

Adding a Modbus TCP Device to the Network



Overview

This section extends the sample Unity Pro application, by describing how to:

- add an STB NIP 2212 Modbus TCP network interface module to your Unity Pro application
- configure the STB NIP 2212 module
- configure a Modbus TCP connection linking the BMX NOC 0401 communication module and the STB NIP 2212 network interface module

NOTE: The instructions in this chapter describe a single, specific device configuration example. Refer to the Unity Pro help files for additional information about alternative configuration choices.

What Is in This Section?

This section contains the following topics:

- [Setting Up Your Network](#)
- [Adding an STB NIP 2212 Remote Device](#)
- [Configuring STB NIP 2212 Properties](#)
- [Connecting to the Advantys STB Island](#)
- [Configuring I/O Items](#)

© 2014 Schneider Electric. All rights reserved.

Setting Up Your Network



[About us](#)

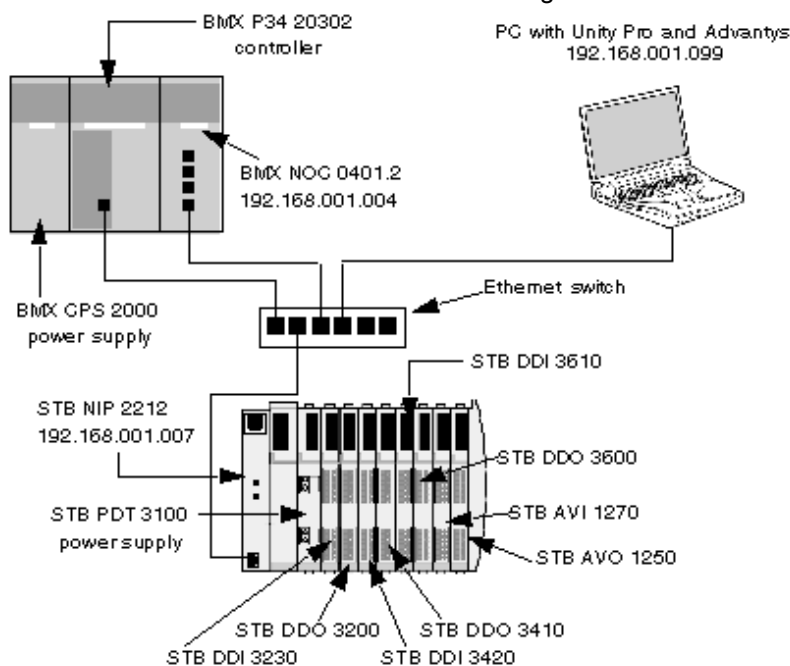
Overview

This sample network includes the following hardware and software:

- a controller rack with:
 - BMX CPS 2000, 100/240 VAC power supply
 - BMX P34 20302 controller
 - BMX NOC 0401.2 Ethernet communication module
- a remote STB Advantys island with:
 - STB NIP 2212 Modbus TCP network interface module
 - STB PDT 3100 power distribution module
 - STB DDI 3230 2 pt digital input module
 - STB DDO 3200 2 pt digital output module
 - STB DDI 3420 4 pt digital input module
 - STB DDO 3410 4 pt digital output module
 - STB DDI 3610 6 pt digital input module
 - STB DDO 3600 6 pt digital output module
 - STB AVI 1270 2 pt analog input module
 - STB AVO 1250 2 pt analog output module
- a PC running both Unity Pro (version 5.0 or higher) and Advantys configuration software (version 5.0 or higher)
- an Ethernet managed switch that is connected to the both the controller and island by means of twisted pair Ethernet cable and RJ45 connectors.

Network Topology

The Ethernet network devices used in this configuration include the following:



To re-create this example:

- use the IP addresses for your own configuration's:
 - PC
 - BMX NOC 0401.2 Ethernet communication module
 - STB NIP 2212 network interface module
- check wiring

NOTE: Unity Pro software running in the PC is used to configure the BMX P34 20302 controller. In this example, the PC is indirectly wired to the CPU's Ethernet port via the Ethernet switch. Alternatively, you could bypass the switch and directly wire the PC to either the CPU's Modbus or USB ports.

© 2014 Schneider Electric. All rights reserved.

Adding an STB NIP 2212 Remote Device



[About us](#)

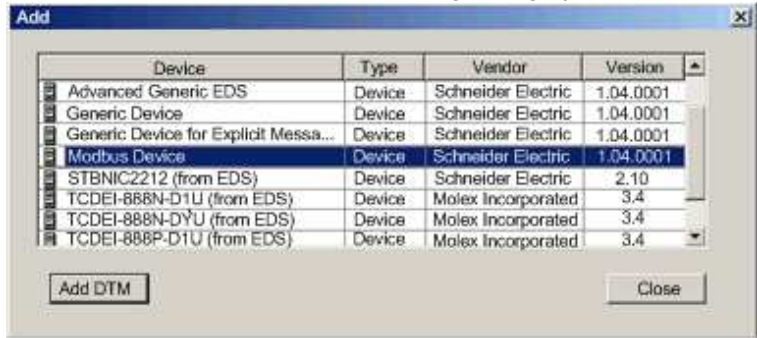
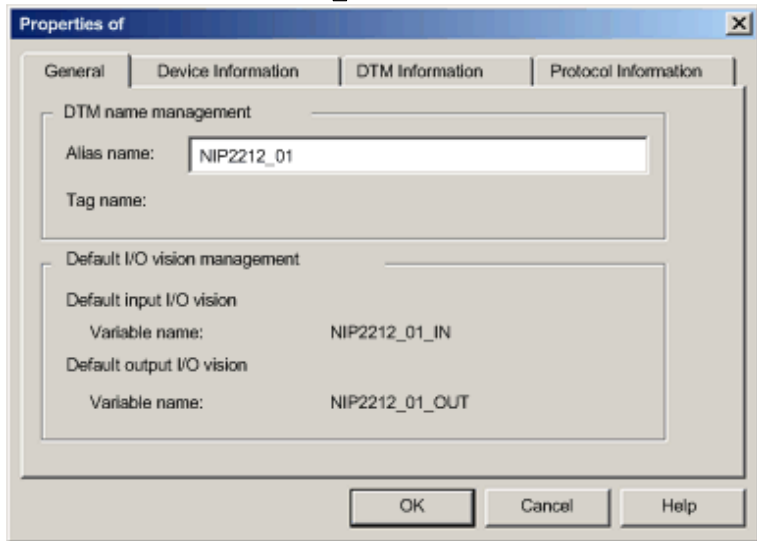
Overview

Use the generic Modbus DTM selection, in the **Add** dialog, to select and insert an STB NIP 2212 module to your project.

Adding an STB NIP 2212 Remote Device

To add the STB NIP 2212 to your project, follow these steps:

Step	Action
1	In the DTM Browser , select the Ethernet communication module node, and then click the right mouse button. A pop-up menu opens.
2	

	<p>In the menu, select Add... The following dialog opens:</p> 
3	<p>In the Add dialog, select the STBNIP2212, then click Add DTM. The Properties window for the STB NIP 2212 network interface module opens.</p>
4	<p>In the General page of the Properties window, edit the default Alias name to read NIP2212_01:</p>  <p>When you edit the Alias name, Unity Pro applies it as the base name for both structure and variable names. NOTE: No additional editing needs to be performed in the pages of this window. Except for the Alias name field, parameters are read-only.</p>
5	<p>Click OK. Unity Pro adds the new STB NIP 2212 network interface module to the DTM Browser, beneath the communication module.</p>
6	<p>Refer to the topic Configuring Properties in the Device Editor for instructions on how to save your configuration edits.</p>

The next step is to configure the device you have just added to the project.

© 2014 Schneider Electric. All rights reserved.

Configuring STB NIP 2212 Properties



 [About us](#)

Overview

Use the pages of the **Device Editor** to view and edit settings for a remote device. To edit the device settings, [disconnect the DTM from the remote device](#).

To display the DTM settings for a remote device, select the device name, which is found under the **Device List** node in the left pane of the **Device Editor**.

For the purposes of this example, which configures an STB NIP 2212 network interface module, select the node named **NIP2212_01**. The **Device Editor** displays the following pages:

- Properties
- Address Setting
- Request Setting

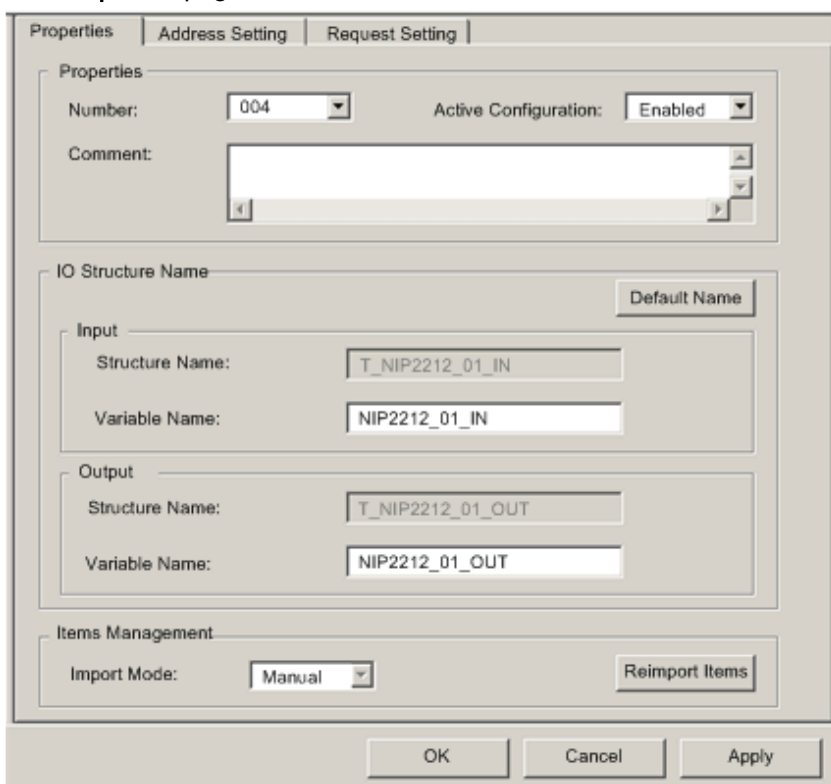
NOTE: Refer to the topic [Configuring Properties in the Device Editor](#) for instructions on how to edit properties.

Configuring the Properties Page

Use the **Properties** page to:

- add the remote device to, or remove it from, the configuration
- edit the base name for variables and data structures used by the remote device
- indicate how input and output items will be created and edited

The **Properties** page for an STB NIP 2212 network interface module looks like this:



The following settings are used in this sample configuration. Use settings that are appropriate for your actual application:

Step	Action	
1	In the Properties section of the page, edit the following:	
	Number	The relative position of the device in the list, from 0...127. For this example, accept the default of 004 .
	Active Configuration	<ul style="list-style-type: none"> • Enabled: adds this device to the Unity Pro project configuration

		<ul style="list-style-type: none"> • Disabled: removes this device from the Unity Pro project configuration <p>Accept the default setting of Enabled.</p>
2	In the IO Structure Name section of the page, edit the following:	
	Input area:	
	Structure Name	(Read-only) Unity Pro automatically assigns a structure name based on the variable name, in this case T_NIP2212_01_IN .
	Variable Name	Accept the auto-generated variable name (based on the alias name): NIP2212_01_IN .
	Output area:	
	Structure Name	(Read-only) Unity Pro automatically assigns a structure name based on the variable name, in this case T_NIP2212_01_OUT .
	Variable Name	Accept the auto-generated variable name (based on the alias name): NIP2212_01_OUT .
	Default Name button	Restores the default variable and structure names. For this example, custom names are used.
3	In the Items Management section of the page, edit the following:	
	Import mode	<ul style="list-style-type: none"> • Automatic: I/O items are taken from the device DTM and updated if the items list in the device DTM changes. Items cannot be edited in the Device Editor. • Manual: I/O items are manually added in the Device Editor. The I/O items list is not affected by changes to the device DTM. <p>In this example, select Manual.</p>
	Reimport Items	Imports the I/O items list from the device DTM, overwriting any manual I/O item edits. Enabled only when Import mode is set to Manual .
4	Click Apply to save your edits, and leave the window open for further edits.	

Configuring the Address Setting Page

Use the **Address Setting** page to:

- configure the IP address for the remote device
- enable, or disable, DHCP client software for the remote device

When the DHCP client software is enabled in the remote device, it will obtain its IP address from the DHCP server in the Ethernet communication module. The **Address Setting** page looks like this:

The following settings are used in this sample configuration. Use settings that are appropriate for your actual application:

Step	Action										
1	In the Address Settings page, edit the following:										
	<table border="1"> <tr> <td>IP Address</td> <td> <p>By default:</p> <ul style="list-style-type: none"> the first three octet values equal the first three octet values of the Ethernet communication module, and the fourth octet value equals this device Number setting—in this case, 004. <p>In this example, the IP address is 192.169.1.7.</p> </td> </tr> <tr> <td>DHCP for this Device</td> <td> <ul style="list-style-type: none"> Enabled activates the DHCP client in this device. The device obtains its IP address from the DHCP service provided by the Ethernet communication module and appears on the auto-generated DHCP client list. Disabled (the default) de-activates the DHCP client in this device. <p>Select Enabled.</p> </td> </tr> <tr> <td>Identified by</td> <td> <p>If DHCP for this Device is Enabled, this indicates the device identifier type:</p> <ul style="list-style-type: none"> MAC Address, or Device Name <p>Select Device Name.</p> </td> </tr> <tr> <td>Identifier</td> <td> <p>If DHCP for this Device is Enabled, the specific device MAC Address or Name value.</p> <p>Type in NIP2212_01.</p> </td> </tr> <tr> <td>Mask</td> <td> <p>The device subnet mask. The default = 255.255.255.0.</p> <p>Accept the default value.</p> </td> </tr> </table>	IP Address	<p>By default:</p> <ul style="list-style-type: none"> the first three octet values equal the first three octet values of the Ethernet communication module, and the fourth octet value equals this device Number setting—in this case, 004. <p>In this example, the IP address is 192.169.1.7.</p>	DHCP for this Device	<ul style="list-style-type: none"> Enabled activates the DHCP client in this device. The device obtains its IP address from the DHCP service provided by the Ethernet communication module and appears on the auto-generated DHCP client list. Disabled (the default) de-activates the DHCP client in this device. <p>Select Enabled.</p>	Identified by	<p>If DHCP for this Device is Enabled, this indicates the device identifier type:</p> <ul style="list-style-type: none"> MAC Address, or Device Name <p>Select Device Name.</p>	Identifier	<p>If DHCP for this Device is Enabled, the specific device MAC Address or Name value.</p> <p>Type in NIP2212_01.</p>	Mask	<p>The device subnet mask. The default = 255.255.255.0.</p> <p>Accept the default value.</p>
IP Address	<p>By default:</p> <ul style="list-style-type: none"> the first three octet values equal the first three octet values of the Ethernet communication module, and the fourth octet value equals this device Number setting—in this case, 004. <p>In this example, the IP address is 192.169.1.7.</p>										
DHCP for this Device	<ul style="list-style-type: none"> Enabled activates the DHCP client in this device. The device obtains its IP address from the DHCP service provided by the Ethernet communication module and appears on the auto-generated DHCP client list. Disabled (the default) de-activates the DHCP client in this device. <p>Select Enabled.</p>										
Identified by	<p>If DHCP for this Device is Enabled, this indicates the device identifier type:</p> <ul style="list-style-type: none"> MAC Address, or Device Name <p>Select Device Name.</p>										
Identifier	<p>If DHCP for this Device is Enabled, the specific device MAC Address or Name value.</p> <p>Type in NIP2212_01.</p>										
Mask	<p>The device subnet mask. The default = 255.255.255.0.</p> <p>Accept the default value.</p>										

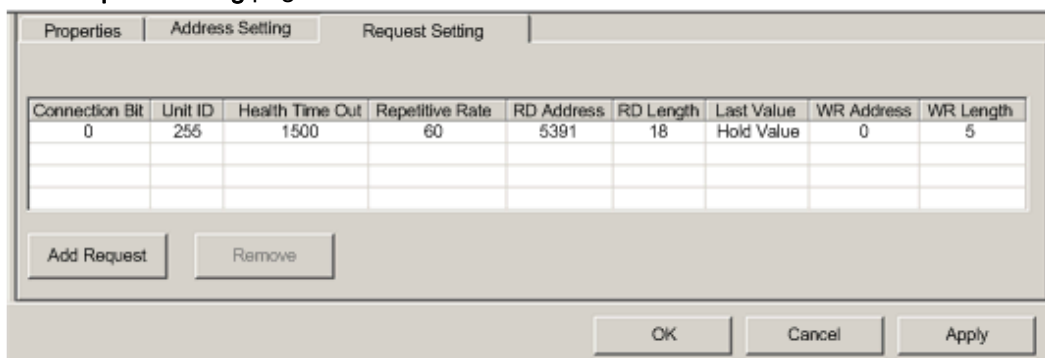
	Gateway	The gateway address used to reach this device. The default of 0.0.0.0 indicates this device is located on the same subnet as the Ethernet communication module. Accept the default value.
2	Click Apply to save your edits, and leave the window open for further edits.	

The next step is to configure the connection between the communication module and the remote device.

Configuring the Request Setting Page

Use the **Request Setting** page to add, configure, and remove Modbus requests for the remote device. Each request represents a separate link between the communication module and the remote device.

The **Request Setting** page for an STB NIP 2212 network interface module looks like this:



The **Add Request** function is enabled only when **Import Mode** is set to **Manual**.

The following settings are used in this sample configuration. Use settings that are appropriate for your actual application:

Step	Action				
1	In the Request Settings page, edit the following:				
	<table border="1"> <tr> <td>Connection Bit</td> <td> (Read-only) The offset for both the health bit and the control bit for this connection. Offset values are auto-generated by the Unity Pro Ethernet Configuration Tool—starting at 0—based on the connection type, in the following order: <ol style="list-style-type: none"> 1. Modbus TCP connections 2. Local Slave connections 3. EtherNet/IP connections NOTE: When this Modbus TCP connection is created, the offset values for the previously created local slave and EtherNet/IP connections are incremented by 1: the local slave connection bit is set to 1, and the EtherNet/IP connection bit value is set to 2. </td> </tr> <tr> <td>Unit ID</td> <td> The number of the device, or module, that is the target of the connection. A value of: <ul style="list-style-type: none"> • 255 (the default) used to access the Ethernet communication module itself • 254 causes no Modbus message to be sent; the module reports an event • 0...253 identifies the device number of the target device, behind a Modbus TCP to Modbus gateway </td> </tr> </table>	Connection Bit	(Read-only) The offset for both the health bit and the control bit for this connection. Offset values are auto-generated by the Unity Pro Ethernet Configuration Tool—starting at 0—based on the connection type, in the following order: <ol style="list-style-type: none"> 1. Modbus TCP connections 2. Local Slave connections 3. EtherNet/IP connections NOTE: When this Modbus TCP connection is created, the offset values for the previously created local slave and EtherNet/IP connections are incremented by 1: the local slave connection bit is set to 1, and the EtherNet/IP connection bit value is set to 2.	Unit ID	The number of the device, or module, that is the target of the connection. A value of: <ul style="list-style-type: none"> • 255 (the default) used to access the Ethernet communication module itself • 254 causes no Modbus message to be sent; the module reports an event • 0...253 identifies the device number of the target device, behind a Modbus TCP to Modbus gateway
Connection Bit	(Read-only) The offset for both the health bit and the control bit for this connection. Offset values are auto-generated by the Unity Pro Ethernet Configuration Tool—starting at 0—based on the connection type, in the following order: <ol style="list-style-type: none"> 1. Modbus TCP connections 2. Local Slave connections 3. EtherNet/IP connections NOTE: When this Modbus TCP connection is created, the offset values for the previously created local slave and EtherNet/IP connections are incremented by 1: the local slave connection bit is set to 1, and the EtherNet/IP connection bit value is set to 2.				
Unit ID	The number of the device, or module, that is the target of the connection. A value of: <ul style="list-style-type: none"> • 255 (the default) used to access the Ethernet communication module itself • 254 causes no Modbus message to be sent; the module reports an event • 0...253 identifies the device number of the target device, behind a Modbus TCP to Modbus gateway 				

	<p>NOTE: When accessing data in the Ethernet communication module itself, use 255. When accessing data in the application running in the PLC, use a value from 0 to 254 (a value of 1 is recommended).</p> <p>Because the remote device itself is the request target, accept the default value of 255.</p>
Health Timeout	The maximum allowed period, in milliseconds, between device responses, from 0...120000 ms in increments of 5 ms. When this setting is exceeded, the health timeout bit is set to 1. The default = 1500 ms. Accept the default value of 1500 .
Repetitive Rate	The data scan rate, from 0...60000 ms, in intervals of 5 ms. The default = 60 ms. Accept the default value of 60 .
RD Address	Address in the remote device of the input data image. The input data image begins at word 45391. Because there is an offset of 40000 in the M340 platform, type in a value of 5391 .
RD Length	The number of words in the remote device, from 0...125, that the communication module will read. Because the Modbus device will be configured for 18 words of input items, type in a value of 18 .
Last Value	The behavior of inputs in the application in the event communication is lost: <ul style="list-style-type: none"> • Hold Value (the default) • Set To Zero Accept the default.
WR Address	Address in the remote device of the output data image. The output data image begins at word 40000. Because there is an offset of 40000 in the M340 platform, type in a value of 0 .
WR Length	The number of words in the remote device, from 0 to 120, to which the communication module will write. Because the Modbus device will be configured for 5 words of output items, type in a value of 5 .
2	Click OK to save your edits, and close the window.

The next step is to connect the Unity Pro project to the Advantys Island.

[© 2014 Schneider Electric. All rights reserved.](#)

Connecting to the Advantys STB Island



[About us](#)

Overview

In this example, you will use the Advantys configuration software running on your PC to:

- connect the Advantys configuration software to the STB NIP 2212 and the 8 I/O modules that comprise the Advantys STB island
- upload Advantys STB island configuration to the Advantys configuration software in your PC
- display a fieldbus image for the Advantys STB island showing the relative location of:
 - input data
 - output data

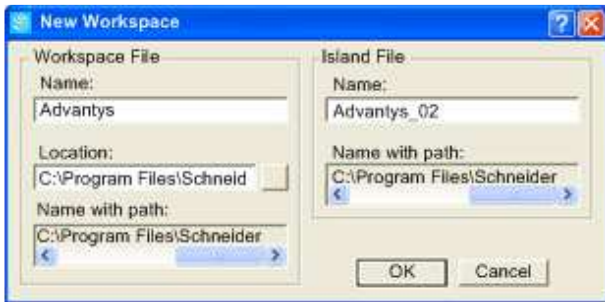
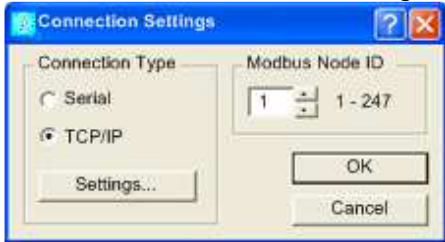

Using the data presented in the fieldbus image, you can use Unity Pro to create input and output

items that map to specific input, output, and output echo data.

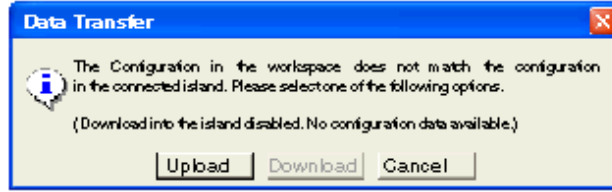
NOTE: Before proceeding with the following instructions, confirm that you have auto-configured the Advantys STB island by pressing the **RST** button on the front of the STB NIP 2212 module.

Making the Connection

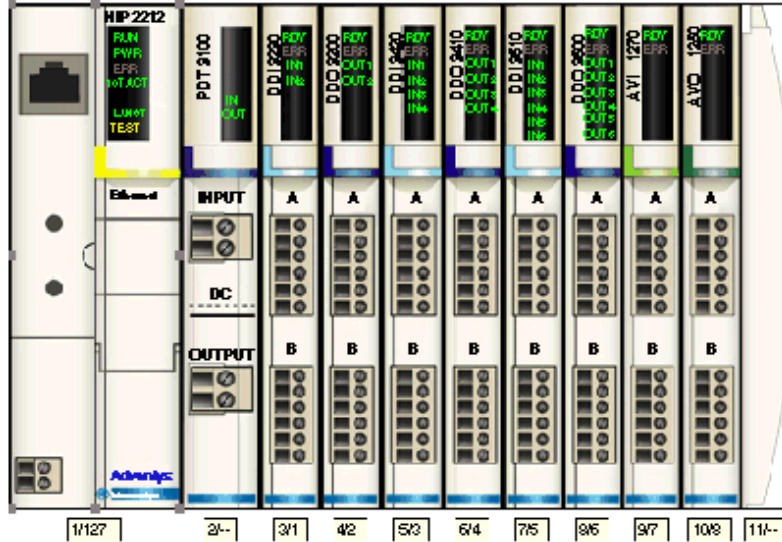
To connect to the STB NIP 2212 and I/O modules using the Advantys configuration software:

Step	Action
1	Startup the Advantys configuration software on your PC. A dialog opens displaying available project types.
2	Select STB .
3	Select File → New Workspace . The New Workspace window opens (below).
4	For this example, type in the following field values: <ul style="list-style-type: none"> • for the field Workspace File type in Advantys • for the field Island File type in Advantys_02 
5	Click OK . The Advantys configuration software displays an empty DIN rail in the center of the screen.
6	Select Online → Connection Settings . The Connection Settings window opens (below).
7	In the Connection Settings window, accept the Modbus Node ID default setting of 1, select TCP/IP , and click the Settings... button:  <p>The TCP/IP Parameters dialog opens (below).</p>
8	In the Remote IP Address field, type in the IP address for the STB NIP 2212, in this example: 192.168.1.7 . 
9	Click OK to close the TCP/IP Parameters dialog, and click OK again to close the Connection Settings dialog.

10 Select **Online** → **Connect**. The **Data Transfer** dialog opens (below):



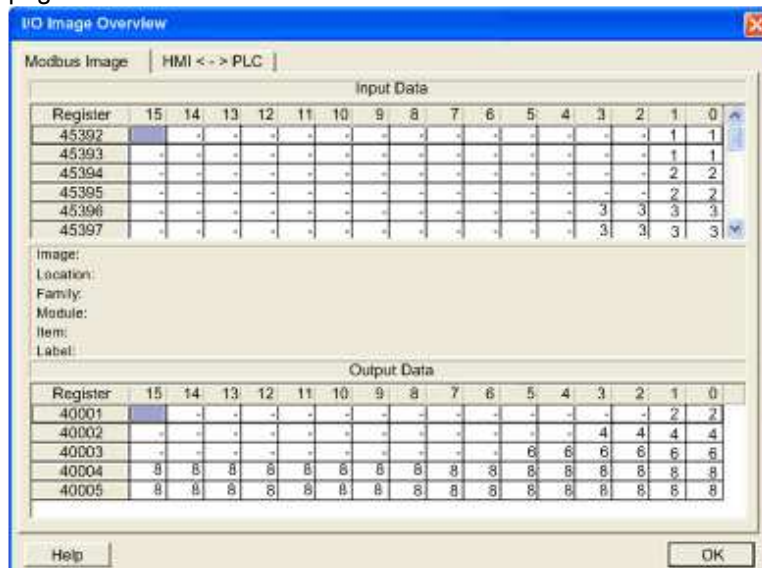
11 Select **Upload** in the **Data Transfer** dialog. The island workspace is populated with island data and shows the STB NIP 2212 and the island modules (below):



Note: A box appears beneath each module containing one or two integers—for example **3/1** . These integers serve the following purpose:

- The left-side integer (3 in this example) identifies the module’s physical position—left to right—among the modules in the rack.
- The right-side integer (1 in this example) identifies the module’s relative position—left to right—among only data producing/receiving modules. If the module is not a data producing/receiving module (e.g. a power supply, or end of segment module) no right-side integer appears.

12 Select **Island** → **I/O Image Overview**. The **I/O Image** window opens to the **Fieldbus Image** page:



Each table cell contains an integer that identifies the relative rack position of a data producing/receiving module with input or output data in that cell. For example:

- the STB DDI 3230 input module is the first data producing or receiving module in the rack; its data and status information is indicated by the integer 1 in bits 0...1 of registers 45392 and 45393 in the **Input Data** table
- the STB DDO 3600 output module is the sixth data producing module in the rack; its output echo and status data is designated by the integer 6 in bits 0 - 5 of register 45402 and in bits 0 - 5 of register 45403 in the **Input Data** table; its output data is designated by the integer 6 in bits 0 - 5 of register 40003 in the **Output Data** table

NOTE:

- Select a cell in either the **Input Data** or **Output Data** tables to display—in the middle of the page—a description of the cell data and its source module.
- Convert the size of the **Input Data** table and the **Output Data** table from words to bytes (i.e. divide by 2), then use that information when setting the **RD Length** (inputs) and **WR Length** (outputs) parameters in the **Request Setting** page for the remote Modbus TCP device.

© 2014 Schneider Electric. All rights reserved.

Configuring I/O Items



 [About us](#)

Overview

The next task in this example is to add I/O items to the configuration of the STB NIP 2212 and its 8 I/O modules. To accomplish this:

- use the **Modbus Image** page of the Advantys configuration software to identify the relative position of each I/O module's inputs and outputs
- use the Unity Pro **Device Editor** to create input and output items, defining each item's:
 - name
 - data type

NOTE: You can manually configure I/O items only when **Input Mode** is set to **Manual**.

I/O Item Types and Sizes

Because the Modbus TCP network interface module transmits data in the form of 16-bit words, in this example you will create every input and output item using the **WORD** data type. This remains true even if the item contains only a few bits of data. Bit-packing is not permitted when, as in this example, the remote device is a Modbus TCP network interface module.

NOTE: When you add more devices to your network, it may be necessary to increase [the size and index location of both inputs and outputs for your Unity Pro project](#).

In this example, the following number and type of items need to be created:

- 18 input words
- 5 output words

Mapping Input and Output Items

Use the **Fieldbus Image** page of the **I/O Image Overview** window in the Advantys configuration software to identify the number and type of I/O items you need to create, as follows:

Step	Action

1	In the Advantys configuration software, select Island → I/O Image Overview . The I/O Image window opens to the Modbus Image page.
2	Select the cell 0 of the first word (45392) in the Input Data table to display—in the middle of the page—a description of the cell data and its source module.
3	Make a note of the register number and item information for that word.
4	Repeat steps 2 and 3 for each word.

NOTE: The Modbus Image presents input and output data in the form of 16-bit words (starting with word 1). You need to maintain this data format as you create input and output items in Unity Pro.

NOTE: When you create items, align items of data type `WORD` and `DWORD`, as follows:

- `WORD` items: align these items on a 16-bit boundary
- `DWORD` items: align these items on a 32-bit boundary

This process yields the following tables of input and output data:

Input Data (Read):

Advantys Modbus Image		Unity Pro Items		STB Module	Description
Register	Bit(s)	Bytes	Bit(s)		
45392	0-1	0	0-1	DDI 3230	input data
		1	not used		
45393	0-1	2	0-1	DDI 3230	input status
		3	not used		
45394	0-1	4	0-1	DDO 3200	output data echo
		5	not used		
45395	0-1	6	0-1	DDO 3200	output status
		7	not used		
45396	0-3	8	0-3	DDI 3420	input data
		9	not used		
45397	0-3	10	0-3	DDI 3420	input status
		11	not used		
45398	0-3	12	0-3	DDO 3410	output data echo
		13	not used		
45399	0-3	14	0-3	DDO 3410	output status
		15	not used		
45400	0-5	16	0-5	DDI 3610	input data
		17	not used		
45401	0-5	18	0-5	DDI 3610	input status
		19	not used		
45402	0-5	20	0-5	DDO 3600	output data echo

		21	not used		
45403	0-5	22	0-5	DDO 3600	output status
		23	not used		
45404	0-15	24	0-7	AVI 1270	input data ch 1
		25	0-7		
45405	0-7	26	0-7	AVI 1270	input status ch 1
		27	not used		
45406	0-15	28	0-7	AVI 1270	input data ch 2
		29	0-7		
45407	0-7	30	0-7	AVI 1270	input status ch 2
		31	not used		
45408	0-7	32	0-7	AVI 1270	output status ch 1
		33	not used		
45409	0-7	34	0-7	AVI 1270	output status ch 2
		35	not used		

Output Data (Write):

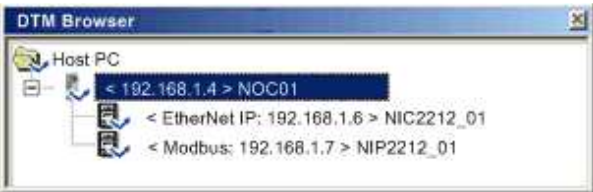

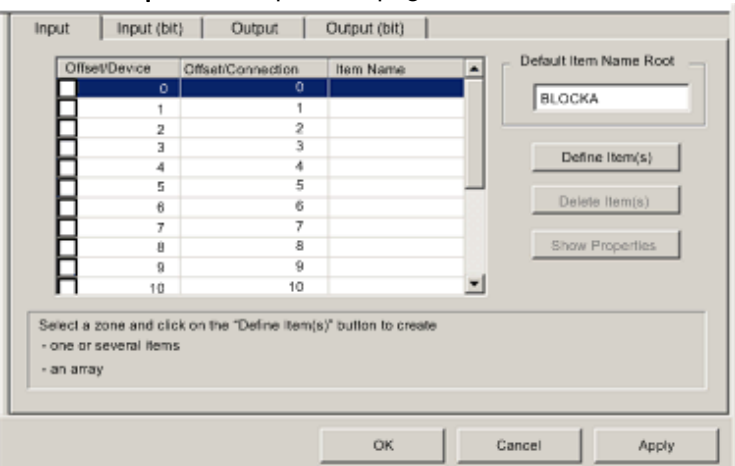
Advantys Modbus Image		Unity Pro Items		STB Module	Description
Register	Bit(s)	Byte	Bit(s)		
40001	0-1	0	0-1	DDO 3200	output data
		1	not used		
40002	0-3	2	0-3	DDO 3410	output data
		3	not used		
40003	0-5	4	0-5	DDO 3600	output data
		5	not used		
40004	0-15	6	0-7	AVO 1250	output data ch 1
		7	0-7		
40005	0-15	8	0-7	AVO 1250	output data ch 2
		9	0-7		

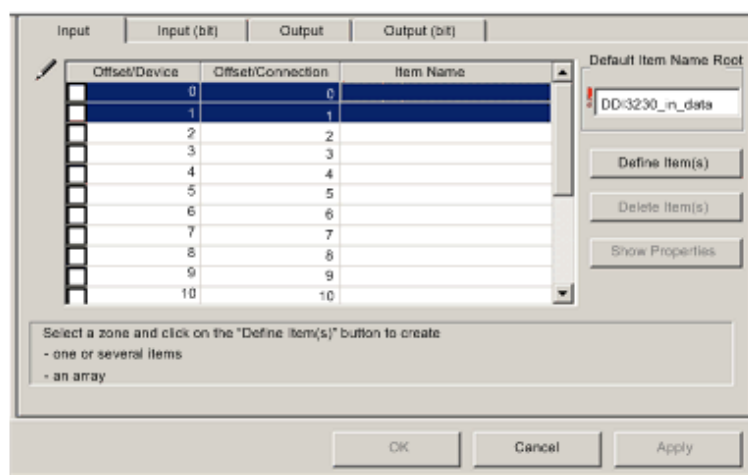
This example shows you how to create 18 words of inputs and 5 words of outputs. This example creates items in the following sequence:

- input word items
- output word items

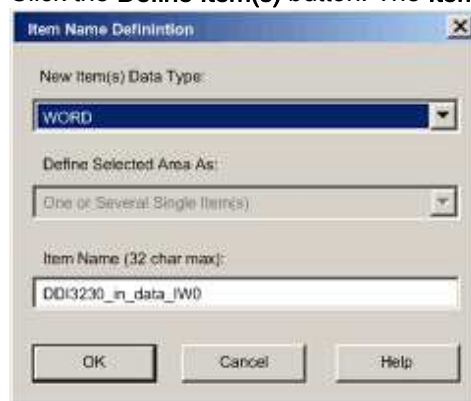
Creating Input Word Items

To create input items for the STB NIP 2212 example, beginning with an input word for the DDI 3230 input module:

Step	Action
1	<p>In the DTM Browser, select the communication module:</p> 
2	<p>Do one of the following:</p> <ul style="list-style-type: none"> • in the main menu, select Edit → Open, or • click the right mouse button, then select Open in the pop-up menu. <p>The Device Editor opens, displaying the DTM for the communication module.</p>
3	<p>In the left pane of the Device Editor, navigate to and select the Items node for the STB NIP 2212 network interface module:</p> 
4	<p>Select the Input tab to open that page:</p>  <p>NOTE: In this example, each row represents a byte. Because the items you create will be a 16-bit word, each item consists of 2 rows.</p>
5	<p>In the Default Item Name Root input box type: DDI3230_in_data.</p>
6	<p>Starting at the beginning of the table, select the first two rows: 0 and 1:</p>

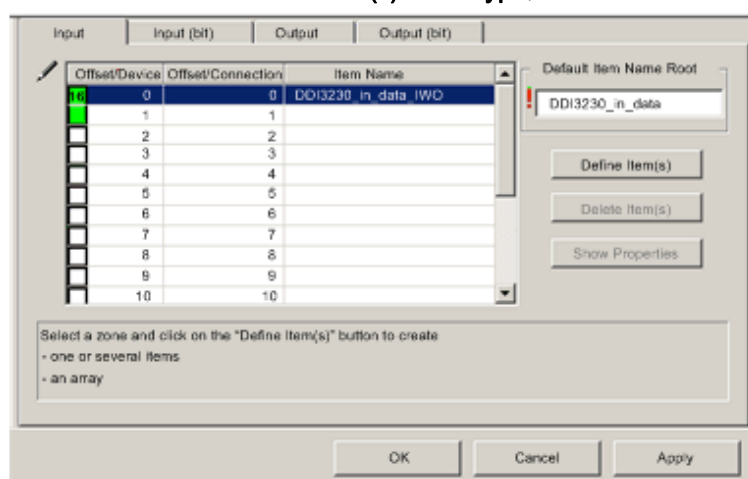


7 Click the **Define Item(s)** button. The **Item Name Definition** dialog opens:



NOTE: The **Define Item(s)** button is enabled only when **Input Mode** is set to **Manual**.

8 Select **WORD** as the **New Item(s) Data Type**, then click **OK**. A new item is created:



9 Click **Apply** to save the new items, and leave the page open.

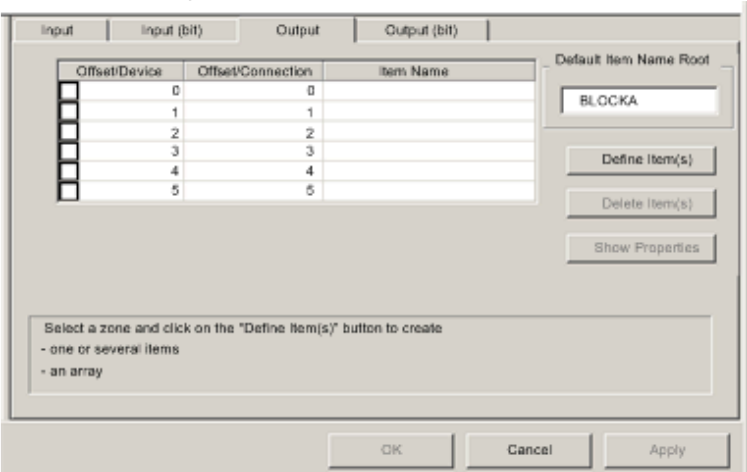
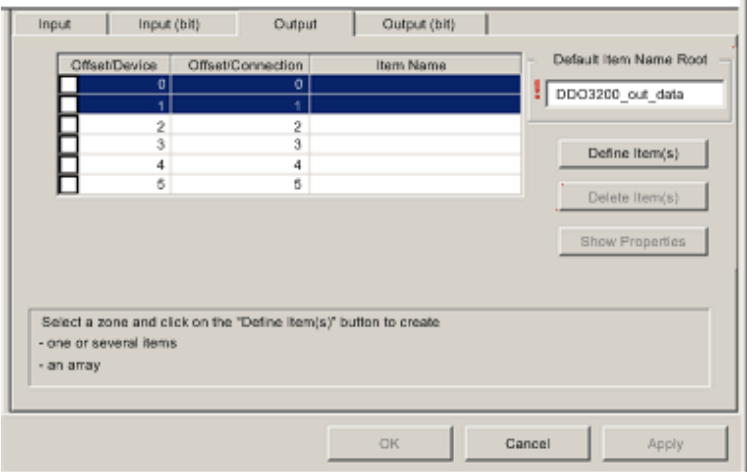
10 Repeat steps 2 - 6 for each new word item you need to create. In this example, that includes the following items:

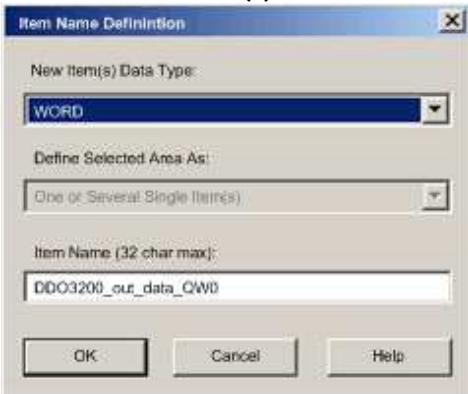
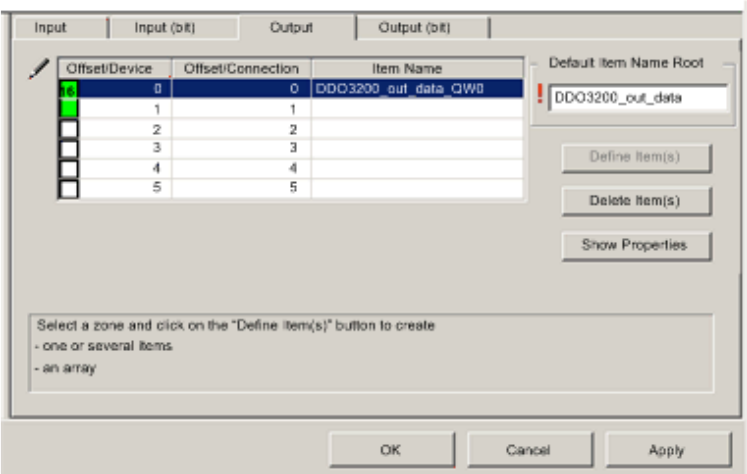
- Rows 2-3, Default Items Name Root: DDI3230_in_st
- Rows 4-5: Default Items Name Root: DDO3200_out_echo
- Rows 6-7: Default Items Name Root: DDO3200_out_st
- Rows 8-9: Default Items Name Root: DDI3420_in_data
- Rows 10-11: Default Items Name Root: DDI3420_in_st
- Rows 12-13: Default Items Name Root: DDO3410_out_echo

	<ul style="list-style-type: none"> • Rows 14-15: Default Items Name Root: DDO3410_out_st • Rows 16-17: Default Items Name Root: DDI3610_in_data • Rows 18-19: Default Items Name Root: DDI3610_in_st • Rows 20-21: Default Items Name Root: DDO3600_out_echo • Rows 22-23: Default Items Name Root: DDO3600_out_st • Rows 24-25: Default Items Name Root: AVI1270_CH1_in_data • Rows 26-27: Default Items Name Root: AVI1270_CH1_in_st • Rows 28-29: Default Items Name Root: AVI1270_CH2_in_data • Rows 30-31: Default Items Name Root: AVI1270_CH2_in_st • Rows 32-33: Default Items Name Root: AVO1250_CH1_out_st • Rows 34-35: Default Items Name Root: AVO1250_CH2_out_st
11	The next task is to create output words.

Creating Output Word Items

To create output items for the STB NIP 2212, example, beginning with an output data word for the DDO 3200 output module:

Step	Action
1	<p>Click on the Output tab to open the following page:</p>  <p>NOTE: In this example, each row represents a byte. Because the items you create will be a 16-bit word, each item consists of 2 rows.</p>
2	In the Default Item Name Root input box type: DDO3200_out_data .
3	<p>Starting at the beginning of the table, select the first 2 rows, 0 and 1:</p> 

<p>4</p>	<p>Click the Define Item(s) button. The Item Name Definition dialog opens:</p>  <p>NOTE: The Define Item(s) button is enabled only when Input Mode is set to Manual.</p>
<p>5</p>	<p>Select WORD as the New Item(s) Data Type, then click OK. A new item is created:</p> 
<p>6</p>	<p>Click Apply to save the new item and leave the page open.</p>
<p>7</p>	<p>Repeat steps 2 - 6 for each new word item you need to create. In this example, that includes the following items:</p> <ul style="list-style-type: none"> • Rows 2-3, Default Items Name Root: DDO3410_out_data • Rows 4-5: Default Items Name Root: DDO3600_out_data • Rows 6-7: Default Items Name Root: AVO1250_CH1_out_data • Rows 8-9: Default Items Name Root: AVO1250_CH2_out_data
<p>8</p>	<p>Click OK to close the Items window.</p>
<p>9</p>	<p>Select File → Save to save your edits.</p>

The next task is to [update the Unity Pro application](#).