Advantys STB

EtherNet/IP Network Interface Applications Guide

10/2019

31008204.04

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

You agree not to reproduce, other than for your own personal, noncommercial use, all or part of this document on any medium whatsoever without permission of Schneider Electric, given in writing. You also agree not to establish any hypertext links to this document or its content. Schneider Electric does not grant any right or license for the personal and noncommercial use of the document or its content, except for a non-exclusive license to consult it on an "as is" basis, at your own risk. All other rights are reserved.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2019 Schneider Electric. All rights reserved.

Table of Contents

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

About the Book

At a Glance

Document Scope

This book describes the Advantys STB EtherNet/IP network interface module (NIM), STB NIC 2212, an ODVA-compliant adapter that allows an Advantys STB island to communicate with an EtherNet/IP fieldbus network. The NIM represents the Advantys STB island as a single node on an EtherNet/IP industrial network.

This guide includes the following information about the STB NIC 2212:

- role in an EtherNet/IP network
- role as the network adapter for the Advantys STB island
- external and internal interfaces
- Flash memory and removable memory
- integrated power supply
- auto-configuration
- saving configuration data
- island bus scanner functionality
- data exchange
- diagnostic messages
- specifications

Validity Note

This document is valid for Advantys 4.5 or later.

The technical characteristics of the devices described in the present document also appear online. To access the information online:

Step	Action
1	Go to the Schneider Electric home page *www.schneider-electric.com*.
2	In the **Search** box type the reference of a product or the name of a product range. ● Do not include blank spaces in the reference or product range. ● To get information on grouping similar modules, use asterisks (*).
3	If you entered a reference, go to the **Product Datasheets** search results and click on the reference that interests you. If you entered the name of a product range, go to the **Product Ranges** search results and click on the product range that interests you.
4	If more than one reference appears in the **Products** search results, click on the reference that interests you.
5	Depending on the size of your screen, you may need to scroll down to see the datasheet.
6	To save or print a datasheet as a .pdf file, click **Download XXX product datasheet**.

The characteristics that are presented in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the document and online information, use the online information as your reference.

Related Documents

Title of Documentation	Reference Number
Advantys STB System Planning and Installation Guide	31002947 (English), 31002948 (French), 31002949 (German), 31002950 (Spanish), 31002951 (Italian)
Advantys STB Analog I/O Modules Reference Guide	31007715 (English), 31007716 (French), 31007717 (German), 31007718 (Spanish), 31007719 (Italian)
Advantys STB Digital I/O Modules Reference Guide	31007720 (English), 31007721 (French), 31007722 (German), 31007723 (Spanish), 31007724 (Italian)
Advantys STB Counter Modules Reference Guide	31007725 (English), 31007726 (French), 31007727 (German), 31007728 (Spanish), 31007729 (Italian)
Advantys STB Special Modules Reference Guide	31007730 (English), 31007731 (French), 31007732 (German), 31007733 (Spanish), 31007734 (Italian)
Advantys STB Configuration Software Quick Start User Guide	31002962 (English), 31002963 (French), 31002964 (German), 31002965 (Spanish), 31002966 (Italian)
Advantys STB Reflex Actions Reference Guide	31004635 (English), 31004636 (French), 31004637 (German), 31004638 (Spanish), 31004639 (Italian)

You can download these technical publications and other technical information from our website at https://www.schneider-electric.com/en/download

Product Related Information

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Only persons with the appropriate expertise in control systems should design, program, install, alter, and apply this product.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Chapter 1
Introduction

Introduction

This chapter describes the Advantys STB NIC 2212 EtherNet/IP network interface module (NIM) and its support for the island as an EtherNet/IP network node.

The chapter begins with an introduction of the NIM and a discussion of its role as the network adapter for the Advantys STB island. There is a brief overview of the island itself, followed by a description of the major characteristics of the EtherNet/IP fieldbus protocol.

Some information in this chapter is specific to the STB NIC 2212 and some is common to all Advantys STB NIMs.

What Is in This Chapter?

This chapter contains the following topics:

What Is a Network Interface Module?

Purpose

Every island requires a network interface module (NIM) in the leftmost location of the primary segment. Physically, the NIM is the first (leftmost) module on the island bus. Functionally, it is the gateway to the island bus. That is, all communications to and from the island bus pass through the NIM. The NIM also has an integrated power supply that provides logic power to the island modules.

The Fieldbus Network

An island bus is a node of distributed I/O on an open fieldbus network, and the NIM is the island’s interface to that network. The NIM supports data transfers over the fieldbus network between the island and the fieldbus master.

The physical design of the NIM makes it compatible with both an Advantys STB island and your specific fieldbus master. Whereas the fieldbus connector on each NIM type may differ, the location on the module front panel is essentially the same.

Communications Roles

Communications capabilities provided on a standard NIM include:

Function	Role
data exchange	The NIM manages the exchange of input and output data between the island and the fieldbus master. Input data, stored in native island bus format, is converted to a fieldbus-specific format that can be read by the fieldbus master. Output data written to the NIM by the master is sent across the island bus to update the output modules and is automatically reformatted.
configuration services	Custom services can be performed by the Advantys configuration software. These services include changing the operating parameters of the I/O modules, fine-tuning island bus performance, and configuring reflex actions. The Advantys Configuration Software runs on a computer attached to the NIM’s CFG interface *(see page 33)*. (For NIMs with Ethernet port connectivity, you can also connect to the Ethernet port.)
human-machine interface (HMI) operations	A serial Modbus HMI panel can be configured as an input and/or output device on the island. As an input device, it can write data that can be received by the fieldbus master; as an output device, it can receive updated data from the fieldbus master. The HMI can also monitor island status, data, and diagnostic information. The HMI panel should be attached to the NIM’s CFG port.

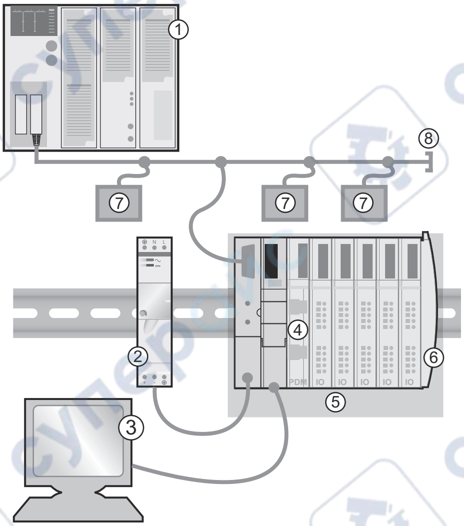
Integrated Power Supply

The NIM's built-in 24-to-5 VDC power supply provides logic power to the I/O modules on the primary segment of the island bus. The power supply requires a 24 VDC external power source. It converts the 24 VDC to 5 V of logic power for the island. Individual STB I/O modules in an island segment generally draw a logic bus current of between 50 and 265 mA. (Consult the *Advantys STB System Planning and Installation Guide* for current limitations at various operating temperatures.) If the logic bus current drawn by the I/O modules totals more than 1.2 A, additional STB power supplies need to be installed to support the load.

The NIM delivers the logic power signal to the primary segment only. Special STB XBE 1300 beginning-of-segment (BOS) modules, located in the first slot of each extension segment, have their own built-in power supplies, which provide logic power to the STB I/O modules in the extension segments. Each BOS module that you install requires 24 VDC from an external power supply.

Structural Overview

The following figure illustrates the multiple roles of the NIM. The figure provides a network view and a physical representation of the island bus:



1 fieldbus master
2 external 24 VDC power supply, the source for logic power on the island
3 external device connected to the CFG port (a computer running the Advantys Configuration Software or an HMI panel)
4 power distribution module (PDM): provides field power to the I/O modules
5 island node
6 island bus terminator plate
7 other nodes on the fieldbus network
8 fieldbus network terminator (if required)

What Is Advantys STB?

Introduction

Advantys STB is an assembly of distributed I/O, power, and other modules that function together as an island node on an open fieldbus network. Advantys STB delivers a highly modular and versatile slice I/O solution for the manufacturing and process industries.

Advantys STB lets you design an island of distributed I/O where the I/O modules can be installed as close as possible to the mechanical field devices that they control. This integrated concept is known as *mechatronics*.

Island Bus I/O

An Advantys STB island can support as many as 32 I/O modules. These modules may be Advantys STB I/O modules, preferred modules, and enhanced CANopen devices.

The Primary Segment

STB I/O modules on an island may be interconnected in groups called segments.

Every island has at least one segment, called the *primary segment*. It is always the first segment on the island bus. The NIM is the first module in the primary segment. The primary segment has to contain at least one Advantys STB I/O module and can support a logic bus current of up to 1.2 A. The segment also contains one or more power distribution modules (PDMs), which distribute field power to the I/O modules.

Extension Segments

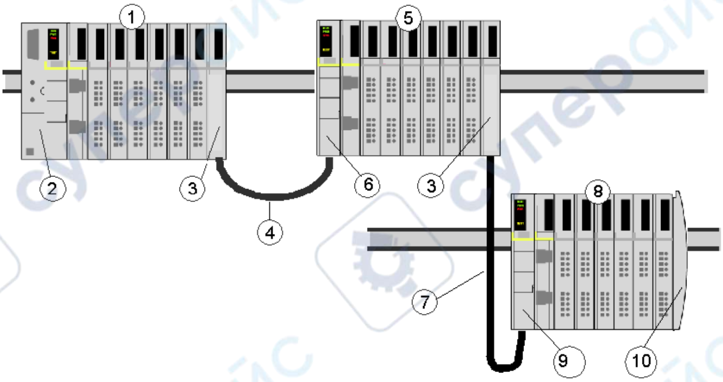
When you are using a standard NIM, Advantys STB I/O modules that do not reside in the primary segment can be installed in *extension segments*. Extension segments are optional segments that enable an island to be a truly distributed I/O system. The island bus can support as many as six extension segments.

Special extension modules and extension cables are used to connect segments in a series. The extension modules are:

- STB XBE 1100 EOS module: the last module in a segment if the island bus is extended
- STB XBE 1300 BOS module: the first module in an extension segment

The BOS module has a built-in 24-to-5 VDC power supply similar to the NIM. The BOS power supply also provides logic power to the STB I/O modules in an extension segment.

Extension modules are connected by lengths of STB XCA 100*x* cable that extend the island communication bus from the previous segment to the next BOS module:



1 primary segment
2 NIM
3 STB XBE 1100 EOS bus extension module(s)
4 1 m length STB XCA 1002 bus extension cable
5 first extension segment
6 STB XBE 1300 BOS bus extension module for the first extension segment
7 4.5 m length STB XCA 1003 bus extension cable
8 second extension segment
9 STB XBE 1300 BOS bus extension module for the second extension segment
10 STB XMP 1100 termination plate

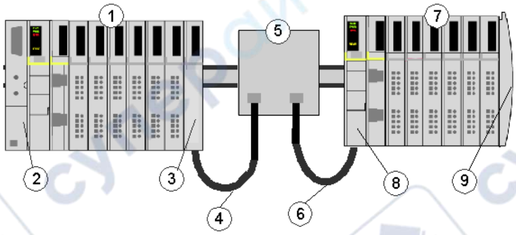
Bus extension cables are available in various lengths, ranging from 0.3 m (1 ft) to 14.0 m (45.9 ft).

Preferred Modules

An island bus can also support those auto-addressable modules referred to as *preferred modules*. Preferred modules do not mount in segments, but they do count as part of the 32-module maximum system limit.

A preferred module can connect to an island bus segment through an STB XBE 1100 EOS module and a length of STB XCA 100*x* bus extension cable. Each preferred module has two IEEE 1394-style cable connectors, one to receive the island bus signals and the other to transmit them to the next module in the series. Preferred modules are also equipped with termination, which has to be enabled if a preferred module is the last device on the island bus and has to be disabled if other modules follow the preferred device on the island bus.

Preferred modules can be chained to one another in a series, or they can connect to Advantys STB segments. As shown in the following figure, a preferred module passes the island bus communications signal from the primary segment to an extension segment of Advantys STB I/O modules:



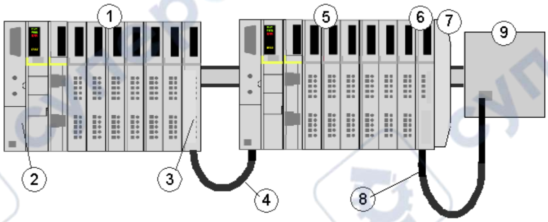
1 primary segment
2 NIM
3 STB XBE 1100 EOS bus extension module
4 1 m length STB XCA 1002 bus extension cable
5 preferred module
6 1 m length STB XCA 1002 bus extension cable
7 extension segment of Advantys STB I/O modules
8 STB XBE 1300 BOS bus extension module for the extension segment
9 STB XMP 1100 termination plate

Enhanced CANopen Devices

You may also install one or more enhanced CANopen devices on an island. These devices are not auto-addressable, and they have to be installed at the end of the island bus. If you want to install enhanced CANopen devices on an island, you need to use an STB XBE 2100 CANopen extension module as the last module in the last segment.

NOTE: If you want to include enhanced CANopen devices in your island, you need to configure the island using the Advantys Configuration Software, and you need to configure the island to operate at 500 kbaud.

Because enhanced CANopen devices cannot be auto-addressed on the island bus, they have to be addressed using physical addressing mechanisms on the devices. The enhanced CANopen devices together with the CANopen extension module form a sub -network on the island bus that needs to be separately terminated at the beginning and end. A terminator resistor is included in the STB XBE 2100 CANopen extension module for one end of the extension sub-network; the last device on the CANopen extension should also be terminated with a 120 Ω resistor. The rest of the island bus needs to be terminated after the CANopen extension module with an STB XMP 1100 termination plate:



1	primary segment
2	NIM
3	STB XBE 1100 EOS bus extension module
4	1 m length STB XCA 1002 bus extension cable
5	extension segment
6	STB XBE 2100 CANopen extension module
7	STB XMP 1100 termination plate
8	typical CANopen cable
9	enhanced CANopen device with 120 Ω termination

Length of the Island Bus

The maximum length of an island bus (the maximum distance between the NIM and the last device on the island) is 15 m (49.2 ft). This length should take into account the extension cables between segments, extension cables between preferred modules, and the space consumed by the devices themselves.

Overview of the STB NIC 2212 Product

Introduction

An Advantys STB island bus assembled with an STB NIC 2212 NIM can function as a slave node on an EtherNet/IP network.

Key Features

Here are the key features of the STB NIC 2212 EtherNet/IP NIM:
- communication with as many as 32 I/O modules
- DIN rail mounting
- IP address assignment (with standard BootP or DHCP tools) or stored IP address
- 496 bytes each of input, output, PLC-to-HMI and HMI-to-PLC data
- configuration through the RS232 serial interface and Ethernet
- input and output data exchanges with EtherNet/IP messaging
- RS232 serial interface HMI connectivity using Modbus messaging
- Ethernet HMI connectivity using EtherNet/IP messaging
- removable memory card for I/O configuration allows copying of the configuration data
- Ethernet communications at 10 or 100 Mb/s communication rate, half- or full-duplex
- Advantys STB island diagnostics information
- auto-configuration through either the RST button or a configuration software command
- HTTP server web pages
- SNMP capability
- FTP server for product updates
- physical diagnostics (LEDs)

Ethernet Connectivity

TCP/IP is the transport layer for the Ethernet LAN on which the STB NIC 2212 Advantys STB island resides. This network architecture enables communications with a wide range of Ethernet TCP/IP control products, such as PLCs, industrial computers, motion controllers, host computers, and operator control stations.

Embedded Web Site

The STB NIC 2212 includes an embedded web site *(see page 85)*, which is a web browser-enabled application. It allows authorized users to view configuration and diagnostic data for the STB NIC 2212. (Users with additional authorization can write data to the STB NIC 2212.)

Ethernet Services

The STB NIC 2212 is configured for the following Ethernet services:
- HTTP embedded web site *(see page 85)*:
  - Port 80 service access point (SAP)
  - browser based IP configuration and troubleshooting
- SNMP—allows remote network management of the STB NIC 2212 through Port 161 SAP.

Introduction to EtherNet/IP Connectivity

Introduction

The STB NIC 2212 allows the Advantys STB island to function as a node on an EtherNet/IP local area network (LAN).

What is EtherNet/IP?

EtherNet/IP (the Ethernet Industrial Protocol) is especially suited to factory applications in which there is an emphasis on control, configuration, and information reporting. The ODVA-specified protocol runs CIP (Common Industrial Protocol) on top of standard Internet protocols, like TCP/IP and UDP. It is an open local (communications) network that enables the interconnectivity of all levels of manufacturing operations from the plant's office to the sensors and actuators on its floor.

NOTE: For more on standard EtherNet/IP specifications and mechanisms, refer to the ODVA home page (http://www.odva.org).

Conformance

The STB NIC 2212 is located on a 10/100 Base-T LAN. This standard is defined by the IEEE 802.3 Ethernet specification. Contention for 10/100 Base-T networks is resolved by using Carrier Sense Multiple Access with Collision Detect (CSMA/CD).

Transmission Rate

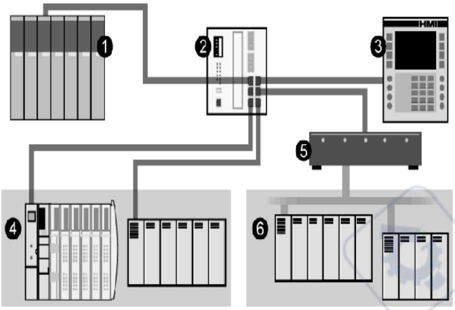
An STB NIC 2212 island node resides on a baseband network with an auto-negotiated transmission rate of 10 or 100 Mbit/s.

Frame Format

The STB NIC 2212 supports both Ethernet II and IEEE 802.3 frame formats. Ethernet II is the default frame type. Refer to the discussion about Frame Format Priorities *(see page 62)*.

Network Topology

This is an example of EtherNet/IP connectivity:



1 PLC or equivalent scanner-class device
2 managed Ethernet switch
3 messaging class device (HMI)
4 adapter class devices (STB NIC 2212, I/O rack)
5 CIP router/bridge
5 CIP devices on any CIP fieldbus

The STB NIC 2212 NIM on the Advantys STB island is an ODVA-compliant adapter class device that supports communication with scanner class devices and EtherNet/IP messaging class devices (HMI, SCADA, etc). Through CIP routers and gateways, the STB NIC 2212 supports those devices even when they operate on other CIP fieldbuses. In addition, devices that support other fieldbus communication (e.g., Modbus TCP) can connect to the same physical medium as the EtherNet/IP NIM.

Object Model

The EtherNet/IP specification is presented in terms of an abstract object model describing device characteristics and the manner in which network connections are established and managed. Each network node is modeled as a collection of objects that describe the node's available communication services and behavior. A device's object model mapping is specific to its implementation on the network.

Device Profiles

The EtherNet/IP device models define the physical connections and promote interoperability among standard devices. Devices that implement the same device model must support common identity and communications status data. Device-specific data is contained in device profiles that are defined for various device types. Typically, a device profile defines the device's:

- object model
- I/O data format
- configurable parameters

The above information is made available to other vendors through the device's EDS. Refer to the section about fieldbus communications support *(see page 63)*.

What's an EDS?

The EDS (electronic data sheet) is a standardized ASCII file that contains information about a network device's communications functionality and the contents of its object dictionary (as defined by ODVA). The EDS also defines device-specific and manufacturer-specific objects.

Your application may or may not require the EDS file. Refer to the connection example chapter *(see page 119)*.

Using the EDS, you can standardize tools to:

- configure EtherNet/IP devices
- design networks for EtherNet/IP devices
- manage project information on different platforms

The parameters of a particular island configuration depend on those objects (parameter, application, communications, emergency, and other objects) that reside on the individual island modules.

Basic and Configured EDS Files

An EDS that describes the island's basic functionality and objects is included with the STB NIC 2212 EtherNet/IP NIM product.

If you wish, you can generate a configuration-specific EDS for your particular island using the (optional) Advantys Configuration Software.

Chapter 2
Physical Description of the STB NIC 2212 NIM Module

Introduction

This chapter describes the Advantys STB EtherNet/IP NIM’s external features, connections, power requirements, and product specifications.

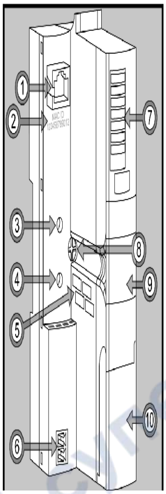
What Is in This Chapter?

This chapter contains the following topics:

External Features of the STB NIC 2212 NIM

Introduction

The physical features of the STB NIC 2212 EtherNet/IP NIM are called out in the illustration below:



The physical features of the NIM are described in the following table:

Feature		Function
1	EtherNet/IP interface	Connect the NIM and the island bus to an EtherNet/IP network through this RJ-45 connector.
2	MAC ID	The MAC ID is a unique 48-bit network ID that is hard-coded in the STB NIC 2212 when manufactured.
3	upper rotary switch	Use the rotary switches to: ● assign the IP address using the BootP or DHCP methods ● assign the IP address using stored or default IP parameters ● clear IP parameters
4	lower rotary switch	
5	space provided to record IP address	Write the IP address that you assign to this STB NIC 2212 here.
6	power supply interface	Connect an external 24 VDC power supply to the NIM through this 2-pin receptacle.
7	LED array	The colored LEDs illuminate in various patterns to visually indicate the operational status of the island.
8	release screw	Turn this screw to remove the NIM from the DIN rail. (See the *Automation Island System Planning and Installation Guide* for details.)
9	removable memory card drawer	Seat a removable memory card in this plastic drawer, then insert it into the NIM.
10	CFG port cover	Lift this hinged flap on the front to access the CFG interface and the RST button.

STB NIC 2212 EtherNet/IP Interface

Introduction

⚠ WARNING

HAZARD OF ELECTRICAL SHOCK OR BURN

Connect the ground wire to the protective earth (PE) terminal before you establish any further connections. When you remove connections, disconnect the ground wire last. The Ethernet cable shield must be connected to the PE ground at the Ethernet switch.

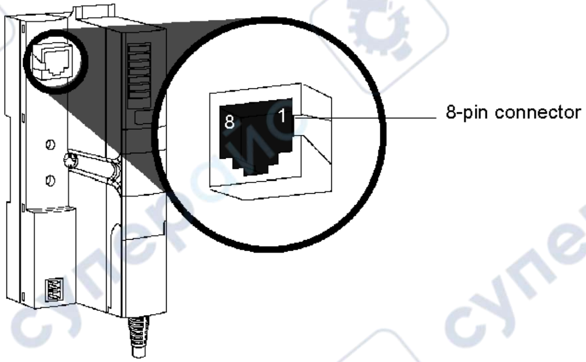
Failure to follow these instructions can result in death, serious injury, or equipment damage.

The fieldbus interface on the STB NIC 2212 NIM is the point of connection between an Advantys STB island and the EtherNet/IP LAN on which the island resides. This fieldbus interface is also called the *EtherNet/IP port*.

The fieldbus interface is a 10/100 Base-T port with an RJ-45 female connector. Category 5 (CAT5) shielded twisted pair (STP) cable connects the STB NIC 2212 to the EtherNet/IP baseband.

Fieldbus Port Connections

The fieldbus interface is located on the front of the EtherNet/IP NIM at the top:



The RJ-45 connector is an 8-pin female connector. The pins connect horizontally along the top. Pin 8 has the leftmost position, and pin 1 is the rightmost. The connector pin-out is described in this table:

Pin	Signal
1	tx+
2	tx-
3	rx+

Pin	Signal
4	reserved
5	reserved
6	rx-
7	reserved
8	reserved

Communications Cable and Connector

The required communications cable is a shielded (STP) twisted pair CAT5 electrical cable with shielded RJ-45 connectors. The cable used with the STB NIC 2212 must terminate with an 8-pin male connector.

The CAT5 cable recommended for connecting the STB NIC 2212 to an Ethernet LAN has the following characteristics:

standard	description	max. length	application	data rate	connector to the fieldbus interface
10Base-T	24-gauge, twisted pair	100 m (328 ft)	data transmission	10 Mbits/s	8-pin male
100Base-T	24-gauge, twisted pair	100 m (328 ft)	data transmission	100 Mbits/s	8-pin male

Notes:
- There are many 8-pin male connectors that are compatible with the RJ-45 fieldbus interface on the STB NIC 2212. Refer to the *Transparent Factory Network Design and Cabling Guide* (490 USE 134 00) for a list of approved connectors.
- For technical specifications of CAT5 cables, refer to FCC Part 68, EIA/TIA-568, TIA TSB-36, and TIA TSB-40.

Per ODVA specifications, the Ethernet connector shield of this device is not connected directly to protective earth.

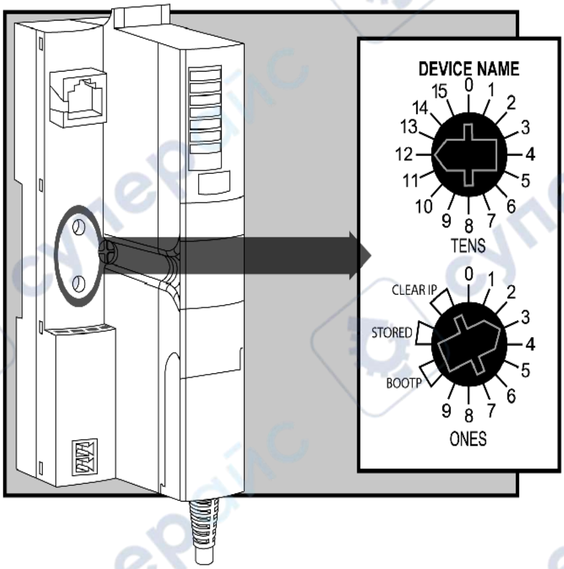
Rotary Switches

Introduction

As the EtherNet/IP adapter for I/O modules on the Advantys STB island, the STB NIC 2212 NIM appears as a single node on the EtherNet/IP network. The NIM must have a unique IP address, easily set with the two rotary switches on the front.

Physical Description

The two rotary switches are on the front of the STB NIC 2212. The upper switch represents the tens digit, and the lower switch represents the ones digit:



Switch Settings for IP Parameter Assignment

The following information summarizes the valid rotary switch settings:

- For a switch-set device name, select a numeric value from 00 to 159. You can use both switches:
  - On the upper switch (tens digit), the available settings are 0 to 15.
  - On the lower switch (ones digit), the available settings are 0 to 9.

  The numeric setting is appended to the STB NIC 2212 part number. For example, an upper switch setting of 12 and a lower switch setting of 3 creates a device name of *STBNIC2212_123*, to which the DHCP server assigns an IP address.
- For a BootP-served IP address, select either of the two BOOTP positions on the bottom switch.
- If you set the bottom switch to either of the two STORED positions, the IP address is assigned through one of the following methods:
  - if the STB NIC 2212 is direct from the factory, it has no software-assigned IP parameters and it uses a MAC-based IP address.
  - a web-configured device name in association with a DHCP server
  - a fixed IP address using the STB NIC 2212 web configuration pages.
- The CLEAR IP settings clear the NIM's internal IP parameters, including the internal device name.(With this selection, the island does not have an IP address.)

NOTE:

- Elsewhere in this guide are detailed descriptions of methods for IP addressing *(see page 58)*.
- For information about how the STB NIC 2212 prioritizes IP addressing options, refer to the IP parameterization flow chart *(see page 61)*.
- The STB NIC 2212 requires a valid IP address to communicate on the EtherNet/IP network and with a host. You must power cycle the STB NIC 2212 to configure the STB NIC 2212 with an IP address set with these rotary switches.

LED Indicators

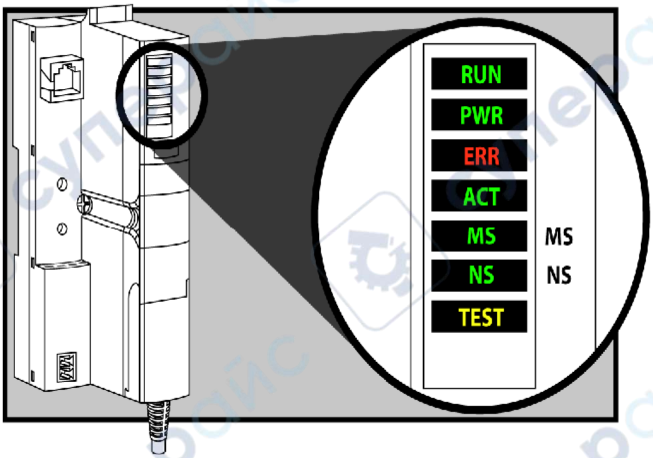
Introduction

The LEDs on the STB NIC 2212 NIM visually indicate the operational status of the island on an Ethernet LAN. The LED array is located on the front of the NIM:

- ACT: This LED indicates whether the Ethernet LAN port are healthy and alive and the transmission speed (10 or 100 Mbps).
- MS: This LED indicates the status of the STB NIC 2212 NIM.
- NS: indicates status of the Ethernet LAN and the EtherNet/IP connectivity.
- RUN/PWR/ERR/TEST: These LEDs indicate the island status or NIM events.

Description

This illustration shows the location and names of the LEDs:



EtherNet/IP Communications LEDs

The ACT (activity), MS (module status), and NS (network status) LEDs indicate these conditions:

Label	Pattern	Meaning
ACT (yellow)	blinking or steady on	10Base-T activity: Packets are being transmitted or received using 10Base-T.
ACT (green)	blinking or steady on	100Base-T activity: Packets are being transmitted or received using 100Base-T.
ACT	off	No activity: There is no Ethernet traffic or no link activity.
MS (green)	steady on	Device operational: The STB NIC 2212 is operating correctly.
	blinking	Standby: The STB NIC 2212 has not been configured.
MS (red)	steady on	Major fault: The STB NIC 2212 has detected a major (non-recoverable) fault.
	blinking	Minor fault: The STB NIC 2212 has detected a minor (recoverable) fault.
MS (green/red)	blinking	Self-test: The STB NIC 2212 is performing its power up self-test.
	steady off	No power: The STB NIC 2212 is not receiving power.
NS (green)	steady on	There is an established EtherNet/IP connection to the STB NIC 2212.
	blinking	The STB NIC 2212 has a valid IP address, but it has no EtherNet/IP connections.
NS (red)	steady on	Duplicate IP: The STB NIC 2212 has detected that its IP address is already in use.
	blinking	Connection timeout: One or more EtherNet/IP connections to the STB NIC 2212 NIM has timed out. Blinking continues until all timed out connections are reestablished or the NIM is reset.
NS (red/green)	blinking	Self-test: The STB NIC 2212 is performing its power up self-test.
	steady off	No IP address or not powered on: The STB NIC 2212 does not have an IP address or is not receiving power.

Advantys STB Island Status LEDs

About the Island Status LEDs

The following table describes:
- the island bus condition(s) communicated by the LEDs
- the colors and blink patterns used to indicate each condition

As you refer to the table, keep in mind the following:
- It is assumed that the *PWR* LED is on continuously, indicating that the NIM is receiving adequate power. If the *PWR* LED is off, logic power *(see page 36)* to the NIM is off or insufficient.
- Individual blinks are approximately 200 ms. There is a 1-second interval between blink sequences. Please note:
  - blinking: blinks steadily, alternating between 200 ms on and 200 ms off.
  - blink 1: blinks once (200 ms), then 1 second off.
  - blink 2: blinks twice (200 ms on, 200 ms off, 200 ms on), then 1 second off.
  - blink *N*: blinks N (some number of) times, then 1 second off.
  - If the *TEST* LED is on, either the Advantys configuration software or an HMI panel is the master of the island bus. If the *TEST* LED is off, the fieldbus master has control of the island bus.

Island Status LED Indicators

RUN (green)	ERR (red)	TEST (yellow)	Meaning
blink: 2	blink: 2	blink: 2	The island is powering up (self test in progress).
off	off	off	The island is initializing. The island is not started.
blink: 1	off	off	The island has been put in the pre-operational state by the RST button. The island is not started.
		blink: 3	The NIM is reading from the removable memory card *(see page 50)*.
		on	The NIM is overwriting its Flash memory with the card's configuration data. (See note 1.)
off	blink: 8	off	The contents of the removable memory card are invalid.
blinking (steady)	off	off	The NIM is configuring *(see page 43)* or auto-configuring *(see page 46)* the island bus. The island bus is not started.
blinking	off	on	Auto-configuration data is being written to Flash memory. (See note 1.)
blink: 3	blink: 2	off	Configuration mismatch detected after power up. At least one mandatory module does not match. The island bus is not started.

RUN (green)	ERR (red)	TEST (yellow)	Meaning
off	blink: 2	off	The NIM has a detected error in module assignment; the island bus is not started.
	blink: 5		invalid internal triggering protocol
off	blink: 6	off	The NIM detects no I/O modules on the island bus.
	blinking (steady)	off	The NIM detects no I/O modules on the island bus ... or ...
			No further communications with the NIM are possible. Probable causes: ● internal condition ● wrong module ID ● device did not auto-address *(see page 44)* ● mandatory module is incorrectly configured *(see page 153)* ● process image is not valid ● device is incorrectly configured *(see page 46)* ● The NIM has detected an anomaly on the island bus. ● receive/transmit queue software overrun
on	off	off	The island bus is operational.
on	blink 3	off	At least one standard module does not match. The island bus is operational with a configuration mismatch.
on	blink: 2	off	There is a serious configuration mismatch (when a module is pulled from a running island). The island bus is now in pre-operational mode because of one or more mismatched mandatory modules.
blink: 4	off	off	The island bus is stopped (when a module is pulled from a running island). No further communications with the island are possible.
off	on	off	Internal condition: The NIM is inoperable.
[any]	[any]	on	Test mode is enabled: The configuration software or an HMI panel can set outputs. (See note 2.)

1 The TEST LED is on temporarily during the Flash overwrite process.

2 The TEST LED is on steadily while the device connected to the CFG port is in control.

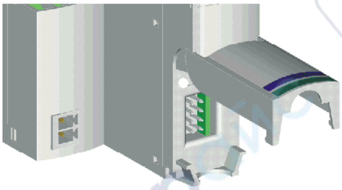
The CFG Interface

Purpose

The CFG port is the connection point to the island bus for either a computer running the Advantys Configuration Software or an HMI panel.

Physical Description

The CFG interface is a front-accessible RS-232 interface located behind a hinged flap on the bottom front of the NIM:



The port uses an 8-pin HE-13 (male) connector.

Port Parameters

The CFG port supports the set of communication parameters listed in the following table. If you want to apply any settings other than the factory default values, use the Advantys Configuration Software:

Parameter	Valid Values	Factory Default Settings
bit rate (baud)	2400/4800/9600/19200/ 38400/ 57600	9600
data bits	7/8	8
stop bits	1 or 2	1
parity	none / odd / even	even
Modbus communications mode	RTU/ASCII	RTU

Always check the data bits. The correct value is “7/8.” (The factory default is “8:”)

NOTE: To restore all of the CFG port’s communication parameters to their factory default settings, push the RST button *(see page 53)* on the NIM. Be aware, however, that this action overwrites all of the island’s current configuration values with factory default values.

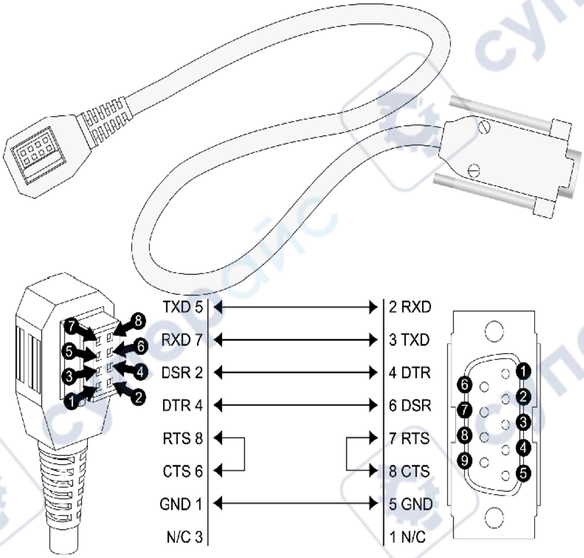
To retain your configuration and still use the RST button to reset your port parameters, write the configuration to an STB XMP 4440 removable memory card *(see page 47)* and insert the card in its drawer in the NIM.

You can also password-protect a configuration *(see page 163)*. If you do this, however, the RST button is disabled and you are unable to use it to reset the port parameters.

Connections

Use an STB XCA 4002 programming cable to connect the computer running the Advantys Configuration Software or a Modbus-capable HMI panel to the NIM through the CFG port.

The STB XCA 4002 is a 2 m (6.23 ft) shielded, twisted-pair cable with an 8-receptacle HE-13 (female) connector on one end that plugs into the CFG port and a 9-receptacle SUB-D (female) connector on the other end that plugs into a computer or an HMI panel:



TXD transmit data
RXD receive data
DSR data set ready
DTR data terminal ready
RTS request to send
CTS clear to send
GND ground reference
N/C not connected

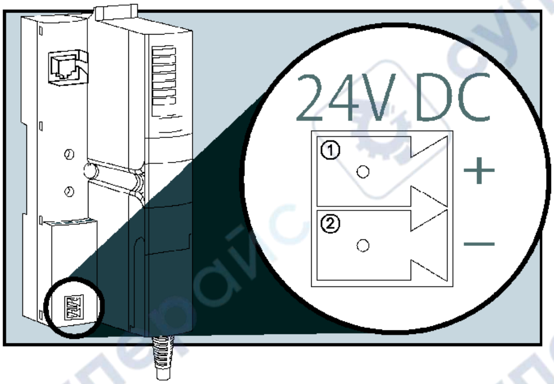
The following table describes the specifications for the programming cable:

Parameter	Description
model	STB XCA 4002
function	connection to a device running the Advantys Configuration Software
	connection to an HMI panel
communications protocol	Modbus, either RTU or ASCII mode
cable length	2 m (6.23 ft)
cable connectors	● 8-receptacle HE-13 (female) ● 9-receptacle SUB-D (female)
cable type	multiconductor

The Power Supply Interface

Physical Description

The NIM's built-in power supply requires 24 VDC from an external SELV-rated power source. The connection between the 24 VDC source and the island is the male 2-pin connector illustrated below:

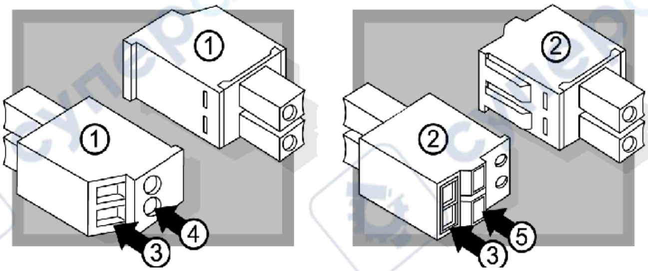


1 connector 1–24 VDC
2 connector 2–common

Connectors

Screw-type and spring-type connectors are provided with the NIM. Replacement connectors are also available.

The following illustrations show two views of each power connector type. A front and back view of the STB XTS 1120 screw type connector is shown on the left, and a front and back view of the STB XTS 2120 spring clamp connector is shown on the right:



1 STB XTS 1120 screw-type power connector
2 STB XTS 2120 spring clamp power connector
3 wire entry slot
4 screw clamp access
5 spring clamp actuation button

Each entry slot accepts a wire in the range 0.14 to 1.5 mm^2 (28 to 16 AWG).

Logic Power

Introduction

Logic power is a 5 VDC power signal on the island bus that the I/O modules require for internal processing. The NIM has a built-in power supply that provides logic power. The NIM sends the 5 V logic power signal across the island bus to support the modules in the primary segment.

External Source Power

The power components are not galvanically isolated. They are intended for use only in systems designed to provide SELV isolation between the supply inputs or outputs and the load devices or system power bus. Use SELV-rated supplies to provide 24 VDC source power to the NIM.

NOTICE

EQUIPMENT DAMAGE

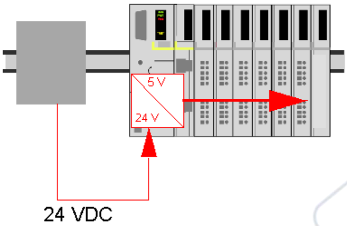
Use only power supplies designed to provide SELV isolation between the supply inputs, outputs, load devices and system power bus.

Failure to follow these instructions can result in equipment damage.

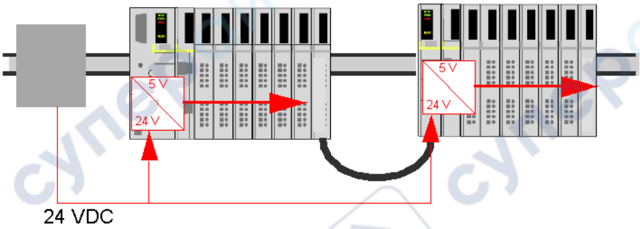
Input from an external 24 VDC power supply *(see page 38)* is needed as the source power for the NIM’s built-in power supply. The NIM’s built-in power supply converts the incoming 24 V to 5 V of logic power. The external supply should be rated *safety extra low voltage* (SELV-rated).

Logic Power Flow

The figure below shows how the NIM's integrated power supply generates logic power and sends it across the primary segment:



The figure below shows how the 24 VDC signal is distributed to an extension segment across the island:



The logic power signal is terminated in the STB XBE 1100 module at the end of the segment (EOS).

Island Bus Loads

The built-in power supply provides logic bus current to the island. If the logic bus current drawn by the I/O modules exceeds the available current, install additional STB power supplies to support the load. Consult the *Advantys STB System Planning and Installation Guide* for the current provided and consumed by Advantys STB modules at various operating temperatures and voltages.

Selecting a Source Power Supply for the Island’s Logic Power Bus

Logic Power Requirements

An external 24 VDC power supply is needed as the source for logic power to the island bus. The external power supply connects to the island’s NIM. This external supply provides the 24 V input to the built-in 5 V power supply in the NIM.

The NIM delivers the logic power signal to the primary segment only. Special STB XBE 1300 beginning-of-segment (BOS) modules, located in the first slot of each extension segment, have their own built-in power supplies, which provide logic power to the STB I/O modules in the extension segments. Each BOS module that you install requires 24 VDC from an external power supply.

Characteristics of the External Power Supply

The power components are not galvanically isolated. They are intended for use only in systems designed to provide SELV isolation between the supply inputs or outputs and the load devices or system power bus. Use SELV-rated supplies to provide 24 VDC source power to the NIM.

NOTICE

EQUIPMENT DAMAGE

Use only power supplies designed to provide SELV isolation between the� supply inputs, outputs, load devices and system power bus.

Failure to follow these instructions can result in equipment damage.

The external power supply needs to deliver 24 VDC source power to the island. The supply that you select can have a low range limit of 19.2 VDC and a high range limit of 30 VDC. The external supply should be rated *safety extra low voltage* (SELV-rated).

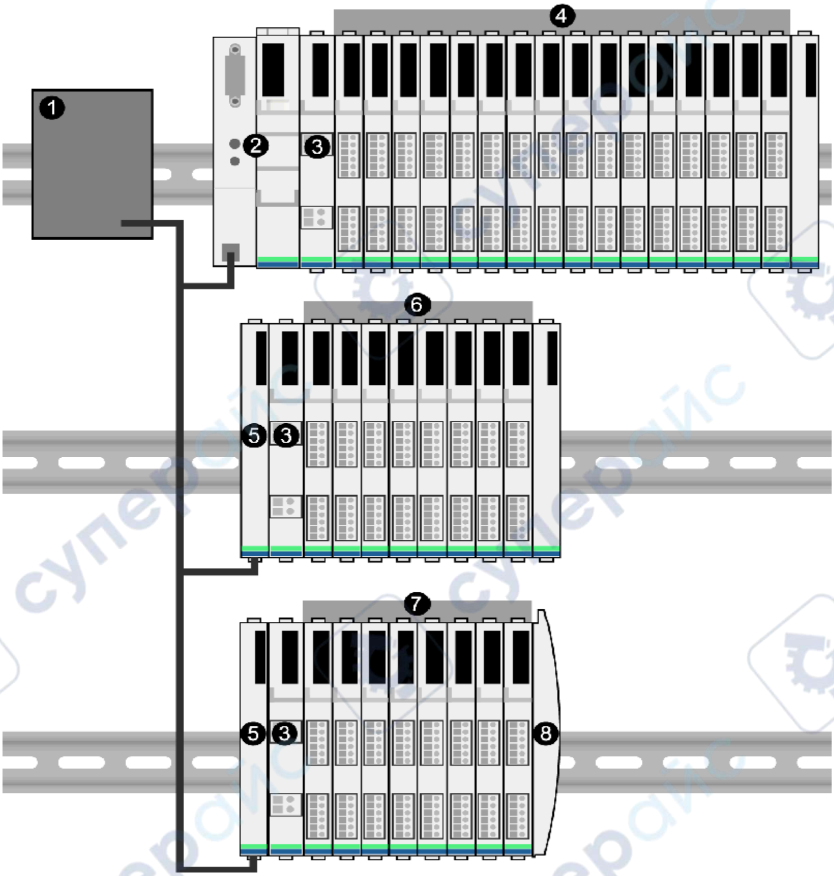
The SELV rating means that, in addition to basic insulation between hazardous voltages and the DC output, a second supplementary insulation layer has been added. As a result, if a single component/insulation does not perform, the DC output does not exceed SELV limits.

Calculating the Wattage Requirement

The amount of power *(see page 37)* that the external power supply should deliver is a function of the number of modules and the number of built-in power supplies installed on the island.

The external supply needs to provide 13 W of power for the NIM and 13 W for each additional STB power supply (like an STB XBE 1300 BOS module). For example, a system with one NIM in the primary segment and one BOS module in an extension segment would require 26 W of power.

Here is an example of an extended island:



1 24 VDC source power supply
2 NIM
3 PDM
4 primary segment I/O modules
5 BOS module
6 first extension segment I/O modules
7 second extension segment I/O modules
8 island bus terminator plate

The extended island bus contains three built-in power supplies:
- the supply built into the NIM, which resides in the leftmost location of the primary segment
- a power supply built into each of the STB XBE 1300 BOS extension modules, which reside in the leftmost location of the two extension segments

In the figure, the external supply would provide 13 W of power for the NIM plus 13 W for each of the two BOS modules in the extension segments (for a total of 39 W).

NOTE: If the 24 VDC source power supply also supplies field voltage to a power distribution module (PDM), you should add the field load to your wattage calculation. For 24 VDC loads, the calculation is simply *amps* x *volts* = *watts*.

Suggested Devices

The external power supply is generally enclosed in the same cabinet as the island. Usually the external power supply is a DIN rail-mountable unit.

We recommend using ABL8 Phaseo power supplies.

Module Specifications

Specifications Detail

The general specifications for the STB NIC 2212 (the EtherNet/IP network adapter for an Advantys STB island bus) appear in the following table:

General Specifications		
dimensions	width	40.5 mm (1.594 in)
	height	130 mm (4.941 in)
	depth	70 mm (2.756 in)
interface and connectors	to the Ethernet LAN	RJ-45 female connector
		CAT5 STP/UTP twisted-pair, electrical cable(s)
	RS-232 *(see page 33)* port for device running the Advantys Configuration Software or an HMI panel *(see page 169)*	8-pin connector HE-13
	to the external 24 VDC power supply	2-pin connector *(see page 35)*
built-in power supply	input voltage	24 VDC nominal
	input power range	19.2 ... 30 VDC
	internal current supply	430 mA@ 24 VDC, consumptive **NOTE:** To provide adequate inrush current capability a minimal rating of 700 mA is recommended for the 24 VDC power supply
	output voltage to the island bus	5 VDC nominal
	output current rating	1.2 A @ 5 VDC
	isolation	no internal isolation **NOTE:** Isolation must be provided by an external 24 VDC source power supply, which must be SELV-rated.
addressable modules supported	per island	32 maximum
segments supported	primary (required)	one
	extension (optional)	6 maximum
standards	Ethernet conformance	IEEE 802.3
	HTTP	Port 80 SAP
	SNMP	Port 161 SAP
	EtherNet/IP	Refer to the specifications *Common Industrial Protocol (CIP) Specification, Edition 3.3* and *EtherNet/IP Adaptation of CIP, Edition 1.4.*
	electromagnetic compatibility (EMC)	EN 61131-2
	MTBF	200,000 hours GB (ground benign)
storage temperature		-40 to 85°C
operating temperature range*		0 to 60°C
number of CIP connections		32
agency certifications		refer to the *Advantys STB System Planning and Installation Guide* (890 USE 171)
*This product supports operation at normal and extended temperature ranges. Refer to the *Advantys STB System Planning and Installation Guide* (890 USE 171) for a complete summary of capabilities and limitations.		

Chapter 3

How to Configure the Island

Introduction

The information in this chapter describes the auto-addressing and auto-configuration processes. An Advantys STB system has an auto-configuration capability in which the actual configuration of I/O modules on the island is read and saved to Flash.

The removable memory card is discussed in this chapter. The card is an Advantys STB option for storing configuration data offline. Factory default settings can be restored to the island bus I/O modules and the CFG port by engaging the RST button.

The NIM is the physical and logical location of all island bus configuration data and functionality.

What Is in This Chapter?

This chapter contains the following topics:

How Do Modules Automatically Get Island Bus Addresses?

Introduction

Each time that the island is powered up or reset, the NIM automatically assigns a unique island bus address to each module on the island that engages in data exchange. All Advantys STB I/O modules and preferred devices engage in data exchange and require island bus addresses.

About the Island Bus Address

An island bus address is a unique integer value in the range 1 through 127 that identifies the physical location of each addressable module on the island. The NIM's address is always 127. Addresses 1 through 32 are available for I/O modules and other island devices.

During initialization, the NIM detects the order in which modules are installed and addresses them sequentially from left to right, starting with the first addressable module after the NIM. No user action is required to address these modules.

Addressable Modules

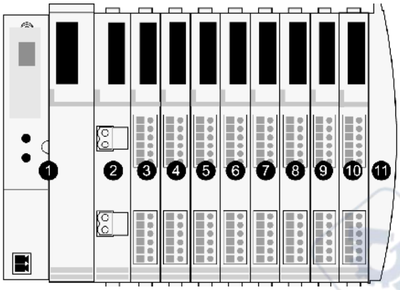
Advantys STB I/O modules and preferred devices are auto-addressable. Enhanced CANopen modules are not auto-addressable. They require manual address settings.

Because they do not exchange data on the island bus, the following are not addressed:

- bus extension modules
- PDMs such as the STB PDT 3100 and STB PDT 2100
- auxiliary power supplies, such as the STB CPS 2111
- termination plate

An Example

For example, if you have an island bus with eight I/O modules:



1 NIM
2 STB PDT 3100 (24 VDC power distribution module)
3 STB DDI 3230 24 VDC (2-channel digital input module)
4 STB DDO 3200 24 VDC (2-channel digital output module)
5 STB DDI 3420 24 VDC (4-channel digital input module)
6 STB DDO 3410 24 VDC (4-channel digital output module)
7 STB DDI 3610 24 VDC (6-channel digital input module)
8 STB DDO 3600 24 VDC (6-channel digital output module)
9 STB AVI 1270 +/-10 VDC (2-channel analog input module)
10 STB AVO 1250 +/-10 VDC (2-channel analog output module)
11 STB XMP 1100 (island bus termination plate)

The NIM would auto-address it as follows. Note that the PDM and the termination plate do not consume island bus addresses:

Module	Physical Location	Island Bus Address
NIM	1	127
STB PDT 3100 PDM	2	not addressed: does not exchange data
STB DDI 3230 input	3	1
STB DDO 3200 output	4	2
STB DDI 3420 input	5	3
STB DDO 3410 output	6	4
STB DDI 3610 input	7	5
STB DDO 3600 output	8	6
STB AVI 1270 input	9	7
STB AVO 1250 output	10	8
STB XMP 1100 termination plate	11	not applicable

Associating the Module Type with the Island Bus Location

As a result of the configuration process, the NIM automatically identifies physical locations on the island bus with specific I/O module types. This feature enables you to hot swap a non-operational module with a new module of the same type.

How to Auto-Configure Default Parameters for Island Modules

Introduction

All Advantys STB I/O modules are shipped with a set of predefined parameters that allow an island to be operational as soon as it is initialized. This ability of island modules to operate with default parameters is known as auto-configuration. Once an island bus has been installed, assembled, and successfully parameterized and configured for your fieldbus network, you can begin using it as a node on that network.

NOTE: A valid island configuration does not require the intervention of the optional Advantys Configuration Software.

About Auto-Configuration

Auto-configuration occurs under these circumstances:

- The island is powered up with a factory default NIM configuration. (If this NIM is subsequently used to create a new island, auto-configuration does not occur when that new island is powered.)
- You push the RST button *(see page 53)*.
- You force an auto-configuration using the Advantys Configuration Software.

As part of the auto-configuration process, the NIM checks each module and confirms that it has been properly connected to the island bus. The NIM stores the default operating parameters for each module in Flash memory.

Customizing a Configuration

In a custom configuration, you can:

- customize the operating parameters of I/O modules
- create reflex actions *(see page 156)*
- add enhanced CANopen standard devices to the island bus
- customize other island capabilities
- configure communication parameters (STB NIP 2311 only)

How to Install the STB XMP 4440 Optional Removable Memory Card

Introduction

The card's performance can be degraded by dirt or grease on its circuitry. Contamination or damage may create an invalid configuration.

Card handling instructions:

- Handle the card with care.
- Inspect for contamination, physical damage, and scratches before installing the card in the NIM drawer.
- If the card gets dirty, clean it with a soft dry cloth.

NOTICE

UNINTENDED EQUIPMENT OPERATION

Do not contaminate or damage the card physically or scratch the card before installing the card in the NIM drawer.

Failure to follow these instructions can result in equipment damage.

The STB XMP 4440 removable memory card is a 32-kbyte subscriber identification module (SIM) that lets you store *(see page 162)*, distribute, and reuse custom island bus configurations. If the island is in edit mode and a removable memory card containing a valid island bus configuration is inserted in the NIM, the configuration data on the card overwrites the configuration data in Flash memory, and is adopted when the island starts up. When the island is in protected mode, it ignores the presence of a removable memory card.

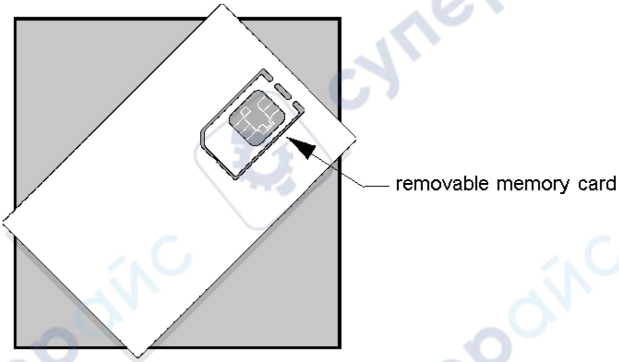
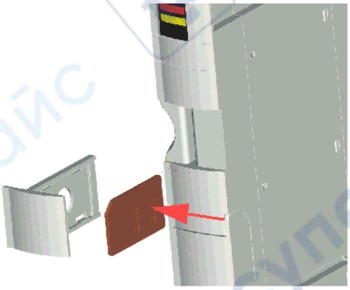
The removable memory card is an optional Advantys STB feature.

Remember:

- Keep the card free of contaminants and dirt.
- Network configuration data, such as the fieldbus baud setting, cannot be saved to the card.

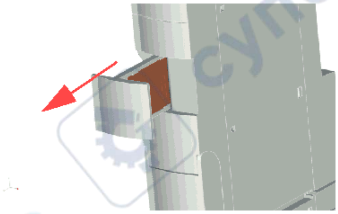
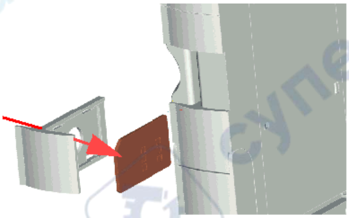
Installing the Card

Use the following procedure to install the memory card:

Step	Action
1	Punch out the removable memory card from the plastic card on which it is shipped.  Check that the edges of the card are smooth after you punch it out.
2	Open the card drawer on the front of the NIM. If it makes it easier for you to work, you may pull the drawer completely out from the NIM housing.
3	Align the chamfered edge (the 45° corner) of the removable memory card with the one in the mounting slot in the card drawer. Hold the card so that the chamfer is in the upper left corner. 
4	Seat the card in the mounting slot, applying slight pressure to the card until it snaps into place. The back edge of the card should be flush with the back of the drawer.
5	Close the drawer.

Removing the Card

Use the following procedure to remove the memory card from the NIM. Avoid touching the circuitry on the card.

Step	Action
1	Open the card drawer. 
2	Push the removable memory card out of the drawer through the round opening at the back. Use a soft but firm object like a pencil eraser. 

Using the STB XMP 4440 Optional Removable Memory Card to Configure the Island

Introduction

A removable memory card is read when an island is powered on or during a reset operation. If the configuration data on the card is valid, the current configuration data in Flash memory is overwritten.

A removable memory card can be *active* only if an island is in *edit* mode. If an island is in protected mode *(see page 163)*, the card and its data are ignored.

Configuration Scenarios

The following discussion describes several island configuration scenarios that use the removable memory card. (The scenarios assume that a removable memory card is already installed in the NIM.):

- initial island bus configuration
- replace the current configuration data in Flash memory in order to:
  - apply custom configuration data to your island
  - temporarily implement an alternative configuration; for example, to replace an island configuration used daily with one used to fulfill a special order
- copying configuration data from one NIM to another, including from a non-operational NIM to its replacement; the NIMs should have the same part number
- configuring multiple islands with the same configuration data

NOTE: Whereas writing configuration data *from* the removable memory card to the NIM does not require use of the optional Advantys Configuration Software, you should use this software to save (write) configuration data *to* the removable memory card in the first place.

Edit Mode

Your island bus should be in edit mode to be configured. In edit mode, the island bus can be written to as well as monitored.

Edit mode is the default operational mode for the Advantys STB island:

- A new island is in edit mode.
- Edit mode is the default mode for a configuration downloaded from the Advantys Configuration Software to the configuration memory area in the NIM.

Additional SIM Features

The removable memory card option in the STB NIP 2311 has an additional feature that allows you to store network parameters. When properly configured, these parameters will be written to flash along with the island parameters on power up.

- Use the configuration software to configure the network communication parameters.
- The communication parameters can be configured only while offline. They take effect after a power cycle of the STB NIP 2311.
- Select the **Enable Editing** check box in the **Ethernet Parameters** tab to enable parameter entries. This check box should remain selected when the configuration is downloaded to the island. If it is deselected before configuration download to the island, these parameters will not be used upon power up.
- Set the **ONES** rotary switch position to **STORED** to use the configured communication parameters.

Initial Configuration and Reconfiguration Scenarios

Use the following procedure to set up an island bus with configuration data that was previously saved *(see page 162)* to a removable memory card. You can use this procedure to configure a new island or to overwrite an existing configuration. (The use of this procedure destroys your existing configuration data.)

Step	Action	Result
1	Install the removable memory card in its drawer in the NIM *(see page 47)*.	
2	Power on the new island bus.	The configuration data on the card is checked. If the data is valid, it is written to Flash memory. The system restarts automatically, and the island is configured with this data. If the configuration data is invalid, it is not used and the island bus stops. If the configuration data was in edit mode, the island bus remains in edit mode. If the configuration data on the card was password-protected *(see page 163)*, your island bus enters protected mode at the end of the configuration process. **NOTE:** If you are using this procedure to reconfigure an island bus and your island is in protected mode, you can use the configuration software to change the island's operational mode to edit.

Using the Card and the RST Function to Reconfigure an Island

You can use a removable memory card in combination with the RST function to overwrite the island's current configuration data. The configuration data on the card can contain custom configuration features. Using the data on the card, you can add password protection to your island, change the I/O module assembly, and change the user-modifiable CFG port settings *(see page 33)*. *Using this procedure destroys your existing configuration data.*

Step	Action	Comment
1	Place the island bus in edit mode.	If your island is in protected mode, you can use the configuration software to change the island's operational mode to *edit*.
2	Press the **RST** button for at least two seconds.	If your configuration data was in edit mode, the island bus remains in edit mode. If the configuration data on the card was protected, your island bus enters protected mode at the end of the configuration process.

Configuring Multiple Island Buses with the Same Data

You can use a removable memory card to make a copy of your configuration data; then use the card to configure multiple island buses. This capability is particularly advantageous in a distributed manufacturing environment or for an OEM (original equipment manufacturer).

NOTE: The island buses may be either new or previously configured, but the NIMs should all have the same part number.

NOTE: If you use the communication parameters feature, moving the removable memory card between islands on the same network will cause duplicate IP addresses. Refer to the LED blink patterns *(see Advantys STB, Standard Dual Port Ethernet Modbus TCP/IP Network Interface Module, Applications Guide)*.

What is the RST Button?

Summary

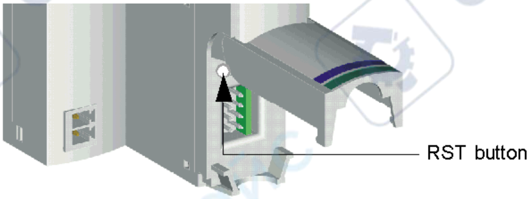
The RST function is basically a Flash memory overwriting operation. This means that RST is functional only after the island has been successfully configured at least once. All RST functionality is performed with the RST button, which is enabled only in edit mode *(see page 50)*.

Physical Description

NOTE: Pushing the RST button reconfigures the island with default settings (no custom parameters).

⚠ WARNING
LOSS OF CONTROL
Do not attempt to restart the island with the RST button.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

The RST button is located immediately above the CFG port *(see page 33)*, and behind the same hinged cover:



Holding down the RST button for 2 seconds or longer causes Flash memory to be overwritten, resulting in a new configuration for the island.

If the island is already auto-configured, there is no consequence other than the island stops during the configuration process. However, island parameters that you previously customized with the Advantys Configuration Software are overwritten by default parameters during the configuration process.

Engaging the RST Button

To engage the RST button, it is recommended that you use a small screwdriver with a flat blade no wider than 2.5 mm (.10 in). Do not use a sharp object that might damage the RST button, nor a soft item like a pencil that might break off and jam the button.

How to Overwrite Flash Memory with the RST Button

Introduction

NOTE: Pushing the RST button *(see page 53)* causes the island bus to reconfigure itself with factory default operating parameters.

⚠ WARNING

LOSS OF CONTROL

Do not attempt to restart the island with the RST button.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The RST function allows you to reconfigure the operating parameters and values of an island by overwriting the current configuration in Flash memory. RST functionality affects the configuration values associated with the I/O modules on the island, the operational mode of the island, and the CFG port parameters.

The RST function is performed by holding down the RST button *(see page 53)* for at least two seconds. The RST button is enabled only in edit mode. In protected mode *(see page 163)*, the RST button is disabled; pressing it has no effect.

NOTE: Pressing the RST button does not affect network settings.

RST Configuration Scenarios

The following scenarios describe some of the ways that you can use the RST function to configure your island:

- Restore factory-default parameters and values to an island, including to the I/O modules and the CFG port *(see page 33)*.
- Add a new I/O module to a previously auto-configured *(see page 46)* island.
  If a new I/O module is added to the island, pressing the RST button forces the auto-configuration process. The updated island configuration data is automatically written to Flash memory.

Overwriting Flash Memory with Factory Default Values

The following procedure describes how to use the RST function to write default configuration data to Flash memory. Follow this procedure if you want to restore default settings to an island. This is also the procedure to use to update the configuration data in Flash memory after you add an I/O module to a previously auto-configured island bus. *Because this procedure overwrites the configuration data, you may want to save your existing island configuration data to a removable memory card before pushing the RST button.*

Step	Action
1	If you have a removable memory card installed, remove it *(see page 49)*.
2	Place the island in edit mode *(see page 50)*.
3	Hold the RST button *(see page 53)* down for at least two seconds.

The Role of the NIM in this Process

The NIM reconfigures the island bus with default parameters as follows:

Stage	Description
1	The NIM auto-addresses *(see page 44)* the I/O modules on the island and derives their factory-default configuration values.
2	The NIM overwrites the current configuration in Flash memory with configuration data that uses the factory-default values for the I/O modules.
3	It resets the communication parameters on its CFG port to their factory-default values *(see page 33)*.
4	It re-initializes the island bus and brings it into operational mode.

Chapter 4
How to Obtain IP Parameters for the STB NIC 2212

About this Chapter

This chapter describes the assignment of IP parameters to the STB NIC 2212 NIM. Each network address must be valid and unique on the Ethernet network.

What Is in This Chapter?

This chapter contains the following topics:

IP Parameter Assignment Methods

Addressing Methods

Set the IP address of the STB NIC 2212 NIM with:

- the rotary switches *(see page 27)*
- the NIM's embedded web site pages *(see page 93)*
- combinations of the above two methods

This table summarizes the addressing methods:

Address Method	Rotary Switch Position	Description
device name	(numeric value)	Use the upper and lower rotary switches to append a device name to the STB NIC 2212 NIM's part number. A DHCP server can then assign it an IP address. **Example**: An upper switch setting of 12 and a lower switch setting of 3 creates a device name of *STBNIC2212_123*, to which the DHCP server assigns an IP address. Also, the NIM's embedded web site provides an alternative method of supplying a device name.
stored	STORED	The lower (ONES) switch is set to one of its STORED positions and the NIM's IP address is assigned by one of the following methods: ● If it's a new module, its MAC-based IP address is applied when it is powered on. ● From a DHCP server with a user-defined device name. ● From the Configure Stored IP web page *(see page 93)*.
BootP server	BOOTP	The lower (ONES) switch is set to one of the BOOTP positions and the module gets IP parameters from a remote BootP server.
clear IP	CLEAR IP	The lower (ONES) switch is set to one of the CLEAR IP positions to clear both the NIM's IP parameters and stored device name from Flash memory and no IP address is assigned. The module then waits for a new IP address assignment (through one of the methods described in this table). Set the switches in accordance with your system and network requirements and repower the module.

How Does the STB NIC 2212 Obtain IP Parameters?

Summary

As a node on a TCP/IP network, the STB NIC 2212 requires a valid 32-bit IP address. The IP address can be:

- assigned by a network server (BootP or DHCP)
- customer-configured using the STB NIC 2212 web pages *(see page 85)*
- the MAC-based default IP address

NOTE: Refer to the IP parameters flow chart *(see page 61)* for information about how the STB NIC 2212 prioritizes IP address assignment options.

Server-Assigned IP Addresses

A server-assigned IP address may be obtained from either a BootP or DHCP server.

Device Name

A device name is a combination of the EtherNet/IP NIM part number STBNIC2212 and a numeric value (e.g., *STBNIC2212_123*).

A device name may be assigned in one of two ways:

- using the numeric settings (00 to 159) on the rotary switches *(see page 27)*
- using a STORED rotary switch position in combination with the NIM’s embedded web site pages.

Stored IP Address

If your STB NIC 2212 does not have a device name, you can configure an IP address on the Configure Stored IP web page *(see page 93)*. Then set the lower rotary switch to a STORED position, and power on the STB NIC 2212.

Deriving an IP Address from a Media Access Control (MAC) Address

The 32-bit default IP address for the STB NIC 2212 is composed of the last four octets of its 48-bit Media Access Control (MAC) address. The MAC address, or Institute of Electrical and Electronics Engineers, Inc. (IEEE) global address is assigned at the factory. The MAC address for an STB NIC 2212 is located on the front bezel under the EtherNet/IP port.

A MAC address is stored in hexadecimal format. Convert the numbers in the MAC address from hexadecimal to decimal notation to derive the *default IP address*. Use the following steps:

Step	Action
1	A MAC address comprises six pairs of hex values, e.g., 00 00 54 10 01 02. Ignore the first two pairs: *00 00*.
2	Identify a pair, e.g., *54*.
3	Multiply the first number, *5* by 16 (5 x 16 = 80).
4	Add the second number, *4* (80 + 4 = 84).

NOTE: There are many resources for converting hex numbers to decimal numbers. We recommend using the Windows calculator in scientific mode.

NOTE: An IP address is derived from the MAC address only when the IP address is not otherwise provided by:

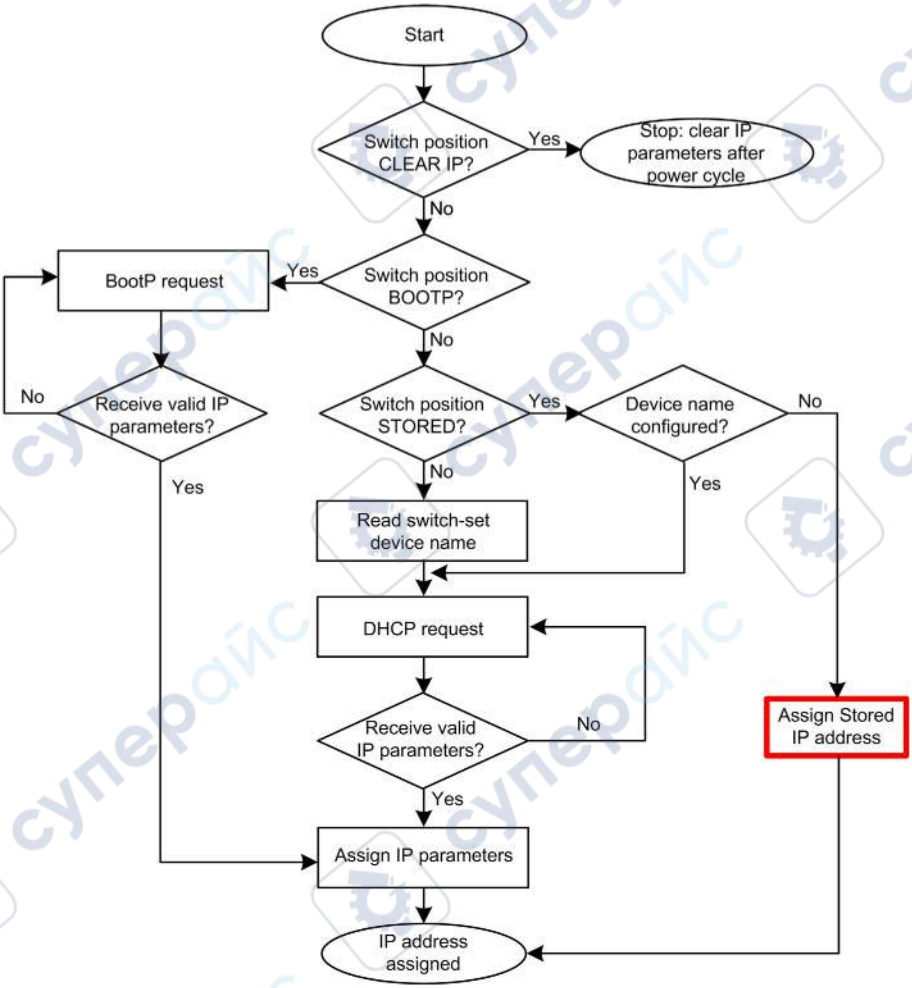
- a BootP server
- a DHCP server
- a user configured IP setting

For more information about how the STB NIC 2212 prioritizes IP address assignment options, refer to the discussions at IP parameters flow chart and fallback IP parameters *(see page 61)*.

How Does the IP Address Assignment Process Work?

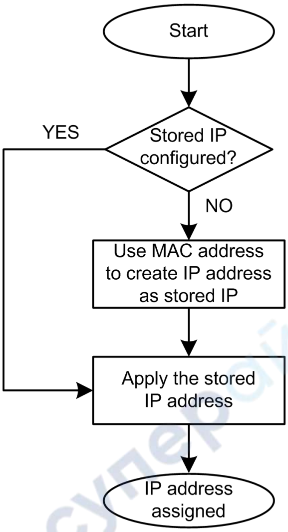
Determining the IP Address

As shown in the following flow chart, the module performs a sequence of checks to determine an IP address:



Assigning a Stored IP Address

If the STB NIC 2212 switch is assigned at STORED position, it will assign itself a stored IP address. The following diagram depicts the **Assigned stored IP address** process, referenced above.



Frame Format Priorities

The module supports communications in the Ethernet II and 802.3 frame formats. Ethernet II is the default.

This table describes the BootP and DHCP behavior of the module when the Auto-select frame format is used:

Server Type	Method
BootP	When communicating with a BootP server, the module makes four Ethernet II frame format requests, followed by four 802.3 frame format requests. If the NIM completes this cycle of requests before it receives IP parameters from the BootP server, it simultaneously: ● assigns fallback IP parameters ● continues to make BootP requests until the BootP server assigns IP parameters
DHCP	When communicating with a DHCP server, the NIM makes four requests using the Ethernet II frame format, then makes four requests using the 802.3 frame format. If the NIM completes making this cycle of requests before it receives an IP address from the DHCP server, it simultaneously: ● assigns a fallback IP address, and ● continues to make DHCP requests until the DHCP server assigns IP parameters

Chapter 5

EtherNet/IP Communications Support

Introduction

This chapter describes how an Advantys STB island node can be accessed from other devices on an EtherNet/IP fieldbus network.

What Is in This Chapter?

This chapter contains the following sections:

Section 5.1
Object Model

Introduction

This section describes the object model for the EtherNet/IP NIM. For general information about the object model for a particular EtherNet/IP device, refer to ODVA specifications.

NOTE: This section includes descriptions of only the most commonly implemented objects (the assembly object and the island bus object). For descriptions of other objects, see Additional Objects in the Object Model *(see page 181)*.

What Is in This Section?

This section contains the following topics:

About the Object Model

Introduction

An EtherNet/IP node is modeled as a collection of objects. Each object provides an abstract representation of a particular component within a product.

An object model defines the device’s:
- I/O data format
- configurable parameters

The above information is made available to other vendors through the device’s EDS.

This chapter describes the STB NIC 2212's implemented objects in terms of:
- supported class attributes
- supported class services
- supported instance attributes
- supported instance services

Further details can be found in Chapter 5 of [28] The CIP Networks Library Volume 2 EtherNet/IP Adaptation of CIP.

Addressing Object Attributes

Objects: Objects provide services and implement behaviors.

Attributes: Attributes (object characteristics) for particular objects are addressed with integer values that correspond to this hierarchy:
- MAC ID (node ID)
- class ID
- instance ID
- attribute ID

Supported Objects

This table lists the EtherNet/IP objects supported by the Advantys STB island:

Object Class	Class ID	Instance ID	Messages	Description
Identity Object *(see page 182)*	1	1	explicit	This object returns the device type, vendor ID, serial number, etc.
Message Router Object *(see page 184)*	2	1	explicit	This object returns information about message router implementation.
Assembly Object *(see page 67)*	4	0x65...0x69 (100...105)	explicit or I/O	This object provides a collection of other object's attributes (frequently used for I/O messaging).
Connection Management Object *(see page 186)*	6	0x01(1)	explicit	This object allows explicit messages to be conducted.
File Object *(see page 188)*	0x37 (55)	0xC8/0xC9 (200/201)	explicit	This object returns the EDS text file and EDS icon file.
Port Object *(see page 190)*	0xF4 (244)	1	explicit	This object returns information about the Ethernet port.
TCP/IP Interface Object *(see page 192)*	0xF5 (245)	1	explicit	This object defines the number of IP address configuration options for the device.
Ethernet Link Object *(see page 194)*	0xF6 (246)	1	explicit	This object tracks configuration and diagnostics information for the Ethernet port.
Island Bus Profile Object *(see page 69)*	0x65 (101)	1	explicit	This object provides diagnostic data and I/O data to/from the EtherNet/IP NIM.

Note: This section includes descriptions of only the most commonly implemented objects (the assembly object and the island bus object). For descriptions of other objects, refer to the appendix Additional Objects in the Object Model *(see page 181)*.

Assembly Object (Class ID 4)

Introduction

The assembly object groups different attributes (data) from a variety of application objects into a single attribute that can be moved with a single message. This message provides the I/O data and status of the Advantys STB EtherNet/IP NIM. Assembly objects can be used to bind input data or output data, as defined from the network's perspective. (That is, an *input* produces data on the network and an *output* consumes data from the network.) For the STB NIC 2212 assembly object:

- The class ID is 4.
- The instance codes are 100, 101, 102, 104, and 105.

Class Attributes (Instance 0)

The assembly object supports these class attributes:

Attribute ID	Name	Access	Description
0x01	Revision	R	This attribute returns the revision of the CIP object (0x02).
0x02	Max Instance	R	This attribute returns the maximum value of the instance number (105).
0x03	Num Instances	R	This attribute returns the number of class instances. The value is between 3 and 5, depending on whether PLC-to-HMI and HMI-to-PLC data are configured.
0x06	Max. Class Attribute	R	This attribute returns the numeric value of the highest class attribute (7).
0x07	Max. Instance Attribute	R	This attribute returns the numeric value of the highest instance attribute (4).

Class Services

The assembly object supports these class services:

Service Code	Name	Description
0x0E	Get Attribute Single	This service returns the value of the specified attribute.

Instance Codes

The Advantys STB EtherNet/IP NIM provides six instances of the assembly object class:

Instance ID	Access	Description
100	R	diagnostic data (see Diagnostic and NIM Status Information *(see page 71)*)
101	R	packed input data (see EtherNet/IP Data Exchange *(see page 79)*)
102	R/W	packed output data (see EtherNet/IP Data Exchange *(see page 79)*)
103	—	reserved
104	R	HMI-to-PLC data (exists only if a data size greater than 0 is configured)
105	R/W	PLC-to-HMI data (exists only if a data size greater than 0 is configured)

Instance Attributes

The assembly object supports these instance attributes:

Attribute ID	Name	Access	Description
1	Number of members	R	This attribute returns a word value of the number of members in the instance.
2	Member List	R	This attribute is an array of structures in which each structure represents one member and consists of: ● *member data size*: a word containing the member data size (in bits) ● *member path size*: a word containing the byte size of the subsequent EPATH: ❍ 0: unused space between members ❍ 0x09: actual members ● *member path*: the EPATH representing the member (For example, "20 04 24 65 30 28 01" is member 1 of instance 101.)
3	Instance data	R/W	This attribute returns instance data as an array of bytes. Access is: ● *read (only)*: input data assemblies ● *read/write*: output data assemblies
4	Instance data size	R	This attribute returns a word representing the instance data size in bytes. (The size depends on the particular I/O modules configured on the bus.)

Instance Services

The assembly object supports these instance services:

Service Code	Name	Description
0x0E	Get Attribute Single	This service returns the value of the specified attribute.
0x010	Set Attribute Single	This service modifies an assembly object instance attribute value.
0x018	Get Member	This service reads a member of an assembly object instance.
0x019	Set Member	This service modifies a member of an assembly object instance.

Island Bus Profile Object (Class ID 0x65)

Introduction

The Island Bus Object is assigned a vendor-specific class ID of 101. The island bus object is an application object that provides the diagnostic data, as well as input and output data from all modules on the island. For the STB NIC 2212 island bus profile object:

- The class code is 0x65 (101).
- The single supported instance is 1.

Class Attributes (Instance 0)

The Island Bus Profile Object supports these class attributes:

Attribute ID	Name	Access	Description
0x01	Revision	R	This attribute returns the revision of the CIP Object (1).
0x02	Max Instance	R	This attribute returns the maximum value of class instances (1).
0x03	Num. Instances	R	This attribute returns the number of class instances (1).
0x06	Max. Class Attribute	R	This attribute returns the value of the highest class attribute (7).
0x07	Max. Instance Attribute	R	This attribute returns value of the highest instance attribute (21).

Class Services

The Island Bus Profile Object supports these class services:

Service Code	Name	Description
0x0E	Get Attribute Single	This service returns the value of the specified instance attribute.
0x01	Get Attribute All	This service returns all class attributes.

Instance Attributes

The Island Bus Object provides explicit access to the following attributes:

Attribute ID	Name	Access	Data Type	Description	Value (from EIP
1	Island Bus State	R	word	communication status	diagnostic data (refer to NIM Status *(see page 77)*)
2	Global Diagnostics	R	word	global diagnostics	
3	Node Configured	R	array of word	indicates configured modules	
4	Node Assembly Fault	R	array of word	indicates incorrectly assembled modules	
5	Node Error	R	array of word	indicates modules with errors	
6	Node Operational	R	array of word	indicates operational modules	
7	Input data size	R	UINT	size of input data in words	unpacked input process image
8	Input data	R	array of word	unpacked input data from island modules	
9	Output data size	R	UINT	size of output data in words	unpacked output process image (including RTP data)
10	Output data	R/W	array of word	unpacked output data to island modules	
13	HMI-to-PLC data size	R	UINT	size of HMI-to-PLC input data in words	HMI-to-PLC input data table (including RTP and virtual analog/digital data)
14	HMI-to-PLC data	R	array of word	HMI-to-PLC input data	
15	PLC-to-HMI data size	R	UINT	size of PLC-to-HMI output data in words	HMI-to-PLC output data table
16	PLC-to-HMI data	R/W	array of word	PLC-to-HMI output data	
21	NIM status	R	word	NIM status word *(see page 77)*	status word

Instance Services

The Island Bus Profile Object supports these instance services:

Service Code	Name	Description
0x0E	Get Attribute Single	This service returns the value of the specified instance attribute.
0x10	Set Attribute Single	This service modifies a (write) instance attribute.

Section 5.2
Diagnostic Data and NIM Status Information

Introduction

The section discusses the diagnostic information that indicates the main states of the Advantys STB island bus and the NIM.

What Is in This Section?

This section contains the following topics:

Diagnostic Data

Introduction

This topic discusses the diagnostic data for the Advantys STB NIC 2212 EtherNet/IP NIM.

Diagnostic Data Structure

The diagnostic data from the Advantys STB system is transmitted through the COS/cyclic I/O connection.

Diagnostic data of the following structure has a fixed length of 20 bytes (10 words):

Diagnostic Information	Data Type	Description
island bus state	word	shows the communication state and diagnostics of the island bus
global diagnostics	word	indicates that the island bus communications are inoperable or that an error has been detected on the network (also reports local island bus errors)
node configured	word array (2)	indicates for each node whether or not an internal error of the device has occurred
node assembly	word array (2)	characterizes every node as deviating from its configured and expected state
node error	word array (2)	characterizes every device that an internal error of the device has occurred and that the internal error is not yet resolved
node operational	word array (2)	characterizes every module station as active or inactive

Island Bus State

The *island bus state* represents the main states of the island bus scanner, the firmware that drives the island bus. This word is composed of a low byte that represents the main communication state and a high byte that contains the actual diagnostics.

Low byte: Each bit in the *island bus state* low byte array indicates a specific event:

Byte Value	Meaning
00h	The island is initializing.
40h	The island bus has been set to pre-operational mode, for example, by the reset function in the Advantys STB configuration software.
60h	*NIM is configuring or auto-configuring*—Communication to all modules is reset.
61h	*NIM is configuring or auto-configuring*—Checking the module ID.
62h	The NIM is auto-addressing the island.
63h	*NIM is configuring or auto-configuring*—Bootup is in progress.
64h	The process image is being set up.

Byte Value	Meaning
80h	Initialization is complete, the island bus is configured, the configuration matches, and the island bus is not started.
81h	*configuration mismatch*: Non-mandatory or unexpected modules in the configuration do not match and the island bus is not started.
82h	*configuration mismatch*: At least one mandatory module does not match and the island bus is not started.
83h	*serious configuration mismatch*: The island bus is set to pre-operational mode and initialization is aborted.
A0h	The configuration matches and the island bus is operating.
A1h	Island is operational with a configuration mismatch. At least one standard module does not match, but all mandatory modules are present and operating.
A2h	*serious configuration mismatch*: The island bus was started but is now in pre-operational mode because of one or more mismatched mandatory module(s).
C0h	Island has been set to pre-operational mode, for example, the stop function in the Advantys STB configuration software.

High byte: Each bit in the *island bus state* high byte array indicates a specific event:

Communication Diagnostic (Bit Number)	Meaning of Value
D8*	1 = low-priority receive queue software overrun
D9*	1 = NIM overrun
D10*	1 = island bus-off error
D11*	1 = error counter in NIM has reached the warning level and the error status bit has been set
D12	1 = NIM error status bit has been reset
D13*	1 = low-priority transfer queue software overrun
D14*	1 = high-priority receive queue software overrun
D15*	1 = high-priority transfer queue software overrun
*the NIM is inoperable	

Island bus state diagnostics can also be accessed through the EtherNet/IP explicit connection by following the path: class 101\instance 1\attribute 1.

Global Diagnostics

Global diagnostics provide the error/status information for internal island bus operations. The *global diagnostics* array is composed of a low byte and a high byte.

Low byte: Each bit in the *global diagnostics* low byte array indicates a specific event:

Bit	Meaning
D0*	No further communications are possible on the island bus.
D1*	module ID error
D2*	Auto-addressing could not be completed.
D3*	Mandatory module configuration error.
D4*	Either the process image configuration is inconsistent or it could not be set during auto-configuration.
D5*	A module has been detected out of order and the NIM can not complete auto-configuration.
D6	Island bus management error detected by the NIM.
D7*	The initialization process in the NIM has detected a module assignment error.
*the NIM is inoperable	

High byte: Each bit in the *global diagnostics* high byte array indicates a specific event:

Bit	Meaning
D8*	internal triggering protocol error
D9*	module data length error
D10*	module configuration error
D11	reserved
D12	timeout error
D13	reserved
D14	reserved
D15	reserved
*the NIM is inoperable	

NOTE: NIM inoperability is caused by internal errors related to either the NIM or a failure in the island configuration software or hardware.

The detection of these errors results in the stopping of the island bus. These are the only ways to get out of this error state:

- Cycle the power.
- Reset the island.
- Clear the error with the Advantys Configuration Software.

The *global diagnostics* can also be accessed through the EtherNet/IP explicit connection by following the path: class 101\instance 1\attribute 2.

Node Configured

Node configured is an array of two words (4 bytes, 32 bits). Each bit represents one specific addressable I/O module on the island bus.

- A value of 1 in a bit position indicates that the corresponding module is configured in the island system.
- A value of 0 indicates that the node is not configured as a slave to the master.

The following table shows the mapping of *node configured* data on EtherNet/IP bytes:

Word	Byte	Bit								Status Data
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	8	7	6	5	4	3	2	1	word offset 0, low byte
	2	16	15	14	13	12	11	10	9	word offset 0, high byte
1	3	24	23	22	21	20	19	18	17	word offset 1, low byte
	4	32	31	30	29	28	27	26	25	word offset 1, high byte

The STB NIC 2212 EtherNet/IP NIM supports a maximum of 32 modules. The two diagnostic words provide the 32 bits that represent the module locations in a typical island configuration. The remaining diagnostic words are available to support island expansion capabilities.

Node configured diagnostics can also be accessed through the EtherNet/IP explicit connection by following the path: class 101\instance 1\attribute 3.

Node Assembly Fault

Node assembly fault is an array of two words (4 bytes, 32 bits). Each bit represents one specific module (node) on the island bus. If the configuration of a module mismatches, the corresponding bit is set:

- A value of 1 in a bit position indicates that the configured module is not present or that the location has not been configured.
- A value of 0 indicates that the correct module is in its configured location.

The following table shows the mapping of *node assembly fault* data on EtherNet/IP bytes:

Word	Byte	Bit								Status Data
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	8	7	6	5	4	3	2	1	word offset 0, low byte
	2	16	15	14	13	12	11	10	9	word offset 0, high byte
1	3	24	23	22	21	20	19	18	17	word offset 1, low byte
	4	32	31	30	29	28	27	26	25	word offset 1, high byte

The STB NIC 2212 EtherNet/IP NIM supports a maximum of 32 modules. The two diagnostic words provide the 32 bits that represent the module locations in a typical island configuration. The remaining diagnostic words are available to support island expansion capabilities.

Node assembly fault diagnostics can also be accessed through the EtherNet/IP explicit connection by following the path: class 101\instance 1\attribute 4.

Node Error

Node error is an array of two words (4 bytes, 32 bits). Each bit represents one specific addressable I/O module on the island bus. After the master receives an error message from a module, the corresponding bit is set:

- A value of 1 in a bit position indicates the presence of the newly received message.
- A value of 0 in a bit position indicates that no values have changed since the last reading of the diagnostic buffer.

The following table shows the mapping of *node error* data on EtherNet/IP bytes:

Word	Byte	Bit								Status Data
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	8	7	6	5	4	3	2	1	word offset 0, low byte
	2	16	15	14	13	12	11	10	9	word offset 0, high byte
1	3	24	23	22	21	20	19	18	17	word offset 1, low byte
	4	32	31	30	29	28	27	26	25	word offset 1, high byte

The STB NIC 2212 EtherNet/IP NIM supports a maximum of 32 modules. The two diagnostic words provide the 32 bits that represent the module locations in a typical island configuration. The remaining diagnostic words are available to support island expansion capabilities.

Node error diagnostics can also be accessed through the EtherNet/IP explicit connection by following the path: class 101\instance 1\attribute 5.

Node Operational

Node operational is an array of two words (4 bytes, 32 bits). Each bit represents one specific addressable I/O module on the island bus.

- A value of 1 in a bit position indicates that the associated module is operating and that no errors were detected.
- A value of 0 in a bit position indicates that the module is not operating because it is not configured or it has an error.

The following table shows the mapping of *node operational* data on EtherNet/IP bytes:

Word	Byte	Bit								Status Data
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	8	7	6	5	4	3	2	1	word offset 0, low byte
	2	16	15	14	13	12	11	10	9	word offset 0, high byte
1	3	24	23	22	21	20	19	18	17	word offset 1, low byte
	4	32	31	30	29	28	27	26	25	word offset 1, high byte

The *node operational* diagnostics can also be accessed through the EtherNet/IP explicit connection by following the path: class 101\instance 1\attribute 6.

NIM Status

Introduction

This topic discusses the status data for the Advantys STB NIC 2212 EtherNet/IP NIM. The status data (along with the I/O data) is transmitted through the polled I/O connection.

NIM Status Data

The NIM status provides the error/status information for internal island bus operations. The NIM status array is composed of a low byte and a high byte. You can access the *NIM status* through the EtherNet/IP explicit connection by following the path: class 101\instance 1\attribute 21.

Low byte: Each bit in the *NIM status* low byte array indicates a specific event:

Bit	Meaning of Value
D0–D5	0 = The value of this reserved bit.
D6	1 = Ethernet traffic overload: the NIM has experienced at least one Ethernet traffic overload condition. This bit will not be reset until the next power cycle.
D7*	1 = Ethernet traffic overload status: the NIM has recently experienced an Ethernet traffic overload condition. This bit will clear automatically, 15 seconds after the register is first read following an overload condition.
*Bits marked with an asterisk (*) in this table indicate that the NIM has stopped operating when the bit is set. The conditions are caused by internal errors related to either the NIM, the island configuration software, or the hardware.	

The detection of these errors results in the stopping of the island bus. The only ways to get out of this error state are to cycle the power, reset the island, or clear the error with the Advantys Configuration Software.

High byte: Each bit in the *NIM status* high byte array indicates a specific event:

Bit	Meaning of Value
D8*	Device inoperable
	1 = the NIM has detected a module on the island bus that is not operational.
	0 = all modules are operational.
D9*	Protected mode
	1 = any global diagnostic bit (except *RESET*) is set
	0 = all global diagnostic bits are set to 0
D10	External error detected
	1 = Fieldbus is not operating as expected.
	0 = fieldbus operating normally.

Bit	Meaning of Value
D11	Protected mode
	1 = NIM in protected mode: The RST button is disabled and the island configuration requires a password to write to it.
	0 = The NIM is not in protected mode: The RST button is enabled and the island configuration is not password-protected.
D12	Removable memory card validity
	1 = the configuration on the card is invalid
	0 = the configuration on the card is valid, the card is absent, or the card is empty.
D13	1 = reflex action functionality has been configured (for NIMs with firmware version 2.0 or greater).
D14	1 = one or more island modules have been hot-swapped (for NIMs with firmware version 2.0 or greater).
D15*	1 = Advantys Configuration Software is controlling the output data of the island’s process image.
	0 = the fieldbus master is controlling the output data of the island’s process image.
*Bits marked with an asterisk (*) in this table indicate that the NIM has stopped operating when the bit is set.	

Section 5.3
Data Exchange

EtherNet/IP Data Exchange

Introduction

This topic discusses the manner in which bit packed process image data is exchanged between the STB NIC 2212 EtherNet/IP NIM and the fieldbus master through a polled connection.

NOTE: In this discussion, *data* and *words* described as *input* and *output* are defined relative to the master. (The master receives *input* data and transmits *output* data.)

Data and Status Objects

Data exchange between the island and the EtherNet/IP fieldbus master involves three types of objects:

- *data objects*: These objects are the operating values the EtherNet/IP master either reads from the input modules or writes to the output modules.
- *status objects*: These objects are module health records sent by I/O modules and read by the EtherNet/IP master.
- *echo output data objects*: These objects are sent by digital output modules to the EtherNet/IP master. These objects are usually a copy of the data objects, but they can contain useful information when a digital output point is configured to handle the result of a reflex action.

The following table shows the relationship between different object types and different module types. It also shows the size of the different objects:

Module Type		Objects in the Input Data Image		Objects in the Output Data Image	
		Objects	Size	Objects	Size
digital input		data	2 bytes or less	does not apply	
		status*	2 bytes or less	does not apply	
digital output		echo output data	2 bytes or less	data	2-byte max
		status*	2 bytes or less	does not apply	
analog input	channel 1	data	2 bytes	does not apply	
		status	1 byte	does not apply	
	channel 2	data	2 bytes	does not apply	
		status	1 byte	does not apply	
analog output	channel 1	status	1 byte	data	2 bytes
	channel 2	status	1 byte	data	2 bytes
*Not available for every module. Check *The Advantys Hardware Components Reference Guide* (890 USE 172) for relevant modules.					

The Internal Process Image

The STB NIC 2212's process image contains memory areas (buffers) for the temporary storage of input and output data. The internal process image is part of the NIM's island bus scanner area.

The island bus manages data exchange in both directions:

- *input data from the island bus*: The island bus scanner operates continuously, gathering data as well as status and confirmation bits and putting them into the process image's input buffer. Access the internal input process image through the EtherNet/IP explicit messaging connection by following this path: class 4, instance 101, attribute 3.
- *output data to the island bus*: The island bus scanner handles output data and places it in the process image's output buffer. The path for the internal output process image is: class 4, instance 102, attribute 3.
- *HMI-to-PLC data*: Access HMI-to-PLC data through the EtherNet/IP explicit messaging connection by following this path: class 4, instance 104, attribute 3.
- *PLC-to-HMI data*: The path for the PLC-to-HMI data is: class 4, instance 105, attribute 3.

NOTE: Input data and output data are assembled in the order of the island bus I/O modules (from left to right).

Word Boundaries and Bit Packing

Every entry in the process image is in a multiple-word format. If modules on the island bus have input or output data entries that are not multiple words, the corresponding word in the process image is moved to the next word boundary.

For example, a module with one bit of output data starts on a word boundary in the process image's output data buffer. The next process image entry starts on the next word boundary, thereby transmitting 15 unused bits of the module's first word. This results in latency during data transmission on the fieldbus.

Bit packing allows bits of data on the fieldbus from different digital I/O modules to be put together in a single byte, resulting in optimized bandwidth.

Bit Packing Rules

The STB NIC 2212 NIM observes these bit packing rules for the external process image:

- The first two bytes of the input process image contain island diagnostics information.
- Bit-packing follows the addressing order of the island bus I/O modules (from left to right) starting with the primary segment.
- The data object (or echo output data object) for a specific module precedes the status object for that module.
- Status objects and data objects for the same or different I/O module can be packed in the same byte if the size of the combined objects is eight bits or less.
- If the combination of objects requires more than eight bits, the objects are placed in separate contiguous bytes. A single object can not be split over two byte boundaries.
- For analog input modules, channel 1 data is followed immediately by channel 1 status, then channel 2 data and channel 2 status.
- The data object for each analog I/O module must start at the word boundary in the process image.

Input and Output Data Exchange

The application of EtherNet/IP's bit packing rules to the sample island assembly result in 6 bytes of output data and 19 bytes of input data. The tables that follow show how digital data is bit-packed for optimization, and how data, status, and echo data (from outputs) appear in the PLC as the same data type (*digital input data*). In these tables, *N* refers to the sample island node numbers. That is, *N1* represents the first addressable node (module) on the sample island bus, *N2* the second, and so forth.

Output Data Exchange

The following table shows how the 6 bytes of the sample island assembly output data process image are organized after applying the bit-packing rules:

Word*	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	empty (set to 0)		N4 output data				N2 output data	
	2	empty (set to 0)		N6 output data					
2	3	N8 (channel 1) analog output data (low byte)							
	4	N8 (channel 1) analog output data (high byte)							
3	5	N8 (channel 2) analog output data (low byte)							
	6	N8 (channel 2) analog output data (high byte)							
*assigned word offset inside PLC									

Input Data Exchange

The following table shows how the 19 bytes of the sample island assembly input data process image are organized after applying the bit packing rules. (The first word contains the NIM status.):

<table>
<tr><th>Word*</th><th>Byte</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td rowspan="2">1</td><td>1</td><td colspan="8">NIM status (low byte) *(see page 77)*</td></tr>
<tr><td>2</td><td colspan="8">NIM status (high byte) *(see page 77)*</td></tr>
<tr><td rowspan="2">2</td><td>3</td><td colspan="2">N2 output status</td><td colspan="2">N2 output echo</td><td colspan="2">N1 input status</td><td colspan="2">N1 input data</td></tr>
<tr><td>4</td><td colspan="4">N3 input status</td><td colspan="4">N3 input data</td></tr>
<tr><td rowspan="2">3</td><td>5</td><td colspan="4">N4 output status</td><td colspan="4">N4 echo output data</td></tr>
<tr><td>6</td><td colspan="2">empty (set to 0)</td><td colspan="6">N5 input data</td></tr>
<tr><td rowspan="2">4</td><td>7</td><td colspan="2">empty (set to 0)</td><td colspan="6">N5 input status</td></tr>
<tr><td>8</td><td colspan="2">empty (set to 0)</td><td colspan="6">N6 echo output data</td></tr>
<tr><td rowspan="2">5</td><td>9</td><td colspan="2">empty (set to 0)</td><td colspan="6">N6 output status</td></tr>
<tr><td>10</td><td colspan="8">empty (set to 0)</td></tr>
<tr><td rowspan="2">6</td><td>11</td><td colspan="8">N7 (channel 1) analog input data (low byte)</td></tr>
<tr><td>12</td><td colspan="8">N7 (channel 1) analog input data (high byte)</td></tr>
<tr><td rowspan="2">7</td><td>13</td><td colspan="8">N7 (channel 1) analog input status</td></tr>
<tr><td>14</td><td colspan="8">empty (set to 0)</td></tr>
<tr><td rowspan="2">8</td><td>15</td><td colspan="8">N7 (channel 2) analog input data (low byte)</td></tr>
<tr><td>16</td><td colspan="8">N7 (channel 2) analog input data (high byte)</td></tr>
<tr><td rowspan="2">9</td><td>17</td><td colspan="8">N7 (channel 2) analog input status</td></tr>
<tr><td>18</td><td colspan="8">N8 (channel 1) analog output status</td></tr>
<tr><td rowspan="2">10</td><td>19</td><td colspan="8">N8 (channel 2) analog output status</td></tr>
<tr><td></td><td colspan="8"></td></tr>
<tr><td colspan="10">*assigned word offset inside PLC</td></tr>
</table>

Chapter 6
STB NIC 2212 Services

Introduction

This chapter describes the services provided by the STB NIC 2212 network interface module.

What Is in This Chapter?

This chapter contains the following sections:

Section 6.1
IP Parameter Assignment

Assignment of IP Parameters from a Server

Introduction

The STB NIC 2212 network interface module can receive an IP address using either the DHCP or BootP service.

For information on how these services are implemented in the STB NIC 2212, including the specific IP assignment sequence *(see page 58)*.

DHCP

DHCP (Dynamic Host Configuration Protocol) manages network addressing parameters for networked devices, in compliance with RFC 1531.

A DHCP server stores a list of device names and associated IP parameter settings for every client device on the network. It dynamically assigns IP addressing settings in response to client requests. A DHCP server responds to both DHCP and BootP (a subset of DHCP) requests.

The STB NIC 2212 network interface module implements DHCP as a client. Its IP parameters can be dynamically assigned by a DHCP IP address server.

BootP

The Bootstrap Protocol (BootP) assigns IP addresses to nodes on an Ethernet network, in compliance with RFC 951. Clients on the network send BootP requests during their initialization sequence.

A BootP server stores a list of MAC addresses and associated IP parameter settings for every client device on the network. After receiving a request, the server responds by assigning IP parameter settings to the BootP client.

The STB NIC 2212 network interface module implements BootP as a client. A BootP client transmits requests on the network every second until it receives a reply from an IP address server.

Section 6.2
Embedded Web Site

Introduction

The STB NIC 2212 NIM has an embedded web site that can be used to configure features and provide diagnostics.

What Is in This Section?

This section contains the following topics:

STB NIC 2212 Web Site Overview

Introduction

Access the NIM's internal web pages with a web browser to view and edit configuration and diagnostic data for the STB NIC 2212.

Browser Requirements

To access the STB NIC 2212 web pages you need:
- Internet Explorer version 5.0 or greater
- Java Runtime Environment version 1.4.2 or greater

Security

The STB NIC 2212 web site provides two levels of security:

- A required web access username and password combination, which (depending upon your configuration password selection) provides either read-only or read/write web site access.
- An optional configuration password; if the configuration password is:
  - *enabled:* The web access username/password combination provides read-only access, and the configuration password provides write access to the STB NIC 2212 web site.
  - *disabled:* The web access username/password combination alone provides both read and write access to the STB NIC 2212 web site.

How to Access the STB NIC 2212 Web Site

Procedure

Use the following steps to access the STB NIC 2212 web site:

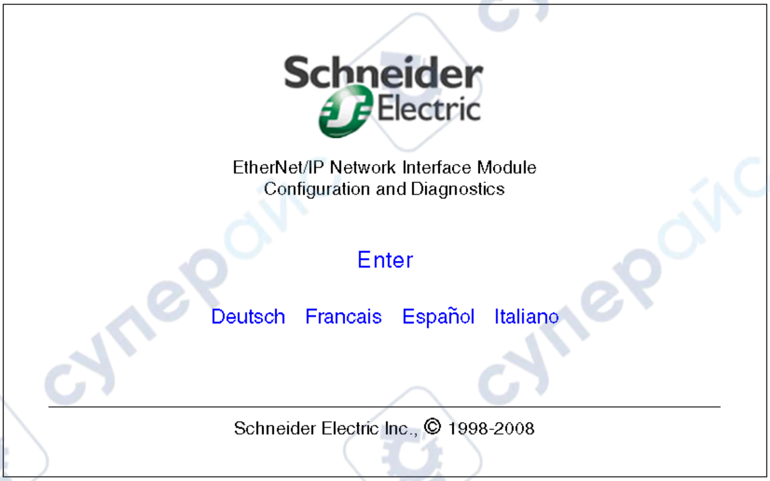
Step	Action	Result
1	Go to your url: **http://configured IP address**	The STB NIC 2212 home page *(see page 88)* appears.
2	Enter your language preference. English is the default language. ● If your language preference is English, click on the **Enter** button. ● To select a different language, click on its name, then click on the Enter button. For instance, for the German language, click Deutsche then Eingabe.	After you choose the language, the web access password dialog box *(see page 89)* appears.
3	Type the user name and the web access password for your STB NIC 2212 site. Then click OK to proceed. **Note:** Initially, the default user name and password are USER. Both are case-sensitive. They should be changed *(see page 90)* for your STB NIC 2212 web site.	After you press OK, the STB NIC 2212 Properties page *(see page 92)* appears.
4	To navigate to a different web page, click on the appropriate navigation tab located on the web page banner *(see page 91)*.	

STB NIC 2212 Web Site Home Page

The Home Page

The home page provides you with access to the STB NIC 2212 web site. It offers a choice of five languages (English, German, French, Spanish, and Italian) in which to view the information.

The home page appears after you enter the IP address of the STB NIC 2212 in your web browser's address field:



EtherNet/IP Network Interface Module
Configuration and Diagnostics

Schneider Electric Inc., © 1998-2008

How to Restrict Web Site Access (Password Protection)

Introduction

Access to the STB NIC 2212 web site is password-protected. All visitors to the STB NIC 2212 site must correctly complete the web access password dialog box that appears immediately after clicking Enter on the STB NIC 2212 home page *(see page 88)*.

NOTE: By default, all users that enter the web site through the password screen have read/write access to the web pages. You can restrict some users to read-only access if you create a configuration password *(see page 100)*, even if they successfully access the site through the password screen.

Default User Name and Password

Sign in with the default user name and password:

- default user name: USER
- default password: USER

You (HTTP) password screen looks like this:

security

User name: USER

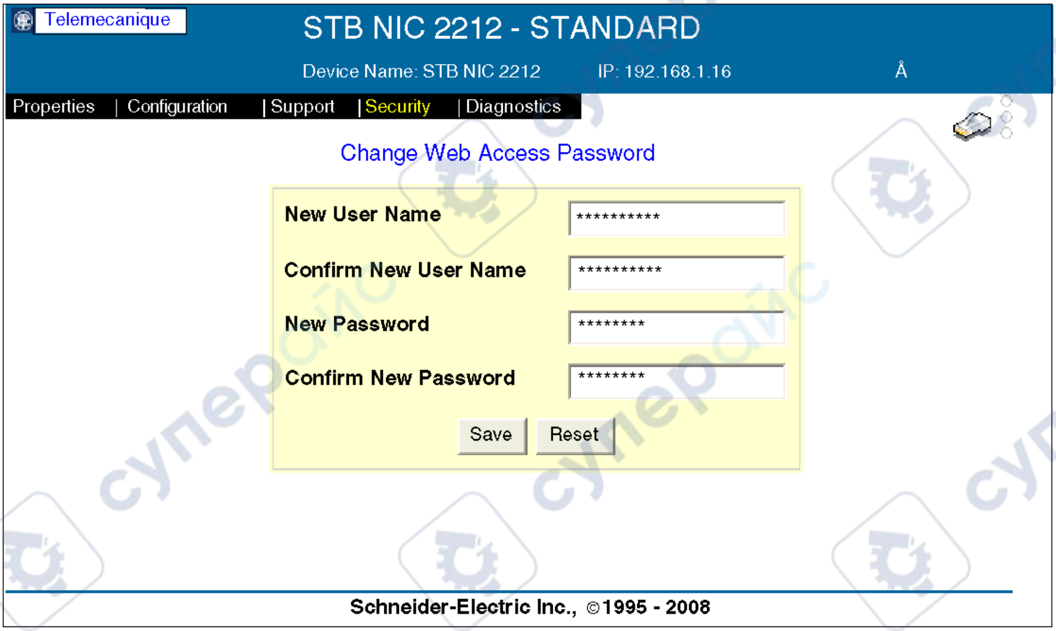
Password: ••••

☐ Remember my password

NOTE: The user name and password are case-sensitive.

Replacing the User Name and Password

You can replace the default user name and password to help limit access to the STB NIC 2212 web site. To do this, click the Security tab in the web page banner *(see page 91)* and select the Change Web Access Password page. This dialog box appears:



STB NIC 2212 - STANDARD

Device Name: STB NIC 2212 IP: 192.168.1.16

Change Web Access Password

New User Name: **********
Confirm New User Name: **********
New Password: ********
Confirm New Password: ********

Schneider-Electric Inc., ©1995 - 2008

Setting Up the Web Access Login

Set up your new user name and password:

Step	Action	Comment
1	Type your new user name in the New User Name field.	● The user name or password can have a maximum of 8 alphanumeric characters or underscores (_). ● The characters are case-sensitive.
2	Type the user name again in the Confirm New User Name field.	
3	Type your web access password in the New Password field.	
4	Type the password again in the Confirm New Password field.	
5	Click the **Save** button.	The new web access user name and password take effect immediately, replacing any previously saved user name and password.

How to Navigate in the STB NIC 2212 Web Site

Page Banner

This banner appears at the top of every STB NIC 2212 web page:

The banner displays:

- *Device Name:* name of this NIM
- *IP:* IP address for this NIM
- communications activity LEDs (under the *Help* menu). LEDs indicate these activities when blinking:
  - top LED: HTTP activity (web pages)
  - middle LED: data refresh through Ethernet activity
  - bottom LED: FTP upload/download activity
- navigation tabs: These tabs (*Properties, Configuration, Support, Security, Diagnostics*) provide access to STB NIC 2212 web pages. (See the table below.)

The navigation tabs are links to the STB NIC 2212 web pages:

Tab	Description
Properties	This tab links to the Properties web page *(see page 92)*.
Configuration	Click these links on the Configuration tab to configure the STB NIC 2212: ● Configure Stored IP *(see page 93)* ● Configure SNMP *(see page 96)* ● Reboot *(see page 98)*
Support	For customer assistance, use the links on the Support tab *(see page 99)*.
Security	Click the links on the Security tab to set the password protection: ● Change Web Access Password *(see page 89)* ● Change Configuration Password *(see page 100)*
Diagnostics	Troubleshoot the STB NIC 2212 through these links on the Diagnostics tab: ● Ethernet Statistics *(see page 102)* ● Network Interface Module Modbus Registers *(see page 103)* ● EtherNet/IP I/O Objects *(see page 105)* ● Modbus I/O Data Values *(see page 106)* ● Island Configuration *(see page 108)* ● Island Parameters *(see page 109)* ● Error Log *(see page 110)*

The Properties Web Page

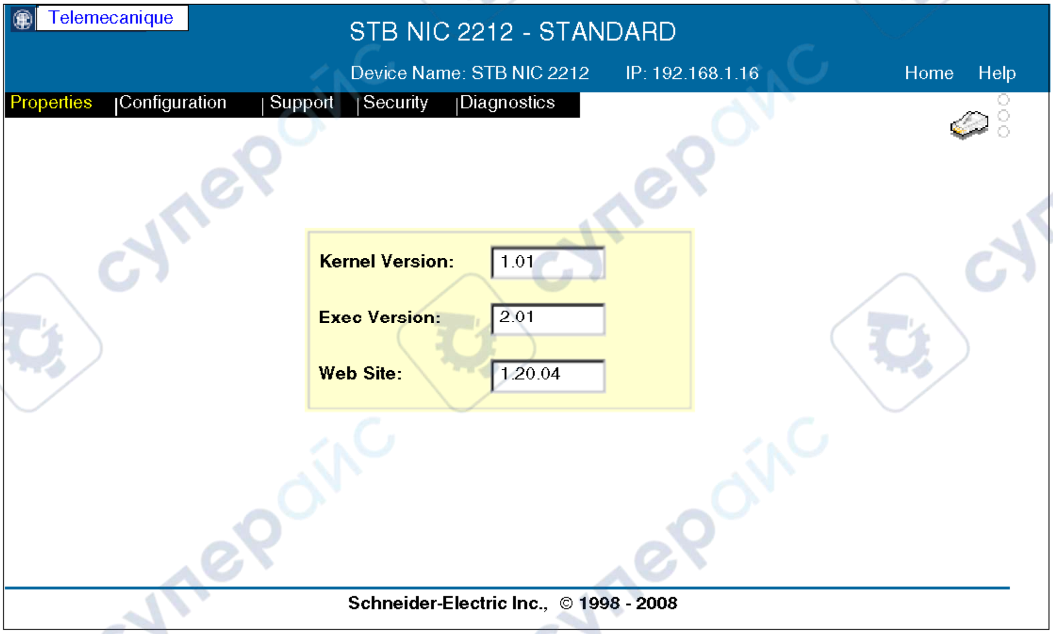
Introduction

The Properties web page displays STB NIC 2212 properties, such as the version of the kernel and the executive, as well as the communications protocols for which the STB NIC 2212 is configured.

NOTE: Access this page through the Properties tab on the web page browser *(see page 91)*.

Properties Web Page

The Properties page appears automatically after the HTTP server authenticates the user name and web access password. This is a sample Properties page:



Kernel Version: 1.01

Exec Version: 2.01

Web Site: 1.20.04

Schneider-Electric Inc., © 1998 - 2008

The Configure Stored IP Web Page

Introduction

To communicate as a node on an Ethernet network, the Ethernet port on the STB NIC 2212 must be configured with a valid IP address. The IP address must be unique on the Ethernet LAN on which the STB NIC 2212 resides.

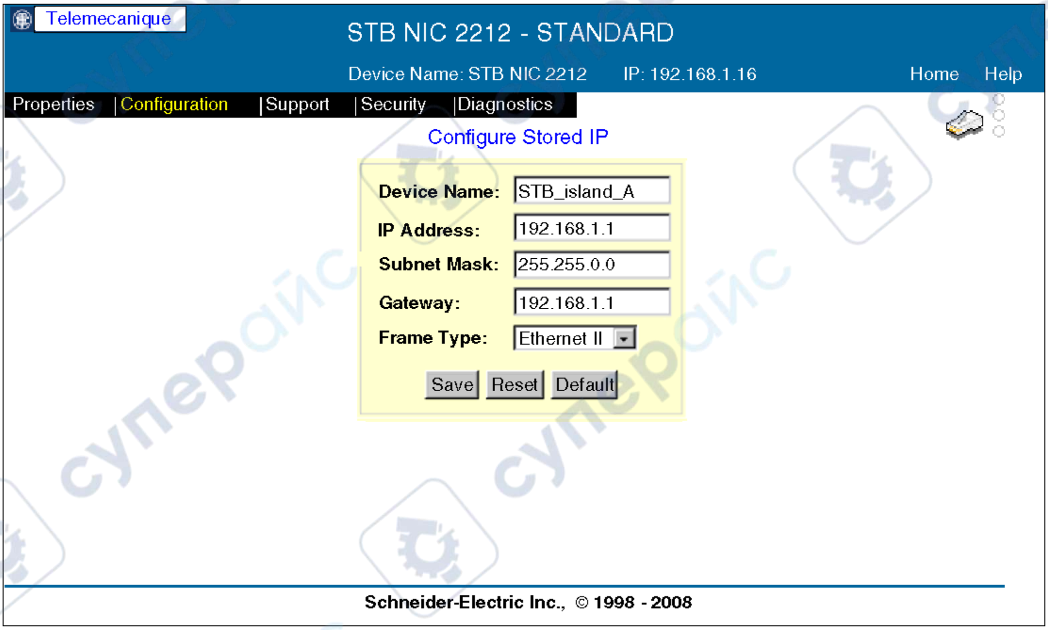
One of the available IP address assignment methods is to configure an IP address yourself, in the Configure Stored IP web page.

NOTE: The stored IP address is used in the fallback IP assignment process *(see page 58)*. The stored IP address is applied only if the STB NIC 2212 is not configured to obtain (or cannot obtain) an IP address from either a BootP or DHCP server.

NOTE: Access this page through the Configuration tab in the page banner *(see page 91)*.

Configure Stored IP Web Page

This is a sample Configure Stored IP web page:



Device Name: STB_island_A

IP Address: 192.168.1.1

Subnet Mask: 255.255.0.0

Gateway: 192.168.1.1

Frame Type: Ethernet II

IP Parameters

The IP address for the STB NIC 2212 has four parameters:

Parameter	Description
Device Name	A device name is the priority IP address assignment method used by the STB NIC 2212. If a device name is assigned, the IP address for the STB NIC 2212 is always associated with it.
IP Address	This is a unique 32-bit address assigned to every node on your network.
Subnet Mask	The subnet mask is 32 bits assigned with the IP address of the STB NIC 2212.
Gateway (optional)	The default gateway, typically a router, is where the host sends frames that are bound for remote networks after the subnet mask compare.
Frame Type	Data format used by a protocol. For example, the STB NIC 2212 can use either the Ethernet II or the IEEE 802.3 frame format. The default frame type is Auto-select.

Command Buttons

The following table describes how to use the command buttons on the Configure Stored IP web page:

Task	Instruction
Display the IP parameters stored in Flash memory	Reset
Display the MAC-based, derived default IP parameters.	Default
Save the IP parameters displayed on the Configure Stored IP web page.	Save
Note: When you make changes to the stored IP configuration, you must reboot the NIM using the Reboot Page *(see page 98)* in order for those changes to take affect. Changes to rotary switch settings require a power cycle of the STB NIC 2212 for the changes to take affect.	

Assign a Stored IP Address

Use this procedure to configure a stored IP address for the STB NIC 2212 after you have established communication (using the BootP address assignment method for instance). *In this case, your STB NIC 2212 can not have a device name.*

Step	Action
1	Open the STB NIC 2212 web site.
2	Click on the Configuration tab of the web page banner *(see page 91)* to display the Configuration menu.
3	Select the Configured IP option.
4	If your STB NIC 2212 has a Device name, remove it by deleting the Device Name setting *(see page 59)*.
5	In the IP address field, type the IP address, subnet mask, gateway address, and framing type that you want to use.

Step	Action
6	Click on the **Save** button to save the address to Flash memory and RAM.
7	Set the lower rotary switch *(see page 27)* to a STORED position, and cycle power to the STB NIC 2212.
8	Cycle the power to the STB NIC 2212.

Restore Default Parameters from the Web

Reconfigure the STB NIC 2212 with its default IP parameters:

Step	Action
1	Open the STB NIC 2212 web site.
2	Click on the Configuration tab to display the Configuration menu.
3	Select the Configured IP option. **Result**: The Configured IP web page *(see page 93)* opens.
4	Click on the Default button. **Notes**: ● The IP address parameters are restored to their default values. ● The address is based on the 48-bit MAC address that was programmed into the STB NIC 2212 when it was manufactured.
5	Click on the Save button to save the address to Flash memory and RAM.
6	Click on the Configuration tab to return to the Configuration menu.

Configuring a Device Name

You can assign, edit, or delete an internally set device name for an STB NIC 2212 in the Device Name web page after you have established communication to the STB NIC 2212 (using the BootP method for instance):

Step	Action
1	Open the STB NIC 2212 web site.
2	Click on the **Configuration** tab to display the Configuration menu.
3	Select the **Device Name** page.
4	Enter the **Device Name**.
5	Click on the **Save** button to save your device name to Flash memory and RAM.
6	Set the lower rotary switch to an STORED position.
7	Cycle the power on the STB NIC 2212.

NOTE: Elsewhere in this guide is a description of device name assignment through the numeric rotary switch settings *(see page 28)*.

The Configure SNMP Web Page

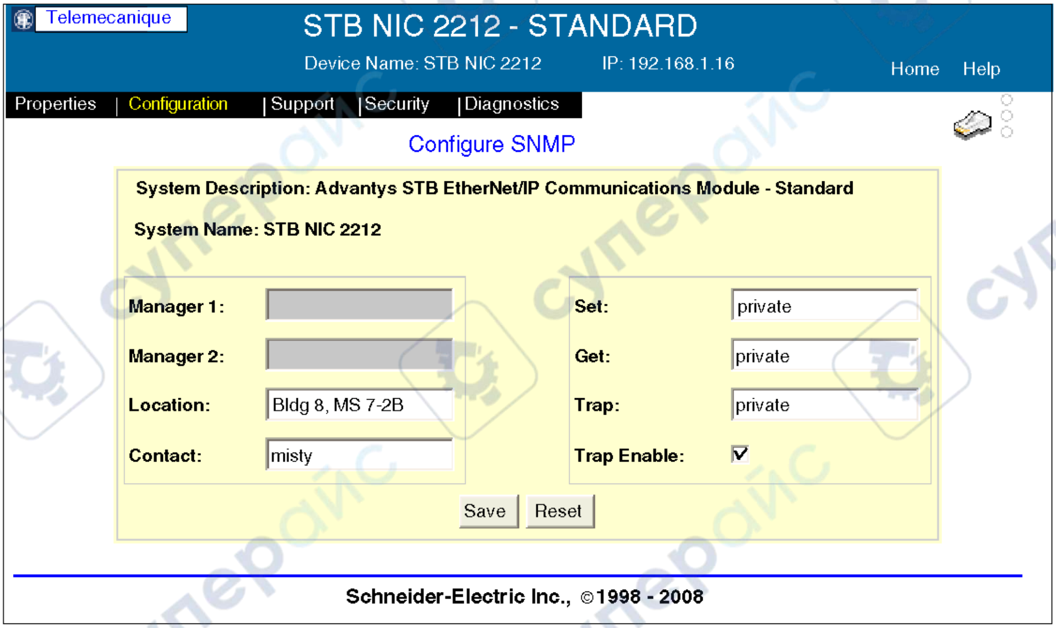
Introduction

The Configure SNMP web page provides access to the parameters used by the SNMP agent contained in the STB NIC 2212.

NOTE: Access this page through the Configuration tab in the page banner *(see page 91)*.

Configure SNMP Web Page

This is a sample Configure SNMP web page:



Device Name: STB NIC 2212 IP: 192.168.1.16

Configure SNMP

System Description: Advantys STB EtherNet/IP Communications Module - Standard

System Name: STB NIC 2212

Manager 1:

Manager 2:

Location: Bldg 8, MS 7-2B

Contact: misty

Set: private

Get: private

Trap: private

Trap Enable: ☑

Schneider-Electric Inc., ©1998 - 2008

This table describes the settings for the SNMP agent:

Purpose	Field Name	Description
Manager	Manager 1	Grayed-out. Available for IP address of SNMP Manager 1.
	Manager 2	Grayed-out. Available for IP address of SNMP Manager 2.
Agent	Location	255-character, case-sensitive alphanumeric string describing the location of the STB NIC 2212 (agent device).
	Contact	255-character, case-sensitive alphanumeric string identifying the contact person for the STB NIC 2212.
Community	Set	100-character, case-sensitive alphanumeric community string used to write the value of a point of information. A SetRequest is used by an SNMP manager to write to the STB NIC 2212. The default community name for the STB NIC 2212 is `public`. **Note:** If you enable an Authentication Failure Trap, assign a private community string for SetRequest.
	Get	100-character, case-sensitive alphanumeric community string, assigned by the user and used by the master to read the value of a point of information provided by the STB NIC 2212. The default community name for the STB NIC 2212 is `public`. **Note:** If you do not assign a private community string for this field, any SNMP manager can read the MIB objects for your STB NIC 2212.
Trap	Trap	The Trap text field contains the community string required by Trap services. By default this value is `public`. **Note:** Trap services are not supported by the STB NIC 2212.
	Trap Enable	This checkbox enables authentication for the community names (Set and Get) defined in the Community fields.

The Reboot Page

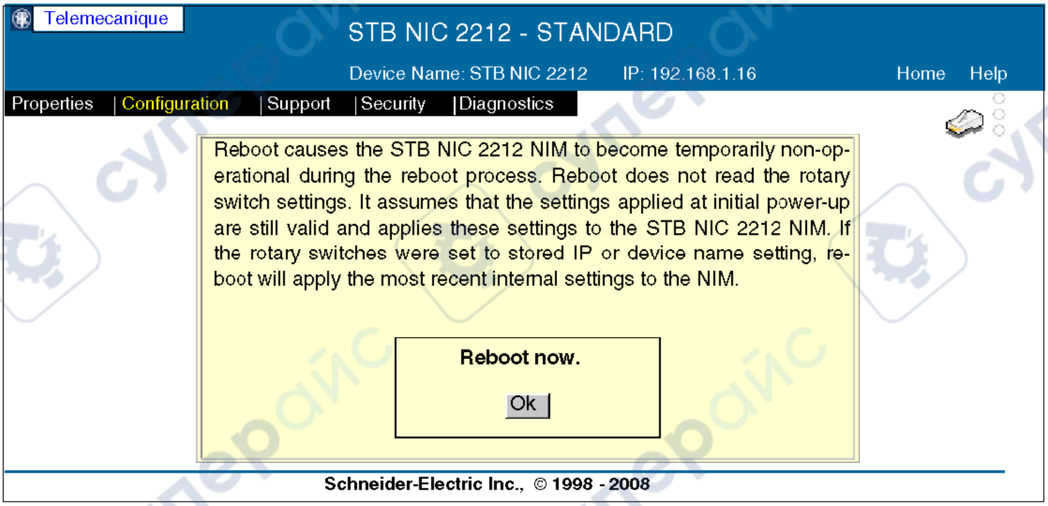
Introduction

The reboot operation temporarily disables the STB NIC 2212.

NOTE: The reboot operation does not read the rotary switches. Instead it applies the island's operating parameters that are stored in Flash memory to island devices, including the STB NIC 2212 NIM. It assumes that the settings that were applied at initial power-up are valid and applies these settings to the NIM. (You can perform this operation only when you are logged in to the web site or when no configuration password is present.) Access this page through the Configuration tab in the page banner *(see page 91)*.

Reboot Page

This is a sample Reboot page:



Reboot causes the STB NIC 2212 NIM to become temporarily non-operational during the reboot process. Reboot does not read the rotary switch settings. It assumes that the settings applied at initial power-up are still valid and applies these settings to the STB NIC 2212 NIM. If the rotary switches were set to stored IP or device name setting, reboot will apply the most recent internal settings to the NIM.

Reboot now.

Ok

Schneider-Electric Inc., © 1998 - 2008

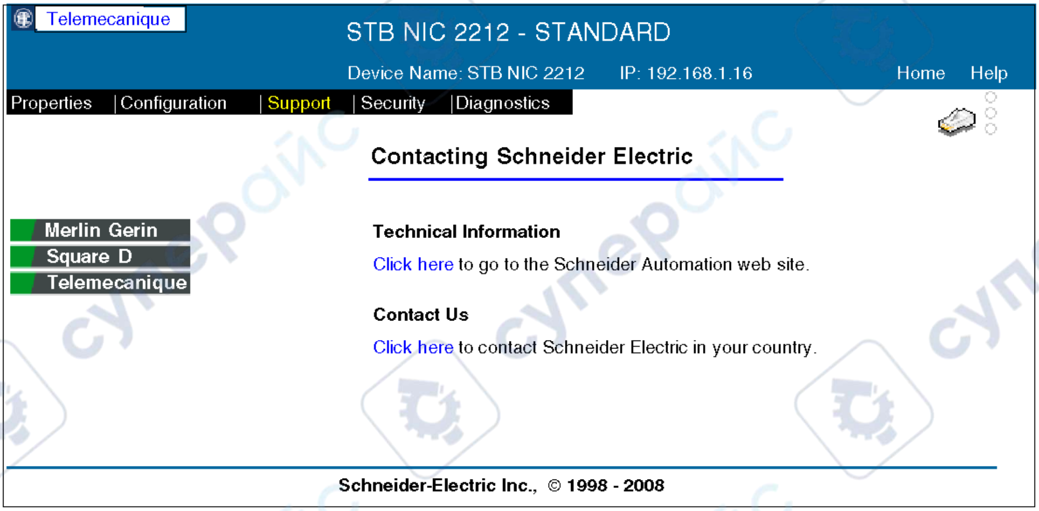
The Support Web Page

Introduction

To contact Schneider Electric about your STB NIC 2212 product, access this page through the Support tab in the page banner *(see page 91)*.

Support Page

This is a sample Support page:



Contacting Schneider Electric

Technical Information

Click here to go to the Schneider Automation web site.

Contact Us

Click here to contact Schneider Electric in your country.

Schneider-Electric Inc., © 1998 - 2008

NOTE: Access this page through the Support navigation tab in the page banner *(see page 91)*.

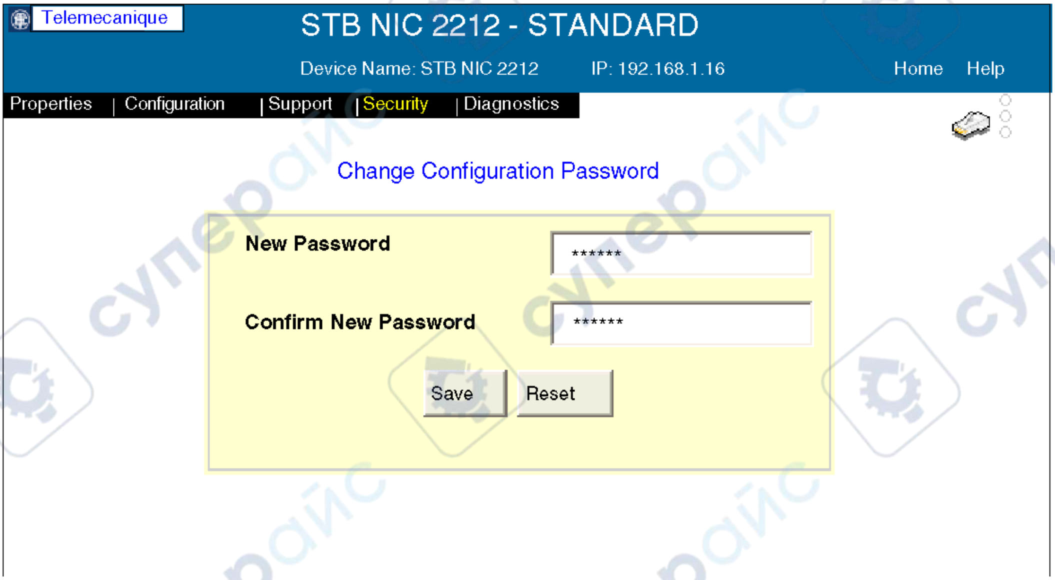
The Change Configuration Password Web Page

Introduction

The configuration password controls write access from the STB NIC 2212 web site to the NIM's Flash memory. There is no default configuration password. Until a configuration password is set, only the web access username and password *(see page 89)* combination is required to access and edit embedded web site parameters.

NOTE: Access this page through the Security tab in the page banner *(see page 91)*.

Set the configuration password on the Change Configuration Password web page:



New Password: ******

Confirm New Password: ******

Setting the Configuration Password

Use the following procedure to set up a configuration password for your STB NIC 2212 web site:

Step	Action	Comment
1	Click on the Security tab.	The Security menu is displayed.
2	In the Security menu, click Change Configuration Password.	The Change Configuration Password page (above) is displayed.
3	In the **New Password** field, type your configuration password.	The password consists of up to 6 alphanumeric characters and is case-sensitive.
4	In the **Confirm New Password** field, re-type the new password	
5	Click **Save**.	The configuration password takes effect immediately.

Login Prompt

After you create a configuration password, you must log in to modify parameters on the web pages. The login prompt (1) is displayed in the web banner (as shown in the following figure). To log in, enter the configuration password in the text box (2) and click Login:

1 Login / Log Out command button
2 configuration password text box

Logging In and Out

After you set the configuration password, use the following login/logout procedure to enable or disable write access to web site data:

Step	Action	Result
1	Type the case-sensitive configuration password in the text box next to the **Login** button and press the Login button.	● The Login button toggles to **Logout**. ● Your entire STB NIC 2212 web session is now write-enabled and the web site is accessible for write operations.
2	After completing your web edits, click on the **Logout** button to end write privileges on your web site.	● The Login button toggles to **Logout**. ● Write protection for your web site is restored.

The Configuration Password in the Configuration Software

The configuration password restricts access to:

- write privileges on the STB NIC 2212 web pages (above)
- control or configure of an Advantys STB island via the Advantys Configuration Software

NOTE: Both activities use the same password at any given time.

Use one of these methods to create or change a configuration password:

- the Change Configuration Password web page (above)
- the Advantys Configuration Software

NOTE:

- Both methods store the configuration password to the NIM’s Flash memory and overwrite any previously configured password.
- You can not remove configuration password protection from the Change Configuration Password web page. The configuration password protection can be removed only with the Advantys Configuration Software.

The Ethernet Statistics Web Page

Introduction

The Ethernet Statistics web page shows status and diagnostic information related to data transmissions to and from the STB NIC 2212 over the Ethernet LAN.

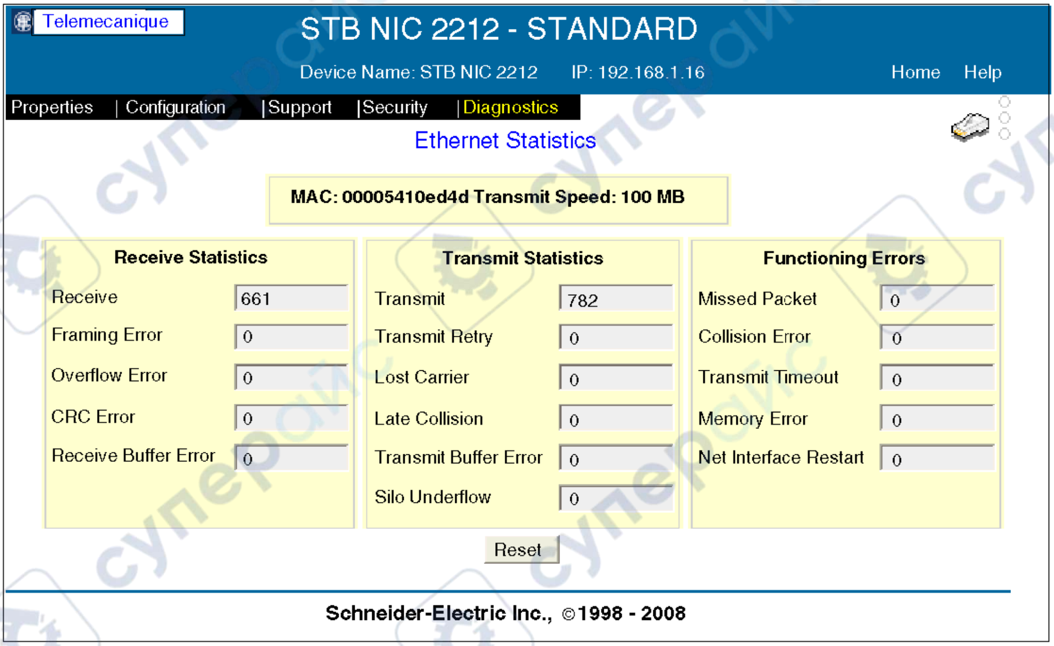
NOTE: Access this page through the Diagnostics tab in the page banner *(see page 91)*.

Refresh Rate

The statistics on this page are updated at the rate of once per second.

Ethernet Statistics Web Page

This is a sample Ethernet Statistics web page:



STB NIC 2212 - STANDARD

Device Name: STB NIC 2212 IP: 192.168.1.16

Ethernet Statistics

MAC: 00005410ed4d Transmit Speed: 100 MB

Receive Statistics

Receive	661
Framing Error	0
Overflow Error	0
CRC Error	0
Receive Buffer Error	0

Transmit Statistics

Transmit	782
Transmit Retry	0
Lost Carrier	0
Late Collision	0
Transmit Buffer Error	0
Silo Underflow	0

Functioning Errors

Missed Packet	0
Collision Error	0
Transmit Timeout	0
Memory Error	0
Net Interface Restart	0

Schneider-Electric Inc., ©1998 - 2008

This screen includes:
- *MAC:* unique MAC address for this STB NIC 2212
- *Ethernet Statistics (Receive Statistics, Transmit Statistics, Functioning Errors)*
- *Help:* Click this button to see a description of each Ethernet statistic.
- *Reset:* Click this button to return all counters to 0 when you are logged in.

The NIM Registers Web Page

Summary

The NIM Registers web page displays information about specific registers in the STB NIC 2212 process image. The register addresses indicate the registers that are displayed.

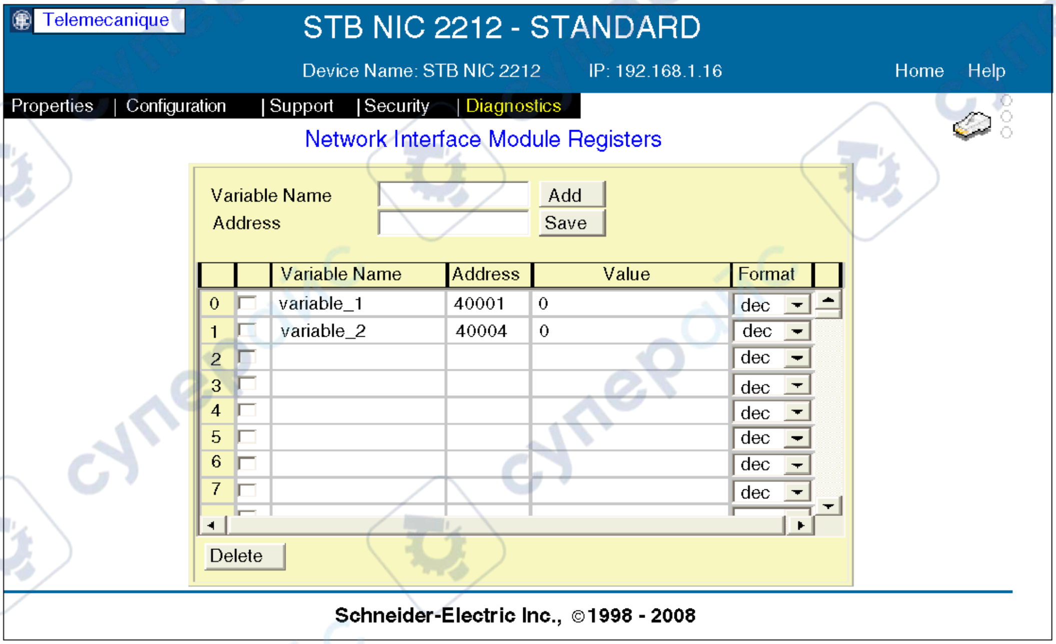
NOTE: Access this page through the Diagnostics tab in the page banner *(see page 91)*.

This page provides a customized read-only view of the STB NIC 2212 process image to everyone viewing the web page.

By supplying personal variable names (maximum 10 characters) and register locations, you can customize this page to show data that is important to you. After the display on the NIM Registers web page is written to Flash memory (by clicking on the Save button on the page), the display on this web page is fixed, providing a common view.

NIM Registers Web Page

This is a sample NIM Registers web page:



		Variable Name	Address	Value	Format
0	☐	variable_1	40001	0	dec
1	☐	variable_2	40004	0	dec
2	☐				dec
3	☐				dec
4	☐				dec
5	☐				dec
6	☐				dec
7	☐				dec

This screen includes:

- *Variable Name:* This column displays the variable name (10-character maximum).
- *Address:* This column shows the register number.
- *Value:* This column shows the current value of the register. (In this example, the values in registers 40001 and 40004 are both 0.)
- *Add:* Click this button to add a (checked) record.
- *Delete:* Click this button to delete a (checked) record.
- *Format:* Pull down to select the preferred data format (decimal, hexadecimal, binary) in this column.
- *Save:* Click this button to send the current view to Flash memory and save the format. This operation overwrites any previously saved views. The Save feature is only available when you are logged in or when no configuration password is present.

NOTE: A complete record requires both a *Variable Name* and an *Address*.

Command Buttons

Use the command buttons on the Network Interface Module Registers web page to accomplish simple tasks:

Task	Instructions
Add a row to the display.	Click the Add button.
Delete one or more rows from the display.	Click the checkbox in front of each row that you want to delete; then, click the Delete button.
Save the NIM’s view of the registers to Flash memory.	Click the Save button. **Note:** The save operation overwrites any previously saved views of the registers.

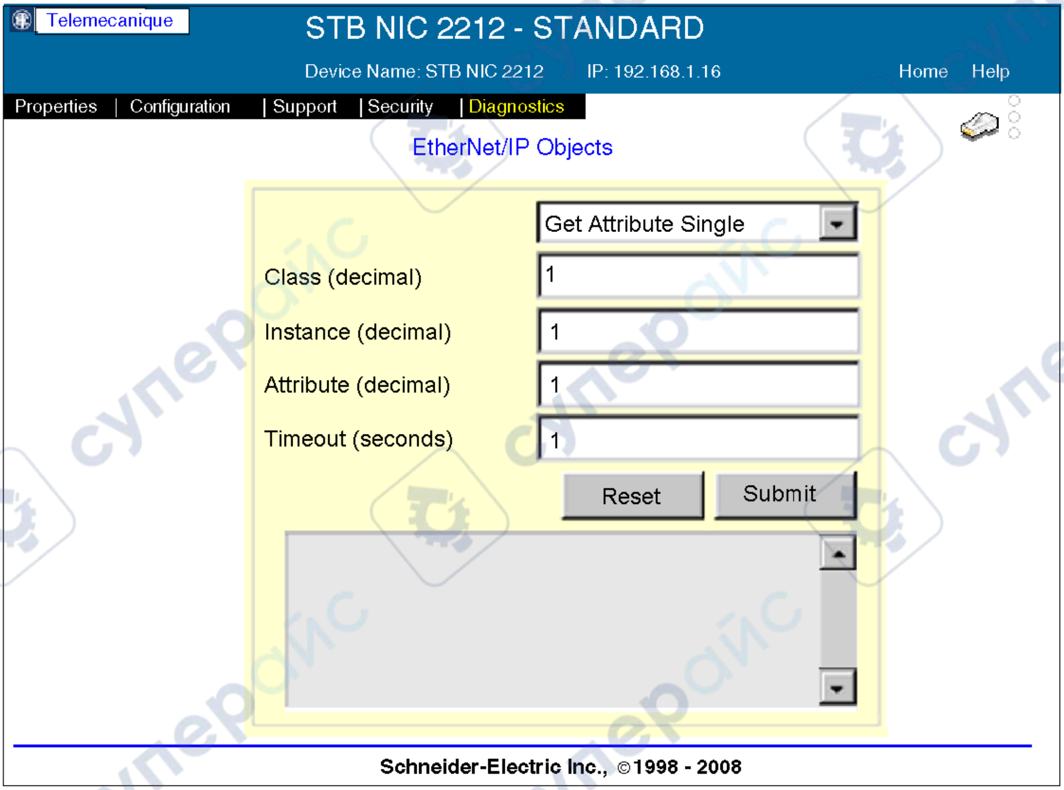
Format Feature

The format feature allows you to select whether the content of the NIM’s registers is displayed in decimal, hexadecimal, or binary notation.

The EtherNet/IP Objects Web Page

Monitor EtherNet/IP Objects

This page reads and displays EtherNet/IP objects. The size of the data that is read is limited to 496 bytes.



STB NIC 2212 - STANDARD

Device Name: STB NIC 2212 IP: 192.168.1.16

EtherNet/IP Objects

Get Attribute Single

Class (decimal) 1

Instance (decimal) 1

Attribute (decimal) 1

Timeout (seconds) 1

Schneider-Electric Inc., ©1998 - 2008

NOTE: Access this page through the Diagnostics tab in the page banner *(see page 91)*.

The Modbus I/O Data Values Web Page

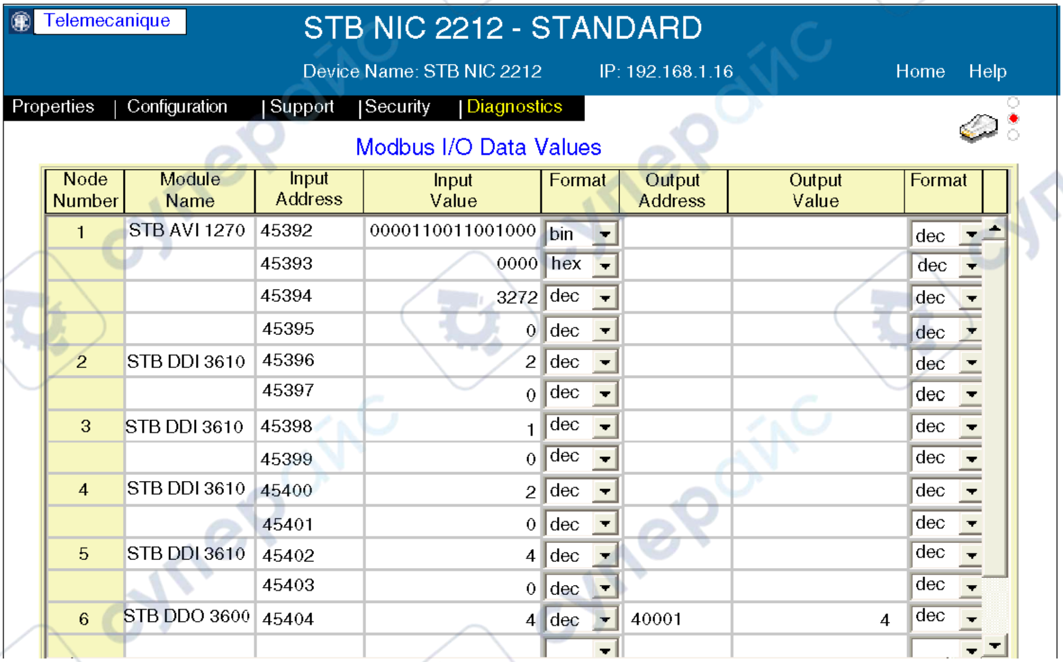
Introduction

The Modbus I/O Data Values web page displays the values stored in the process image output data area and input data area for a Modbus view of the I/O modules currently configured on the island. The order of information on this web page is the order of the I/O modules in the configuration.

NOTE: Access this page through the Diagnostics tab in the page banner *(see page 91)*.

Modbus I/O Data Values Web Page

This is a sample Modbus I/O Data Values web page:



Modbus I/O Data Values

Node Number	Module Name	Input Address	Input Value	Format	Output Address	Output Value	Format
1	STB AVI 1270	45392	0000110011001000	bin			dec
		45393	0000	hex			dec
		45394	3272	dec			dec
		45395	0	dec			dec
2	STB DDI 3610	45396	2	dec			dec
		45397	0	dec			dec
3	STB DDI 3610	45398	1	dec			dec
		45399	0	dec			dec
4	STB DDI 3610	45400	2	dec			dec
		45401	0	dec			dec
5	STB DDI 3610	45402	4	dec			dec
		45403	0	dec			dec
6	STB DDO 3600	45404	4	dec	40001	4	dec

These columns appear on the screen:
- *Node Number:* STB NIC 2212's island bus node address
- *Module Name:* STB module name
- *Input Address:* Modbus register location(s) for input and status data
- *Input Value*
- *Format* (two columns): pull-down menu for the preferred data format (decimal, hexadecimal, binary)
- *Output Address:* Modbus register location(s) for output data
- *Output Value*

The EtherNet/IP I/O Data Values Web Page

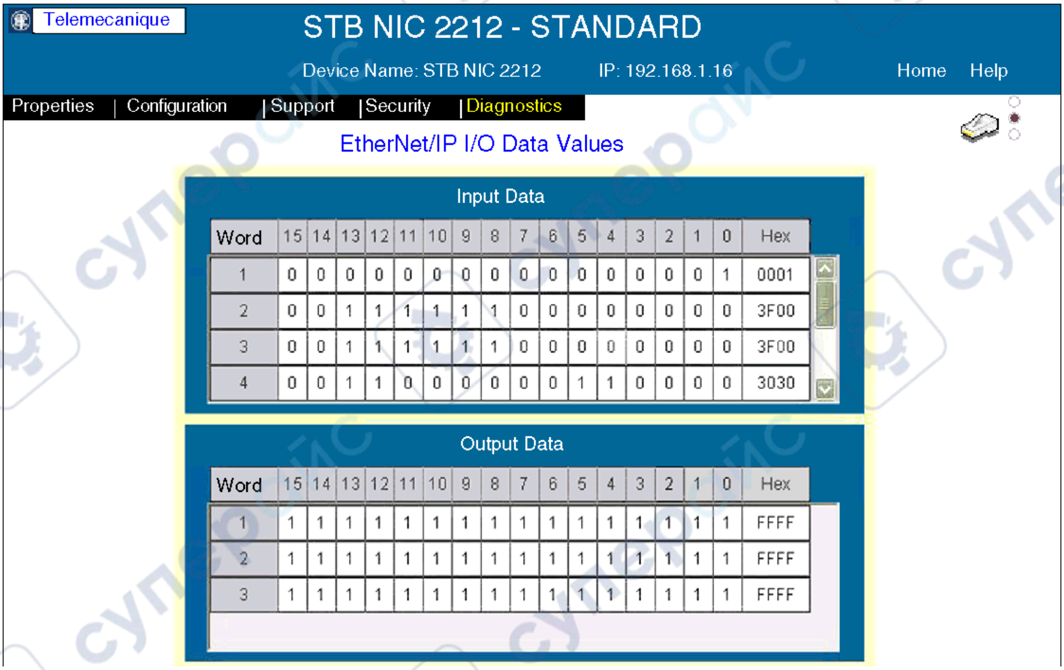
Introduction

The EtherNet/IP I/O Data Values page is a read-only view of input and output data values in the fieldbus view of the process image. These values apply to the I/O modules in the current island configuration.

NOTE: Access this page through the Diagnostics tab in the page banner *(see page 91)*.

EtherNet/IP I/O Data Values Page

This is a sample page:



EtherNet/IP I/O Data Values

Input Data

Word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001
2	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	3F00
3	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	3F00
4	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	3030

Output Data

Word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFF
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFF
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFF

The Input Data table shows packed input values for:

- I/O modules
- RTP

The Output Data table shows packed output values for:

- I/O modules
- virtual modules
- RTP

The order of the data is according to the sequence of I/O modules in the island configuration.

The Island Configuration Web Page

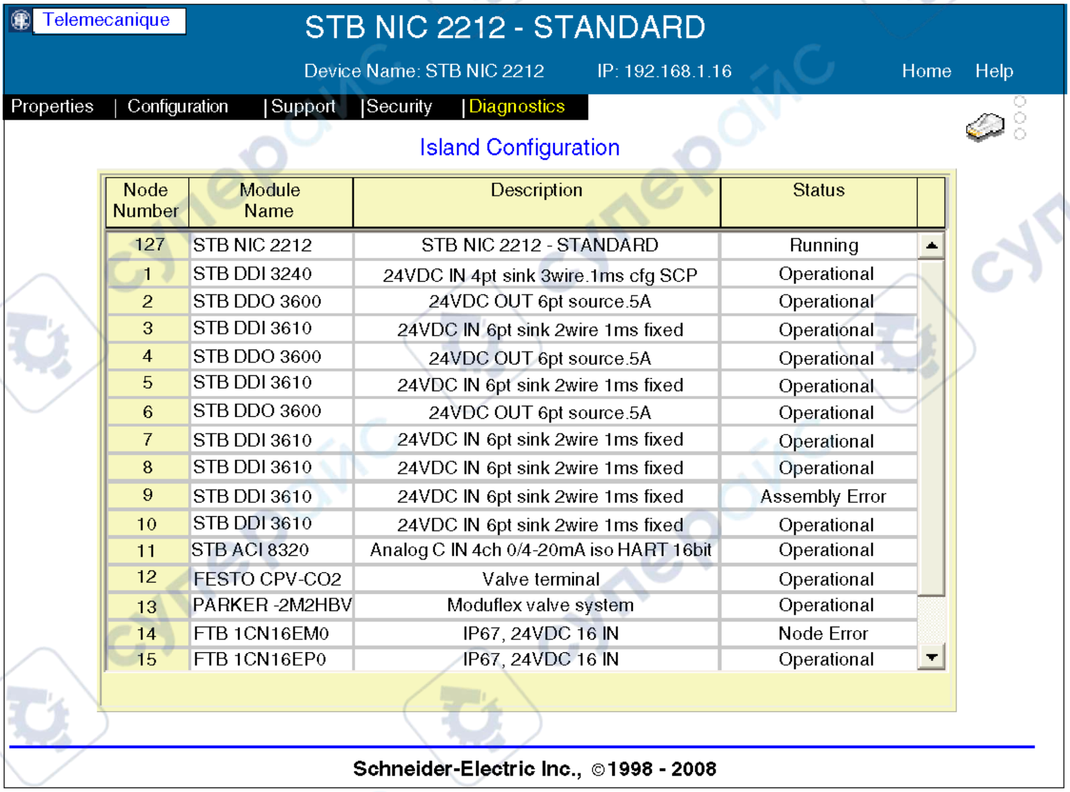
Introduction

The Island Configuration web page shows the configuration and status for each addressable module in the island configuration stored in the NIM. The modules are listed in order of their configuration starting with the STB NIC 2212 NIM.

NOTE: Access this page through the Diagnostics tab in the page banner *(see page 91)*.

Island Configuration Web Page

This is a sample Island Configuration web page:



Island Configuration

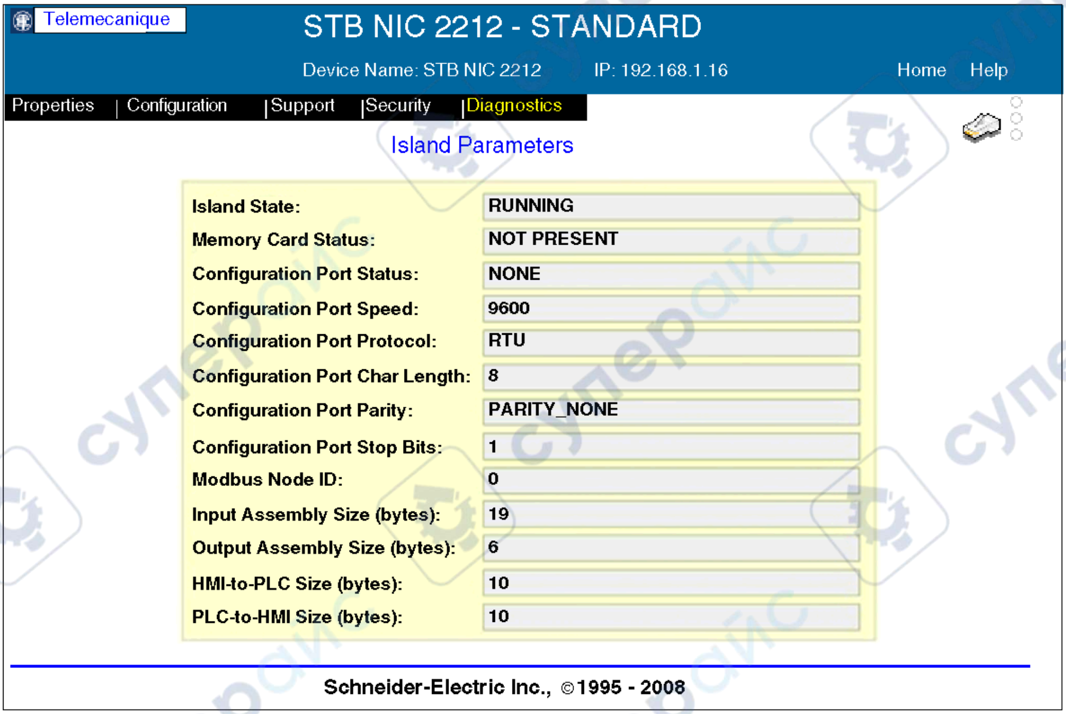
Node Number	Module Name	Description	Status
127	STB NIC 2212	STB NIC 2212 - STANDARD	Running
1	STB DDI 3240	24VDC IN 4pt sink 3wire.1ms cfg SCP	Operational
2	STB DDO 3600	24VDC OUT 6pt source.5A	Operational
3	STB DDI 3610	24VDC IN 6pt sink 2wire 1ms fixed	Operational
4	STB DDO 3600	24VDC OUT 6pt source.5A	Operational
5	STB DDI 3610	24VDC IN 6pt sink 2wire 1ms fixed	Operational
6	STB DDO 3600	24VDC OUT 6pt source.5A	Operational
7	STB DDI 3610	24VDC IN 6pt sink 2wire 1ms fixed	Operational
8	STB DDI 3610	24VDC IN 6pt sink 2wire 1ms fixed	Operational
9	STB DDI 3610	24VDC IN 6pt sink 2wire 1ms fixed	Assembly Error
10	STB DDI 3610	24VDC IN 6pt sink 2wire 1ms fixed	Operational
11	STB ACI 8320	Analog C IN 4ch 0/4-20mA iso HART 16bit	Operational
12	FESTO CPV-CO2	Valve terminal	Operational
13	PARKER -2M2HBV	Moduflex valve system	Operational
14	FTB 1CN16EM0	IP67, 24VDC 16 IN	Node Error
15	FTB 1CN16EP0	IP67, 24VDC 16 IN	Operational

Schneider-Electric Inc., ©1998 - 2008

The Island Parameters Web Page

Sample Island Parameters Web Page

The Island Parameters web page displays a read-only list of select island parameters and their current values. This is a sample Island Parameters page:



Island Parameters

Island State:	RUNNING
Memory Card Status:	NOT PRESENT
Configuration Port Status:	NONE
Configuration Port Speed:	9600
Configuration Port Protocol:	RTU
Configuration Port Char Length:	8
Configuration Port Parity:	PARITY_NONE
Configuration Port Stop Bits:	1
Modbus Node ID:	0
Input Assembly Size (bytes):	19
Output Assembly Size (bytes):	6
HMI-to-PLC Size (bytes):	10
PLC-to-HMI Size (bytes):	10

Schneider-Electric Inc., ©1995 - 2008

NOTE: Access this page through the Diagnostics tab in the page banner *(see page 91)*.

The Error Log Web Page

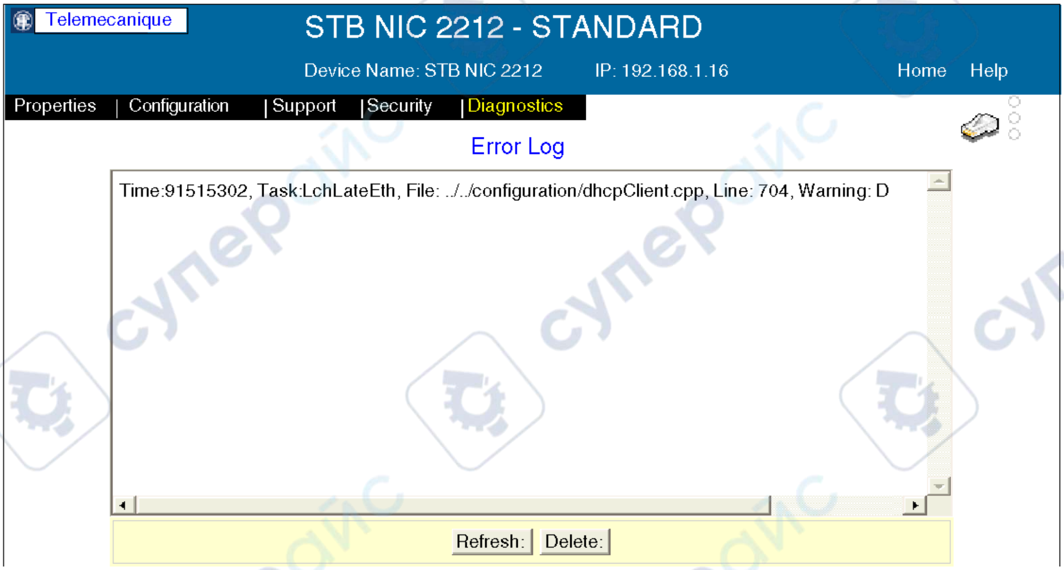
Introduction

The Error Log web page reports island-wide information that is collected during Advantys STB island operations.

NOTE: Access this page through the Diagnostic tab in the page banner *(see page 91)*.

Error Log Web Page

This is a sample Error Log web page:



```
Time:91515302, Task:LchLateEth, File: ../../configuration/dhcpClient.cpp, Line: 704, Warning: D
```

Error Log Operations

The operations associated with the Error Log web page are described in the following table:

Task	Instruction	Comment
Display the Error Log web page.	Click on the Diagnostics tab to display the Diagnostics menu *(see page 91)*. Then select the Error Log option.	The Error Log page is displayed.
Update the display.	Click on the Refresh button.	The error log is not updated automatically. It can only be updated manually. Therefore, you can read messages while other messages are incoming.
Delete the log.	Click on the Delete button.	You must have read/write authorization to delete the error log. **Note:** Deleting the error log on the web page removes it from Flash memory.

Section 6.3
SNMP Services

Introduction

The STB NIC 2212 supports the simple network management protocol (SNMP).

What Is in This Section?

This section contains the following topics:

SNMP Device Management

Introduction

The STB NIC 2212 contains a Simple Network Management Protocol (SNMP) Version 1.0 *agent* that is capable of supporting up to three concurrent SNMP connections.

User Datagram Protocol (UDP)

On the STB NIC 2212, SNMP services are delivered via the UDP/IP stack. UDP is the transport protocol used by the SNMP application in its communications with the STB NIC 2212.

NOTE: BootP and DHCP applications also use UDP as their transport layer when communicating with the STB NIC 2212.

SNMP Agents and Managers

The SNMP network management model uses the following definitions:

- manager: the client application program running on the master
- agent: the server application running on a network device, in this case, the STB NIC 2212

The SNMP manager initiates communications with the agent. An SNMP manager can query, read data from, and write data to other host devices. An SNMP manager uses UDP to establish communications with an *agent device* via an "open" Ethernet interface.

When the STB NIC 2212 is successfully configured with SNMP, the STB NIC 2212 agent and the SNMP manager devices recognizes each other on the network. The SNMP manager can then transmit data to and retrieve data from the STB NIC 2212.

Network Management Application

SNMP software allows an SNMP manager (remote PC) to monitor and control the STB NIC 2212. Specifically, SNMP services are used to monitor and manage:

- performance
- detected errors
- configuration
- security

SNMP Protocol Data Units (PDUs)

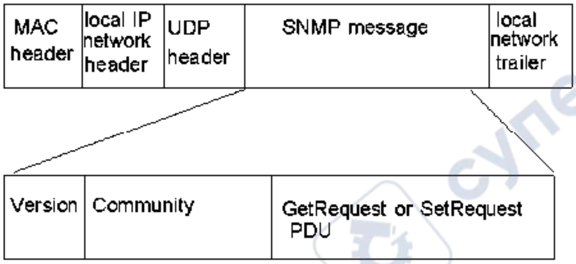
Protocol Data Units (PDUs) within SNMP carry requests and responses between the manager and the STB NIC 2212 agent. The following PDUs are used:

- GetRequest: An SNMP manager uses the "Get" PDU to read the value of one or more management information base (MIB) objects from the STB NIC 2212 agent.
- SetRequest: An SNMP manager uses the "Set" PDU to write a value to one or more objects resident on the STB NIC 2212 agent.

These PDUs are used in conjunction with MIB objects to get and set information contained in an Object Identifier (OID).

SNMP PDU Structure

An SNMP message is the innermost part of a typical network transmission frame:



Version & Community Identifiers

The STB NIC 2212 is configured with SNMP, Version 1.0. When setting up the SNMP agent function for your STB NIC 2212 *(see page 96)*, you should configure private community name(s) for GetRequest and SetRequest.

NOTE: If you do not configure private community names for GetRequest and SetRequest, any SNMP manager can read the MIB objects for your STB NIC 2212.

The community name is an identifier that you assign to your SNMP network when you set up the SNMP manager. Community names for the SNMP manager and agent must agree before SNMP processing can occur.

Configuring the SNMP Agent

The SNMP agent is configured using the STB NIC 2212’s embedded web site. Refer to the Configure SNMP Web Page *(see page 96)* topic for instructions on how to configure SNMP settings.

About the Schneider Private MIBs

Introduction

The following information describes the Schneider Electric private MIB, and the Transparent Factory Ethernet (TFE) and other subtrees that apply to the STB NIC 2212.

The STB NIC 2212 uses the MIB II standard.

Management Information Base (MIB)

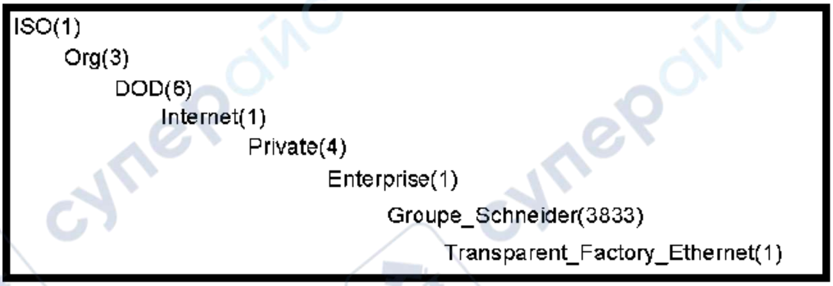
The Management Information Base (MIB) is an international communications database in which each object that SNMP accesses is listed with a unique name and its definition. The SNMP manager and agent applications both access the MIB.

Each MIB contains a finite number of objects. A management station (PC) running an SNMP application uses Sets *(see Advantys STB, Standard Ethernet Modbus TCP/IP Network Interface Module, Applications Guide)* and Gets *(see Advantys STB, Standard Ethernet Modbus TCP/IP Network Interface Module, Applications Guide)* to set system variables and to retrieve system information.

Schneider Private MIB

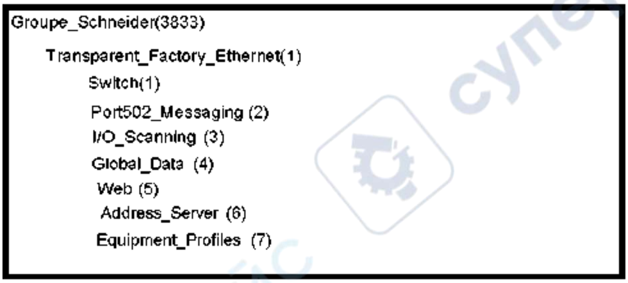
Schneider Electric has a private MIB, Groupe_Schneider (3833). The 3833 represents a private enterprise number (PEN) assigned to Groupe_Schneider by the Internet Assigned Numbers Authority (IANA). The number represents a unique object identifier (OID) for Groupe_Schneider.

The OID for the root of the Groupe_Schneider subtree is `1.3.6.1.4.1.3833`. This OID represents a path to the TFE subtree as follows:



Transparent Factory Ethernet (TFE) Subtree

Under the Groupe_Schneider MIB is a Transparent Factory Ethernet (TFE) private MIB that is controlled by the TFE SNMP embedded component. All SNMP managers that communicate with an Advantys STB island through an SNMP agent use the object names and definitions exactly as they appear in the TFE private MIB:



The TFE private MIB is a subtree of the Groupe_Schneider private MIB. The TFE SNMP component controls Groupe_Schneider's private MIB function. The Groupe_Schneider private MIB manages and monitors all of the Advantys STB system components through its associated network communications services.

The TFE MIB provides data to manage the main TFE communications services for the communication components that are part of the TFE architecture. The TFE MIB does not define specific management applications and policies.

The Transparent_Factory_Ethernet(1) defines groups that support TFE services and devices:

Service	Description
Port 502_Messaging(2)	This subtree defines objects for managing explicit client/server communications.
web(5)	This subtree defines objects for managing embedded web server activity.
equipment_profiles(7)	This subtree identifies objects for each type of device in the TFE product portfolio.
NOTE: Numbers such as 1, 2, 5, and 7 are OIDs.	

Transparent Factory Ethernet (TFE) MIB Subtree

Introduction

The Transparent Factory Ethernet (TFE) private is a subtree of the Groupe_Schneider private MIB. The TFE SNMP component controls Groupe_Schneider's private MIB function. Via its associated network communications services, the Groupe_Schneider private MIB manages and monitors the Advantys STB system components.

The TFE MIB provides data to manage the main TFE communications services for the communication components that are part of the TFE architecture. The TFE MIB does not define specific management applications and policies.

Transparent Factory Ethernet (TFE) MIB Subtree

The Transparent_Factory_Ethernet(1) defines groups that support TFE services and devices:

Service	Description
Port 502_Messaging(2)	This subtree defines objects for managing explicit client/server communications.
web(5)	This subtree defines objects for managing embedded web server activity.
equipment_profiles(7)	This subtree identifies objects for each type of device in the TFE product portfolio.
Note: Numbers such as 1, 2, 5, and 7 are OIDs.	

Web MIB Subtree

Introduction

The Web MIB subtree, OID 5, defines objects for managing embedded web server activity.

Web MIB Subtree

The following table describes the objects in the Web subtree that support Ethernet services used by the Advantys STB system:

Service	Indication	Available Values
webStatus(1)	global status of the web service	1: idle
		2: operational
webPassword(2)	switch to enable/disable use of web passwords	1: *disabled* (see note)
		2: enabled
webSuccessfulAccess(3)	total number of successful accesses to the STB NIC 2212 web site	
webFailedAttempts(4)	total number of unsuccessful attempts to access the STB NIC 2212 web site	
Note: When disabled, there is no web password prompt when you attempt to log in to the web pages. This completely bypasses the web password.		

Equipment Profile Subtree

Introduction

The equipmentProfile subtree (OID 3833.1.7) identifies objects for every device type in the TFE product portfolio.

Equipment Profiles MIB Subtree

The following table describes the objects contained in the equipmentProfile MIB subtree (group) that are common to all TFE products:

Service	Description	Comment
profileProductName(1)	displays the commercial name of the communication product as a string	e.g., STB NIC 2212
profileVersion(2)	displays software version of STB NIC 2212	e.g., Vx.y or V1.1
profileCommunicationServices(3)	displays list of communication services supported by the profile	e.g., Port502Messging, Web
profileGlobalStatus(4)	indicates global_status of the STB NIC 2212	available values ● 1: nok ● 2: ok
profileConfigMode(5)	indicates the IP configuration mode of the STB NIC 2212	available values ● 1: local: the IP configuration is created locally ● 2: DHCP-served: the IP configuration is created remotely by a DHCP server
profileRoleName(6)	indicates device name for IP address management	if none, value is *NO DEVICE NAME*
profileBandwidthMgt(7)	indicates the status of bandwidth management	value is disabled
profileBandwidthDistTable(8)		not available
profileLEDDisplayTable(9)	displays a table giving the name and state of each module’s LEDs	refer to the STB NIC 2212 LEDs discussion
profileSlot(10)		value=127
profileCPUType(11)		ADVANTYS STB
profileTrapTableEntries Max(12)		managers not required; value is 0
profileTrapTable(13)		not used
profileSpecified(14)		.1.3.6.1.4.1.3833.1.7.255.*x*
profileIPAddress(15)		IP address in use
profileNetMask(16)	subnet mask associated with SNMP agent’s IP address	–
profileIPGateway(17)	default gateway IP address for the SNMP agent	–
profileMacAddress(18)	Ethernet media dependent address of the SNMP agent	–

Chapter 7
Connection Examples for Advantys STB Islands on EtherNet/IP Networks

Overview

Follow the examples in this chapter to connect and commission Advantys STB islands on EtherNet/IP networks.

What Is in This Chapter?

This chapter contains the following sections:

Section 7.1

Connecting an Advantys STB Island to a ControlLogix Master Using RSLogix 5000

Introduction

This connection example tells you how to connect, configure, and commission an Advantys island with an STB NIC 2212 module for communications with a ControlLogix processor. The example uses RSLogix 5000 programming software. Communication to the Advantys island is through an Ethernet switch.

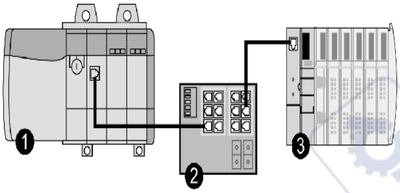
What Is in This Section?

This section contains the following topics:

About this Connection Example

Introduction

This connection example tells you how to connect, configure, and commission an Advantys island on an EtherNet/IP network:



1	ControlLogix rack including a processor and an EtherNet/IP communications module
2	Ethernet managed switch
3	Advantys STB island with an STB NIC 2212 EtherNet/IP network adapter

This example includes instructions for configuring two connections:
- Class1 connection: Add the Advantys island I/O configuration to the RSLogix 5000 project.
- Class3 connection: Place an explicit message in the RSLogix 5000 ladder logic. (This connection also applies to PLC-to-HMI and HMI-to-PLC data.)

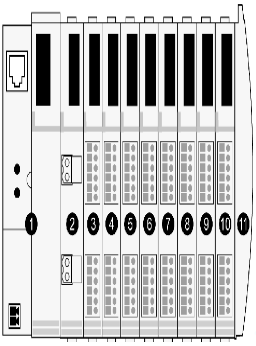
Before You Begin

Here are some characteristics of this example:
- Communication between the network and the Advantys island is through an STB NIC 2212 NIM.
- A CompactLogix L32E processor is implemented as the master.
- RSLogix 5000 (v. 13.04 or higher) programming software is used to program the processor for device control.
- A 1756-ENBT/A bridge is used in this example.
  NOTE: The older 1756-ENET/B bridge does not work with the STB NIC 2212 NIM.
- This example uses a simple Advantys island that has been auto-configured by pressing the RST button. Therefore, the example island does not contain enhanced CANopen modules, such as ATV motor controllers or third-party CANopen devices. (You can use enhanced modules if you configured the island from within the Advantys Configuration Software before downloading the configuration to the NIM.)

NOTE:
The described configuration method is essentially identical for:
- configuration with RSLogix versions 13 through 16
- CompactLogix or ControlLogix processors from Allen-Bradley

Sample Island

This sample island used in this connection example is a representative island bus assembly with an STB NIC 2212 NIM module as the EtherNet/IP network adapter:



1 STB NIC 2212: network interface module
2 STB PDT 3100: 24 VDC power distribution module (PDM)
3 STB DDI 3230: 24 VDC 2-channel digital input module (2 bits data, 2 bits status)
4 STB DDO 3200: 24 VDC 2-channel digital output module (2 bits data, 2 bits of echo output data, 2 bits status)
5 STB DDI 3420: 24 VDC 4-channel digital input module (4 bits data, 4 bits status)
6 STB DDO 3410: 24 VDC 4-channel digital output module (4 bits data, 4 bits of echo output data, 4 bits status)
7 STB DDI 3610: 24 VDC 6-channel digital input module (6 bits data, 6 bits status)
8 STB DDO 3600: 24 VDC 6-channel digital output module (6 bits data, 6 bits of echo output data, 6 bits status)
9 STB AVI 1270: +/-10 VDC 2-channel analog input module (16 bits data–channel 1, 16 bits data–channel 2, 8 bits status–channel 1, 8 bits status–channel 2)
10 STB AVO 1250: +/-10 VDC 2-channel analog output module (16 bits data–channel 1, 16 bits data–channel 2, 8 bits status–channel 1, 8 bits status–channel 2)
11 STB XMP 1100: island bus termination plate

The I/O modules in the sample assembly have the following island bus addresses:

I/O Model	Module Type	Module's Island Bus Address
STB DDI 3230	2-channel digital input	1
STB DDO 3200	2-channel digital output	2
STB DDI 3420	4-channel digital input	3
STB DDO 3410	4-channel digital output	4
STB DDI 3610	6-channel digital input	5
STB DDO 3600	7-channel digital output	6
STB AVI 1270	2-channel analog input	7
STB AVO 1250	2-channel analog output	8

The PDM and the termination plate are not addressable *(see page 44)*.

Assign an IP Address to the STB NIC 2212

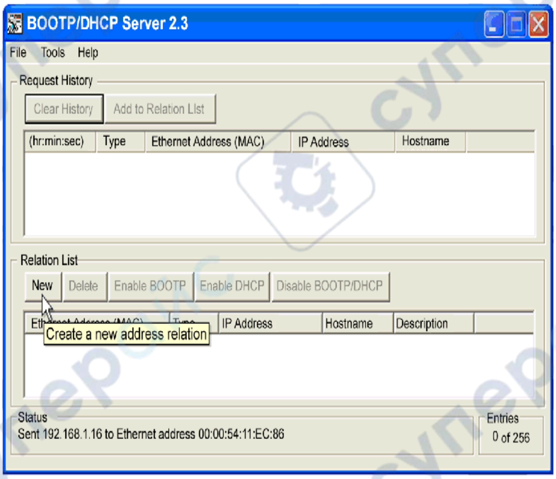
Introduction

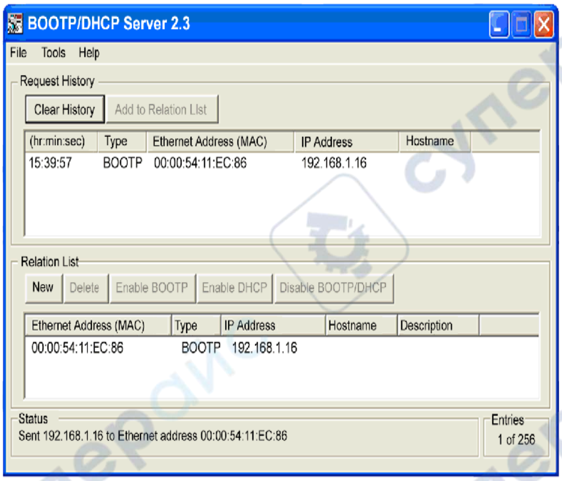
The STB NIC 2212 NIM comes from the factory without IP parameters. Therefore, it is necessary to assign an IP configuration to the NIM before you can configure the Advantys island for operation on an EtherNet/IP network.

NOTE: Make sure your island if fully assembled before you begin this procedure.

Assign IP Address via BootP

Use the BootP method to get an initial IP configuration for the island:

Step	Action
1	Set the ONES rotary switch on the STB NIC 2212 NIM to either of its BOOTP positions.
2	Start the Rockwell BootP/DHCP server on your PC.
3	In the server's Relation List, press New (Create a new address relation):  Sent 192.168.1.16 to Ethernet address 00:00:54:11:EC:86 **Note:** The Enable BOOTP, Enable DHCP, and Disable BOOP/DHCP button are not supported.
4	In the appropriate columns, enter the NIM's MAC address and the desired IP address. Press OK to see the new relation appear in the Relation List. **Notes**: ● The NIM's MAC address is stamped on the NIM enclosure. ● You do not have to put entries in the Hostname and Description columns.

Step	Action
5	With the rotary switch still in the BOOTP position, apply power to the NIM:  (hr:min:sec) · Type · Ethernet Address (MAC) · IP Address · Hostname 15:39:57 · BOOTP · 00:00:54:11:EC:86 · 192.168.1.16 Ethernet Address (MAC) · Type · IP Address · Hostname · Description 00:00:54:11:EC:86 · BOOTP · 192.168.1.16 Sent 192.168.1.16 to Ethernet address 00:00:54:11:EC:86 1 of 256 **Result**: After a moment, the NIM’s BootP-derived IP address appears in the Request History.
6	Open Internet Explorer and enter the NIM’s IP address in the address bar. **Result**: The appearance of the NIM’s home page verifies that a BootP address was successfully assigned to the NIM.
7	Close the BOOTP/DHCP Server application.

Next...

In the next procedure, you will configure the NIM’s BootP-derived IP parameters as an internally stored IP address. (A stored address means you don’t have to run an address server every time you cycle power to the NIM.) The procedure starts on the NIM’s home page.

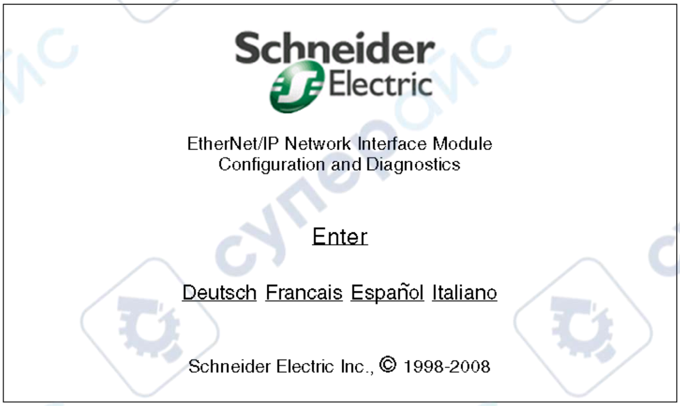
Configure a Stored IP Address

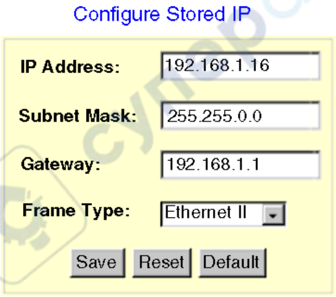
Objective

Configure a stored IP address so that you don’t have to run an address server every time you cycle power to the NIM.

IP Address Configuration

Store the BootP-derived IP parameters to the NIM:

Step	Action
1	Click Enter on the NIM’s home page:  Schneider Electric EtherNet/IP Network Interface Module Configuration and Diagnostics Enter Deutsch Francais Español Italiano Schneider Electric Inc., © 1998-2008 **Result**: You are prompted for a username and password.
2	Enter the username USER and the password USER. **Note**: Check the "remember my password" box if you want your PC to remember the username and password for a particular IP address.
3	Press OK. **Result**: After you press OK, the NIM’s Properties page appears.
4	Click on the Configuration tab to access the Configuration page.

Step	Action
5	Click the Configured IP link on the Configuration page to change the NIM's IP parameters. Using the dotted decimal format, enter the BootP-assigned IP address and the subnet mask:  Configure Stored IP **IP Address:** 192.168.1.16 **Subnet Mask:** 255.255.0.0 **Gateway:** 192.168.1.1 **Frame Type:** Ethernet II **Note:** You can enter an appropriate gateway address or leave it blank.
6	Click Save. **Result**: If there are no errors, the Save Successful screen appears.
7	Click OK on the Save Successful screen.
8	Move the NIM's ONES rotary switch to either of the two STORED positions. **Note**: The STORED position configures the NIM with the stored IP parameters at the next power cycle. (Until then, the NIM continues to operate with the BootP parameters.)
9	Close Internet Explorer.

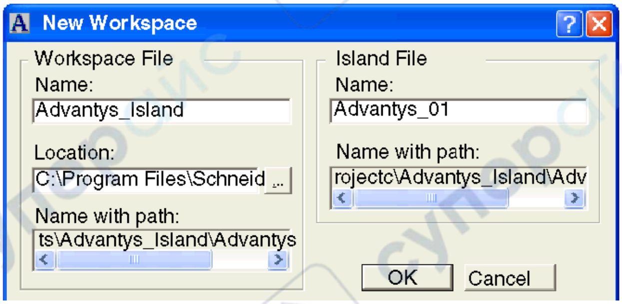
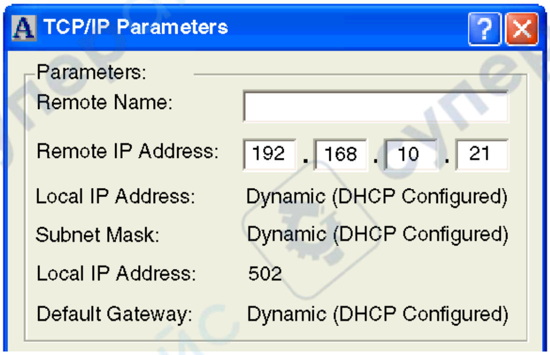
Next...

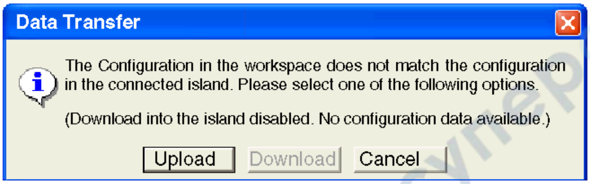
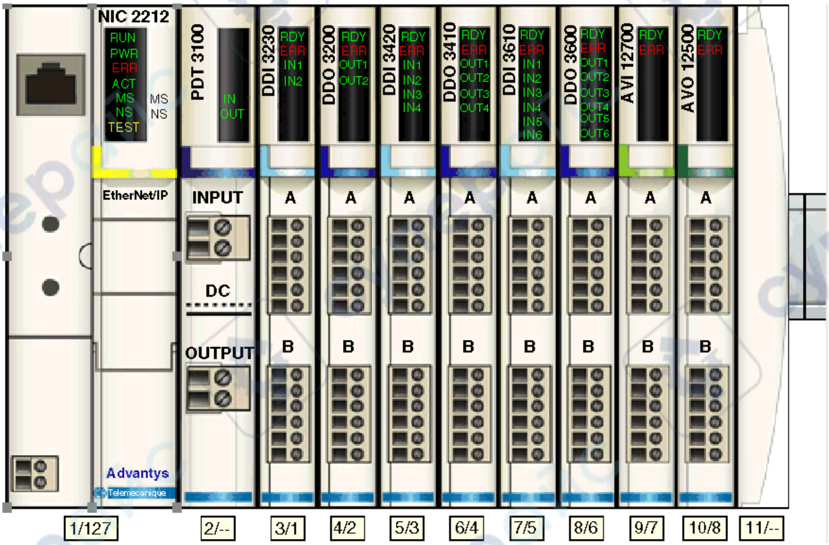
This completes the basic configuration of the Advantys island. The next step is to use the Advantys Configuration Software to determine the size of the I/O process image (in bytes). Use this size in the RSLogix 5000 project.

Determine and View the NIM’s I/O Process Image Size in the RSLogix Project

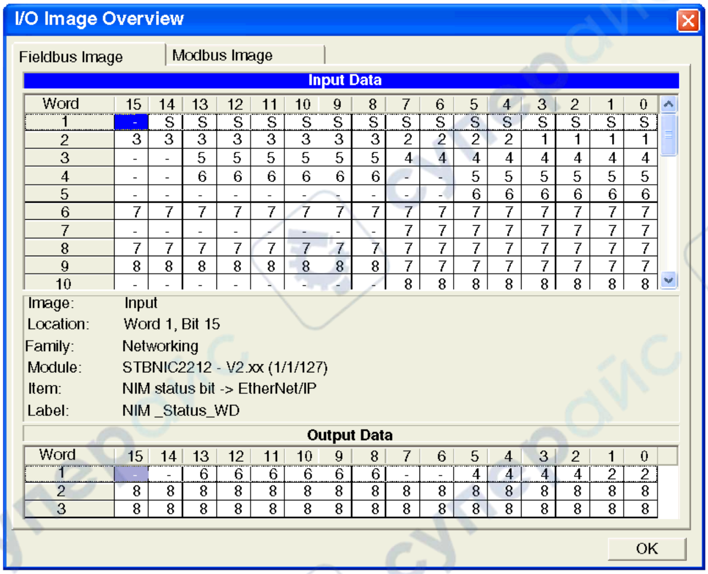
Determine the Data Size

Determine the size of the island I/O process image in the RSLogix project:

Step	Action
1	Press and hold the RST button for at least 2 seconds to properly configure the island.
2	Start the Advantys Configuration Software (ACS).
3	At the first prompt, select the STB icon. **Result**: The island development environment appears.
4	In the File menu, scroll to New Workspace. **Result**: The New Workspace screen appears.
5	Enter names for the new Workspace File and the Island File and press OK:  Workspace File Name: Advantys_Island Location: C:\Program Files\Schneid Name with path: ts\Advantys_Island\Advantys Island File Name: Advantys_01 Name with path: rojectc\Advantys_Island\Adv **Result**: The island rail in the New workspace is empty.
6	In the Online menu, scroll to Connection Settings. **Note**: You are starting to establish a connection between the ACS and the island that allows you to populate the rail with island modules.
7	On the Connection Settings screen, set the Connection Type to TCP/IP and press the Settings button.
8	The TCP/IP Parameters screen appears. Enter the NIM’s configured IP address in the Remote IP Address:  Parameters: Remote Name: Remote IP Address: 192 . 168 . 10 . 21 Local IP Address: Dynamic (DHCP Configured) Subnet Mask: Dynamic (DHCP Configured) Local IP Address: 502 Default Gateway: Dynamic (DHCP Configured) **Note**: You can leave the Remote Name field empty.
9	Press OK to close the TCP/IP Parameters screen, then press OK to close the Connections Settings screen.
10	In the Online menu, scroll to Connect. **Result**: The Data Transfer screen appears.

Step	Action
11	Press Upload:  **Data Transfer** The Configuration in the workspace does not match the configuration in the connected island. Please select one of the following options. (Download into the island disabled. No configuration data available.) **Result**: This action uploads the island configuration to the workspace.
12	The workspace display is populated with island data and shows the STB NIC 2212 and island modules:  **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 position (left to right) in the rack. ● The right-side integer (1 in this example) identifies the module's relative position (left to right) among the I/O modules. If the module is not an I/O module (that is, if it is a power supply or end of segment module) no right-side integer appears.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>13</td><td>
In the Island menu, scroll to I/O Image Overview. When the screen appears, select the Fieldbus Image tab:



Input Data

<table>
<tr><th>Word</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>1</td><td>-</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td></tr>
<tr><td>2</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>2</td><td>2</td><td>2</td><td>2</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>3</td><td>-</td><td>-</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>4</td><td>-</td><td>-</td><td>6</td><td>6</td><td>6</td><td>6</td><td>6</td><td>6</td><td>-</td><td>-</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>5</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>6</td><td>6</td><td>6</td><td>6</td><td>6</td><td>6</td></tr>
<tr><td>6</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td></tr>
<tr><td>7</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td></tr>
<tr><td>8</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td></tr>
<tr><td>9</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td></tr>
<tr><td>10</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
</table>

Image: Input
Location: Word 1, Bit 15
Family: Networking
Module: STBNIC2212 - V2.xx (1/1/127)
Item: NIM status bit -> EtherNet/IP
Label: NIM _Status_WD

Output Data

<table>
<tr><th>Word</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>1</td><td>-</td><td>-</td><td>6</td><td>6</td><td>6</td><td>6</td><td>6</td><td>6</td><td>-</td><td>-</td><td>4</td><td>4</td><td>4</td><td>4</td><td>2</td><td>2</td></tr>
<tr><td>2</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>3</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
</table>

Each table cell contains one of the following alpha-numeric indicators:

- **S**: indicates a status bit for the NIC 2212 network interface module
- **An integer**: identifies the relative physical position of an island I/O module (from left to right) with input or output data in that cell. For example:
  - The DDI 3230 input module is the first I/O module in the rack. Its data is designated by the integer 1 in bits 0 - 3 of word 2 in the Input Data table.
  - The DDO 3600 output module is the sixth data producing module in the rack. Its status and output echo data are designated by the integer 6 in bits 8 - 13 of word 4 and in bits 0 - 5 of word 5 in the Input Data table. Its output data is designated by the integer 6 in bits 8 - 13 of word 1 in the Output Data table.

Notes:

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 (that is, multiply by 2 and check for a remainder), then use that data as values for the *Input Size* (19) and *Output Size* (6) parameters when configuring the adapter's general connection properties.
</td></tr>
<tr><td>14</td><td>
Scroll to the end of the Input Data and Output Data tables.
Note: The total number of Input Data and Output Data words are last values in the table.
</td></tr>
<tr><td>15</td><td>
Convert the 16-bit word values on the Fieldbus Image tab (to the nearest byte).
Notes:

- RSLogix accepts only 8-bit byte data.
- In the example, the Input Data displaces 10 words, but only 19 bytes. The last byte is not used. Therefore, 19 bytes is the value you will enter later for the Input Assembly Instance in RSLogix 5000. In the same fashion, the 3 full words that the Output Data displaces are entered as 6 bytes in the Output Assembly Instance for RSLogix 5000 when you insert a generic Ethernet module *(see page 130)*.
</td></tr>
</table>

Next...

In the next procedure, you will add the island I/O configuration to the RSLogix 5000 project.

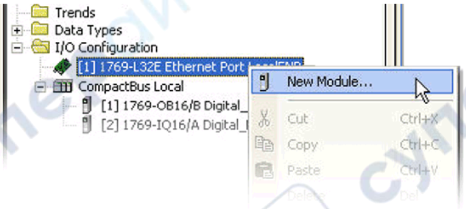
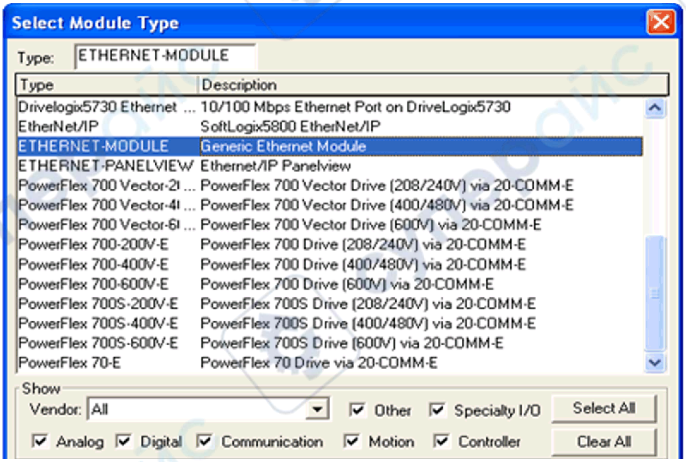
Add the Island I/O Configuration to the RSLogix Project (Class1 Connection)

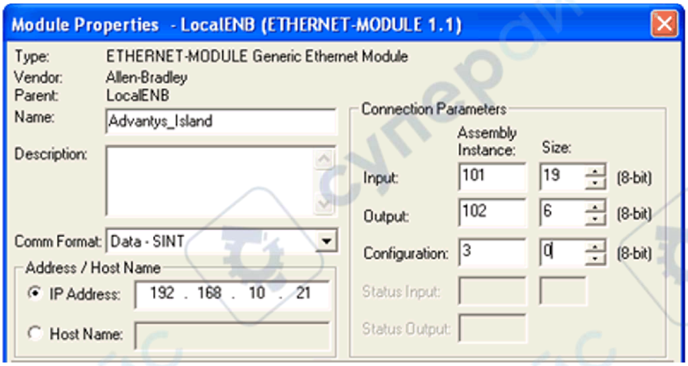
Objective

As the master, the CompactLogix processor must contain the island I/O configuration. The example makes a Class1 connection as part of the basic configuration of the RSLogix project. (This example uses RSLogix version 13.04 and the CompactLogix L32E processor.)

Insert a Generic Ethernet Module

Start this procedure with an RSLogix project that already includes the configurations for the CompactLogix L32E processor and an Ethernet port:

Step	Action
1	Right-click on the Ethernet port in the I/O Configuration tree, and scroll to New Module:  **Result**: After the selection of New Module, the Select Module Type screen appears.
2	Select ETHERNET-MODULE/Generic Ethernet Module and press OK:  Select Module Type Type: ETHERNET-MODULE Type — Description Drivelogix5730 Ethernet ... — 10/100 Mbps Ethernet Port on DriveLogix5730 EtherNet/IP — SoftLogix5800 EtherNet/IP ETHERNET-MODULE — Generic Ethernet Module ETHERNET-PANELVIEW — EtherNet/IP Panelview PowerFlex 700 Vector-2t ... — PowerFlex 700 Vector Drive (208/240V) via 20-COMM-E PowerFlex 700 Vector-4t ... — PowerFlex 700 Vector Drive (400/480V) via 20-COMM-E PowerFlex 700 Vector-6t ... — PowerFlex 700 Vector Drive (600V) via 20-COMM-E PowerFlex 700-200V-E — PowerFlex 700 Drive (208/240V) via 20-COMM-E PowerFlex 700-400V-E — PowerFlex 700 Drive (400/480V) via 20-COMM-E PowerFlex 700-600V-E — PowerFlex 700 Drive (600V) via 20-COMM-E PowerFlex 700S-200V-E — PowerFlex 700S Drive (208/240V) via 20-COMM-E PowerFlex 700S-400V-E — PowerFlex 700S Drive (400/480V) via 20-COMM-E PowerFlex 700S-600V-E — PowerFlex 700S Drive (600V) via 20-COMM-E PowerFlex 70-E — PowerFlex 70 Drive via 20-COMM-E Show Vendor: All ☑ Other ☑ Specialty I/O ☑ Analog ☑ Digital ☑ Communication ☑ Motion ☑ Controller **Result**:The Module Properties screen appears. (RSLogix 5000 interprets the entire Advantys island as a single module.) **Note**: With RSLogix 5000 v.16, it is possible to select a CIP bridge. This is useful if you are doing more than just polling I/O data. For instance, you can include I/O, HMI-to-PLC, PLC-to-HMI, and diagnostic data in a bridge to a single IP address.

Step	Action
3	Assign values in the Module Properties screen:  Module Properties - LocalENB (ETHERNET-MODULE 1.1) Type: ETHERNET-MODULE Generic Ethernet Module Vendor: Allen-Bradley Parent: LocalENB Name: Advantys_Island Description: Comm Format: Data - SINT Address / Host Name ● IP Address: 192 . 168 . 10 . 21 ○ Host Name: Connection Parameters Assembly Instance: / Size: Input: 101 / 19 (8-bit) Output: 102 / 6 (8-bit) Configuration: 3 / 0 (8-bit) Status Input: Status Output: Assign these values: ● Name: Enter a module name. RSLogix 5000 uses the name to generate controller tags for the I/O. ● Comm Format: Scroll to Data – SINT (because the island's I/O data format is treated as bytes). (See note 1.) ● IP Address: Enter the address previously configured for the NIM. ● Connection Parameters: For Input, enter an Assembly Instance of 101 with a size that corresponds to the previously calculated number of bytes (19). For Output, enter an Assembly Instance of 102 with a size that corresponds to the previously calculated number of bytes (6). For Configuration, enter an Assembly Instance of 3 and leave the size at 0. (See note 1.) ● Next: Press Next if you want to change the Requested Packet Interval to something other than 10 ms. (The minimum RPI setting is 4 ms.) ● Finish: Press Finish to complete the Class1 connection for the Advanty island's I/O data. **Note 1**: When the island configuration contains only input bytes: ● set the Comm Format to Input Data – SINT ● set the Output (Assembly Instance) to 254 **Note 2:** If you configured a CIP bridge in the previous step and would like to configure more than just I/O data, then the instances for HMI-to-PLC, PLC-to-HMI, and diagnostics data are 104, 105, and 100, respectively.

To Add a Listen Only Connection

To establish a Class I Listen Only connection to the island, set the Output Assembly Instance to 255 and the Output Size to 0 in the Module Properities screen (see above).

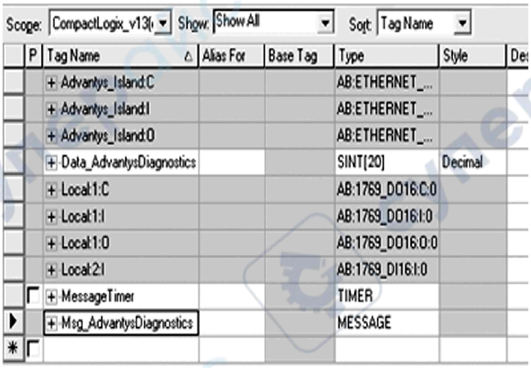
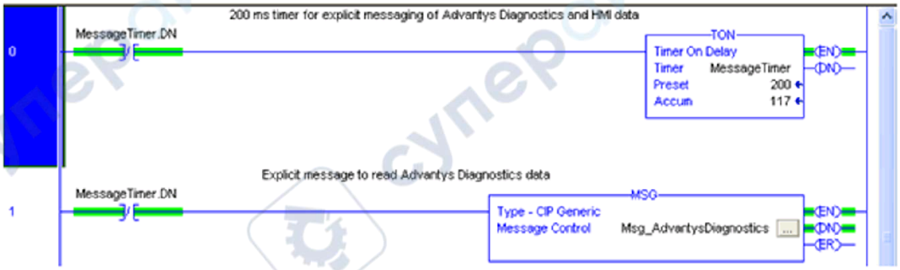
Next...

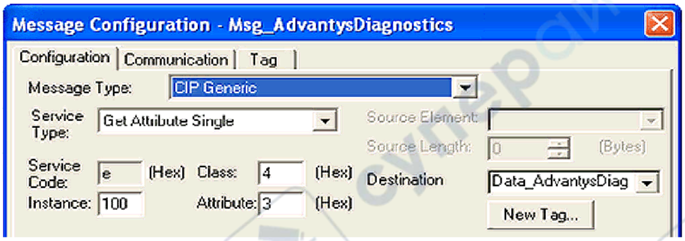
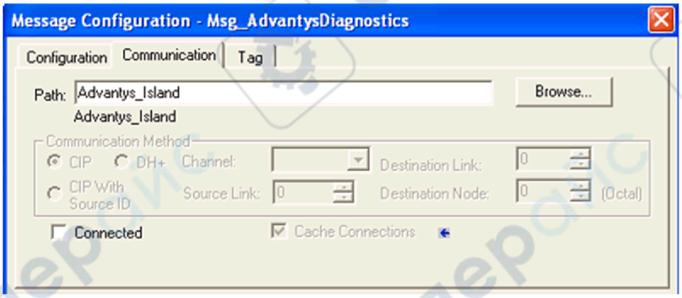
This completes the Class1 connection for the I/O data. Your connection becomes active when the ControlLogix process enters RUN mode. Next, you will add a Class3 connection (for explicit messages) to the RSLogix 5000 ladder logic.

Add an Explicit Message to RSLogix 5000 Ladder Logic (Class3 Connection)

Insert an Explicit Message Block

Establish an explicit Class3 connection between the Advantys island and the ControlLogix processor to poll diagnostic data:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Enter the timer, message, and SINT tags in the RSLogix 5000 controller tags table.
Notes:
- RSLogix 5000 requires this information to poll diagnostic data.
- The diagnostic tag type is a 20-element array of SINTs. (The tag types for the message (MESSAGE) and timer (TIMER) are obvious.)</td></tr>
<tr><td>2</td><td>The RSLogix 5000 controller tags table looks like this:

Scope: CompactLogix_v13(Show: Show All Sort: Tag Name
<table>
<tr><th>P</th><th>Tag Name</th><th>Alias For</th><th>Base Tag</th><th>Type</th><th>Style</th><th>Des</th></tr>
<tr><td></td><td>+ Advantys_Island:C</td><td></td><td></td><td>AB:ETHERNET_...</td><td></td><td></td></tr>
<tr><td></td><td>+ Advantys_Island:I</td><td></td><td></td><td>AB:ETHERNET_...</td><td></td><td></td></tr>
<tr><td></td><td>+ Advantys_Island:O</td><td></td><td></td><td>AB:ETHERNET_...</td><td></td><td></td></tr>
<tr><td></td><td>+ Data_AdvantysDiagnostics</td><td></td><td></td><td>SINT[20]</td><td>Decimal</td><td></td></tr>
<tr><td></td><td>+ Local:1:C</td><td></td><td></td><td>AB:1769_DO16:C:0</td><td></td><td></td></tr>
<tr><td></td><td>+ Local:1:I</td><td></td><td></td><td>AB:1769_DO16:I:0</td><td></td><td></td></tr>
<tr><td></td><td>+ Local:1:O</td><td></td><td></td><td>AB:1769_DO16:O:0</td><td></td><td></td></tr>
<tr><td></td><td>+ Local:2:I</td><td></td><td></td><td>AB:1769_DI16:I:0</td><td></td><td></td></tr>
<tr><td>☐</td><td>+ MessageTimer</td><td></td><td></td><td>TIMER</td><td></td><td></td></tr>
<tr><td></td><td>+ Msg_AdvantysDiagnostics</td><td></td><td></td><td>MESSAGE</td><td></td><td></td></tr>
<tr><td>☐</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td>3</td><td>After the tag assignment, the function blocks are added to the ladder logic:

200 ms timer for explicit messaging of Advantys Diagnostics and HMI data
Explicit message to read Advantys Diagnostics data
</td></tr>
</table>

Step	Action
4	Double-click on the button in the MSG function block to access the Message Configuration screen:  Message Configuration - Msg_AdvantysDiagnostics Message Type: CIP Generic Service Type: Get Attribute Single Source Element: Source Length: 0 (Bytes) Service Code: e (Hex) Class: 4 (Hex) Destination: Data_AdvantysDiag Instance: 100 Attribute: 3 (Hex)
5	Enter values for these fields: ● Message Type: Scroll to CIP Generic. ● Service Type: Scroll to Get Attribute Single. ● Service Code: Service code *e* is selected in conjunction with the service type Get Attribute Single. ● Class/Instance/Attribute: Enter the values in the figure (4, 100, 3). You can use the same class and attribute for HMI/PLC data by changing only the instance for HMI-to-PLC data (instance = 104) and PLC-to-HMI data (instance = 105). (Create separate data tags for each message.) ● Destination: Scroll to the tag you previously created for the diagnostic data.
6	On the Communication tab, select the name you gave the Advantys island as a Generic Ethernet Module and press OK:  Message Configuration - Msg_AdvantysDiagnostics Path: Advantys_Island Advantys_Island Communication Method ● CIP ○ DH+ Channel: Destination Link: 0 ○ CIP With Source ID Source Link: 0 Destination Node: 0 (Octal) ☐ Connected ☑ Cache Connections

You're Done

This concludes the connection example. You can now download the configuration to the CompactLogix processor and put it into RUN mode.

Section 7.2

Connecting an Advantys STB Island to a Quantum Master Using Unity Pro

Introduction

In this section, use Unity Pro to establish EtherNet/IP communications between the NOC 771 00 communications module in a Quantum PLC rack and an Advantys island with an STB NIC 2212 NIM.

What Is in This Section?

This section contains the following topics:

About this Connection Example

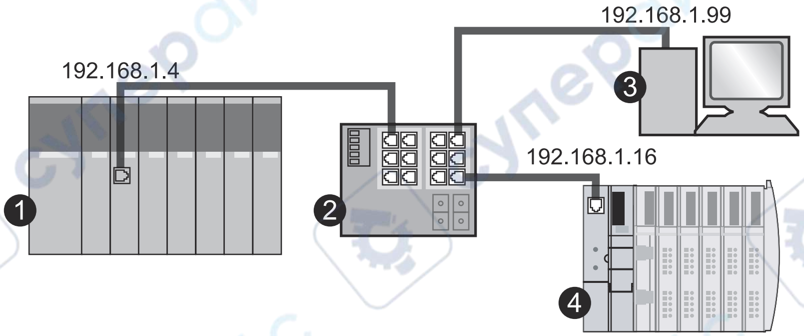
Introduction

This connection example tells you how to set up your Unity Pro project to establish communications between an NOC 771 00 communications module in a Quantum PLC rack and an Advantys STB island with an STB NIC 2212 NIM. There are instructions for configuring two connections:

- Class1 connection: This connection is for transmitting and receiving I/O data.
- Class3 connection: The connection is for explicit messages that are initiated by a PLC program. (A Class3 connection is also used for collecting diagnostics data in PLC-to-HMI and HMI-to-PLC communications.)

Connection Figure

The following figure shows the EtherNet/IP connections for this example:



1 Quantum PLC with an EtherNet/IP interface module (IP address 192.168.1.4)
2 Ethernet managed switch
3 PC with Ethernet card, Advantys Configuration Software, and Unity Pro configuration software (IP address 192.168.1.99)
4 Advantys island with STB NIC 2212 EtherNet/IP NIM (IP address 192.168.1.16)

This sample network implements connections to a switch through shielded (STP), twisted-pair Cat 5 cables, as required for CE compliance.

NOTE:

- A direct connection between the PC host (with an Ethernet card) and the STB NIC 2212 NIM can be made through a Cat 5 crossover cable.
- Compatible switch, hub, connector, and cable selections are described in the *Transparent Factory Network Design and Cabling Guide*.

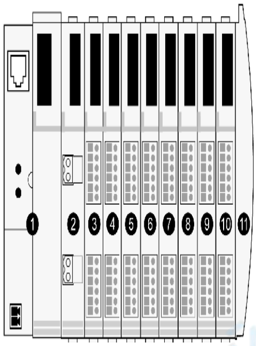
Before You Begin

Here are some characteristics of this example:

- Communication between the network and the Advantys island is through an STB NIC 2212 NIM.
- A Quantum CPU 534 14A implemented as the master.
- The Quantum CPU is upgraded to Unity 2.50 firmware.
- An NOC 771 00 EtherNet/IP communication module sends messages between the Quantum CPU and the network.
- Unity Pro version 3.1.x.x is used.
- This example uses a simple Advantys island that has been auto-configured by pressing the RST button. Therefore, the example island does not contain enhanced CANopen modules, such as ATV motor controllers or third-party CANopen devices. (You can use enhanced modules if you configured the island from within the Advantys Configuration Software before downloading to configuration to the NIM.)

Sample Island

This sample island is used in the connection example. It is a representative island bus assembly with an STB NIC 2212 NIM module as the EtherNet/IP network adapter:



1 STB NIC 2212: network interface module
2 STB PDT 3100: 24 VDC power distribution module (PDM)
3 STB DDI 3230: 24 VDC 2-channel digital input module (2 bits data, 2 bits status)
4 STB DDO 3200: 24 VDC 2-channel digital output module (2 bits data, 2 bits of echo output data, 2 bits status)
5 STB DDI 3420: 24 VDC 4-channel digital input module (4 bits data, 4 bits status)
6 STB DDO 3410: 24 VDC 4-channel digital output module (4 bits data, 4 bits of echo output data, 4 bits status)
7 STB DDI 3610: 24 VDC 6-channel digital input module (6 bits data, 6 bits status)
8 STB DDO 3600: 24 VDC 6-channel digital output module (6 bits data, 6 bits of echo output data, 6 bits status)
9 STB AVI 1270: +/-10 VDC 2-channel analog input module (16 bits data–channel 1, 16 bits data–channel 2, 8 bits status–channel 1, 8 bits status–channel 2)
10 STB AVO 1250: +/-10 VDC 2-channel analog output module (16 bits data–channel 1, 16 bits data–channel 2, 8 bits status–channel 1, 8 bits status–channel 2)
11 STB XMP 1100: island bus termination plate

The I/O modules in the sample assembly have the following island bus addresses:

I/O Model	Module Type	Module’s Island Bus Address
STB DDI 3230	2-channel digital input	1
STB DDO 3200	2-channel digital output	2
STB DDI 3420	4-channel digital input	3
STB DDO 3410	4-channel digital output	4
STB DDI 3610	6-channel digital input	5
STB DDO 3600	7-channel digital output	6
STB AVI 1270	2-channel analog input	7
STB AVO 1250	2-channel analog output	8

The PDM and the termination plate are not addressable *(see page 44)*.

Determine I/O Data Block Sizes

Objective

You must place the size of the island's I/O data image in your Unity Pro project to configure the NOC 771 00 to send and receive I/O data. You can also use an island configuration with a known I/O data image size. You could also construct the island within the ACS to generate the I/O data image or connect to the ACS to the island via the Ethernet port after the NIM gets an IP address.

The procedure uses a serial method to avoid having to download a configuration to the processor twice.

NOTE: Make sure your island is fully assembled before you begin this procedure.

About the Device Name

In this example, the DHCP assignment of the IP address is based on the NIM device name. There are three methods of assigning a device name:
- direct assignment on the NIM's web pages
- assignment through the rotary switch setting
- assignment through the numeric values on the rotary switches

This example uses the numeric settings on the rotary switches

The device name includes:
- the NIM part name (STBNIC2212)
- an underscore (_)
- the value represented by the selections on the rotary switches.

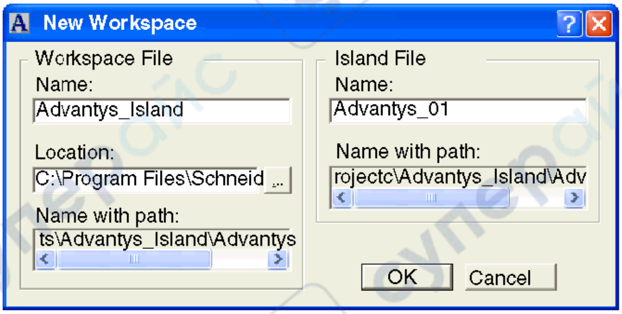
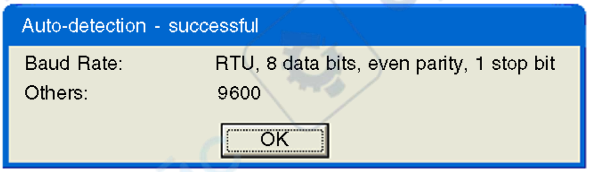
Assign a Device Name

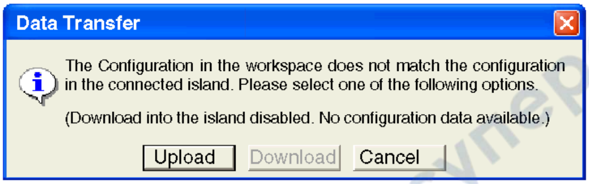
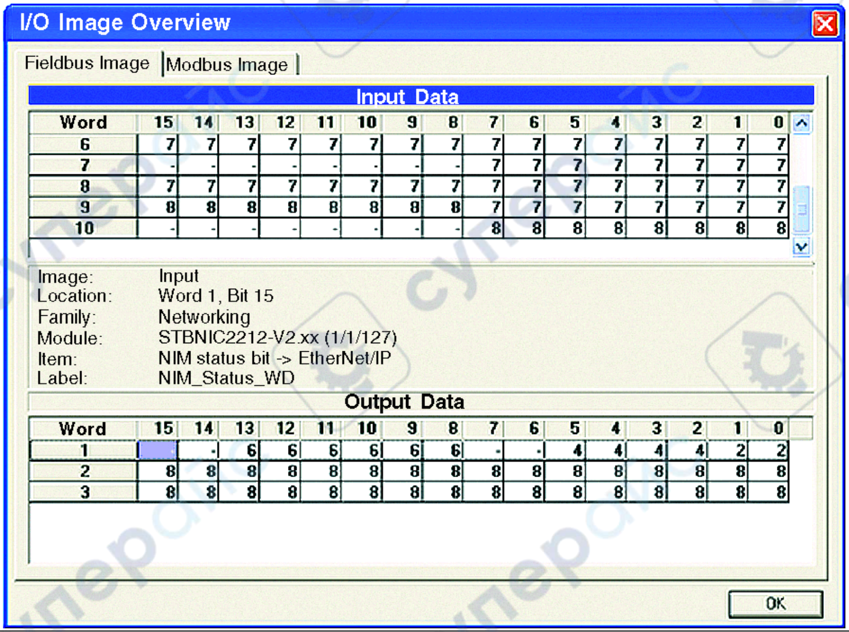
Create a device name for this example:

Step	Action
1	Set the TENS switch to 1.
2	Set the ONES switch to 6. **Note**: The two switches represent a combined value of 16 (016).
3	Power up the NIM. **Result**: The NIM now starts sending continuous DHCP discovery requests with the device name STBNIC2212_016 until it receives a reply.
4	Connect the Advantys Configuration Software serial cable between the NIM and the PC.

Determine Data Size

Use the Advantys Configuration Software to determine the size of the I/O process image in bytes, information that is required to configure the island within Unity Pro:

Step	Action
1	Press and hold the RST button for at least 2 seconds to properly configure the island.
2	Start the Advantys Configuration Software (ACS).
3	At the first prompt, select the STB icon. **Result**: The island development environment appears.
4	In the File menu, scroll to New Workspace. **Result**: The New Workspace screen appears.
5	Enter names for the new Workspace File and the Island File and press OK:  New Workspace Workspace File Name: Advantys_Island Location: C:\Program Files\Schneid Name with path: ts\Advantys_Island\Advantys Island File Name: Advantys_01 Name with path: rojectc\Advantys_Island\Adv **Result**:The island rail in the New workspace is empty.
6	In the Online menu, scroll to Connection Settings. **Note**: You are starting to establish a connection between the ACS and the island that allows you to populate the rail with island modules.
7	On the Connection settings screen, set the Connection Type to Serial and press the Settings button. **Result**: The Serial Parameters screen appears.
8	Press Auto Detection. **Result**: Through auto detection, the ACS tries to establish communications to the NIM via the serial cable. The ACS tries various combinations of connection parameters beginning with the NIM default parameters of 9600 baud, RTU, 8 data bits, 1 stop bit, and even parity.
9	Press OK in the Auto-detection – successful screen, and continue to press OK in response to subsequent screens:  Auto-detection - successful Baud Rate: RTU, 8 data bits, even parity, 1 stop bit Others: 9600 **Result**: The Auto-detection - successful screen appears when the ACS achieves communications to the NIM.
10	In the Online menu, scroll to Connect. **Result**: The Data Transfer screen appears.

Step	Action
11	Press Upload:  **Data Transfer** The Configuration in the workspace does not match the configuration in the connected island. Please select one of the following options. (Download into the island disabled. No configuration data available.) **Result**: This action uploads the island configuration to the workspace.
12	In the Island menu, select I/O Image Overview. When the screen appears, select the Fieldbus Image tab: 

I/O Image Overview

Input Data

Word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
6	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	-	-	-	-	-	-	-	-	7	7	7	7	7	7	7	7
8	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
9	8	8	8	8	8	8	8	8	7	7	7	7	7	7	7	7
10	-	-	-	-	-	-	-	-	8	8	8	8	8	8	8	8

Image: Input
Location: Word 1, Bit 15
Family: Networking
Module: STBNIC2212-V2.xx (1/1/127)
Item: NIM status bit -> EtherNet/IP
Label: NIM_Status_WD

Output Data

Word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1		-	6	6	6	6	6	6	-	-	4	4	4	4	2	2
2	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
3	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8

13	Scroll to the end of the Input and Output Data tables to see the total number of Input Data and Output Data words. **Notes**: ● The ACS displays data in 16-bit words that must be converted into 8-bit bytes (down to the nearest byte) for use in the Unity Pro project. ● In the example I/O Image Overview screen, the Input Data displaces 10 words, but only 19 bytes. The last byte is not used. Therefore, enter 19 bytes for the Input Assembly Instance in the Unity Pro project. The Output Data displaces 3 full words, so enter 6 bytes for the Output Assembly Instance.

Next...

Continue to the next topic to configure a class 1 connection for I/O data.

Configuring Unity Pro to Use Advantys Island I/O Data (Class1 Connection)

Objective

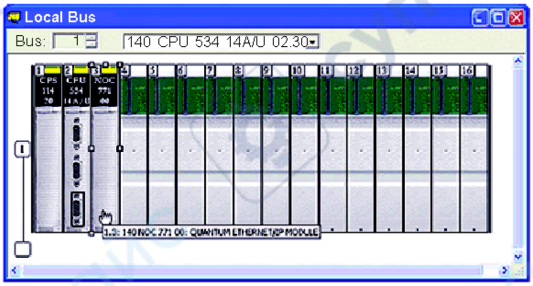
In previous procedures, you gathered the I/O data image sizes with the Advantys Configuration Software and created a DHCP device name to configure the Advantys island within Unity Pro. This procedure uses that information to facilitate the transmission and reception of I/O data between the Advantys island and the NOC 771 00 communications module in the Quantum PLC rack.

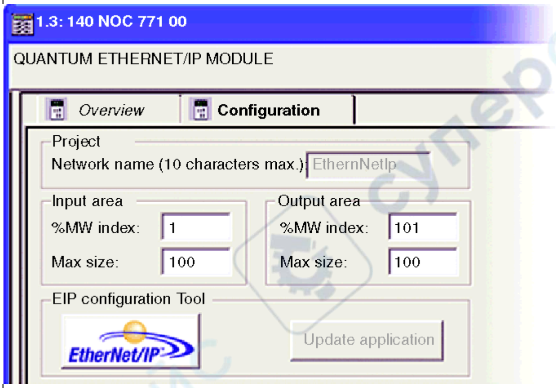
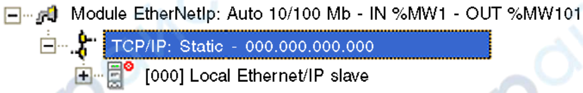
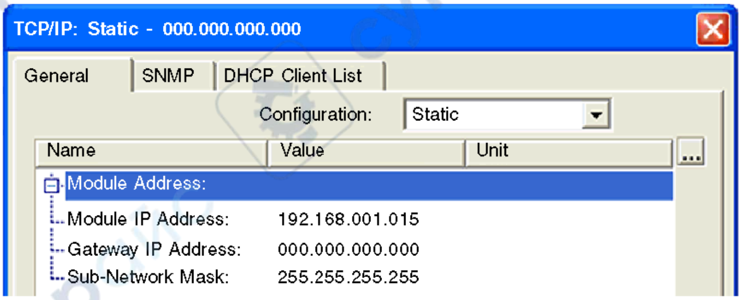
To make this Class1 connection, perform these procedures in this order:

- Assign an IP address to the NOC 771 00 communications module.
- Assign an IP address to the Advantys NIM through DHCP using a device name derived from the numeric values set on the rotary switches.
- Configure the NOC 771 00 to send and receive I/O data within a Unity Pro project.

NOC 771 00 IP Address Assignment

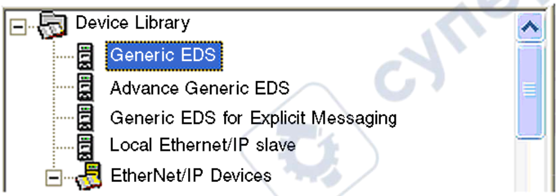
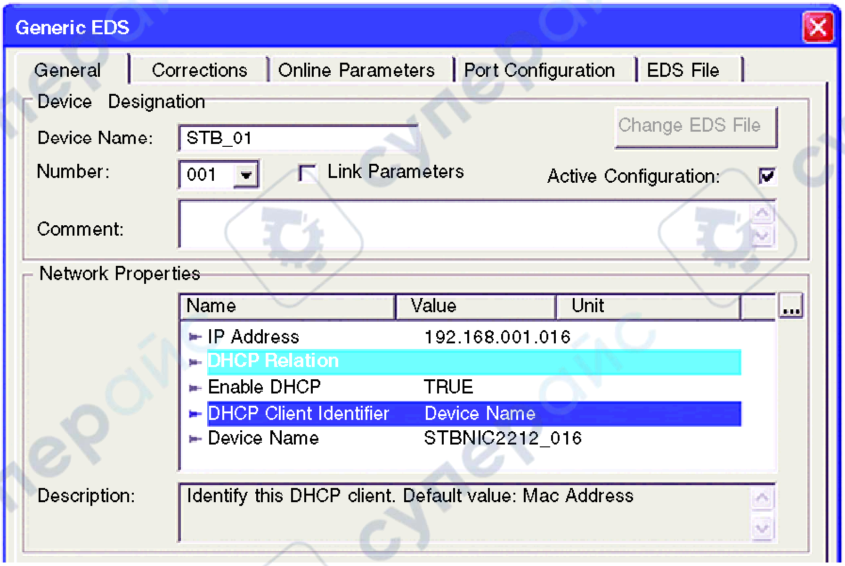
Assign an IP address to the NOC 771 00 communications module:

Step	Action
1	In the Unity project, double-click on the NOC 771 00 module:  **Result:** The Quantum EtherNet/IP Module Configuration screen appears.

Step	Action
2	On this screen...  1.3: 140 NOC 771 00 QUANTUM ETHERNET/IP MODULE Overview | Configuration Project Network name (10 characters max.): EthernNetIp Input area %MW index: 1 Max size: 100 Output area %MW index: 101 Max size: 100 EIP configuration Tool EtherNet/IP Update application enter... ● a Network Name for the NOC 771 00 ● the %MW memory locations for the I/O ● the memory allotment size **Note:** In this example we start at %MW1 for inputs and %MW101 for outputs, each having 100 words available. When you are done, validate your entries with the checkbox tool button.
3	Click on the EIP configuration Tool button. **Note:** This starts a separate program for configuring the network portion of the NOC 771 00 and your Advantys island. An information box appears stating the Unity configuration has been updated.
4	Click OK in the main workspace and double-click on the NOC 771 00 TCP/IP address entry:  Module EtherNetIp: Auto 10/100 Mb - IN %MW1 - OUT %MW101 TCP/IP: Static - 000.000.000.000 [000] Local Ethernet/IP slave
5	When the General tab in the NOC 771 00 module properties screen appears, populate the Module IP Address field with the module's IP address and press OK:  TCP/IP: Static - 000.000.000.000 General | SNMP | DHCP Client List Configuration: Static Name | Value | Unit Module Address: Module IP Address: 192.168.001.015 Gateway IP Address: 000.000.000.000 Sub-Network Mask: 255.255.255.255 **Note:** In this example the IP address of the NOC 771 00 is static, enter the values, then press OK.

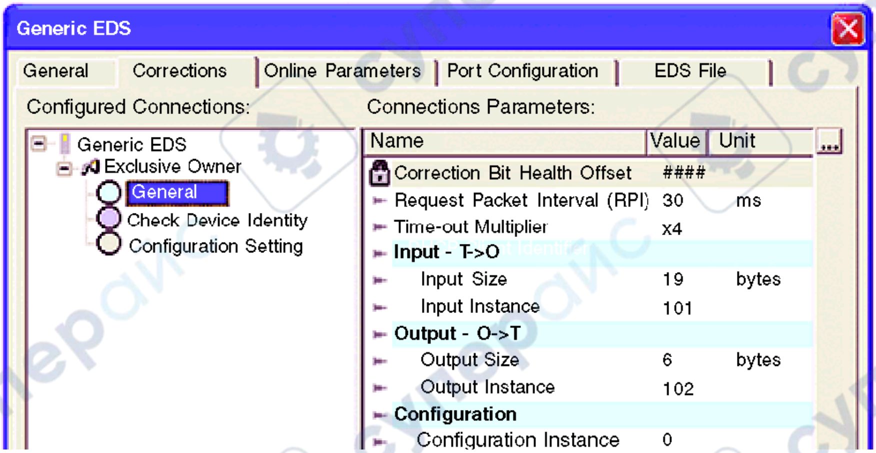
Assign the NIM's IP Address

Assign an IP address to the Advantys NIM through DHCP using a device name derived from the numeric values set on the rotary switches:

Step	Action
1	Drag the Generic EDS from the device library to the main workspace:  **Result:** The General tab on the Generic EDS properties screen appears.
2	Select the General tab:  Device Name: STB_01 Number: 001 ☐ Link Parameters Active Configuration: ☑ Comment: Name / Value / Unit IP Address: 192.168.001.016 DHCP Relation Enable DHCP: TRUE DHCP Client Identifier: Device Name Device Name: STBNIC2212_016 Description: Identify this DHCP client. Default value: Mac Address
3	Enter the device name of your choice in the Device Name field of the Device Designation area. Then make these selections (in this order) on the Network Properties area: ● IP address: In the Network Properties table, double-click on IP address. ● Enable DHCP: In the Network Properties table, double-click on Enable DHCP and select TRUE to enable DHCP. ● DHCP Client Identifier: In the Network Properties table, double-click on DHCP Client Identifier and select Device Name. This tells the NOC 771 00 to assign an address based on the NIM’s device name. ● Device Name: In the Network Properties table, double-click on Device Name and enter STBNIC2212_ xxx, where xxx are the three digits that correspond to the rotary switch positions on the NIM. In this case, we use STBNIC2212_016 to correspond to the last octet of the IP address. In this example, the NOC 771 00 acts as a DHCP server. It assigns a predefined address according to the DHCP device name discovery request from the NIM.

NOC 771 00 in Unity Pro

Use this procedure to configure the NOC 771 00 to send and receive I/O data within a Unity Pro project:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Select the Connections tab:


Generic EDS

General | Corrections | Online Parameters | Port Configuration | EDS File

Configured Connections:

Generic EDS

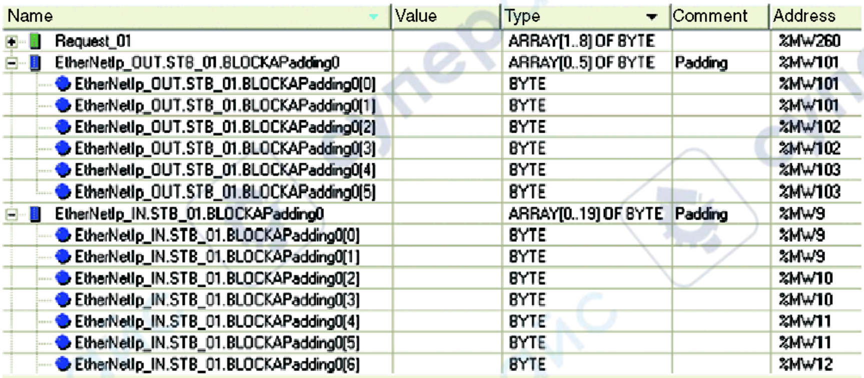
Exclusive Owner

General

Check Device Identity

Configuration Setting

Connections Parameters:
<table>
<tr><th>Name</th><th>Value</th><th>Unit</th></tr>
<tr><td>Correction Bit Health Offset</td><td>####</td><td></td></tr>
<tr><td>Request Packet Interval (RPI)</td><td>30</td><td>ms</td></tr>
<tr><td>Time-out Multiplier</td><td>x4</td><td></td></tr>
<tr><td>Input - T->O</td><td></td><td></td></tr>
<tr><td>Input Size</td><td>19</td><td>bytes</td></tr>
<tr><td>Input Instance</td><td>101</td><td></td></tr>
<tr><td>Output - O->T</td><td></td><td></td></tr>
<tr><td>Output Size</td><td>6</td><td>bytes</td></tr>
<tr><td>Output Instance</td><td>102</td><td></td></tr>
<tr><td>Configuration</td><td></td><td></td></tr>
<tr><td>Configuration Instance</td><td>0</td><td></td></tr>
</table>
</td></tr>
<tr><td>2</td><td>Make these selections:
● Input Size: Double-click on Input Size and insert 19 bytes, the input data block size that you calculated in the ACS I/O image overview.
● Input Instance: Double-click on Input Instance and enter 101.
● Output Size: Double-click on Output Size and insert 6 bytes, the output data block size that you calculated in the ACS I/O image overview.
● Output Instance: Double-click on Output Instance and enter 102.
Note: To establish a Listen Only connection to the island, set the Output Instance value to 255 and the Output Size value to 0.</td></tr>
<tr><td>3</td><td>Click OK.
Result: This closes the Generic EDS screen and brings you back to the main workspace.</td></tr>
<tr><td>4</td><td>From the File menu, select Save, then Exit.
Result:: This closes the EIP Configuration tool screen and brings you back to Unity Pro’s EtherNet/IP module configuration screen.</td></tr>
<tr><td>5</td><td>Press Update application.
After the Unity Pro update, the button is greyed out. Then you can do one of the following:
● Build and download the Unity Pro project and continue to the next step.
● Configure a master block to poll NIM diagnostic data in the Configure Function Blocks procedure.</td></tr>
</table>

Step	Action
6	Open an existing animation table or start a new one to write to the outputs and view the inputs of the Advantys island: 

Name	Value	Type	Comment	Address
Request_01		ARRAY[1..8] OF BYTE		%MW260
EtherNetIp_OUT.STB_01.BLOCKAPadding0		ARRAY[0..5] OF BYTE	Padding	%MW101
EtherNetIp_OUT.STB_01.BLOCKAPadding0[0]		BYTE		%MW101
EtherNetIp_OUT.STB_01.BLOCKAPadding0[1]		BYTE		%MW101
EtherNetIp_OUT.STB_01.BLOCKAPadding0[2]		BYTE		%MW102
EtherNetIp_OUT.STB_01.BLOCKAPadding0[3]		BYTE		%MW102
EtherNetIp_OUT.STB_01.BLOCKAPadding0[4]		BYTE		%MW103
EtherNetIp_OUT.STB_01.BLOCKAPadding0[5]		BYTE		%MW103
EtherNetIp_IN.STB_01.BLOCKAPadding0		ARRAY[0..19] OF BYTE	Padding	%MW9
EtherNetIp_IN.STB_01.BLOCKAPadding0[0]		BYTE		%MW9
EtherNetIp_IN.STB_01.BLOCKAPadding0[1]		BYTE		%MW9
EtherNetIp_IN.STB_01.BLOCKAPadding0[2]		BYTE		%MW10
EtherNetIp_IN.STB_01.BLOCKAPadding0[3]		BYTE		%MW10
EtherNetIp_IN.STB_01.BLOCKAPadding0[4]		BYTE		%MW11
EtherNetIp_IN.STB_01.BLOCKAPadding0[5]		BYTE		%MW11
EtherNetIp_IN.STB_01.BLOCKAPadding0[6]		BYTE		%MW12

MBP_MSTR Configuration for Explicit Messages (Class3 Connection)

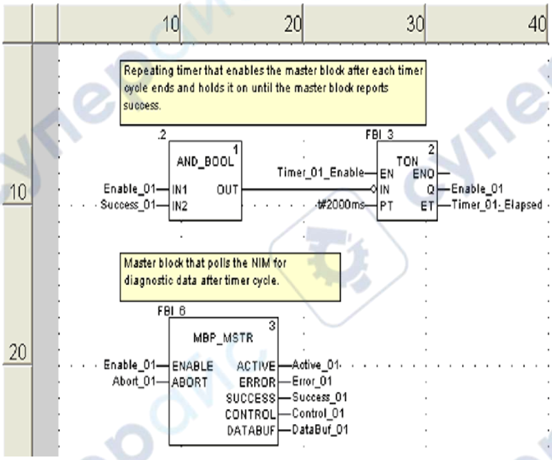
Objective

The following is an example of MBP_MSTR function block configuration for explicit messaging. Explicit messages are generally used to poll non-I/O data from the NIM. Each block of transmitted data requires its own MBP_MSTR function block. (This same procedure applies to PLC-to-HMI and HMI-to-PLC data blocks.)

Insert and Configure Function Blocks

Insert and configure the function blocks to facilitate the Class3 connection:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Insert a master (MBP_MSTR) function block and a self-resetting section timer (TON) into a programming section:

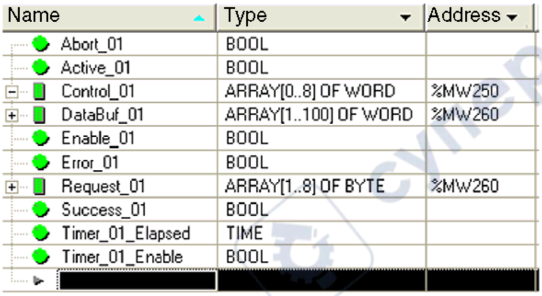


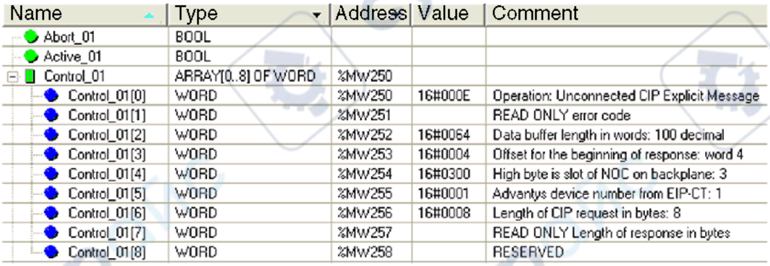
Repeating timer that enables the master block after each timer cycle ends and holds it on until the master block reports success.

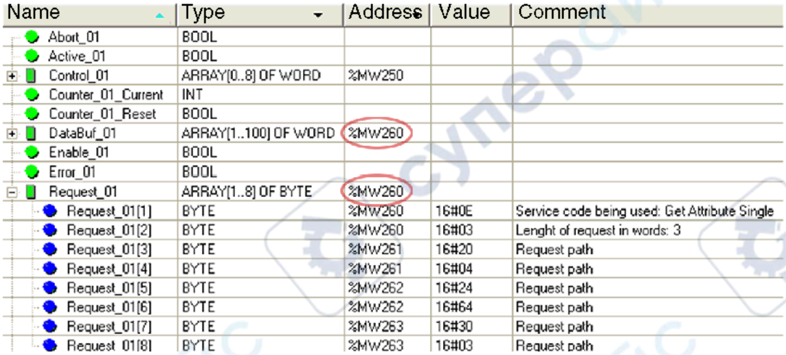
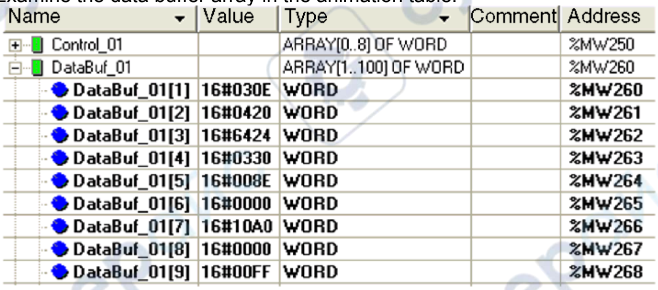
Master block that polls the NIM for diagnostic data after timer cycle.

Function block logic:

- TON: Explicit messaging begins when Timer_01_Enable is set to TRUE (1) and the first timer cycle has ended.
- MBP_MSTR: When the master block is successfully completed, the Enable_01 and Success_01 bits reset the timer to begin again. Enable_01 drops to FALSE (0), repeating the cycle.
</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>Assign variables in the Elementary Variables table:

<table>
<tr><th>Name</th><th>Type</th><th>Address</th></tr>
<tr><td>Abort_01</td><td>BOOL</td><td></td></tr>
<tr><td>Active_01</td><td>BOOL</td><td></td></tr>
<tr><td>Control_01</td><td>ARRAY[0..8] OF WORD</td><td>%MW250</td></tr>
<tr><td>DataBuf_01</td><td>ARRAY[1..100] OF WORD</td><td>%MW260</td></tr>
<tr><td>Enable_01</td><td>BOOL</td><td></td></tr>
<tr><td>Error_01</td><td>BOOL</td><td></td></tr>
<tr><td>Request_01</td><td>ARRAY[1..8] OF BYTE</td><td>%MW260</td></tr>
<tr><td>Success_01</td><td>BOOL</td><td></td></tr>
<tr><td>Timer_01_Elapsed</td><td>TIME</td><td></td></tr>
<tr><td>Timer_01_Enable</td><td>BOOL</td><td></td></tr>
</table>
Observe these function block pin definition formats:

These pins are boolean: Enable_01, Abort_01, Active_01, Error_01, Success_01, Timer_01_Enable
These pins are word arrays of 10 and 100 elements, respectively: Control_01, DataBuf_01
Request parameters are defined in bytes. Therefore, instead of entering parameters directly into the DataBuf_01 word array, you may want to enter request parameters in the Request_01 byte array and let Unity Pro combine the parameters in words. (That’s why the DataBuf_01 and Request_01 arrays point to the same memory location, %MW260.)
</td></tr>
<tr><td>3</td><td>Within the Data Editor, populate the control and request arrays:

<table>
<tr><th>Name</th><th>Type</th><th>Address</th><th>Value</th><th>Comment</th></tr>
<tr><td>Abort_01</td><td>BOOL</td><td></td><td></td><td></td></tr>
<tr><td>Active_01</td><td>BOOL</td><td></td><td></td><td></td></tr>
<tr><td>Control_01</td><td>ARRAY[0..8] OF WORD</td><td>%MW250</td><td></td><td></td></tr>
<tr><td>Control_01[0]</td><td>WORD</td><td>%MW250</td><td>16#000E</td><td>Operation: Unconnected CIP Explicit Message</td></tr>
<tr><td>Control_01[1]</td><td>WORD</td><td>%MW251</td><td></td><td>READ ONLY error code</td></tr>
<tr><td>Control_01[2]</td><td>WORD</td><td>%MW252</td><td>16#0064</td><td>Data buffer length in words: 100 decimal</td></tr>
<tr><td>Control_01[3]</td><td>WORD</td><td>%MW253</td><td>16#0004</td><td>Offset for the beginning of response: word 4</td></tr>
<tr><td>Control_01[4]</td><td>WORD</td><td>%MW254</td><td>16#0300</td><td>High byte is slot of NOC on backplane: 3</td></tr>
<tr><td>Control_01[5]</td><td>WORD</td><td>%MW255</td><td>16#0001</td><td>Advantys device number from EIP-CT: 1</td></tr>
<tr><td>Control_01[6]</td><td>WORD</td><td>%MW256</td><td>16#0008</td><td>Length of CIP request in bytes: 8</td></tr>
<tr><td>Control_01[7]</td><td>WORD</td><td>%MW257</td><td></td><td>READ ONLY Length of response in bytes</td></tr>
<tr><td>Control_01[8]</td><td>WORD</td><td>%MW258</td><td></td><td>RESERVED</td></tr>
</table>
Note: The read-only and reserved elements are left blank.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>Examine the Data Editor and notice that the Request_01 array is in the same memory location (circled) as the first four words of the DataBuf_01 array. The Request_01 array is populated with these values:

<table>
<tr><th>Name</th><th>Type</th><th>Address</th><th>Value</th><th>Comment</th></tr>
<tr><td>Abort_01</td><td>BOOL</td><td></td><td></td><td></td></tr>
<tr><td>Active_01</td><td>BOOL</td><td></td><td></td><td></td></tr>
<tr><td>Control_01</td><td>ARRAY[0..8] OF WORD</td><td>%MW250</td><td></td><td></td></tr>
<tr><td>Counter_01_Current</td><td>INT</td><td></td><td></td><td></td></tr>
<tr><td>Counter_01_Reset</td><td>BOOL</td><td></td><td></td><td></td></tr>
<tr><td>DataBuf_01</td><td>ARRAY[1..100] OF WORD</td><td>%MW260</td><td></td><td></td></tr>
<tr><td>Enable_01</td><td>BOOL</td><td></td><td></td><td></td></tr>
<tr><td>Error_01</td><td>BOOL</td><td></td><td></td><td></td></tr>
<tr><td>Request_01</td><td>ARRAY[1..8] OF BYTE</td><td>%MW260</td><td></td><td></td></tr>
<tr><td>Request_01[1]</td><td>BYTE</td><td>%MW260</td><td>16#0E</td><td>Service code being used: Get Attribute Single</td></tr>
<tr><td>Request_01[2]</td><td>BYTE</td><td>%MW260</td><td>16#03</td><td>Lenght of request in words: 3</td></tr>
<tr><td>Request_01[3]</td><td>BYTE</td><td>%MW261</td><td>16#20</td><td>Request path</td></tr>
<tr><td>Request_01[4]</td><td>BYTE</td><td>%MW261</td><td>16#04</td><td>Request path</td></tr>
<tr><td>Request_01[5]</td><td>BYTE</td><td>%MW262</td><td>16#24</td><td>Request path</td></tr>
<tr><td>Request_01[6]</td><td>BYTE</td><td>%MW262</td><td>16#64</td><td>Request path</td></tr>
<tr><td>Request_01[7]</td><td>BYTE</td><td>%MW263</td><td>16#30</td><td>Request path</td></tr>
<tr><td>Request_01[8]</td><td>BYTE</td><td>%MW263</td><td>16#03</td><td>Request path</td></tr>
</table></td></tr>
<tr><td>5</td><td>Download the project to the Quantum processor.
Note: The DHCP server in the NOC 771 00 assigns an IP address to the NIM, which has been awaiting a response to its DHCP discovery request.</td></tr>
<tr><td>6</td><td>Start the timer in the section.
Note: When the timer reaches its preset value, the master block (MBP_MSTR) starts and sends a request, populating the data buffer (DataBuf_01) with NIM diagnostic data.</td></tr>
<tr><td>7</td><td>Examine the data buffer array in the animation table.

<table>
<tr><th>Name</th><th>Value</th><th>Type</th><th>Comment</th><th>Address</th></tr>
<tr><td>Control_01</td><td></td><td>ARRAY[0..8] OF WORD</td><td></td><td>%MW250</td></tr>
<tr><td>DataBuf_01</td><td></td><td>ARRAY[1..100] OF WORD</td><td></td><td>%MW260</td></tr>
<tr><td>DataBuf_01[1]</td><td>16#030E</td><td>WORD</td><td></td><td>%MW260</td></tr>
<tr><td>DataBuf_01[2]</td><td>16#0420</td><td>WORD</td><td></td><td>%MW261</td></tr>
<tr><td>DataBuf_01[3]</td><td>16#6424</td><td>WORD</td><td></td><td>%MW262</td></tr>
<tr><td>DataBuf_01[4]</td><td>16#0330</td><td>WORD</td><td></td><td>%MW263</td></tr>
<tr><td>DataBuf_01[5]</td><td>16#008E</td><td>WORD</td><td></td><td>%MW264</td></tr>
<tr><td>DataBuf_01[6]</td><td>16#0000</td><td>WORD</td><td></td><td>%MW265</td></tr>
<tr><td>DataBuf_01[7]</td><td>16#10A0</td><td>WORD</td><td></td><td>%MW266</td></tr>
<tr><td>DataBuf_01[8]</td><td>16#0000</td><td>WORD</td><td></td><td>%MW267</td></tr>
<tr><td>DataBuf_01[9]</td><td>16#00FF</td><td>WORD</td><td></td><td>%MW268</td></tr>
</table>
Note: The requested data begins at element 7 of DataBuf_01. (The first four elements contain the CIP request that you assigned in the Request_01 array, and elements 5 and 6 contain the header of the CIP response.) The value in element 7 (10A0) reports a healthy NIM. This table shows that the Request_01 data in the DataBuf_01 array is arranged in the "little endian" format, where the least significant byte is stored in the smallest memory address. That is, 0E in element 1 is in the lower byte location of word DataBuf_01[1] and 03 in element 2 is the upper byte.</td></tr>
</table>

You’re Done

If you’ve followed the configuration steps in this section, these communications are possible:

- The Quantum processor can transfer I/O data to and from the Advantys island. (The NOC 771 00 and STB NIC 2212 NIM communicate over a Class1 connection.)
- The Quantum processor can poll diagnostic data according the section timer’s pre-set time through a Class3 connection. (Additional MBP_MSTR function blocks can be used to send PLC-to-HMI data and receive HMI-to-PLC data.)

Chapter 8

Advanced Configuration Features

Introduction

This chapter describes the advanced and/or optional configuration features that you can add to an Advantys STB island.

What Is in This Chapter?

This chapter contains the following topics:

Configurable Parameters for the STB NIC 2212

Introduction

The following information describes how to the configure parameters for the STB NIC 2212 using the Advantys Configuration Software.

Users can configure these operating parameters:

- data size (in words) of PLC output data transmitted to the HMI panel and HMI input data sent to the PLC
- maximum node ID for the last module assembled on the island bus, including CANopen devices

General Information

For general information about the NIM module (model name, version number, vendor code, etc.), do the following:

Step	Action	Comment
1	Open your island configuration with the Advantys Configuration Software.	The STB NIC 2212 is the leftmost module in your island bus assembly.
2	Double-click on the NIM in the island editor.	The *module editor* window appears.
3	Select the *General* tab.	General information about the STB NIC 2212 is available from this tab.

Accessing Configurable Parameters

Configure HMI/PLC data, enable RTP, and change the maximum node ID in the *module editor* window:

Step	Action	Comment
1	Double-click on the STB NIC 2212 in the island editor.	The *module editor* window appears. It has two tabs for changing parameters, *Parameters* and *Options*.
2	Select the *Parameters* tab.	The *Parameters* tab includes configuration parameters for HMI/PLC data.
3	In the *Parameter name* column, expand the *Additional Info Store List* by clicking on the plus (+) sign.	When the column is expanded, the configurable parameters for HMI/PLC data appear.
4	Select the *Options* tab.	The *Options* tab includes configuration parameters that enable RTP and change the maximum node ID.

Selecting the Display Format

By default, the values for the configurable NIM parameters use decimal notation. You can change the display format to hexadecimal notation, and vice-versa:

Step	Action	Comment
1	Double-click the NIM in the island editor.	The *module editor* window appears.
2	Select the *Parameters* tab.	—
3	Click on the checkbox in front of *Hexadecimal* at the top right of the module editor window.	The display values for the configurable parameters are in hexadecimal notation.
Note: To return to the default decimal notation, click the checkbox to disable hexadecimal notation.		

Reserved Sizes (HMI and PLC)

HMI-to-PLC: The network interprets data from the HMI as input and reads it from the input data table in the process image. This table is shared with data from all input modules on the island bus. When the reserved size (HMI to PLC) is selected, the range of available data sizes (in words) is displayed. Space that you reserve for HMI to PLC data must not exceed the maximum value shown (248 words).

PLC-to-HMI: The network transmits data to the HMI as output by writing it to the output data table in the process image. This table is shared with data for all output modules on the island bus. When the reserved size (PLC to HMI) is selected, the range of available data sizes (in words) is displayed. Space that you reserve for the PLC to HMI data must not exceed the maximum value shown (248 words).

Data Transfer: To transfer data to the PLC from a Modbus HMI panel attached to the CFG port:, you must reserve space for that data:

Step	Action	Result
1	In the *module editor* window, select the *Parameters* tab.	—
2	In the *Parameter name* column, expand the *Additional Info Store List* by clicking on the plus (+) sign.	The configurable NIM parameters are displayed.
3	Double-click in the *Value* column next to the *Reserved Size (Words) of HMI to PLC table*.	The value is highlighted.
4	Type a value for the data size to be reserved for data sent from the HMI panel to the PLC.	The value *plus* the data size of your island cannot exceed the maximum value. If you accept the default value (0), no space is reserved in the HMI table in the process image.
5	Repeat steps 2-4 to select a value for the *Reserved Size (Words) of PLC to HMI table* row.	—
6	Click on the *OK* button to save your work.	—
7	Click on the *Apply* button to configure the NIM with these values.	—

CANopen Device Node IDs

From the *Options* tab, you can set the maximum node ID of the last module on the island bus. The last module may be a standard CANopen device. Standard CANopen devices follow the last segment of STB I/O modules. CANopen modules are addressed by counting backwards from the value that you specify here. The ideal node ID sequence is sequential.

For example, if you have an island with five STB I/O modules and three CANopen devices, a maximum node ID of at least 8 (5 + 3) is required. This results in node IDs of 1 through 5 for STB I/O modules and 6 through 8 for standard CANopen devices. Using the default ID of 32 (the maximum number of modules the island can support) results in node IDs of 1 through 5 for STB I/O modules and 30 through 32 for standard CANopen devices. Unless required, high addresses are not desirable if one (or more) of your standard CANopen devices has a limited address range.

Assigning the Max. Node ID (CANopen Devices)

To assign the highest node ID used by a CANopen device on the island bus:

Step	Action	Comment
1	In the *module editor* window, select the *Options* tab.	Configurable parameters are located under this tab.
2	In the box next to *Max. node ID on the CANopen extension*, enter a node ID.	This node ID represents the last CANopen module on the island bus.

Configuring Mandatory Modules

Summary

As part of a custom configuration, you can assign *mandatory* status to any I/O module or preferred device on an island. The mandatory designation indicates that the module or device should operate in your application. If the NIM does not detect a healthy mandatory module at its assigned address during normal operations, the NIM stops the entire island.

NOTE: The Advantys Configuration Software is required if you want to designate an I/O module or a preferred device as a mandatory module.

Specifying Mandatory Modules

By default, the Advantys STB I/O modules are in a non-mandatory (*standard*) state. Mandatory status is enabled by clicking on the mandatory checkbox on a module or preferred device's **Options** tab. Depending on your application, any number of modules that your island supports can be designated as mandatory modules.

Effects on Island Bus Operations

The following table describes the conditions under which mandatory modules affect island bus operations and the NIM's response:

Condition	Response
A mandatory module is not operating during normal island bus operations.	The NIM stops the island bus. The island enters fallback mode *(see page 160)*. I/O modules and preferred devices assume their fallback values.
You attempt to hot swap a mandatory module.	The NIM stops the island bus. The island enters fallback mode. I/O modules and preferred devices assume their fallback values.
You are hot swapping a standard I/O module that resides to the left of a mandatory module on the island bus, and the island loses power.	When power is restored, the NIM attempts to address the island modules but should stop at the empty slot where the standard module used to reside. Because the NIM is now unable to address the mandatory module, it generates a mandatory mismatch condition. The island does not start when this condition is present.

Recovering from a Mandatory Stop

NOTE:

Pushing the RST button *(see page 53)*,

- causes the island bus to reconfigure itself with factory-default operating parameters, which do not support mandatory I/O status.
- loads the island's default configuration data, while recovering from a mandatory stop.

⚠ WARNING

LOSS OF CONTROL

- Do not attempt to restart the island by pushing the RST button.
- If a module is inoperable, replace it with the same module type.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Hot Swapping a Mandatory Module

If the NIM has stopped island bus operations because it cannot detect a healthy mandatory module, you can recover island bus operations by installing a healthy module of the same type. The NIM automatically configures the replacement module to match the removed module. Assuming that other modules and devices on the island bus are correctly configured and conform to their configuration data as written to Flash memory, the NIM starts of restarts normal island bus operations.

Prioritizing a Module

Summary

Using the Advantys configuration software, you can assign priority to digital input modules in your island assembly. Prioritization is a method of fine tuning the NIM's I/O scan of the island bus. The NIM will scan modules with priority more frequently than other island modules.

Limitations

You can prioritize only modules with digital inputs. You cannot prioritize output modules or analog modules. You can prioritize only 10 modules for a given island.

What Is a Reflex Action?

Summary

Reflex actions are small routines that perform dedicated logical functions directly on the Advantys island bus. They allow output modules on the island to act on data and drive field actuators directly, without requiring the intervention of the fieldbus master.

A typical reflex action comprises one or two function blocks that perform:

- Boolean AND or exclusive-OR operations
- comparisons of an analog input value to user-specified threshold values
- up- or down-counter operations
- timer operations
- the triggering of a latch to hold a digital value high or low
- the triggering of a latch to hold an analog value at a specific value

The island bus optimizes reflex response time by assigning the highest transmission priority to its reflex actions. Reflex actions take some of the processing workload off the fieldbus master, and they offer a faster, more efficient use of system bandwidth.

How Reflex Actions Behave

For outputs that are configured to respond to reflex actions, the output state represented in the island's network interface module (NIM) may not represent the actual states of the outputs.

- Turn off field power before you service any equipment connected to the island.
- For digital outputs, view the echo register for the module in the process image to see the actual output state.
- For analog outputs, there is no echo register in the process image. To view an actual analog output value, connect the analog output channel to an analog input channel.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Check whether the output state represented in the Network Interface Module represents the actual states of the output.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Reflex actions are designed to control outputs independently of the fieldbus master controller. They may continue to turn outputs on and off even when power is removed from the fieldbus master. Use prudent design practices when you use reflex actions in your application.

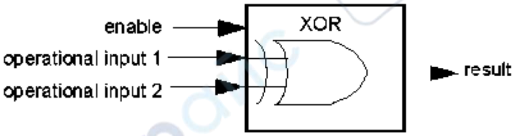
Configuring a Reflex Action

Each block in a reflex action should be configured using the Advantys configuration software.

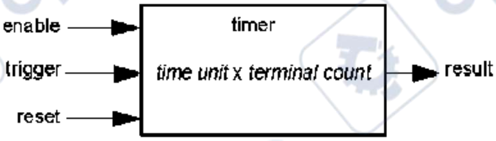
Each block should be assigned a set of inputs and a result. Some blocks also require that you specify one or more user-preset values—a compare block, for example, requires that you preset threshold values and a delta value for hysteresis.

Inputs to a Reflex Action

The inputs to a reflex block include an enable input and one or more operational inputs. The inputs may be constants or they may come from other I/O modules on the island, from virtual modules or outputs from another reflex block. For example, an XOR block requires three inputs—the enable and two digital inputs that contain the Boolean values to be XORed:



Some blocks, such as the timers, require reset and/or trigger inputs to control the reflex action. The following example shows a timer block with three inputs:



The trigger input starts the timer at 0 and accumulates *time units* of 1, 10, 100 or 1000 ms for a specified number of counts. The reset input causes the timer accumulator to be reset.

An input to a block may be a Boolean value, a word value, or a constant, depending on the type of reflex action it is performing. The enable input is either a Boolean or a constant *always enabled* value. The operational input to a block such as a digital latch should be a Boolean, whereas the operational input to an analog latch should be a 16-bit word.

You will need to configure a source for the block’s input values. An input value may come from an I/O module on the island or from the fieldbus master via a virtual module in the NIM.

NOTE: All inputs to a reflex block are sent on a change-of-state basis. After a change-of-state event has occurred, the system imposes a 10 ms delay before it accepts another change of state (input update). This feature is provided to minimize jitter in the system.

Result of a Reflex Block

Depending on the type of reflex block that you use, it will output either a Boolean or a word as its result. Generally, the result is mapped to an *action module*, as shown in the following table:

Reflex Action	Result	Action Module Type
Boolean logic	Boolean value	digital output
integer compare	Boolean value	digital output
counter	16-bit word	first block in a nested reflex action
timer	Boolean value	digital output
digital latch	Boolean value	digital output
analog latch	16-bit word	analog output

The result from a block is usually mapped to an individual channel on an output module. Depending on the type of result that the block produces, this action module may be an analog channel or a digital channel.

When the result is mapped to a digital or analog output channel, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device.

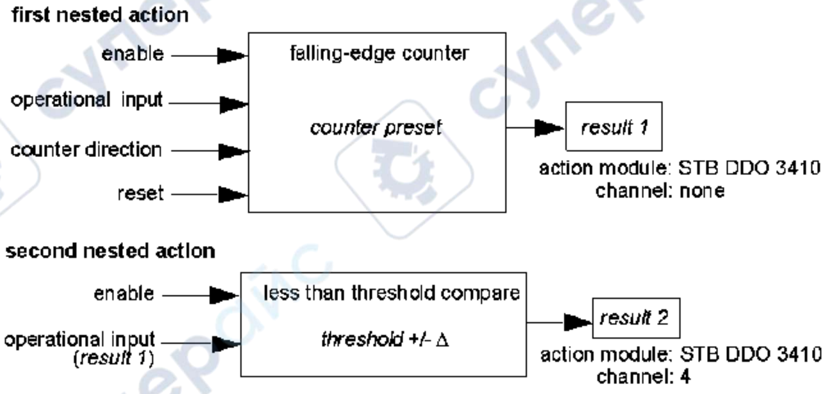
The exception is when a reflex block is the first of two actions in a nested reflex action.

Nesting

The Advantys configuration software allows you to create nested reflex actions. One level of nesting is supported—i.e., two reflex blocks, where the result of the first block is an operational input to the second block.

When you nest a pair of blocks, you need to map the results of both to the same action module. Choose the action module type that is appropriate for the result of the second block. This may mean that in some cases you will need to choose an action module for the first result that does not seem to be appropriate according to the table above.

For example, say you want to combine a counter block and a compare block in a nested reflex action. You want the result of the counter to be the operational input to the compare block. The compare block will then produce a Boolean as its result:



Result 2 (from the compare block) is the result that the nested reflex action will send to an actual output. Because the result of a compare block needs to be mapped to a digital action module, *result 2* is mapped to channel 4 on an STB DDO 3410 digital output module.

Result 1 is used only inside the module—it provides the 16-bit operational input to the compare block. It is mapped to the same STB DDO 3410 digital output module that is the action module for the compare block.

Instead of specifying a physical channel on the action module for *result 1*, the channel is set to *none*. In effect, you are sending *result 1* to an internal reflex buffer where it is stored temporarily until it is used as the operational input to the second block. You are not really sending an analog value to a digital output channel.

Number of Reflex Blocks on an Island

An island can support up to 10 reflex blocks. A nested reflex action consumes two blocks.

An individual output module can support up to two reflex blocks. Supporting more than one block requires that you manage your processing resources efficiently. If you are not careful with your resources, you may be able to support only one block on an action module.

Processing resources are consumed quickly when a reflex block receives its inputs from multiple sources (different I/O modules on the island and/or virtual modules in the NIM). To preserve processing resources:

- use the *always enabled* constant as the enable input whenever possible
- use the same module to send multiple inputs to a block whenever possible

Island Fallback Scenarios

Introduction

In the event of a communications interruption on the island or between the island and the fieldbus, output data is put into a fallback state. In this state, output data is replaced with pre-configured fallback values. This makes known the module's output data values when the system recovers from this condition.

Fallback Scenarios

There are several scenarios in which Advantys STB output modules go into their fallback states:

- detected loss of fieldbus communications: Communications with the PLC are lost.
- detected loss of island bus communications: There is an internal island bus communications interruption, indicated by a missing heartbeat message from either the NIM or a module.
- change of operating state: The NIM may command the island I/O modules to switch from a running to a non-running (stopped or reset) state.
- missing or non-operating mandatory module: The NIM detects this condition for a mandatory island module.

NOTE: If a mandatory (or any other) module is not operating, it needs to be replaced. The module itself does not go into its fallback state.

In these fallback scenarios, the NIM disables the heartbeat message.

Heartbeat Message

The Advantys STB system relies on a heartbeat message to verify the integrity and continuity of communications between the NIM and the island modules. The health of island modules and the overall integrity of the Advantys STB system are monitored through the transmission and reception of these periodic island bus messages.

Because island I/O modules are configured to monitor the NIM's heartbeat message, output modules go into their fallback states if they do not receive a heartbeat message from the NIM within the defined interval.

Fallback States for Reflex Functions

Only an output module channel to which the result of a reflex action *(see page 156)* has been mapped can operate in the absence of the NIM's heartbeat message.

When modules that provide input for reflex functionality are not operating or are removed from the island, the channels that hold the result of those reflex actions go into their fallback states.

In most cases, an output module that has one of its channels dedicated to a reflex action goes to its configured fallback state if the module loses communication with the fieldbus master. The only exception is a two-channel digital output module that has both of its channels dedicated to reflex actions. In this case, the module may continue to solve logic after a detected loss of fieldbus communication. For more information about reflex actions, refer to the *Reflex Actions Reference Guide*.

Configured Fallback

To define a customized fallback strategy for individual modules, you are required to use the Advantys Configuration Software. Configuration is done channel by channel. You can configure a single module’s multiple channels with different fallback parameters. Configured fallback parameters (implemented only during a communications interruption) are part of the configuration file stored in the NIM’s non-volatile Flash memory.

Fallback Parameters

You can select either of two fallback modes when configuring output channels with the Advantys Configuration Software:

- *hold last value*: In this mode, outputs retain the last values they were assigned before the fallback condition was triggered.
- *predefined value*: In this (default) mode, you can select either of two fallback values:
  - 0 (default)
  - some value in acceptable range

The permissible values for fallback parameters in the *predefined value* mode for discrete and analog modules and reflex functions appear in the following table:

Module Type	Fallback Parameter Values
discrete	0/off (default)
	1/on
analog	0 (default)
	not 0 (in range of acceptable analog values)

NOTE: In an auto-configured system, default fallback parameters and values are used.

Saving Configuration Data

Introduction

The Advantys configuration software allows you to save configuration data created or modified with this software to the NIM’s Flash memory and/or to the removable memory card *(see page 47)*. Subsequently, this data can be read from Flash memory and used to configure your physical island.

NOTE: If your configuration data is too large, you will receive a message when you attempt to save it.

How to Save a Configuration

The following procedure describes the steps you use to save a configuration data file to Flash memory directly and to a removable memory card. For more detailed procedural information, use the configuration software’s online help feature:

Step	Action	Comment
1	Connect the device running the Advantys Configuration Software to the CFG port *(see page 33)* on the NIM.	For NIM modules that support Ethernet communications, you can connect the device directly to the Ethernet port.
2	Launch the configuration software.	
3	Download the configuration data that you want to save from the configuration software to the NIM.	A successful download stores the configuration data to the NIM's flash memory.
4	Install the card *(see page 48)* in the host NIM, then use the **Store to SIM card** command.	Saving the configuration data to the removable memory card is optional. This operation overwrites old data on the SIM card.

Write-Protecting Configuration Data

Introduction

As part of a custom configuration, you can password-protect an Advantys STB island. Only authorized persons have write privileges to the configuration data currently stored in Flash memory:

- Use the Advantys Configuration Software to password-protect an island's configuration.
- For some modules, it is possible to password-protect the island configuration through an embedded web site.

The island runs normally in protected mode. All users have the ability to monitor (read) the activity on the island bus. If a configuration is write-protected, access to it is restricted in the following ways:

- An unauthorized user is unable to overwrite the current configuration data in Flash memory.
- The RST button *(see page 53)* is disabled, and pushing it has no effect on island bus operations.
- The presence of a removable memory card *(see page 47)* is ignored. The configuration data currently stored in Flash cannot be overwritten by data on the card.

NOTE: The STB NIP 2311 NIM reads from the removable memory card, if one is present in the module.

Password Characteristics

A password should meet the following criteria:

- It should be between 0 and 6 characters in length.
- Only alphanumeric ASCII characters are permitted.
- The password is case-sensitive.

If password protection is enabled, your password is saved to Flash memory (or to a removable memory card) when you save the configuration data.

NOTE: A password-protected configuration is inaccessible to anyone who does not know the password. Your system administrator is responsible for keeping track of the password and the list of authorized users. If the assigned password is lost or forgotten, you are unable to change the island's configuration.
If the password is lost and you need to reconfigure the island, you need to perform a destructive reflash of the NIM. This procedure is described on the Advantys STB product Web site at *www.schneiderautomation.com*.

A Modbus View of the Island's Data Image

Summary

A block of Modbus registers is reserved in the NIM to hold and maintain the island's data image. Overall, the data image holds 9999 registers. The registers are divided into contiguous groups (or blocks), each dedicated to a specific purpose.

Modbus Registers and Their Bit Structure

Registers are16-bit constructs. The most significant bit (MSB) is bit 15, which is displayed as the leftmost bit in the register. The least significant bit (LSB) is bit 0, displayed as the rightmost bit in the register:

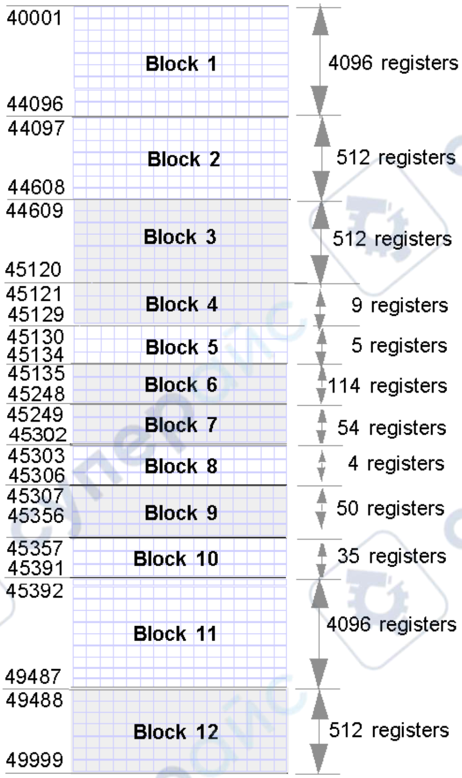
MSB															LSB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The bits can be used to display operating data or device/system status.

Each register has a unique reference number, starting at 40001. The content of each register, represented by its 0/1 bit pattern, may be dynamic, but the register reference and its assignment in the control logic program remain constant.

The Data Image

The 9999 contiguous registers in the Modbus data image start at register 40001. This figure shows the subdivision of data into sequential blocks:



Block 1 output data process image (4096 registers available)
Block 2 fieldbus master-to-HMI output table (512 registers available)
Block 3 reserved (512 registers available)
Block 4 9-register block reserved for future read/write use
Block 5 5-register RTP Request Block
Block 6 114-register block reserved for future read/write use
Block 7 54-register block reserved for future read/write use
Block 8 4-register RTP Response Block
Block 9 50-register block reserved for future read-only use
Block 10 35 predefined island bus status registers
Block 11 input data/status process image (4096 registers available)
Block 12 HMI-to-fieldbus master input table (512 registers available)

Each block has a fixed number of registers reserved for its use. Whether or not the registers reserved for that block are used in an application, the number of registers allocated to that block remains constant. This permits you to know at all times where to begin looking for the type of data of interest to you.

For example, to monitor the status of the I/O modules in the process image, look at the data in block 11 beginning at register 45392.

Reading Register Data

All the registers in the data image can be read by an HMI panel connected to the island at the NIM's CFG port *(see page 33)*. The Advantys configuration software reads this data, and displays blocks 1, 2, 5, 8, 10, 11, and 12 in the Modbus Image screen in its I/O Image Overview.

Writing Register Data

Some registers, usually configured number of registers in block 12 (registers 49488 through 49999) of the data image, may be written to by an HMI panel *(see page 169)*.

The Advantys configuration software or an HMI panel may also be used to write data to the registers in block 1 (registers 40001 through 44096). The configuration software or the HMI panel should be the island bus master in order for it to write to the data image—i.e., the island should be in *test* mode.

The Island's Process Image Blocks

Summary

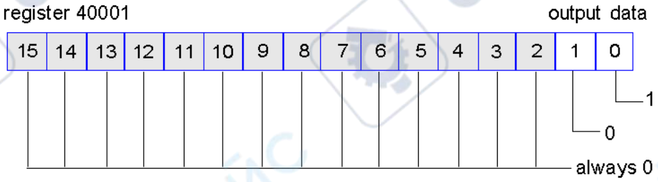
Two blocks of registers in the island's data image *(see page 165)* are the focus for this discussion. The first block is the output data process image, which starts at register 40001 and goes to register 44096. The other block is the input data and I/O status process image, which also consumes 4096 registers (45392 through 49487). The registers in each block are used to report island bus device status and to dynamically exchange input or output data between the fieldbus master and the island's I/O modules.

Output Data Process Image

The output data block (registers 40001 through 44096) handles the output data process image. This process image is a Modbus representation of the control data that has just been written from the fieldbus master to the NIM. Only data for the island's output modules is written to this block.

Output data is organized in 16-bit register format. One or more registers are dedicated to the data for each output module on the island bus.

For example, say you are using a two-channel digital output module as the first output module on your island bus. Output 1 is on and output 2 is off. This information would be reported in the first register in the output data process image, and it would look like this:



where:

- Normally, a value of 1 in bit 0 indicates that output 1 is on.
- Normally, a value of 0 in bit 1 indicates that output 2 is off.
- The remaining bits in the register are not used.

Some output modules, such as the one in the example above, utilize a single data register. Others may require multiple registers. An analog output module, for example, would use separate registers to represent the values for each channel, and might use the 11 or 12 most significant bits to display analog values in IEC format.

Registers are allocated to output modules in the output data block according to their addresses on the island bus. Register 40001 contains the data for the first output module on the island (the output module closest to the NIM).

Output Data Read/Write Capabilities

The registers in the output data process image are read/write-capable.

You can read (i.e., monitor) the process image using an HMI panel or the Advantys Configuration Software. The data content that you see when you monitor the output data image registers is updated in near-real time.

The island’s fieldbus master also writes updated control data to the output data process image.

Input Data and I/O Status Process Image

The input data and I/O status block (registers 45392 through 49487) handles the input data and I/O status process image. Every I/O module on the island bus has information that needs to be stored in this block.

- Each digital input module reports data (the on/off status of its input channels) in one register of input data and I/O status block, then reports its status in the next register.
- Each analog input module uses four registers in the input data and I/O status block. It represents the analog data for each channel in separate registers and the status of each channel in separate registers. Analog data is usually represented with 11- or 12-bit resolution in the IEC format; status in an analog input channel is usually represented by a series of status bits that report the presence or absence of an out-of-range value in a channel.
- Each digital output module reports an echo of its output data to a register in the input data and I/O status block. Echo output data registers are essentially copies of the register values that appear in the output data process image. This data is usually not of much interest, but it can be useful in the event that a digital output channel has been configured for a reflex action. In this case, the fieldbus master can see the bit value in the echo output data register even though the output channel is being updated inside the island bus.
- Each analog output module uses two registers in the input data and I/O status block to report status. Status in an analog output channel is usually represented by a series of status bits that report the presence or absence of an out-of-range value in a channel. Analog output modules do not report data in this block.

A detailed view of how the registers in the input-data and I/O status block are implemented is shown in the process image example.

The HMI Blocks in the Island Data Image

Summary

An HMI panel that communicates using the Modbus protocol can be connected to the CFG port *(see page 33)* on the NIM. Using the Advantys configuration software, you can reserve one or two blocks of registers in the data image *(see page 164)* to support HMI data exchange. When an HMI panel writes to one of these blocks, that data is accessible to the fieldbus master (as inputs). Data written by the fieldbus master (as outputs) is stored in a different reserved block of registers that the HMI panel can read.

HMI Panel Configuration

Advantys STB supports the ability of an HMI panel to act as:

- an input device, which writes data to the island's data image that is read by the fieldbus master
- an output device, which can read data written by the fieldbus master to the island's data image
- a combined I/O device

HMI Input Data Exchange

Input data to the fieldbus master can be generated by the HMI panel. Input controls on an HMI panel might be elements such as:

- push buttons
- switches
- a data entry keypad

To use an HMI panel as an input device on the island, you need to enable the HMI-to-fieldbus master block in the island's data image *(see page 165)* and specify the number of registers in this block that you want to use for HMI-to-fieldbus master data transfers. You should use the Advantys configuration software to make these configuration adjustments.

The HMI-to-fieldbus master block can comprise up to 512 registers, ranging from register 49488 to 49999. (Your actual register limit will be dictated by your fieldbus.) This block follows immediately after the standard input data and I/O status process image *(see page 168)* block (registers 45392 through 49487) in the island's data image.

The HMI panel writes the input data to a specified number of registers in the HMI-to-fieldbus master block. The NIM manages the transfer of the HMI data in these registers as part of the overall input data transfer—it converts the 16-bit register data to a fieldbus-specific data format and transfers it together with the standard input data and I/O status process image to the fieldbus. The fieldbus master sees and responds to HMI data as if it were standard input data.

HMI Output Data Exchange

In turn, output data written by the fieldbus master can be used to update enunciator elements on the HMI panel. Enunciator elements might be:

- display indicators
- buttons or screen images that change color or shape
- data display screens (for example, temperature read-outs)

To use the HMI panel as an output device, you need to enable the fieldbus-to-HMI block in the island's data image *(see page 165)* and specify the number of registers in this block that you want to use. You need to use the Advantys configuration software to make these adjustments to your configuration.

The fieldbus master-to-HMI block can comprise up to 512 registers, ranging from register 44097 to 44608. This block follows immediately after the standard output data process image *(see page 167)* block (registers 40001 through 44096) in the island's data image.

The fieldbus master writes output update data in native fieldbus format to the HMI data block concurrent with writing this data to the output data process image area. The output data is placed in the fieldbus master-to-HMI block. Upon request by the HMI via a Modbus *read* command, the role of the NIM is to receive this output data, convert it to16-bit Modbus format, and send it over the Modbus connection at the CFG port to the HMI panel.

NOTE: The *read* command enables all Modbus registers to be read, not just those in the block reserved for fieldbus master-to-HMI data exchange.

Test Mode

Summary

Test Mode indicates that the output data of the STB island's process image is not controlled by a fieldbus master device, but is instead controlled by either the Advantys Configuration Software or an HMI. When the STB island is operating in Test Mode, the fieldbus master cannot write the STB island's outputs, but can continue to read its inputs and diagnostic data.

Test Mode is configured off-line, downloaded with the island configuration, then activated online.

Select Test Mode Settings in the **Online** menu to open the Test Mode configuration window, where you can select a test mode setting. Test Mode settings are stored with other STB island configuration settings both in the NIM's flash memory and in a SIM card, if one is attached to the NIM.

When Test Mode is activated, the NIM's TEST LED is lit, and bit #5 of the NIM Status word in register 45391 is set to 1.

NOTE: Detected loss of Modbus communications does not affect Test Mode.

There are three Test Mode settings:

- Temporary Test Mode
- Persistent Test Mode
- Password Test Mode

The following sections describe the process and effect of activating Test Mode.

Temporary Test Mode

When operating online, use the Advantys Configuration Software (not an HMI) to activate Temporary Test Mode, by selecting **Test Mode** in the **Online** menu.

Once activated, Temporary Test Mode is deactivated by:

- de-selecting **Test Mode** in the **Online** menu
- cycling power to the NIM
- selecting **Reset** in the **Online** menu
- performing Autoconfiguration
- downloading a new island configuration to the NIM (or inserting a SIM card with a new island configuration into the NIM and cycling power to the NIM).

Temporary Test Mode is the default Test Mode configuration setting.

Persistent Test Mode

Use the Advantys Configuration Software to configure the STB island for Persistent Test Mode. When the download of this configuration is complete, Persistent Test Mode is activated. Thereafter, the STB island operates in Test Mode each time power is cycled to the island. When Persistent Test Mode is activated, the STB island's process image output data is controlled exclusively by either the HMI or the configuration software. The fieldbus master no longer controls these outputs.

Persistent Test Mode is deactivated by:

- downloading a new island configuration to the NIM (or inserting a SIM card with a new island configuration into the NIM and cycling power to the NIM)
- performing Autoconfiguration.

Password Test Mode

Use the Advantys Configuration Software to enter a password to the STB island's configuration settings. The password should have an integer value from 1 to 65535 (FFFF hex).

After the changed configuration (including the password) has been downloaded, you can activate Password Test Mode only by using an HMI to send a single Modbus Register write command to send the password value to Modbus Register 45120.

After Password Test Mode is activated, the STB island's process image output data is controlled by either the HMI or the configuration software. In this case, the fieldbus master no longer controls these outputs.

Password Test Mode, once activated, is deactivated by:

- cycling power to the NIM
- selecting **Reset** in the **Online** menu
- performing Autoconfiguration
- downloading a new island configuration to the NIM (or inserting a SIM card with a new island configuration into the NIM and cycling power to the NIM)
- using an HMI to issue a single Modbus register write command to send the password value to Modbus Register 45121 (STB NIC 2212 and STB NIP 2311 NIMs only)

NOTE: Activate Password Test Mode nly by using the NIM's configuration port. All attempts to enter Password Test Mode using the fieldbus (via NIM models STB NMP 2212 or STB NIP 2212) are unsuccessful.

Run-Time Parameters

Introduction

For STB modules, the Advantys Configuration Software provides the RTP (run-time parameters) feature. It can be used for monitoring and modifying selected I/O parameters and Island bus status registers of the NIM while the Island is running. This feature is available only in standard STB NIMs with firmware version 2.0 or later.

RTP has to be configured using the Advantys Configuration Software before it can be used. RTP is not configured by default. Configure RTP by selecting **Configure run-time Parameters** in the **Options** tab of the NIM Module Editor. This allocates the necessary registers within the NIM's data process image to support this feature.

Request and Response Blocks

Once configured, use the RTP feature by writing up to 5 reserved words in the NIM's output data process image (the RTP request block) and by reading the value of 4 reserved words in the NIM's input data process image (the RTP response block). The Advantys Configuration Software displays both blocks of reserved RTP words in the Island's **I/O Image Overview** dialog box, both in the **Modbus Image** tab and (for NIMs with a separate fieldbus image) in the **Fieldbus Image** tab. In each tab, the blocks of reserved RTP words appear after the block of process I/O data and before the block of HMI data (if any).

NOTE: The Modbus address values of the RTP request and response blocks are the same in all standard NIMs. The fieldbus address values of the RTP request and response blocks depend upon the network type. Use the **Fieldbus Image** tab of the **I/O Image Overview** dialog box to obtain the location of the RTP registers. For Modbus Plus and Ethernet networks, use the Modbus register numbers.

Exceptions

Any parameter you modify using the RTP feature does not retain its modified value if one of the following events occurs:

- Power is cycled to the NIM.
- A **Reset** command is issued to the NIM using the Advantys Configuration Software.
- A **Store to SIM Card** command is issued using the Advantys Configuration Software.
- The module whose parameter has been modified is hot-swapped.
  If a module is hot-swapped, as indicated by the HOT_SWAP indicator bit, you can use the RTP feature to detect which module has been hot-swapped and to restore the parameters to their previous values.

Test Mode

When the NIM is operating in test mode, the NIM's output data process image (including the RTP request block) can be controlled either by the Advantys Configuration Software or by an HMI (depending upon the test mode configured). Standard Modbus commands can be used to access the RTP words. If the NIM is in test mode, the fieldbus master cannot write to the RTP request block in the NIM's output data process image.

RTP Request Block Words Definitions

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Write all bytes in the RTP request block before you set the `toggle+CMD` and `toggle+length` bytes to the same new value.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The following table lists RTP request block words:

Modbus Address	Upper Byte	Lower Byte	Data Type	Attribute
45130	sub-index	`toggle + length`	unsigned 16	RW
45131	index (high data byte)	index (low data byte)	unsigned 16	RW
45132	data byte 2	data byte 1 (LSB)	unsigned 16	RW
45133	data byte 4 (MSB)	data byte 3	unsigned 16	RW
45134	`toggle + CMD`	Node ID	unsigned 16	RW
NOTE: The RTP request block is also presented in the manufacturer specific area of the CANopen fieldbus as an object with a dedicated index of 0x4101 and sub-index 1 to 5 (data type = unsigned 16, attribute = RW).				

The NIM performs range checking on the above bytes as follows:

- index (high / low byte): 0x2000 to 0xFFFF for write; 0x1000 to 0xFFFF for read
- `toggle + length`: length = 1 to 4 bytes; the most significant bit contains the toggle bit
- `toggle + CMD`: CMD = 1 to 0x0A (see the table *Valid Commands*, below); most significant bit contains toggle bit
- Node ID: 1 to 32 and 127 (the NIM itself)

The `toggle+CMD` and `toggle+length` bytes are at either end of the RTP request register block. The NIM processes the RTP request when the same value is set in the respective toggle bits of these two bytes. The NIM processes the same RTP block again only when both values have changed to a new identical value. We recommend that you configure new matching values for the two toggle bytes (`toggle+CMD` and `toggle+length`) only after you have constructed the RTP request between them.

RTP Response Block Words Definitions

The following list shows RTP response block words:

Modbus Address	Upper Byte	Lower Byte	Data Type	Attribute
45303	status (the most significant bit is used to indicate whether RTP service is enabled: MSB=1 means enabled)	`toggle + CMD echo`	unsigned 16	RO
45304	data byte 2	data byte 1 (LSB)	unsigned 16	RO
45305	data byte 4 (MSB)	data byte 3	unsigned 16	RO
45306	-	`toggle + CMD echo`	unsigned 16	RO
NOTE: The RTP response block is also presented in the manufacturer specific area of the CANopen fieldbus as an object with a dedicated index of 0x4100 and sub-index 1 to 4 (data type = unsigned 16, attribute = RO).				

The `toggle+CMD echo` bytes are located at the end of the register range to let you validate the consistency of the data wrapped within these bytes (in case RTP response block words are not updated in a single scan). The NIM updates the status byte and the 4 data bytes (if applicable) before updating the `toggle+CMD echo` bytes in Modbus register 45303 and 45306 to equal the value of the `toggle+CMD` byte of the corresponding RTP request. First check that both `toggle+CMD` bytes match the `toggle+CMD` byte in the RTP request block before making use of the data inside the RTP response block.

Valid RTP Commands

The following list shows valid commands (CMDs):

Command (CMD)	Code (Except the msb)	Valid Node IDs	Allowed State of the Addressed Node	Data Bytes
Enable RTP (Only After RTP Has Been Configured Using the Advantys Configuration Software)	0x08	127	N/A	-
Disable RTP	0x09	127	N/A	-
Reset Hot-Swap Bit	0x0A	1-32	N/A	-
Read Parameter	0x01	1-32, 127	pre-operational operational	data bytes in response, length to be given
Write Parameter	0x02	1-32	operational	data bytes in request, length to be given

The most significant bit of an RTP request block's `toggle+CMD` byte is the toggle bit. A new command is identified when the value of this bit changes and matches the value of the toggle bit in the `toggle+length` byte.

A new RTP request is processed only if the preceding RTP request has finished. Overlapping RTP requests are not allowed. A new RTP request made before the completion of a preceding request is ignored.

To determine when an RTP command has been processed and its response is complete, check the values of the `toggle+CMD echo` bytes in the RTP response block. Continue to check both `toggle+CMD` bytes in the RTP response block until they match the RTP request block's `toggle+CMD` byte. Once they match, the contents of the RTP response block is valid.

Valid RTP Status Messages

The following list shows valid status messages:

Status Byte	Code	Comment
Success	0x00 or 0x80	0x00 for successful completion of a Disable RTP command
Command not Processed due to Disabled RTP	0x01	-
Illegal CMD	0x82	-
Illegal Data Length	0x83	-
Illegal Node ID	0x84	-
Illegal Node State	0x85	Access is denied because a node is absent or not started.
Illegal Index	0x86	-
RTP Response Has More Than 4 Bytes	0x87	-
No Communication Possible on the Island Bus	0x88	-
Illegal Write to Node 127	0x89	-
SDO Aborted	0x90	If there is a detected error in SDO protocol, the data bytes in the response contain the SDO abort code according to DS301.
General Exception Response	0xFF	This is a status event of a type other than those specified above.

The most significant bit of the status byte in the RTP response block indicates whether RTP is enabled (1) or disabled (0).

Virtual Placeholder

Summary

The virtual placeholder feature lets you create a standard island configuration and depopulated variations of that configuration that share the same fieldbus process image, thereby letting you maintain a consistent PLC or fieldbus master program for various island configurations. The depopulated islands are physically built using only those modules that are not marked as *not present*, thus saving cost and space.

As part of an Advantys STB island custom configuration, you can set *Virtual Placeholder* status for any STB I/O or preferred third-party module whose node address is assigned by the NIM during auto-addressing.

After a module has been assigned Virtual Placeholder status, you can physically remove it from its Advantys STB island base, while retaining the island's process image. All modules that physically remain in the Advantys STB island configuration will retain their previous node addresses. This lets you physically alter the design of your island, without having to edit your PLC program.

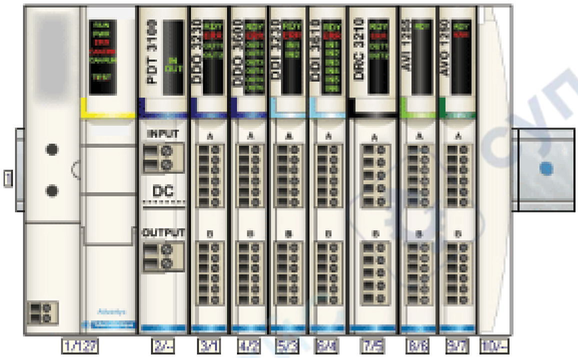
NOTE: Advantys configuration software is required to set Virtual Placeholder status.

Setting Virtual Placeholder Status

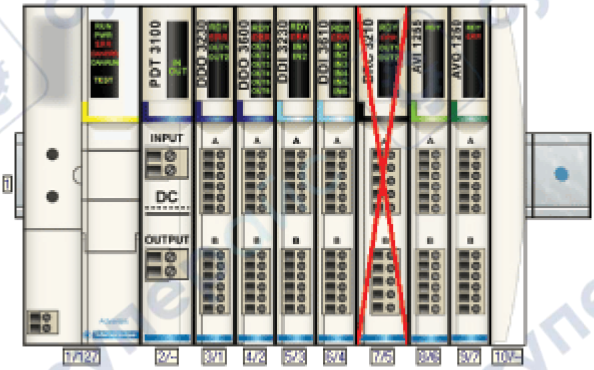
To set Virtual Placeholder status:

Step	Action
1	Open the STB I/O or preferred third-party module's property window.
2	In the Options tab, select **Not Present**.
3	Click **OK** to save your settings. The Advantys STB configuration software marks the virtual placeholder module with a red "X" (as shown below).

For example, the following island configuration contains a NIM, a PDM, 2 digital Input modules, 2 digital output modules, a digital relay output module, an analog input module, and an analog output module:



After you assign Virtual Placeholder status to the DRC 3210 digital relay output module (by selecting **Not Present** in its Options tab), the Advantys STB configuration software marks the virtual placeholder module with a red "X" as shown below:



For example, when you physically construct the above configuration, you would build the island without the DRC-3210 and its base.

NOTE: Any reflex output, that is configured to use a virtual placeholder module as an input, will constantly be in fallback.

Appendices

Appendix A
Additional Objects in the Object Model

Introduction

This chapter discusses the objects in the object model.

NOTE:
Descriptions of the more commonly implemented objects are in the Object Model section *(see page 64)*. They are:

- assembly object *(see page 67)*
- island bus object *(see page 69)*

What Is in This Chapter?

This chapter contains the following topics:

Identity Object (Class ID 1)

Introduction

The identity object provides the configuration and status of the STB NIC 2212 NIM’s physical connection to the network.

Class Attributes (Instance 0)

The identity object supports these class attributes:

Attribute ID	Name	Access	Description
0x01	Revision	R	This attribute returns the object implementation revision (0x01).
0x02	Max Instance	R	This attribute returns the largest instance number (0x01).
0x03	Number of Instances	R	This attribute returns the number of object instances (0x01).
0x04	Optional Instance Attribute list	R	The first two bytes contain the number of optional instance attributes. Each subsequent byte pair represents the number of a different optional instance attribute.
0x06	Max Class Attribute	R	This attribute returns the largest class attributes value (0x07).
0x07	Max Instance Attribute	R	This attribute returns the largest instance attributes value (0x11).

Class Services

The identity object supports these class services:

Service Code	Name	Description
0x0E	Get Attribute Single	This service returns the value of the specified attribute.
0x01	Get Attribute All	This service returns the value of all class attributes.

Instance Attributes (Instance 1)

The identity object supports these instance attributes:

Attribute ID	Name	Access	Description
0x01	Vendor ID	R	This attribute returns a value of 243 (as assigned to Schneider Electric by ODVA).
0x02	Device Type	R	This attribute returns a Communication Adapter value (0x0C).
0x03	Device Code	R	This attribute returns the device code (2213).
0x04	Identity Revision	R	This attribute returns major and minor class revisions.
0x05	Identity Status	R	This attribute returns the EtherNet/IP status of the device.
0x06	Device Serial Number	R	This attribute returns a number used in conjunction with the vendor ID to form a unique identifier for each device.
0x07	Product Name	R	The first byte contains the number of ASCII characters in the ASCII string that follows. The string contains the dynamically built device name, "STBNIC2212 In<xx> Out<yy," where xx is the packed input process image size in bytes and yy is the packed output process image size in bytes. Values xx and yy are not 0 padded and take as many decimal digits as needed. For example, xx can be 8 or 48 or 488.

Instance Services

The identity object supports these instance services:

Service Code	Name	Description
0x01	Get Attribute All	This instance returns the value of all instance attributes of access type R.
0x0E	Get Attribute Single	This instance returns the value of the specified Identity attribute of access type R.
0x10	Set Attribute Single	This instance modifies the instance attribute value with access types W or R/W.
0x05	Reset	This instance emulates the NIM device reset.

Message Router Object (Class ID 2)

Introduction

The message router object directs explicit messages among objects within a device.

Class Attributes (Instance 0)

The message router object supports these class attributes:

Attribute ID	Name	Access	Description
0x01	Revision	R	This attribute returns the implementation revision (0x01).
0x02	Max Instance	R	This attribute returns the largest instance number (0x01).
0x03	Number of Instances	R	This attribute returns the number of object instances (0x01).
0x04	Optional Attribute List	R	This attribute returns the number and list of any implemented optional attributes. It returns 0x00 0x00 when no such attributes are supported.
0x05	Optional Service List	R	This attribute returns the number and list of any implemented optional services. It returns 0x00 0x00 when no such services are supported.
0x06	Max Class Attribute	R	This attribute returns the largest class attribute value (0x07).
0x07	Max Instance Attribute	R	This attribute returns the largest instance attribute value (0x02).

Class Services

The message router object supports these class services:

Service Code	Name	Description
0x0E	Get Attribute Single	This service returns the value of the specified attribute.
0x01	Get Attribute All	This service returns the value of all class attributes.

Instance Attributes (Instance 1)

The message router object supports these instance attributes:

Attribute ID	Name	Access	Description
0x01	Implemented Object List	R	The first two bytes of the implemented objects list contain the number of implemented objects. Each subsequent two bytes in the list represent another implemented class number. The list contains these objects: • Identity Object *(see page 182)* • Message Router Object *(see page 184)* • Assembly Object *(see page 67)* • Connection Management Object *(see page 186)* • PCCC (class ID 0x67) • Port Object *(see page 190)* • TCP/IP Interface Object *(see page 192)* • Ethernet Link Object *(see page 194)* • Island Bus Profile Object *(see page 69)* • File Object *(see page 188)*
0x02	Max Connection Number Supported	R	This attribute returns the maximum number of concurrent CIP connections (class 1 and class 3) that are supported (32).

Instance Services

The message router object supports these instance services:

Service Code	Name	Description
0x01	Get Attribute All	This service returns the value of all instance attributes.
0x0E	Get Attribute Single	This service returns the value of the specified instance attribute.

Connection Management Object (Class ID 6)

Introduction

The connection object class allocates and manages the internal resources associated with both I/O and explicit messaging connections. The Advantys STB EtherNet/IP NIM supports the predefined master/slave connection set and the unconnected message manager (UCMM) for dynamic establishment of messaging connections. For the STB NIC 2212 NIM connection management object:

- The class ID is 6.
- The single instance ID is 1.

Class Attributes (Instance 0)

The connection management object supports these class attributes:

Attribute ID	Name	Access	Description
0x01	Revision	R	This attribute returns the implemented version of the connection manager (0x01).
0x02	Max Instance	R	This attribute returns the value of the largest instance number (0x01).
0x03	Number of Instances	R	This attribute returns the number of object instances (0x01).
0x04	Optional Attribute list	R	This attribute returns the number and list of the optional attributes. The first word contains the number of attributes that follow. Every subsequent word contains another attribute code. The optional attributes are: ● the total number of incoming requests to open connections ● the number of connection requests that are rejected because the format of Forward Open is unexpected ● the number of connection requests that are rejected because of insufficient resources ● the number of connection requests that are rejected because the parameter value is sent with the Forward Open ● the number of received Forward Close requests ● the number of invalid (format) Forward Close requests ● the number of Forward Close requests that do not match an open connection ● the number of connections that are timed out because: ❍ the other side stopped producing ❍ there is a network disconnection
0x06	Max Class Attribute	R	This attribute returns the value of the highest class attribute (0x07).
0x07	Max Instance Attribute	R	This attribute returns the value of the largest instance attribute (0x08).

Class Services

The connection management object supports these class services:

Service Code	Name	Description
0x0E	Get Attribute Single	This service returns the value of the specified attribute.
0x01	Get Attribute All	This service returns the value of all class attributes.

Instance Attributes (Instance 1)

The connection management object supports these instance attributes:

Attribute ID	Name	Access	Description
0x01	Incoming Forward Open Request	R	This attribute returns the number of incoming open connection requests.
0x02	Forward Open Format Failure count	R	This attribute returns the number of Forward Open requests that are rejected because of bad formatting in the Forward Open.
0x03	Forward Open Resource Failure count	R	This attribute returns the number of Forward Open requests that are rejected because of insufficient resources.
0x04	Forward Open Parameter Value count	R	This attribute returns the number of Forward Open requests that are rejected because the parameter value is sent with the Forward Open.
0x05	Incoming Forward Close requests count	R	This attribute returns the number of incoming close connection requests.
0x06	Forward Close Format Failure count	R	This attribute returns the number of Forward Close requests that have an invalid format.
0x07	Forward Close Matching Failure count	R	This attribute returns the number of Forward Close requests that don’t match an active connection.
0x08	Timed out Connections count	R	This attribute returns the number of connections that are timed out because: ● the other side stopped producing ● there is a network disconnection

Instance Services

The connection management object supports these instance services:

Service Code	Name	Description
0x01	Get Attribute All	This service returns the value of all instance attributes.
0x0E	Get Attribute Single	This service returns the value of the specified instance attribute.
0x54	Forward Open	This service opens a new connection.
0x4E	Forward Close	This service closes an existing connection.
0x52	Unconnected Send	This service sends an unconnected multi-hop request.

File Object (Class ID 0x37)

Class Attributes (Instance 0)

The file object supports these class attributes:

Attribute ID	Name	Access	Description
0x01	Revision	R	This attribute returns the implemented File Object version (0x01).
0x02	Max Instance	R	This attribute returns the value of the largest instance number (0xC9).
0x03	Number of Instances	R	This attribute returns the number of object instances (0x02).
0x06	Max Class Attribute	R	This attribute returns the highest class attribute's value (0x20).
0x07	Max Instance Attribute	R	This attribute returns the value of the largest instance attribute (0xB).
0x20	Instance List	R	This attribute returns information on all configured instances, including Instance Number, Instance Name, and Instance File Name.

Class Services

The file object supports these class services:

Service Code	Name	Description
0x0E	Get Attribute Single	This service returns the value of the specified attribute.

Instance Codes

The following instances are implemented:

Instance	Description
0xC8	This instance returns the uncompressed version of the device EDS text file: ● The returned instance name attribute name is "EDS and Icon Files." ● The contents of the EDS file is adjusted dynamically by the STB NIC 2212 (based on the current Island Bus configuration). ● The connection data sizes in the EDS file is adjusted to reflect the actual assembly instance sizes.
0xC9	The returned instance name is "Related EDS and Icon Files." The returned file name attribute is "STBNIC2212.gz," a ZLIB-encoded file that contains only one file, "STBNIC2212.ico."

Instance Attributes

The file object supports no instance attributes.

Instance Services

The file object supports these instance services:

Service Code	Name	Description
0x0E	Get Attribute Single	This service returns data for the above instance attributes.
0x10	Set Attribute Single	This service returns Set data for the Get/Set instance attributes listed above (only for read/write files).
0x4B	Initiate Upload	This service starts the upload process. The request contains the Maximum File Size that the client can upload. The response contains the actual File Size, which is never more than the Maximum File Size and the Transfer Size (the number of bytes transferred with each Upload Transfer request).
0x4F	Upload Transfer	This service uploads another chunk of file data. The request contains the Transfer Number (which is incremented with each subsequent transfer). The response contains the matching Transfer Number, Transfer Type, File Data, and (for the last transfer) the Checksum word. Transfer Type indicates: ● the first packet ● the intermediate packet ● the last packet ● the only packet ● this transfer should be aborted
0x4C	Initiate Download	This service starts downloading read/write files. The request contains the Total Download size, Instance Format Version, File Revision, and File Name. The response contains: ● *Burn Size*: the number of bytes before a save to non-volatile memory occurs ● *Burn Time*: the number of seconds that are allocated for the non-volatile save ● *Transfer Size*: the number of bytes sent with each Download Transfer request
0x50	Download Transfer	The services downloads an additional chunk of file data. The request contains: ● *Transfer Number*: increments with each subsequent transfer ● *Transfer Type*: indicates the first, intermediate, last, or only packet, and whether the transfer should be aborted ● *File Data* ● *Checksum word*: last transfer
0x51	Clear File	This service clears the contents of the read/write file: ● *File Instance State*: set to File Empty ● *File Size*: set to 0

Port Object (Class ID 0xF4)

Class Attributes (Instance 0)

The port object supports these class attributes:

Attribute ID	Name	Access	Description
0x01	Revision	R	This attribute returns the implemented Port Object revision (0x01).
0x02	Max Instance	R	This attribute returns the value of the largest instance number (0x01).
0x03	Number of Instances	R	This attribute returns the number of object instances (0x01).
0x06	Max Class Attribute	R	This attribute returns the value of the largest class attribute (0x09).
0x07	Max Instance Attribute	R	This attribute returns the value of the largest instance attribute (0x07).
0x08	Request Entry Port Instance number	R	This attribute returns the instance of the port (0x01) through which the request enters the device. (The EtherNet/IP port supports on 1 instance.)
0x09	Port Instance Info List	R	This attribute returns an array of structures that contain instance attributes 1 (port type) and 2 (port number). It returns an array with the only instantiated port set to type TCP/IP (0x04) and the port number of the TCP/IP port (0x02).

Class Services

The port object supports these class services:

Service Code	Name	Description
0x0E	Get Attribute Single	This service returns the value of the specified attribute.
0x01	Get Attribute All	This service returns the value of all class attributes.

Instance Attributes (Instance 1)

The port supports these instance attributes:

Attribute ID	Name	Access	Description
0x01	Port Type	R	This attribute returns the Port Type identifier (0x04 for the TCP/IP port type).
0x02	Port Number	R	This attribute returns the Port Number (0x02 for the TCP/IP port number).
0x03	Port Path	R	The first word contains the number of path words that follow. The EPATH that points to the TCP/IP port follows. It returns six bytes: 0x02, 0x00, 0x20, 0xF5, 0x24, and 0x01. They point to the class of TCP/IP, instance 1.
0x04	Port Name	R	This attribute returns SHORT_STRING, where the first byte is the length of the string in bytes. It is followed by the ASCII string "Ethernet Port."
0x07	Node Address	R	This attribute returns 0x10 0x00, indicating that a node address in not used on the Ethernet network.

Instance Services

The port object supports these instance services:

Service Code	Name	Description
0x0E	Get Attribute All	This service returns the value of all instance attributes.
0x01	Get Attribute Single	This service returns the value of the specified instance attribute.

TCP/IP Interface Object (Class ID 0xF5)

Introduction

The TCP/IP Interface Object defines the number of IP address configuration options for the device.

NOTE: Some parameters that this object sets and reads can also be configured through the embedded web pages *(see page 85)*.

Class Attributes (Instance 0)

The TCP/IP interface object supports these class attributes:

Attribute ID	Name	Access	Description
0x01	Revision	R	This attribute returns the implemented revision of the TCP/IP object (0x01).
0x02	Max Instance	R	This attribute returns 0x01, indicating that there is only one host IP address.
0x03	Number of Instances	R	This attribute returns the number of object instances (0x01).
0x06	Max Class Attribute	R	This attribute returns the value of the largest class attribute (0x07).
0x07	Max Instance Attribute	R	This attribute returns the value of the largest instance attribute (0x06).

Class Services

The TCP/IP interface object supports these class services:

Service Code	Name	Description
0x0E	Get Attribute Single	This service returns the value of the specified attribute.
0x01	Get Attribute All	This service returns the value of all class attributes.

Instance Attributes (Instance 1)

The TCP/IP interface object supports these instance attributes:

Attribute ID	Name	Access	Description
0x01	Configuration Status	R	This attribute indicates whether the TCP/IP object (including all parameters) is configured.
0x02	Configuration Capability	R	This attribute indicates whether the TCP/IP object (with all parameters) can be configured using DHCP or BootP, and whether it can resolve the host names using the DNS server. It returns 5 to indicate support for DHCP and BootP.
0x03	Configuration Control	R	This attribute indicates the implemented IP address method: Stored IP (0), BootP (1), or DHCP (2).
0x04	Physical Link	R	This attribute returns the electronic path to the physical link object, which is the Ethernet Link class. The first word contains the size of the EPATH in words. The path that follows specifies instance 1 of the Ethernet Link object (0x20 0xF6 0x24 0x01).
0x05	Interface Configuration	R	This attribute returns all TCP/IP parameters, including: ● DWORD: contains the device IP address ● DWORD: contains the subnet mask ● DWORD: contains the gateway address ● DWORD: contains the name server IP address ● DWORD: contains the second name server IP address ● WORD: contains the number of ASCII characters in the domain name ● ASCII: a string that contains the domain name
0x06	Host Name	R	The first word contains the number of ASCII bytes in the device host name. The ASCII host name string follows.

Instance Services

The TCP/IP interface object supports these instance services:

Service Code	Name	Description
0x01	Get Attribute All	This service returns the value of all instance attributes.
0x0E	Get Attribute Single	This service returns the value of the specified instance attribute.

Ethernet Link Object (Class ID 0xF6)

Introduction

The Ethernet Link Object tracks configuration and diagnostics information for the Ethernet port.

Class Attributes (Instance 0)

The Ethernet Link Object supports these class attributes:

Attribute ID	Name	Access	Description
0x01	Revision	R	This attribute returns the implemented TCP/IP object revision (0x02).
0x02	Max Instance	R	This attribute returns 0x01, indicating that there is only one host IP address.
0x03	Number of Instances	R	This attribute returns the number of object instances (0x01).
0x06	Max Class Attribute	R	This attribute returns the value of the largest class attribute (0x07).
0x07	Max Class Attribute	R	This attribute returns the value of the largest instance attribute (0x03).

Class Services

The Ethernet Link Object supports these class services:

Service Code	Name	Description
0x0E	Get Attribute Single	This service returns the value of the specified attribute.
0x01	Get Attribute All	This service returns the value of all instance attributes.

Instance Attributes (Instance 1)

The Ethernet Link Object supports these instance attributes:

Attribute ID	Name	Access	Description
0x01	Interface Speed	R	This instance returns the interface speed. It depends on the Ethernet speed: ● 0x0A: 10 Mbps ● 0x64: 100 Mbps
0x02	Interface Flags	R	These flags indicate interface information: ● *Bit 0 (link status)*: The link status indicates when the Ethernet 802.3 communications interface is connected to an active network. 0 indicates an inactive link. 1 indicates an active link. ● *Bit 1 (half/full duplex)*: 0 indicates that the interface is running in half duplex. 1 indicates that the interface is running in full duplex. (Note that the value of the half/full duplex flag is indeterminate when the link status flag is 0.) ● *Bits 2...31 (reserved)*: These bits are set to 0.
0x03	MAC Address	R	This service returns device's 6-byte MAC address.

Instance Services

The Ethernet Link Object supports these instance services:

Service Code	Name	Description
0x01	Get Attribute All	This service returns the value of all instance attributes.
0x0E	Get Attribute Single	This service returns the value of the specified instance attribute.

Glossary

!

100Base-T

An adaptation of the IEEE 802.3u (Ethernet) standard, the 100Base-T standard uses twisted-pair wiring with a maximum segment length of 100 m (328 ft) and terminates with an RJ-45 connector. A 100Base-T network is a baseband network capable of transmitting data at a maximum speed of 100 Mbit/s. "Fast Ethernet" is another name for 100Base-T, because it is ten times faster than 10Base-T.

10Base-T

An adaptation of the IEEE 802.3 (Ethernet) standard, the 10Base-T standard uses twisted-pair wiring with a maximum segment length of 100 m (328 ft) and terminates with an RJ-45 connector. A 10Base-T network is a baseband network capable of transmitting data at a maximum speed of 10 Mbit/s.

802.3 frame

A frame format, specified in the IEEE 802.3 (Ethernet) standard, in which the header specifies the data packet length.

A

agent

1. SNMP – the SNMP application that runs on a network device.
2. Fipio – a slave device on a network.

analog input

A module that contains circuits that convert analog DC input signals to digital values that can be manipulated by the processor. By implication, these analog inputs are direct. That means a data table value directly reflects the analog signal value.

analog output

A module that contains circuits that transmit an analog DC signal proportional to a digital value input to the module from the processor. By implication, these analog outputs are direct. That means a data table value directly controls the analog signal value.

application object

In CAN-based networks, application objects represent device-specific functionality, such as the state of input or output data.

ARP

The ARP (address resolution protocol) is the IP network layer protocol, which uses ARP to map an IP address to a MAC (hardware) address.

auto baud

The automatic assignment and detection of a common baud rate as well as the ability of a device on a network to adapt to that rate.

auto-addressing

The assignment of an address to each Island bus I/O module and preferred device.

auto-configuration

The ability of Island modules to operate with predefined default parameters. A configuration of the Island bus based completely on the actual assembly of I/O modules.

B

basic I/O

Low-cost Advantys STB input/output modules that use a fixed set of operating parameters. A basic I/O module cannot be reconfigured with the Advantys Configuration Software and cannot be used in reflex actions.

basic network interface

A low-cost Advantys STB network interface module that supports up to 12 Advantys STB I/O modules. A basic NIM does not support the Advantys Configuration Software, reflex actions, nor the use of an HMI panel.

basic power distribution module

A low-cost Advantys STB PDM that distributes sensor power and actuator power over a single field power bus on the Island. The bus provides a maximum of 4 A total power. A basic PDM includes a 5 A fuse.

BootP

BootP (bootstrap protocol) is an UDP/IP protocol that allows an internet node to obtain its IP parameters based on its MAC address.

BOS

BOS stands for beginning of segment. When more than 1 segment of I/O modules is used in an Island, an STB XBE 1200 or an STB XBE 1300 BOS module is installed in the first position in each extension segment. Its job is to carry Island bus communications to and generate logic power for the modules in the extension segment. Which BOS module has to be selected depends on the module types that shall follow.

bus arbitrator

A master on a Fipio network.

C

CAN

The CAN (controller area network) protocol (ISO 11898) for serial bus networks is designed for the interconnection of smart devices (from multiple manufacturers) in smart systems for real-time industrial applications. CAN multi-master systems provide high data integrity through the implementation of broadcast messaging and advanced diagnostic mechanisms. Originally developed for use in automobiles, CAN is now used in a variety of industrial automation control environments.

CANopen protocol

An open industry standard protocol used on the internal communication bus. The protocol allows the connection of any enhanced CANopen device to the Island bus.

CI

This abbreviation stands for command interface.

CiA

CiA (CAN in Automation) is a non-profit group of manufacturers and users dedicated to developing and supporting CAN-based higher layer protocols.

CIP

Common Industrial Protocol. Networks that include CIP in the application layer can communicate seamlessly with other CIP-based networks. For example, the implementation of CIP in the application layer of an Ethernet TCP/IP network creates an EtherNet/IP environment. Similarly, CIP in the application layer of a CAN network creates a DeviceNet environment. Devices on an EtherNet/IP network can therefore communicate with devices on a DeviceNet network via CIP bridges or routers.

COB

A COB (communication object) is a unit of transportation (a message) in a CAN-based network. Communication objects indicate a particular functionality in a device. They are specified in the CANopen communication profile.

configuration

The arrangement and interconnection of hardware components within a system and the hardware and software selections that determine the operating characteristics of the system.

CRC

cyclic redundancy check. Messages that implement this detected error mechanism have a CRC field that is calculated by the transmitter according to the message's content. Receiving nodes recalculate the field. Disagreement in the two codes indicates a difference between the transmitted message and the one received.

CSMA/CS

carrier sense multiple access/collision detection. CSMA/CS is a MAC protocol that networks use to manage transmissions. The absence of a carrier (transmission signal) indicates that a network channel is idle. Multiple nodes may try to simultaneously transmit on the channel, which creates a collision of signals. Each node detects the collision and immediately terminates transmission. Messages from each node are retransmitted at random intervals until the frames are successfully transmitted.

D

DDXML

Device Description eXtensible Markup Language

device name

A customer-driven, unique logical personal identifier for an Ethernet NIM. A device name (or *role name*) is created when you combine the numeric rotary switch setting with the NIM (for example, STBNIP2212_010).

After the NIM is configured with a valid device name, the DHCP server uses it to identify the island at power up.

DeviceNet protocol

DeviceNet is a low-level, connection-based network that is based on CAN, a serial bus system without a defined application layer. DeviceNet, therefore, defines a layer for the industrial application of CAN.

DHCP

dynamic host configuration protocol. A TCP/IP protocol that allows a server to assign an IP address based on a device name (host name) to a network node.

differential input

A type of input design where two wires (+ and -) are run from each signal source to the data acquisition interface. The voltage between the input and the interface ground are measured by two high-impedance amplifiers, and the outputs from the two amplifiers are subtracted by a third amplifier to yield the difference between the + and - inputs. Voltage common to both wires is thereby removed. When ground differences exist, use differential signalling instead of single ended signalling to help reduce cross channel noise.

digital I/O

An input or output that has an individual circuit connection at the module corresponding directly to a data table bit or word that stores the value of the signal at that I/O circuit. It allows the control logic to have discrete access to the I/O values.

DIN

Deutsche industrial norms. A German agency that sets engineering and dimensional standards and now has worldwide recognition.

Drivecom Profile

The Drivecom profile is part of CiA DSP 402 (profile), which defines the behavior of drives and motion control devices on CANopen networks.

E

economy segment

A special type of STB I/O segment created when an STB NCO 1113 economy CANopen NIM is used in the first location. In this implementation, the NIM acts as a simple gateway between the I/O modules in the segment and a CANopen master. Each I/O module in an economy segment acts as a independent node on the CANopen network. An economy segment cannot be extended to other STB I/O segments, preferred modules or enhanced CANopen devices.

EDS

electronic data sheet. The EDS is a standardized ASCII file that contains information about a network device's communications functionality and the contents of its object dictionary. The EDS also defines device-specific and manufacturer-specific objects.

EIA

Electronic Industries Association. An organization that establishes electrical/electronic and data communication standards.

EMC

electromagnetic compatibility. Devices that meet EMC requirements can operate within a system's expected electromagnetic limits without interruption.

EMI

electromagnetic interference. EMI can cause an interruption or disturbance in the performance of electronic equipment. It occurs when a source electronically transmits a signal that interferes with other equipment.

EOS

This abbreviation stands for end of segment. When more than 1 segment of I/O modules is used in an Island, an STB XBE 1000 or an STB XBE 1100 EOS module is installed in the last position in every segment that has an extension following it. The EOS module extends Island bus communications to the next segment. Which EOS module has to be selected depends on the module types that shall follow.

Ethernet

A LAN cabling and signaling specification used to connect devices within a defined area, e.g., a building. Ethernet uses a bus or a star topology to connect different nodes on a network.

Ethernet II

A frame format in which the header specifies the packet type, Ethernet II is the default frame format for NIM communications.

EtherNet/IP

EtherNet/IP (the Ethernet Industrial Protocol) is especially suited to factory applications in which there is a need to control, configure, and monitor events within an industrial system. The ODVA-specified protocol runs CIP (the Common Industrial Protocol) on top of standard Internet protocols, like TCP/IP and UDP. It is an open local (communications) network that enables the interconnectivity of all levels of manufacturing operations from the plant's office to the sensors and actuators on its floor.

F

fallback state

A known state to which an Advantys STB I/O module can return in the event that its communication connection is not open.

fallback value

The value that a device assumes during fallback. Typically, the fallback value is either configurable or the last stored value for the device.

FED_P

Fipio extended device profile. On a Fipio network, the standard device profile type for agents whose data length is more than 8 words and equal to or less than 32 words.

Fipio

Fieldbus Interface Protocol (FIP). An open fieldbus standard and protocol that conforms to the FIP/World FIP standard. Fipio is designed to provide low-level configuration, parameterization, data exchange, and diagnostic services.

Flash memory

Flash memory is nonvolatile memory that can be overwritten. It is stored on a special EEPROM that can be erased and reprogrammed.

FRD_P

Fipio reduced device profile. On a Fipio network, the standard device profile type for agents whose data length is two words or less.

FSD_P

Fipio standard device profile. On a Fipio network, the standard device profile type for agents whose data length is more than two words and equal to or less than 8 words.

full scale

The maximum level in a specific range—e.g., in an analog input circuit the maximum allowable voltage or current level is at full scale when any increase beyond that level is over-range.

function block

A function block performs a specific automation function, such as speed control. A function block comprises configuration data and a set of operating parameters.

function code

A function code is an instruction set commanding 1 or more slave devices at a specified address(es) to perform a type of action, e.g., read a set of data registers and respond with the content.

G

gateway

A program or hardware that passes data between networks.

global_ID

global_identifier. A 16-bit integer that uniquely identifies a device's location on a network. A global_ID is a symbolic address that is universally recognized by all other devices on the network.

GSD

generic slave data (file). A device description file, supplied by the device's manufacturer, that defines a device's functionality on a Profibus DP network.

H

HMI

human-machine interface. An operator interface, graphical, for industrial equipment.

hot swapping

Replacing a component with a like component while the system remains operational. When the replacement component is installed, it begins to function automatically.

HTTP

hypertext transfer protocol. The protocol that a web server and a client browser use to communicate with one another.

I

I/O base

A mounting device, designed to seat an Advantys STB I/O module, connect it on a DIN rail, and connect it to the Island bus. It provides the connection point where the module can receive either 24 VDC or 115/230 VAC from the input or output power bus distributed by a PDM.

I/O module

In a programmable controller system, an I/O module interfaces directly to the sensors and actuators of the machine/process. This module is the component that mounts in an I/O base and provides electrical connections between the controller and the field devices. Normal I/O module capacities are offered in a variety of signal levels and capacities.

I/O scanning

The continuous polling of the Advantys STB I/O modules performed by the COMS to collect data bits, status, nd diagnostics information.

IEC

International Electrotechnical Commission Carrier. Founded in 1884 to focus on advancing the theory and practice of electrical, electronics, and computer engineering, and computer science. EN 61131-2 is the specification that deals with industrial automation equipment.

IEC type 1 input

Type 1 digital inputs support sensor signals from mechanical switching devices such as relay contacts and push buttons operating in normal environmental conditions.

IEC type 2 input

Type 2 digital inputs support sensor signals from solid state devices or mechanical contact switching devices such as relay contacts, push buttons (in normal or harsh environmental conditions), and 2- or 3-wire proximity switches.

IEC type 3 input

Type 3 digital inputs support sensor signals from mechanical switching devices such as relay contacts, push buttons (in normal-to-moderate environmental conditions), 3-wire proximity switches and 2-wire proximity switches that have:

- a voltage drop of no more than 8 V
- a minimum operating current capability less than or equal to 2.5 mA
- a maximum off-state current less than or equal to 1.5 mA

IEEE

Institute of Electrical and Electronics Engineers, Inc. The international standards and conformity assessment body for all fields of electrotechnology, including electricity and electronics.

IGMP

(Internet group management protocol). This Internet standard for multicasting allows a host to subscribe to a particular multicast group.

industrial I/O

An Advantys STB I/O module designed at a moderate cost for typical continuous, high-duty-cycle applications. Modules of this type often feature standard IEC threshold ratings, providing user-configurable parameter options, on-board protection, good resolution, and field wiring options. They are designed to operate in moderate-to-high temperature ranges.

input filtering

The amount of time that a sensor has to hold its signal on or off before the input module detects the change of state.

input polarity

An input channel’s polarity determines when the input module sends a 1 and when it sends a 0 to the master controller. If the polarity is *normal*, an input channel sends a 1 to the controller when its field sensor turns on. If the polarity is *reverse*, an input channel sends a 0 to the controller when its field sensor turns on.

input response time

The time it takes for an input channel to receive a signal from the field sensor and put it on the Island bus.

INTERBUS protocol

The INTERBUS fieldbus protocol observes a master/slave network model with an active ring topology, having all devices integrated in a closed transmission path.

IOC object

Island operation control object. A special object that appears in the CANopen object dictionary when the remote virtual placeholder option is enabled in a CANopen NIM. It is a 16-bit word that provides the fieldbus master with a mechanism for issuing reconfiguration and start requests.

IOS object

Island operation status object. A special object that appears in the CANopen object dictionary when the remote virtual placeholder option is enabled in a CANopen NIM. It is a 16-bit word that reports the success of reconfiguration and start requests or records diagnostic information in the event that a request is not completed.

IP

internet protocol. That part of the TCP/IP protocol family that tracks the internet addresses of nodes, routes outgoing messages, and recognizes incoming messages.

IP Rating

Ingress Protection rating according to IEC 60529. Each IP rating requires the following standards to be met with respect to a rated device:

- IP20 modules are protected against ingress and contact of objects larger than 12.5 mm. The module is not protected against harmful ingress of water.
- IP67 modules are completely protected against ingress of dust and contact. Ingress of water in harmful quantity is not possible when the enclosure is immersed in water up to 1 m.

L

LAN

local area network. A short-distance data communications network.

light industrial I/O

An Advantys STB I/O module designed at a low cost for less rigorous (e.g., intermittent, low-duty-cycle) operating environments. Modules of this type operate in lower temperature ranges with lower qualification and agency requirements and limited on-board protection; they have limited or no user-configuration options.

linearity

A measure of how closely a characteristic follows a straight-line function.

LSB

least significant bit, least significant byte. The part of a number, address, or field that is written as the rightmost single value in conventional hexadecimal or binary notation.

M

MAC address

media access control address. A 48-bit number, unique on a network, that is programmed into each network card or device when it is manufactured.

mandatory module

When an Advantys STB I/O module is configured to be mandatory, it should be present and healthy in the Island configuration for the Island to be operational. If a mandatory module is inoperable or is removed from its location on the Island bus, the Island goes to a pre-operational state. By default, all I/O modules are not mandatory. You should use the Advantys Configuration Software to set this parameter.

master/slave model

The direction of control in a network that implements the master/slave model is from the master to the slave devices.

Modbus

Modbus is an application layer messaging protocol. Modbus provides client and server communications between devices connected on different types of buses or networks. Modbus offers many services specified by function codes.

MOV

metal oxide varistor. A 2-electrode semiconductor device with a voltage-dependant nonlinear resistance that drops markedly as the applied voltage is increased. It is used to suppress transient voltage surges.

MSB

most significant bit, most significant byte. The part of a number, address, or field that is written as the leftmost single value in conventional hexadecimal or binary notation.

N

N.C. contact

normally closed contact. A relay contact pair that is closed when the relay coil is de-energized and open when the coil is energized.

N.O. contact

normally open contact. A relay contact pair that is open when the relay coil is de-energized and closed when the coil is energized.

NEMA

National Electrical Manufacturers Association

network cycle time

The time that a master requires to complete a single scan of the configured I/O modules on a network device; typically expressed in microseconds.

NIM

network interface module. This module is the interface between an Island bus and the fieldbus network of which the Island is a part. A NIM enables all the I/O on the Island to be treated as a single node on the fieldbus. The NIM also provides 5 V of logic power to the Advantys STB I/O modules in the same segment as the NIM.

NMT

network management. NMT protocols provide services for network initialization, diagnostic control, and device status control.

O

object dictionary

Part of the CANopen device model that provides a map to the internal structure of CANopen devices (according to CANopen profile DS-401). A device's object dictionary (also called the *object directory*) is a lookup table that describes the data types, communications objects, and application objects the device uses. By accessing a particular device's object dictionary through the CANopen fieldbus, you can predict its network behavior and build a distributed application.

ODVA

Open Devicenet Vendors Association. The ODVA supports the family of network technologies that are built on the Common Industrial Protocol (EtherNet/IP, DeviceNet, and CompoNet).

open industrial communication network

A distributed communication network for industrial environments based on open standards (EN 50235, EN50254, and EN50170, and others) that allows the exchange of data between devices from different manufacturers.

output filtering

The amount that it takes an output channel to send change-of-state information to an actuator after the output module has received updated data from the NIM.

output polarity

An output channel's polarity determines when the output module turns its field actuator on and when it turns the actuator off. If the polarity is *normal*, an output channel turns its actuator on when the master controller sends it a 1. If the polarity is *reverse*, an output channel turns its actuator on when the master controller sends it a 0.

output response time

The time it takes for an output module to take an output signal from the Island bus and send it to its field actuator.

P

parameterize

To supply the required value for an attribute of a device at run-time.

PDM

power distribution module. A module that distributes either AC or DC field power to a cluster of I/O modules directly to its right on the Island bus. A PDM delivers field power to the input modules and the output modules. It is important that all the I/O installed directly to the right of a PDM be in the same voltage group—either 24 VDC, 115 VAC, or 230 VAC.

PDO

process data object. In CAN-based networks, PDOs are transmitted as unconfirmed broadcast messages or sent from a producer device to a consumer device. The transmit PDO from the producer device has a specific identifier that corresponds to the receive PDO of the consumer devices.

PE

protective ground. A return line across the bus to keep improper currents generated at a sensor or actuator device out of the control system.

peer-to-peer communications

In peer-to-peer communications, there is no master/slave or client/server relationship. Messages are exchanged between entities of comparable or equivalent levels of functionality, without having to go through a third party (like a master device).

PLC

programmable logic controller. The PLC is the brain of an industrial manufacturing process. It automates a process as opposed to relay control systems. PLCs are computers suited to survive the harsh conditions of the industrial environment.

PowerSuite Software

PowerSuite Software is a tool for configuring and monitoring control devices for electric motors, including ATV31x, ATV71, and TeSys U.

preferred module

An I/O module that functions as an auto-addressable device on an Advantys STB Island but is not in the same form factor as a standard Advantys STB I/O module and therefore does not fit in an I/O base. A preferred device connects to the Island bus via an EOS module and a length of a preferred module extension cable. It can be extended to another preferred module or back into a BOS module. If it is the last device on the Island, it should be terminated with a 120 Ω terminator.

premium network interface

A premium NIM has advanced features over a standard or basic NIM.

prioritization

An optional feature on a standard NIM that allows you to selectively identify digital input modules to be scanned more frequently during a the NIM’s logic scan.

process I/O

An Advantys STB I/O module designed for operation at extended temperature ranges in conformance with IEC type 2 thresholds. Modules of this type often feature high levels of on-board diagnostics, high resolution, user-configurable parameter options, and higher levels of agency approval.

process image

A part of the NIM firmware that serves as a real-time data area for the data exchange process. The process image includes an input buffer that contains current data and status information from the Island bus and an output buffer that contains the current outputs for the Island bus, from the fieldbus master.

producer/consumer model

In networks that observe the producer/consumer model, data packets are identified according to their data content rather than by their node address. All nodes *listen* on the network and consume those data packets that have appropriate identifiers.

Profibus DP

Profibus Decentralized Peripheral. An open bus system that uses an electrical network based on a shielded 2-wire line or an optical network based on a fiber-optic cable. DP transmission allows for high-speed, cyclic exchange of data between the controller CPU and the distributed I/O devices.

Q

QoS

(quality of service). The practice of assigning different priorities to traffic types for the purpose of regulating data flow on the network. In an Industrial network, QoS can help provide a predictable level of network performance.

R

reflex action

A simple, logical command function configured locally on an Island bus I/O module. Reflex actions are executed by Island bus modules on data from various Island locations, like input and output modules or the NIM. Examples of reflex actions include compare and copy operations.

repeater

An interconnection device that extends the permissible length of a bus.

reverse polarity protection

Use of a diode in a circuit to help protect against damage and unintended operation in the event that the polarity of the applied power is accidentally reversed.

rms

root mean square. The effective value of an alternating current, corresponding to the DC value that produces the same heating effect. The rms value is computed as the square root of the average of the squares of the instantaneous amplitude for 1 complete cycle. For a sine wave, the rms value is 0.707 times the peak value.

role name

A customer-driven, unique logical personal identifier for an Ethernet NIM. A role name (or *device name*) is created when you:

- combine the numeric rotary switch setting with the NIM (for example, STBNIP2212_010), or . . .
- edit the **Device Name** setting in the NIM's embedded web server pages

After the NIM is configured with a valid role name, the DHCP server uses it to identify the island at power up.

RSTP

(rapid spanning tree protocol). Allows a network design to include spare (redundant) links that provide automatic backup paths when an active link becomes inoperable, without loops or manual enabling/disabling of backup links. Loops should be avoided because they result in flooding the network.

RTD

resistive temperature detect. An RTD device is a temperature transducer composed of conductive wire elements typically made of platinum, nickel, copper, or nickel-iron. An RTD device provides a variable resistance across a specified temperature range.

RTP

run-time parameters. RTP lets you monitor and modify selected I/O parameters and Island bus status registers of the NIM while the Advantys STB Island is running. The RTP feature uses 5 reserved output words in the NIM’s process image (the RTP request block) to send requests, and 4 reserved input words in the NIM’s process image (the RTP response block) to receive responses. Available only in standard NIMs running firmware version 2.0 or higher.

Rx

reception. For example, in a CAN-based network, a PDO is described as an RxPDO of the device that receives it.

S

SAP

service access point. The point at which the services of 1 communications layer, as defined by the ISO OSI reference model, is made available to the next layer.

SCADA

supervisory control and data acquisition. Typically accomplished in industrial settings by means of microcomputers.

SDO

service data object. In CAN-based networks, SDO messages are used by the fieldbus master to access (read/write) the object directories of network nodes.

segment

A group of interconnected I/O and power modules on an Island bus. An Island should have at least 1 segment and, depending on the type of NIM used, may have as many as 7 segments. The first (leftmost) module in a segment needs to provide logic power and Island bus communications to the I/O modules on its right. In the primary or basic segment, that function is filled by a NIM. In an extension segment, that function is filled by an STB XBE 1200 or an STB XBE 1300 BOS module.

SELV

safety extra low voltage. A secondary circuit designed so that the voltage between any 2 accessible parts (or between 1 accessible part and the PE terminal for Class 1 equipment) does not exceed a specified value under normal conditions or under single-fault conditions.

SIM

subscriber identification module. Originally intended for authenticating users of mobile communications, SIMs now have multiple applications. In Advantys STB, configuration data created or modified with the Advantys Configuration Software can be stored on a SIM (referred to as the “removable memory card”) and then written to the NIM’s Flash memory.

single-ended inputs

An analog input design technique whereby a wire from each signal source is connected to the data acquisition interface, and the difference between the signal and ground is measured. For the success of this design technique, 2 conditions are imperative: the signal source should be grounded, and the signal ground and data acquisition interface ground (the PDM lead) should have the same potential.

sink load

An output that, when turned on, receives DC current from its load.

size 1 base

A mounting device, designed to seat an STB module, install it on a DIN rail, and connect it to the Island bus. It is 13.9 mm (0.55 in.) wide and 128.25 mm (5.05 in.) high.

size 2 base

A mounting device, designed to seat an STB module, install it on a DIN rail, and connect it to the Island bus. It is 18.4 mm (0.73 in.) wide and 128.25 mm (5.05 in.) high.

size 3 base

A mounting device, designed to seat an STB module, install it on a DIN rail, and connect it to the Island bus. It is 28.1 mm (1.11 in.) wide and 128.25 mm (5.05 in.) high.

slice I/O

An I/O module design that combines a small number of channels (between 2 and 6) in a small package. The idea is to allow a system developer to purchase just the right amount of I/O and to be able to distribute it around the machine in an efficient, mechatronics way.

SM_MPS

state management_message periodic services. The applications and network management services used for process control, data exchange, diagnostic message reporting, and device status notification on a Fipio network.

SNMP

simple network management protocol. The UDP/IP standard protocol used to manage nodes on an IP network.

snubber

A circuit generally used to suppress inductive loads—it consists of a resistor in series with a capacitor (in the case of an RC snubber) and/or a metal-oxide varistor placed across the AC load.

source load

A load with a current directed into its input; has to be driven by a current source.

standard I/O

Any of a subset of Advantys STB input/output modules designed at a moderate cost to operate with user-configurable parameters. A standard I/O module may be reconfigured with the Advantys Configuration Software and, in most cases, may be used in reflex actions.

standard network interface

An Advantys STB network interface module designed at moderate cost to support the configuration capabilities, multi-segment design and throughput capacity suitable for most standard applications on the Island bus. An Island run by a standard NIM can support up to 32 addressable Advantys STB and/or preferred I/O modules, up to 12 of which may be standard CANopen devices.

standard power distribution module

An Advantys STB module that distributes sensor power to the input modules and actuator power to the output modules over two separate power buses on the Island. The bus provides a maximum of 4 A to the input modules and 8 A to the output modules. A standard PDM requires a 5 A fuse for the input modules and an 8 A fuse for the outputs.

STD_P

standard profile. On a Fipio network, a standard profile is a fixed set of configuration and operating parameters for an agent device, based on the number of modules that the device contains and the device's total data length. There are 3 types of standard profiles: Fipio reduced device profile (FRD_P), Fipio standard device profile (FSD_P), and the Fipio extended device profile (FED_P).

stepper motor

A specialized DC motor that allows discrete positioning without feedback.

subnet

A part of a network that shares a network address with the other parts of a network. A subnet may be physically and/or logically independent of the rest of the network. A part of an internet address called a subnet number, which is ignored in IP routing, distinguishes the subnet.

surge suppression

The process of absorbing and clipping voltage transients on an incoming AC line or control circuit. Metal-oxide varistors and specially designed RC networks are frequently used as surge suppression mechanisms.

T

TC

thermocouple. A TC device is a bimetallic temperature transducer that provides a temperature value by measuring the voltage differential caused by joining together two different metals at different temperatures.

TCP

transmission control protocol. A connection-oriented transport layer protocol that provides full-duplex data transmission. TCP is part of the TCP/IP suite of protocols.

telegram

A data packet used in serial communication.

TFE

transparent factory Ethernet. Schneider Electric’s open automation framework based on TCP/IP.

Tx

transmission. For example, in a CAN-based network, a PDO is described as a TxPDO of the device that transmits it.

U

UDP

user datagram protocol. A connectionless mode protocol in which messages are delivered in a datagram to a destination computer. The UDP protocol is typically bundled with the Internet Protocol (UPD/IP).

V

varistor

A 2-electrode semiconductor device with a voltage-dependant nonlinear resistance that drops markedly as the applied voltage is increased. It is used to suppress transient voltage surges.

voltage group

A grouping of Advantys STB I/O modules, all with the same voltage requirement, installed directly to the right of the appropriate power distribution module (PDM) and separated from modules with different voltage requirements. Install modules with different voltage requirements in different voltage groups.

VPCR object

virtual placeholder configuration read object. A special object that appears in the CANopen object dictionary when the remote virtual placeholder option is enabled in a CANopen NIM. It provides a 32-bit subindex that represents the actual module configuration used in a physical Island.

VPCW object

virtual placeholder configuration write object. A special object that appears in the CANopen object dictionary when the remote virtual placeholder option is enabled in a CANopen NIM. It provides a 32-bit subindex where the fieldbus master can write a module reconfiguration. After the fieldbus writes to the VPCW subindex, it can issue a reconfiguration request to the NIM that begins the remote virtual placeholder operation.

W

watchdog timer

A timer that monitors a cyclical process and is cleared at the conclusion of each cycle. If the watchdog runs past its programmed time period, it reports a time-out.

Index