interface is required, go to *Application Socket Configuration (Patent Pending)* on Page 16.
8. Verify **Reset Port** and **Save in Flash** are selected and click on **Submit**.

If all is set up correctly, the DeviceMaster UP will place the data packets into the location specified in **Step7**. See the example program in section *Example Programs* on Page 8 or Siemens **Step7** documentation for more information on how this is done. The first WORD is an integer received representing the sequence number. This is incremented with each new data packet. The next WORD indicate the length, which is the number of bytes of data received. The rest is data. For

example, the first "packet" received from the DeviceMaster UP might have a sequence number of 1 (0x0001), a length of 7 bytes (0x0007) and seven data bytes (for instance **ABCDEFG**, or {0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47} in ASCII text).

Note: *If the buffer specified in* ***Step7*** *is not large enough, the data is truncated and an error is logged.*

The format of data sent is:

Field Name	Offset in BYTES	Data Type	Data Value(s)
Sequence number	0	WORD	0-65536 (0xFFFF Hex)
Data length (in BYTES)	2	WORD	0-220
Data	4	Array of BYTEs	User controlled

Configuring Read/Write Devices

The previous two sections explained how to receive serial or TCP/IP data on the PNIO controller through the DeviceMaster UP. This section explains how to write data to the serial or TCP/IP device as well.

Follow the procedures in *Configuring Read-Only Serial Devices* on Page 2 or *Configuring Read-Only Ethernet TCP/IP Devices* on Page 4 and use the following procedure to complete the procedure for read/write devices.

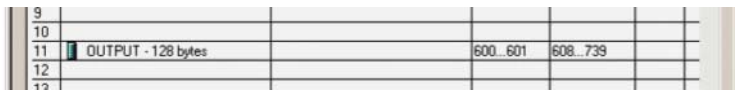
Note: *Make sure that you have performed Steps 1 through 5 in the Installation Overview on Page 1.*

1. Access the *Server Configuration* web page using one of the following methods:
   - Entering the IP address in your web browser
   - Right-click the DeviceMaster UP in PortVision DX and click **Web Manager**
2. Open the embedded web page for the serial or socket port.
   - **Serial ports**: Set up the transmit serial packet identification.

     - If desired, set the **STX (Start of transmission) Tx Append** in decimal format for the PLC and/or the application. This will append the STX byte(s) to your transmitted message. Refer to your *serial device's User Manual* for this setting.
     - If desired, set the **ETX (End of transmission) Tx Append** in decimal format for the PLC and/or the application. This will append the ETX byte(s) to your transmitted message. Refer to your *serial device's User Manual* for this setting.
3. If any embedded web page settings have changed, verify **Reset Port** and **Save in Flash** are selected and click **Submit**.
4. Configure your controller to send data out the serial or socket ports.

   Data Format: To send data from the PLC, simply write a packet to the **OUTPUT** memory configured in **Step7**. The **OUPUT** memory packets are in the same format as the incoming packets – both have 16-bit sequence numbers and lengths – only the direction of the data flow has changed. The following steps explain this in greater detail.
   - **Writing Out the Serial Port**
     - **Use the correct slot.** To write data out the serial port, use the output module configured for Slot 11. Just as Slot 1 is used for incoming serial data, Slot 11 is used for outgoing serial data.

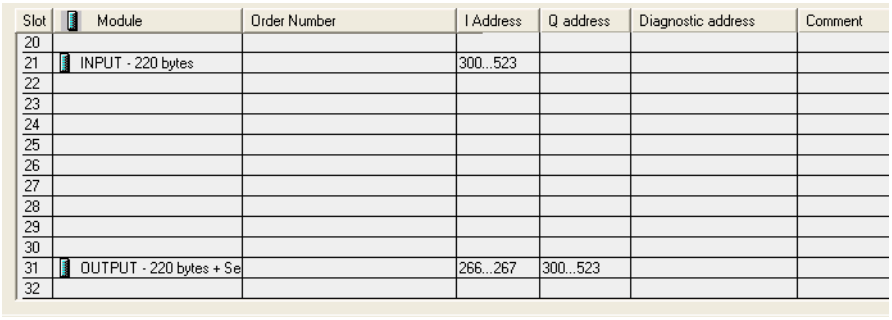
     - **Fill in the data.** Write the data into the output memory. The data that will be written out the serial port starts 4 bytes into the memory buffer. You can write as much data as the memory buffer will hold.
     - **Controlling how much data is sent.** Although the maximum amount of data sent is limited by the size of the output module configured in **Step7**, the actual amount of data written is controlled by the length field

(see table below) in the output buffer. The length field is two bytes, and starts at the 2nd byte.

Field Name	Offset in BYTES	Data Type	Data Value(s)
Sequence number	0	WORD	0-65536 (0xFFFF Hex)
Data length (in BYTES)	2	WORD	0-220
Data	4	Array of BYTEs	User controlled

- **Triggering the send.** The data will not be sent through the serial port until the DeviceMaster UP sees that the sequence number in the Output data is incremented. The DeviceMaster UP is informed of the new sequence number during the next I/O cycle. So, with a cycle time of 32ms (specified in **Step7** the DeviceMaster UP will begin sending the data within 32ms. The data will be only sent once, and no further data will be sent until the sequence number is again updated. This prevents the same data from being sent over and over out the serial port.
- **Verifying that the data has been sent.** If you wish to confirm that the last data written to the output data area has been sent out the serial port, check the input memory for the same output module. This 16-bit integer is the last data packet that was sent out the serial port. So, to confirm that the current packet has been sent, compare the sequence number (a WORD) in the output data with the input data for that module. In the screen shot (above), we would compare PIW 600 with QW 608. If they are the same, the data has been sent.

- **Writing Out the Socket Port**
  - **Use the correct slot.** To write data out the socket port, place an Output module in Slot 31. Slot 31 is the only slot that can be used to write Output data to the socket. The input and output modules for socket 1 are shown below with 220 byte buffers.



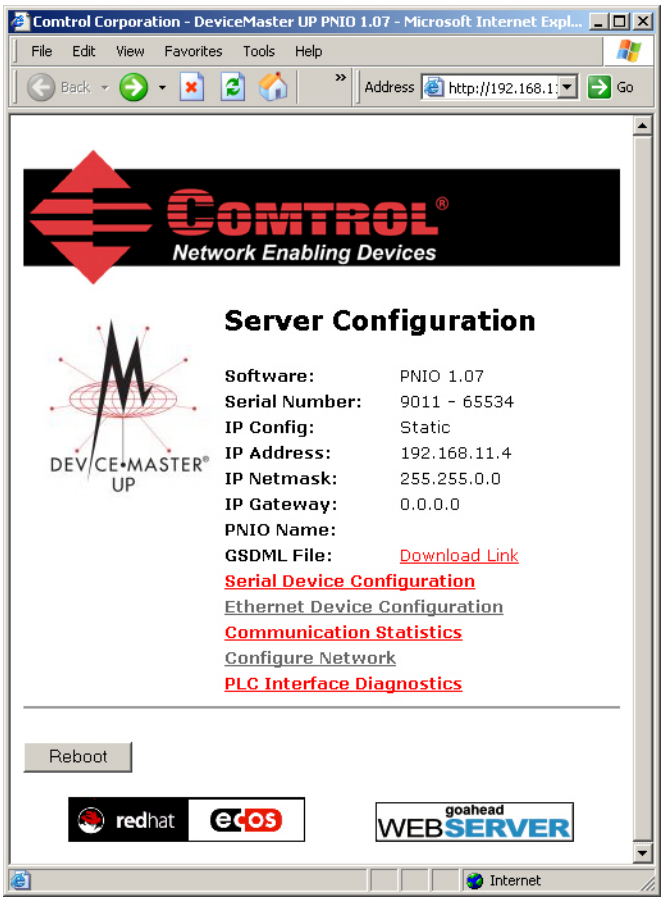
Slot	Module	Order Number	I Address	Q address	Diagnostic address	Comment
20						
21	INPUT - 220 bytes		300...523			
22						
23						
24						
25						
26						
27						
28						
29						
30						
31	OUTPUT - 220 bytes + Se		266...267	300...523		
32						

  - **The remaining steps are exactly the same as those for writing out a serial port.** Only the slot changes.

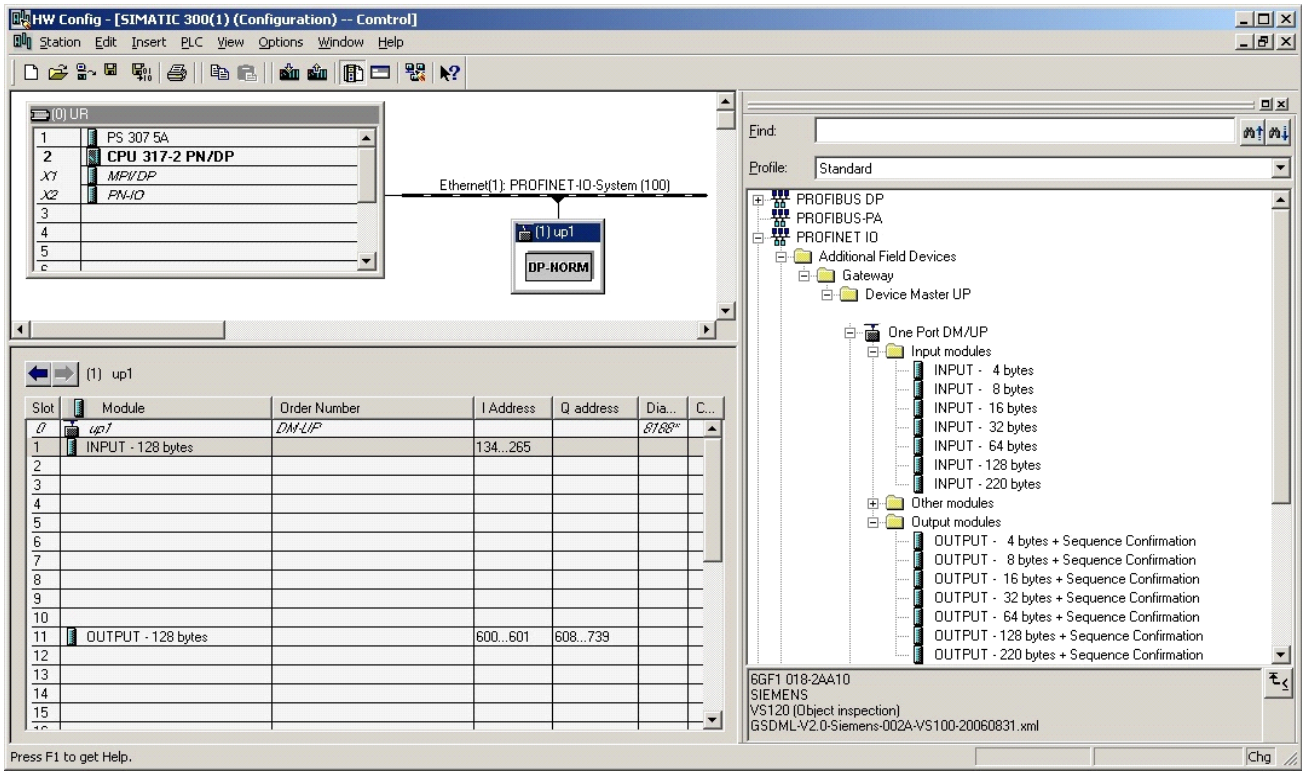
Example Programs

The example program in this section shows how to create a loopback program on the PLC. This loopback program will simply take any information it receives from the DeviceMaster UP, on the serial port or TCP/IP port, and writes it back out the same interface.

- The first step is to create your configuration with the controller, add the PROFINET IO network, install the GSD file for PNIO then install the input and output modules.
- In order to install the GSD you will first need a copy of it. This can be obtained by downloading it from the device's home page. To do this, type the device's IP address into your browser, then right click on the **Download Link** next to the **GSDML File**: text, then save the file to your Desktop or another location where **Step7** can find it. The home page is shown below:



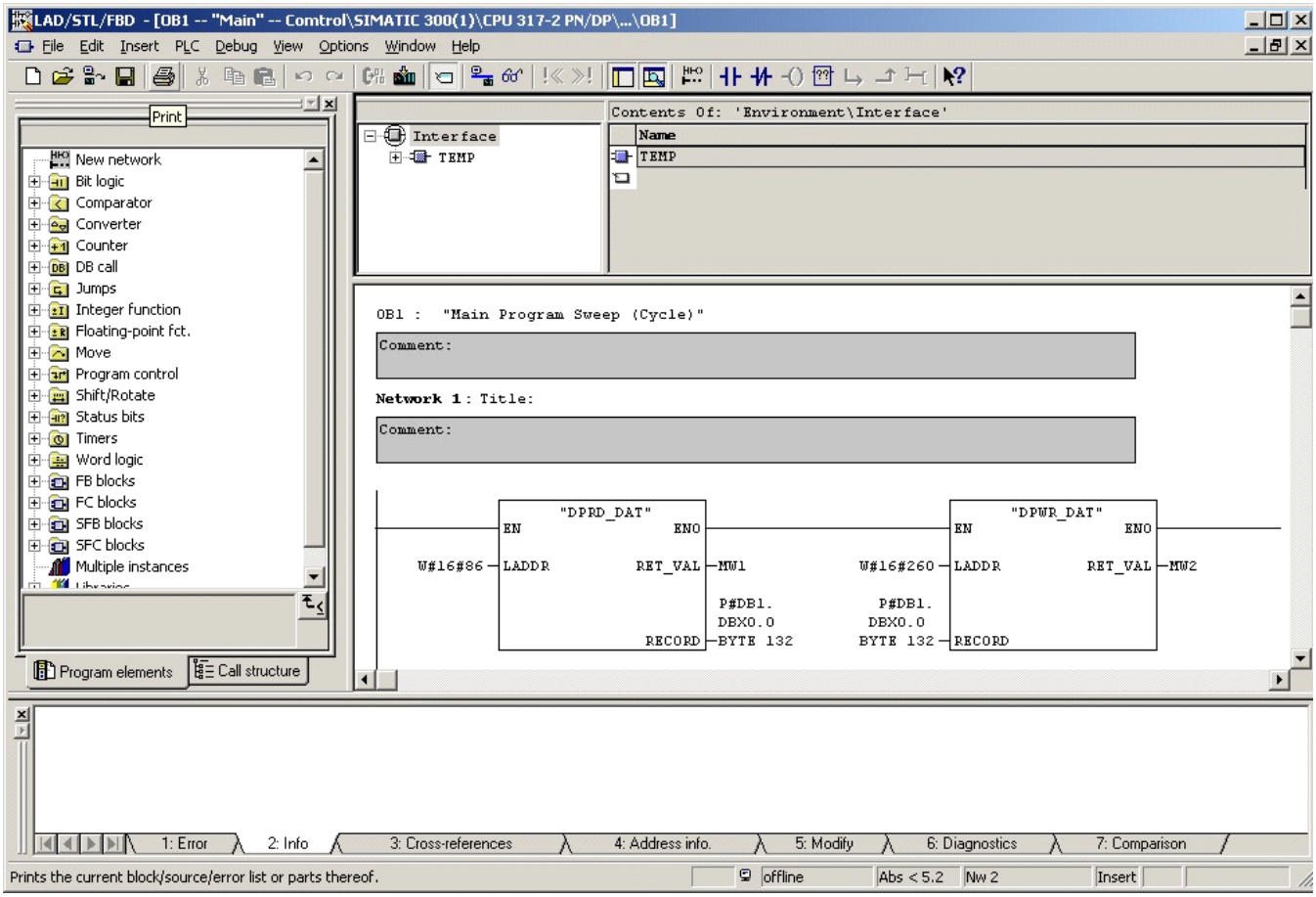
- Once you have installed the GSD, you can add the DeviceMaster UP to your network and add the input and output modules for your serial IO in Slots 1 and 11 as shown in the following screen shot:



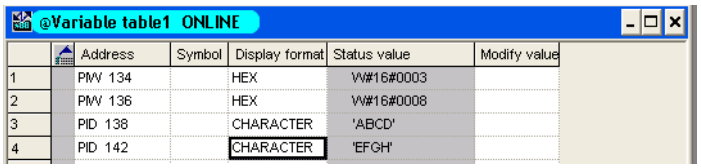
Note: *In the screen shot above, the serial port input slot (Slot 1) has a 128 BYTE input module assigned to address 134 and the output is assigned to the output address for 600. These are important because this is where your PLC program will interact with the DeviceMaster UP.*

- It is also worth noting that neither of these addresses is in the Process Image (PI) for this controller. Instead we will access this data through some SFC blocks that we will use to copy data into and out of the output and input addresses respectively. We could use the Process Image, but because of the size of the buffer, the default mapping is to outside the PI.

- All that is left is to write the code that copies the input data to the output data. In this example, we have mapped the DeviceMaster UP outside the process image, so getting all the data from the input to the output consistently (that is all the data in the buffer is from a single IO cycle) we use **SFC14 DPRD_DAT** as shown below:



- The **DPRD_DAT** function (SFC 14) copies all 132 bytes at address 134 decimal (or 0x86 in Hex) to DB1.DBX0.0. Then all this data is copied from DB1 to the output data located at address 608 decimal (or 0x260 in Hex). Note that there are 132 bytes being copied is that the full buffer includes a WORD for the sequence number (i.e. a counter identifying the sequence of the packet) and a WORD for the amount of data in that buffer (i.e. the maximum amount of data is 128 BYTEs, but the actual amount may be less – either way we copy the entire buffer).
- Perhaps this will be more clear if we examine some of the memory on the PLC in a running system. Consider the following watch window captured from **Step7** during system operation.



	Address	Symbol	Display format	Status value	Modify value
1	PIW 134		HEX	W#16#0003	
2	PIW 136		HEX	W#16#0008	
3	PID 138		CHARACTER	'ABCD'	
4	PID 142		CHARACTER	'EFGH'	

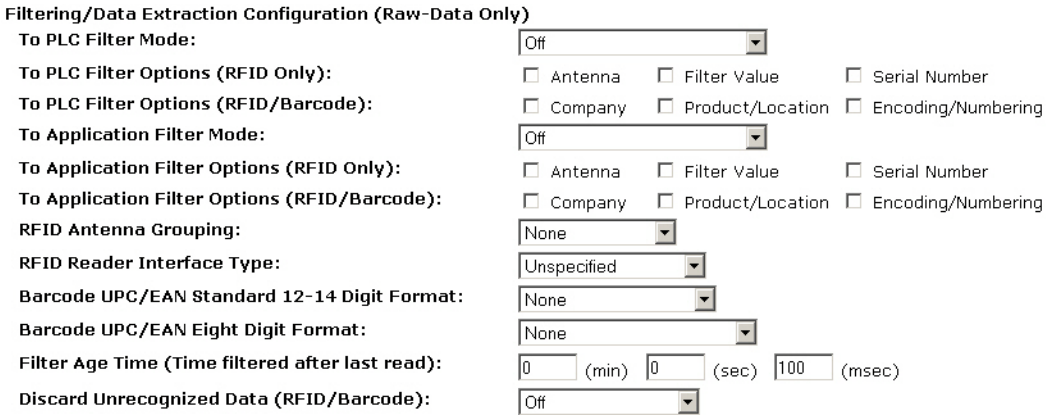
- Having sent two "packets" we can now examine the data to see if we received what we expected. In this case, we typed **ABCDEFG** (followed by an <Enter> key) twice to send the **ABCDEFG** packet twice. So we can verify the following?
  - We can see that the PLC has received notification that the 3rd packet received (note the sequence number 3 inline 1 of the window)?
  - The packet was 8 bytes long (note the length field equals 8 in line 2).?
  - If we look at the input data in PID 138 and PID 142 we see **ABCDEFGH**?
  - We see that the 3rd packet has also been sent back out the output port because the device tells us that it wrote packet 3 at input address PIW 600 on line 5.

This demonstrates that the program is effectively sending any data packet that is received by the DeviceMaster UP's serial port back out it's serial port shortly thereafter. The actual delay is influenced by the cycle I/O specified for the DeviceMaster UP in **Step7** and the timing of the data entry. By changing the size of the modules inserted into the slots in the DeviceMaster UP and changing the way data is written into the output memory, you can customize this application as necessary.

Filtering/Data Extraction Configuration (Patent Pending)

Select your filtering mode(s):

- Use *String* Filtering if:
  - Your received data can be no greater than 128 bytes in length.
  - Your received data is not in EPCglobal or barcode UPC/EAN formats or you do not want the DeviceMaster UP to extract the RFID tag or barcode parameters.
  - You want to filter and eliminate duplicate received messages.
- Use *RFID* filtering if:
  - You have an Alien or Intermec RFID reader or another reader that can provide RFID tag data is ASCII hex format similar to either an Alien or Intermec reader.
  - Your data is in EPCglobal format and you want the DeviceMaster UP to extract the RFID tag data parameters and filter based on those parameters.
- Use *Barcode* filtering if your barcode data is in UPC-A, UPC-E, EAN-13, JAN, EAN-14, or EAN-8 formats and you want the DeviceMaster UP to extract the barcode data parameters and filter based on those parameters.



PLC Filtering/Data Extraction

Under the *Filtering/Data Extraction Configuration* section corresponding to the desired serial or socket port:

1. Set **To PLC Filter Mode** to the desired mode.
2. **For String (128 char max)**: set the **Filter Age Time** to how long after the last read you want an entry to be filtered. In other words, after receiving a tag that matches the filtering criteria, no other tags that match the same criteria will be sent until the amount of time specified in the **Filter Age Time** has expired.
3. Go to the appropriate discussion for your environment.
   - *RFID (EPCglobal Formats)* on Page 12
   - *Barcode (UPC/EAN Formats)* on Page 13

RFID (EPCglobal Formats)

Use the following procedure to configure PLC filtering and data extraction properties for RFID devices using an EPCglobal format.

1. Set any or all of the **To PLC Filter Options (RFID Only)** filtering options.
2. Set any or all of the **To PLC Filter Options (RFID/Barcode)** filtering options.

   Note: *You must select at least one filtering option for filtering/data extraction to function.*
3. If **Antenna Grouping** is desired, set **RFID Antenna Grouping** option to reflect your antenna configuration. The antenna grouping allows you to treat a subset of the antennas on a RFID reader as a single RFID reader. See the *DeviceMaster UP Filtering and Data Extraction Reference Guide* for more detail.
4. Set the **RFID Reader Interface Type** to that of your RFID Reader configuration. If your RFID Reader is not listed, refer to the *DeviceMaster UP Filtering and Data Extraction Reference Guide* for the supported RFID reader interfaces. If your RFID reader format matches one of the listed formats, then set the **RFID Reader Interface Type** to that format.
5. Set the **Filter Age Time** to how long after the last read you want an entry to be filtered. In other words, after receiving a tag that matches the filtering criteria, no other tags that match the same criteria will be sent until the amount of time specified in the **Filter Age Time** has expired.
6. If you want the DeviceMaster UP to discard any non-RFID tag messages, set the **Discard Unrecognized Data** to either **To-PLC** or **To-PLC/Application**.

   Refer to the *DeviceMaster UP Filtering and Data Extraction Reference Guide* for more information.

   To PLC RFID Data Format

   When the PLC interface is operating in RFID filtering mode, all data sent to the PLC will be in the following format:

Field	Data Type	Description
Produced data sequence number	WORD Values = 0-65535 (FFFF Hex)	Sequence number that is incremented with each new message.
Length of RFID message	WORD Values = 20-148	Length in bytes of following data.
Company Code	DWORD[2]	Company Code extracted from tag data. Depending on encoding scheme, this field may include Company Prefixes, Company Prefix Indexes, or Government Managed Identifier.
Product/Location Code	DWORD[2]	Product Code extracted from tag data. Depending on encoding scheme, this field may include the Item Reference, Location Reference, Asset Reference, Object Class, or be set to zero.
Serial Number	DWORD[2]	Serial Number extracted from tag data. Depending on the encoding scheme, this field may include the Serial Number or Individual Asset Reference.
Encoding Scheme	WORD	Encoding Scheme from tag data.
Filtering Value	WORD	Filtering value from tag data.
Antenna Number	WORD	Antenna number on RFID reader/scanner.
Tag Data Length	WORD	Length of RFID tag string in bytes.
Tag Data	BYTE[128]	Tag data string (variable length field). May also include non-tag messages, which can optionally be sent to the PLC and/or application.

Barcode (UPC/EAN Formats)

Use the following procedure to configure PLC filtering and data extraction properties for barcode devices using a UPC/EAN format.

1. Set any or all of the **To PLC** filter options **(RFID/Barcode)** filtering options.

   Note: *You must select at least one for the filtering/data extraction to function.*

2. If you are using *standard* twelve to fourteen-digit UPC/EAN barcodes, set the **Barcode UPC/EAN 12-14 Digit Format** to match that of your barcodes. The **Company-5/Product-5** is the most popular format.
3. If you are using *eight*-digit UPC/EAN barcodes, set the **Barcode UPC/EAN 8 Digit Format** to match that of your barcodes.
4. If you want the DeviceMaster UP to discard any non-RFID tag messages, set the **Discard Unrecognized Data** to either **To-PLC** or **To-PLC/Application**.

Refer to the *DeviceMaster UP Filtering and Data Extraction Reference Guide* for more information.

To PLC Barcode Data Format

When the PLC interface is operating in barcode filtering mode, all data sent to the PLC will be in the following format:

Field	Size	Description
Produced data sequence number	WORD Values = 0-65535 (FFFF Hex)	Sequence number that is incremented with each new message.
Length	WORD Values = 12-140	Length in bytes of following data.
Company Code	DWORD	Company Code.
Product Code	DWORD	Product Code.
Numbering Code	WORD	Numbering code (from first byte(s) of barcode data).
Barcode Data Length	WORD	Length of barcode string in bytes.
Barcode Data	BYTE[128]	Barcode data string (variable length field).

Note: *The Company Code will be set to zero for all EAN-8 codes.*

Application Filtering/Data Extraction

Access the *Filtering/Data Extraction Configuration* section corresponding to the desired serial or socket port:

1. Set **To Application Filter Mode** to the desired mode.
2. For **String (128 char max)**: set the **Filter Age Time** to how long after the last read you want an entry to be filtered.
3. Use the appropriate procedure for your environment:
   - *RFID (EPCglobal Formats)* on Page 14
   - *Barcode (UPC/EAN Formats)* on Page 15

RFID (EPCglobal Formats)

Use the following procedure to configure application filtering and data extraction properties for RFID devices using an EPCglobal format.

1. Set any or all of the **To Application Filter Options (RFID Only)** filtering options.
2. Set any or all of the **To Application Filter Options (RFID/Barcode)** filtering options.

   Note: *You must select at least one filtering option for filtering/data extraction to function.*
3. If **Antenna Grouping** is desired, set **RFID Antenna Grouping** option to reflect your antenna configuration. The antenna grouping allows you to treat a subset of the antennas on a RFID reader as a single RFID reader. See the *DeviceMaster UP Filtering and Data Extraction Reference Guide* for more detail.
4. Set the **RFID Reader Interface Type** to that of your RFID reader configuration. If your RFID reader is not listed, refer to the *DeviceMaster UP Filtering and Data Extraction Reference Guide* for the supported RFID reader interfaces. If your RFID reader format matches one the listed formats, the set the **RFID Reader Interface Type** to that format.
5. Set the **Filter Age Time** to how long after the last read you want an entry to be filtered. In other words, after receiving a tag that matches the filtering criteria, no other tags that match the same criteria will be sent until the amount of time specified in the **Filter Age Time** has expired.
6. If you want the DeviceMaster UP to discard any non-RFID tag messages, set the **Discard Unrecognized Data** to either **To-Application** or **To-PLC/Application**.

Refer to the *DeviceMaster UP Filtering and Data Extraction Reference Guide* for more information.

To Application RFID Data Format

When the application interface is operating in RFID filtering mode, all data sent to the application is in the following format:

Field	Data Type	Description
Company Code	DWORD[2]	Company Code extracted from tag data. Depending on encoding scheme, this field may include Company Prefixes, Company Prefix Indexes, or Government Managed Identifier.
Product/Location Code	DWORD[2]	Product Code extracted from tag data. Depending on encoding scheme, this field may include the Item Reference, Location Reference, Asset Reference, Object Class, or be set to zero.
Serial Number	DWORD[2]	Serial Number extracted from tag data. Depending on the encoding scheme, this field may include the Serial Number or Individual Asset Reference.
Encoding Scheme	WORD	Encoding Scheme from tag data.
Filtering Value	WORD	Filtering Value from tag data.
Antenna Number	WORD	Antenna Number on RFID reader/scanner.
Tag Data Length	WORD	Length of RFID tag string in bytes.
Tag Data	BYTE[128]	Tag data string (variable length field). May also include non-tag messages, which can optionally be sent to the PLC and/or application

Note: *The RFID parameters will be sent to the application in big-endian format. All parameters, with the exception of the tag data string, will have to be byte-swapped for use on a little-endian system.*

Barcode (UPC/EAN Formats)

Use the following procedure to configure application filtering and data extraction properties for barcode devices using a UPC/EAN format.

1. Set any or all of the **To Application Filter Options (RFID/Barcode)** filtering options.

   Note: *You must select at least one for the filtering/data extraction to function.)*

2. If you are using *standard* twelve to fourteen-digit UPC/EAN barcodes, set the **Barcode UPC/EAN 12-14 Digit Format** to match that of your barcodes. The **Company-5/Product-5** is the most popular format.
3. If you are using *eight*-digit UPC/EAN barcodes, set the **Barcode UPC/EAN 8 Digit Format** to match that of your barcodes.
4. If you want the DeviceMaster UP to discard any non-RFID tag messages, set the **Discard Unrecognized Data** to either **To-Application** or **To-PLC/Application**.

Refer to the *DeviceMaster UP Filtering and Data Extraction Reference Guide* for more information.

To Application Barcode Data Format

When the application interface is operating in **barcode** filtering mode, all data sent to the application is in the following format:

Field	Size	Description
Company Code	DWORD	Company Code
Product Code	DWORD	Product Code
Numbering Code	WORD	Numbering Code (from first byte(s) of barcode data)
Barcode Data Length	WORD	Length of barcode string in bytes
Barcode Data	BYTE[128]	Barcode data string (variable length field)

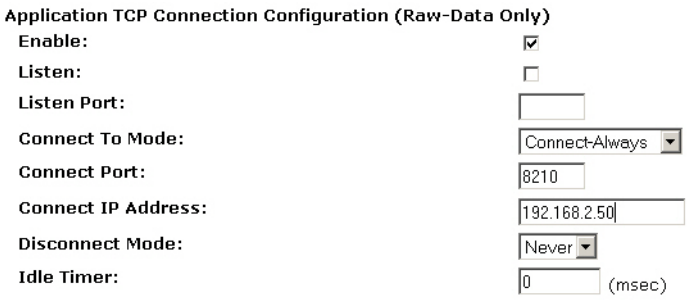
Note: *The Company Code will be set to zero for all EAN-8 codes.*

The Barcode parameters will be sent to the application in big-endian format. All parameters, with the exception of the barcode data string, will have to be byte-swapped for use on a little-endian system.

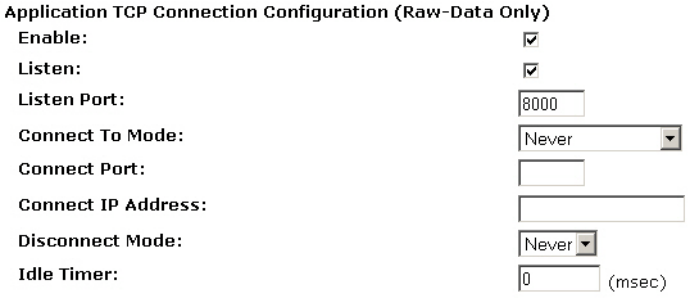
Application Socket Configuration (Patent Pending)

Access the *Application TCP Connection Configuration* section corresponding to the desired serial or socket port:

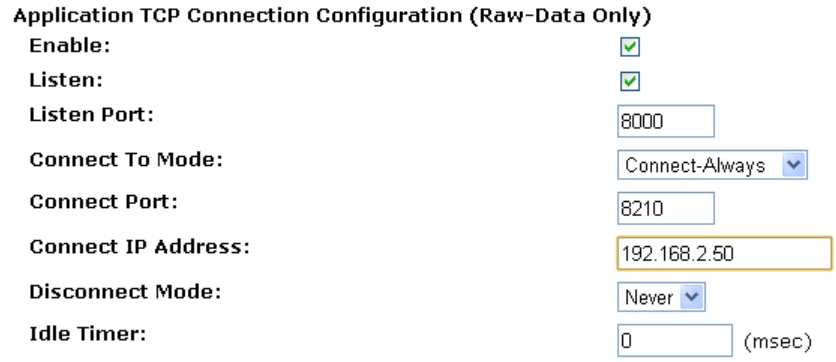
1. Select **Enable**.
2. If your Ethernet TCP/IP application requires another device to connect to it, configure the socket port on the DeviceMaster UP to **Connect** mode:
   a. Leave **Listen** unselected.
   b. Set **Connect To Mode** to **Connect-Always**.
   c. Set the **Connect Port** to the socket port number of your Ethernet application.
   d. Set the **Connect IP Address** to the IP address of your Ethernet application.
   e. Set **Disconnect Mode** to **Never**.



3. If your Ethernet TCP/IP application is configured to connect to another device, configure the socket port on the DeviceMaster UP to **Listen** mode:
   a. Select **Listen**.
   b. Use the default **Listen Port** on the DeviceMaster UP of ***8xxx*** or designate your own.
   c. Set **Connect To Mode** to **Never**.
   d. Set **Disconnect Mode** to **Never**.
   e. Configure your Ethernet application to connect to the DeviceMaster UP at the DeviceMaster UP IP address and **Listen Port**.



4. If you do not know if your application will connect to another Ethernet device, but do know your application's socket port and IP address, you can do the following to enable both the **Listen** and **Connect** modes:
   a. Select **Listen**.
   b. Use the default **Listen Port** on the DeviceMaster UP of ***8xxx*** or designate your own.
   c. Set **Connect To Mode** to **Connect-Always**.
   d. Set the **Connect Port** to the socket port number of your Ethernet application.
   e. Set the **Connect IP Address** to the IP address of your Ethernet application.
   f. Set **Disconnect Mode** to **Never**.
   g. Optionally configure your Ethernet application to connect to the DeviceMaster UP at the DeviceMaster UP IP address and **Listen Port**.



Troubleshooting

You can refer to the following information to troubleshoot problems. In addition, you may want to refer to the *DeviceMaster UP Hardware and Configuration Guide*.

Issue	Possible Corrective Action
What are the slot numbers for serial and Ethernet input and output?	Use the slot number to determine whether the input and output is for a serial port or a TCP/IP port. Serial input is received in Slot 1, serial output is sent through Slot 11, TCP/IP input is received in Slot 21 and TCP/IP output is sent through Slot 31. This will eventually allow us to handle I/O for Ports 2-4 in Slot 2, 3 and 4 or there corresponding ports in the 10s, 20s and 30s range.
What if the system fault light is on?	• Look at the hardware diagnostics for the CPU that the DeviceMaster UP should be connected to. • Examine any faults and take corrective action. • Ensure that your device is assigned the correct PROFINET IO *Device Name* using **HW Config**. • Ensure that all the components are properly plugged in and powered on.
What if I never see data appearing in PLC Input memory?	• Verify that there are no faults having to do with the DeviceMaster UP. • Verify that the PLC is in **RUN** mode. • Verify that you are looking at the correct Input address as assigned in the **HW Config** tool.
What if after writing data to output memory it doesn't appear?	• Verify that you are writing to the Output Memory address you configured in **HW Config**. • Verify that you are setting the Sequence Number and Length fields correctly in the Output Memory. The DeviceMaster UP Output Memory begins with two WORD values; a 16-bit Sequence Number and 16-bit Length field at the beginning of every “packet” of serial or Ethernet data. The sequence number allows you to control when a new packet is sent by incrementing the outgoing sequence number. See Step 4 in *Configuring Read/Write Devices* on Page 6. • Verify that no system faults are occurring on PLC or DeviceMaster UP.
What if I miss some data from the PLC – i.e. it never seems to arrive?	PNIO is a cyclic I/O protocol. That is, data is exchanged between the PLC and the DeviceMaster UP every *N* milliseconds. The frequency of the update is controlled (as with all PNIO devices) in the **IO Cycle** tab of the **Properties** window for the DeviceMaster UP. This is accessed by double-clicking on the device in the **HW Config** application from within **Step7**. Update rates from 8ms to 512ms are available in power of two increments. • If you are not getting the information from the serial device soon enough, you may have the update rate configured to be too slow in the **HW Config**. • Is the connection dropping? Verify that you are not seeing system faults in the history.
I can't find the DeviceMaster UP device in **HW Config**!	It should be under PROFINET IO – *Additional Field Devices – Gateway*. If it is not there, try reinstalling the GSDML file.

Issue	Possible Corrective Action
Why can't I see the incoming data even though there are no fault indications on the PLC? Why are my Input Memory addresses so high?	The DeviceMaster UP delivers relatively large amounts of data to the PLC (i.e. we are not simply delivering 8 digital I/Os or a few analog values) **Step7** frequently puts the Input Data from our device outside the "Process Image". *What is the Process Image?* The Process Image is a special area of memory that is always kept consistent. That is, all the data in the buffer is from a single I/O cycle, and can not have part of a new packet in it if new data arrives while the program is processing it. However, the PI is also limited in size, and so large (e.g. 128 byte) buffers are often stored outside the PI. *Reading Consistent Data From Outside The Process Image* In order to ensure that the data being processed in their logic is consistent (i.e. it is all from a single serial packet) they should copy their data to a data block before processing it. This is done using **DPRD_DAT** (SFC14), see the example on Page 10. Siemens PLCs require a slightly different form of addressing the input data that is not in the Process Image. Namely, instead of expressing the address as IW600 (input word at address 600) one must type PIW600 (pointer input word 600). This is true for the Variable Table provided by **Step7** to monitor memory on the PLC and for some PLC instructions as well. *Writing Consistent Data Outside The Process Image* To write data consistently, use the **DPWR_DAT** (SFC15) to copy data from a DB (data buffer) to the output memory, see *Example Programs* on Page 8.

Technical Support

Review the *Troubleshooting* on Page 17 section before contacting Technical Support. If you need technical support, contact Comtrol using one of the following methods.

Contact Method	Corporate Headquarters
Downloads	http://www.comtrol.com/support
Web site	http://www.comtrol.com
Phone	(763) 957-6000