Advantys STB

Standard DeviceNet Network Interface Module

Applications Guide

8/2009

www.schneider-electric.com

Schneider Electric

Schneider Electric assumes no responsibility for any errors that may appear in this document. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

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.

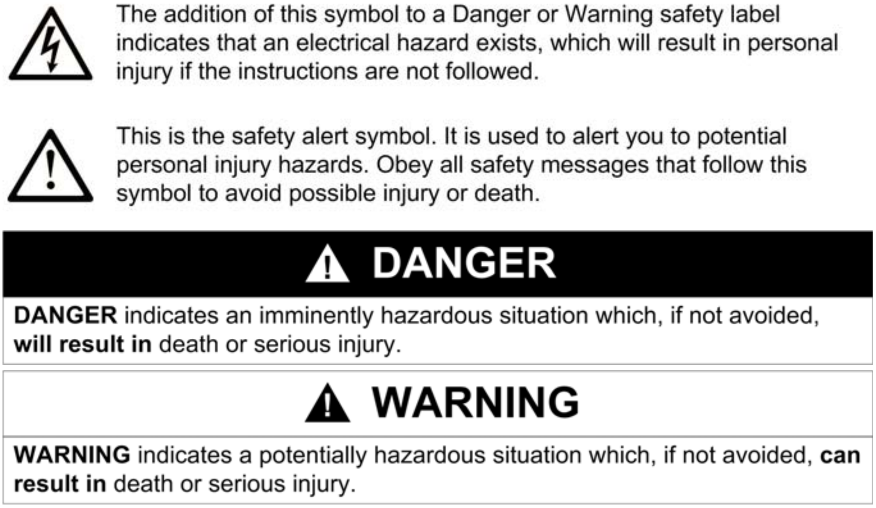
Table of Contents

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, 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 an imminently hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

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

⚠ **CAUTION**

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

CAUTION

CAUTION, used without the safety alert symbol, indicates a potentially hazardous situation which, if not avoided, **can result in** equipment damage.

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 the 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 standard network interface module, STB NDN 2212, for the DeviceNet open fieldbus. This NIM represents the Advantys STB island as a single node on a DeviceNet industrial network.

This guide includes the following information about the STB NDN 2212.

- role in a DeviceNet network
- role as the gateway to 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.

Related Documents

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

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

User Comments

We welcome your comments about this document. You can reach us by e-mail at techcomm@schneider-electric.com.

Introduction

1

Introduction

This chapter describes the STB NDN 2212 Advantys STB DeviceNet network interface module (NIM) and its support for the island as a DeviceNet network node.

The chapter begins with an introduction of the NIM and a discussion of its role as the gateway to the Advantys STB island. There is a brief overview of the island itself, followed by a description of the major characteristics of the DeviceNet fieldbus protocol.

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

What's 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 36)*. (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 must be attached to the NIM's CFG port.

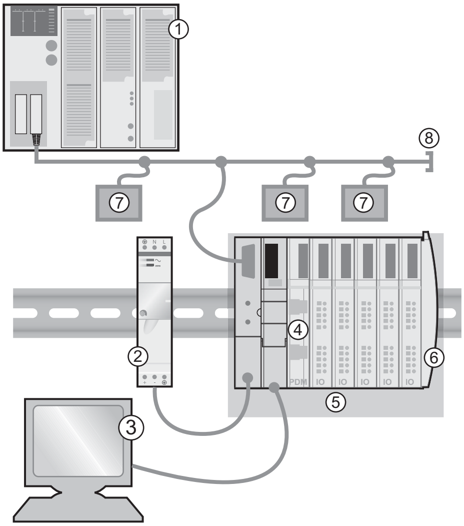
Integrated Power Supply

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 must 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

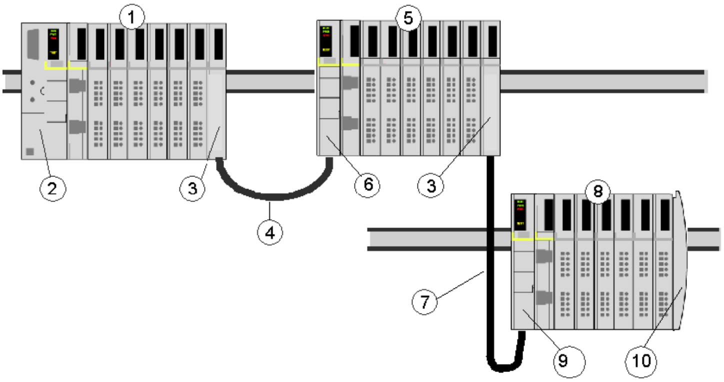
When you are using a standard NIM, Advantys STB I/O modules that do not reside in the primary segment can be installed in *extension segments*. Extension segments are optional segments that enable an island to be a truly distributed I/O system. The island bus can support as many as six extension segments.

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

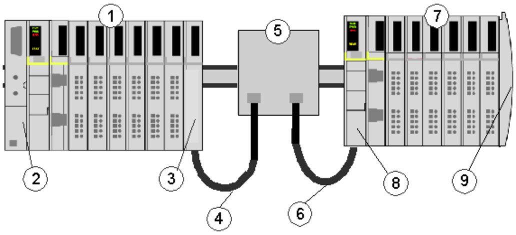
Bus extension cables are available in various lengths, ranging from 0.3 m (1 ft) to 14.0 m (45.9 ft).

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 must be enabled if a preferred module is the last device on the island bus and must 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:



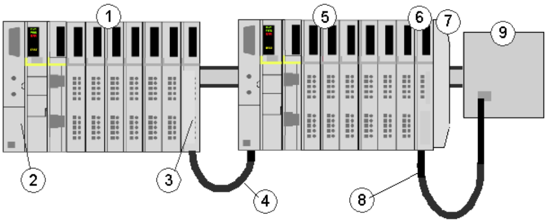
1 primary segment
2 NIM
3 STB XBE 1100 EOS bus extension module
4 1 m length STB XCA 1002 bus extension cable
5 preferred module
6 1 m length STB XCA 1002 bus extension cable
7 extension segment of Advantys STB I/O modules
8 STB XBE 1300 BOS bus extension module for the extension segment
9 STB XMP 1100 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 must 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 must 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 must 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 must take into account the extension cables between segments, extension cables between preferred modules, and the space consumed by the devices themselves.

About DeviceNet

Introduction

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.

ODVA (Open DeviceNet Vendor Association) creates specifications for DeviceNet networks and devices.

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

Physical Layer

DeviceNet's data link layer is defined by the CAN specification and by the implementation of widely available CAN controller chips. CAN also implements a differentially driven (common return), two-wire bus line.

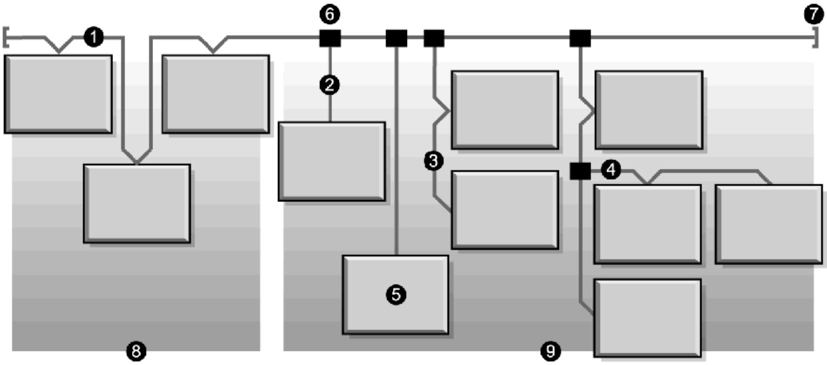
DeviceNet's physical layer contains two twisted pairs of shielded wires. One twisted pair is for transferring data and one is for supplying power. This results in simultaneous support for devices that receive power from the network (like sensors) and those that are self-powered (like actuators). Devices can be added or removed from the bus line without powering down the fieldbus.

Network Topology

DeviceNet supports a trunk line/drop line network configuration. The implementation of multiple, branched, zero, and daisy chained drops should be established during system design.

Power taps allow the connection of DeviceNet-compliant power supplies from a variety of manufacturers. The network must be terminated at each end with 120 Ω resistors.

A sample DeviceNet network topology is shown in the following figure:



1 trunk line
2 drop line (0 to 6 m)
3 daisy chain drop-off
4 branched drop-off
5 network node
6 trunk line tap junction
7 terminating resistor
8 zero drop
9 short drops

Transmission Media

Your implementation of thick, thin, or flat cables for trunk lines and drop lines should be established during system design. Thick cables are generally used for trunk lines. Thin cables can be used for trunk or drop lines.

Maximum Network Lengths

End-to-end network distance varies with data rate and cable size. The following table shows the range of bauds that the STB NDN 2212 DeviceNet NIM supports for CAN devices and the resulting maximum length of the DeviceNet network.

Cable Type	**125 kbits/s**	**250 kbits/s**	**500 kbits/s**
Thick Trunk	500 m	250 m	100 m
Thin Trunk	100 m	100 m	100 m
Flat Trunk	420 m	200 m	75 m
Maximum Drop Length	6 m	6 m	6 m
Cumulative Drop Length*	156 m	78 m	39 m
*The sum of the length of all drop lines.			

Node Limitations

A DeviceNet network is limited to 64 addressable nodes (node IDs 0 to 63).

Network Model

Like any broadcast communications network, DeviceNet operates within a producer/consumer model. Each data packet's identifier field defines the data priority and allows for efficient data transfer among multiple users. All nodes *listen* on the network for messages with identifiers that apply to their functionality. Messages sent by producer devices will be accepted only by designated consumer devices.

DeviceNet supports strobed, polled, cyclic, change of state, and application-triggered data exchange.

DeviceNet allows users to implement a master/slave, multimaster, or peer-to-peer network architecture (or some combination thereof), depending on the device's flexibility and your application requirements.

Connections

Because DeviceNet is a connection-based network, connections must be established between particular devices before the transfer of data between them can commence. Connections are established through either the unconnected message manager (UCMM) or an unconnected port. (The STB NDN 2212 Advantys STB DeviceNet NIM is a UCMM-capable device.)

The connection ID is defined in the CAN message's 11-bit identifier. The identifier field is divided into four prioritized message groups:

- *group 1*—Responses from DeviceNet nodes are typically in the form of these high-priority I/O messages *(see page 22)*.
- *group 2*—Generally, these medium-priority messages are used for simple master/slave messages.
- *group 3*—These low-priority messages are usually used for explicit messages *(see page 22)*.
- *group 4*—These messages of the lowest priority are reserved for future use.

Object Model

The DeviceNet specification is presented in terms of an abstract object model *(see page 64)* 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.

Messaging

The following connection types are established with DeviceNet's connection-based model:

- *I/O messaging*—I/O messages contain application-specific data. They are communicated across single or multicast connections between an application producer and its corresponding consuming application. Because I/O messages carry time-critical messages, they have high-priority identifiers.
- *explicit messaging connections*—Explicit messaging connections provide point-to-point communication paths between two particular devices. You can use explicit messaging connections to configure nodes and diagnose problems. Explicit messages contain I/O data only; they do not contain device-specific information.

Device Profiles

DeviceNet's 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 (electronic data sheet).

What's an EDS?

The EDS 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 OVDA). The EDS also defines device-specific and manufacturer-specific objects.

Using the EDS, you can standardize tools to:

- configure DeviceNet devices
- design networks for DeviceNet 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.

The STB NDN 2212 NIM Module

2

Introduction

This chapter describes the Advantys STB DeviceNet NIM's external features, connections, power requirements, and product specifications.

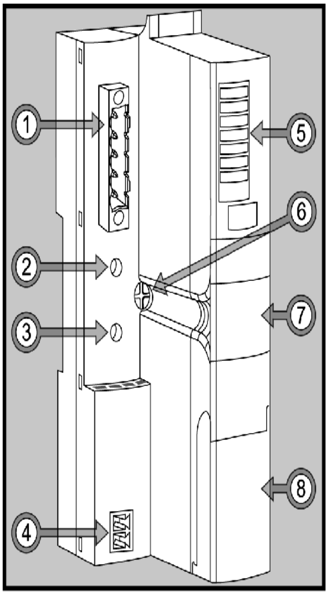
What's in this Chapter?

This chapter contains the following topics:

External Features of the STB NDN 2212 NIM

Introduction

The physical features of the STB NDN 2212 DeviceNet NIM are called out in the illustration below:



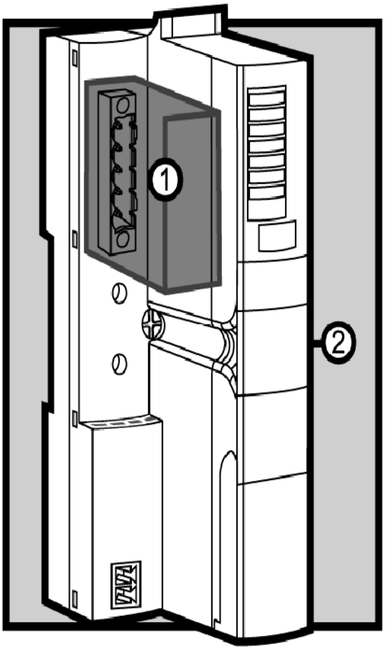
The features are described in the following table:

Feature		Function
1	fieldbus interface *(see page 26)*	a 5-pin open style connector used to connect the NIM and the island bus to a DeviceNet fieldbus
2	upper rotary switch	the two rotary switches *(see page 28)* are used together to specify the NIM’s node ID on the DeviceNet fieldbus
3	lower rotary switch	
4	power supply interface	a two-pin receptacle for connecting an external 24 VDC power supply to the NIM
5	LED array	colored LEDs *(see page 31)* that illuminate in various patterns to visually indicate the operational status of the island bus
6	release screw	a mechanism that needs to be turned if you need to remove the NIM from the DIN rail (see the *Automation Island System Planning and Installation Guide* for details)

Feature		Function
7	removable memory card *(see page 54)* drawer	a plastic drawer in which a removable memory card can be seated and inserted in the NIM
8	CFG port *(see page 36)* cover	a liftable lid on the NIM's front panel that covers the CFG interface and the RST button

Housing Shape

The L-shaped external housing of the NIM is designed to accommodate the attachment of a fieldbus connector without raising the depth profile of the island:



1 space reserved for the network connector
2 NIM housing

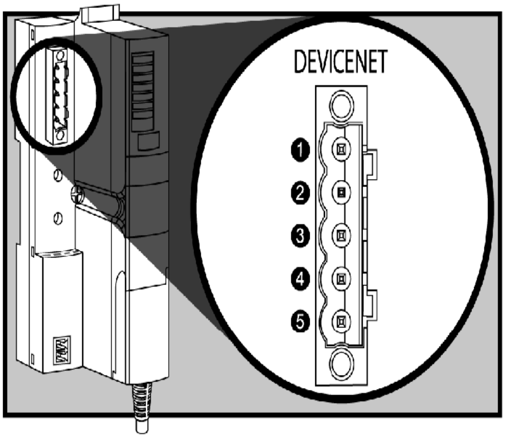
STB NDN 2212 Fieldbus Interface

Summary

The fieldbus interface on the STB NDN 2212 NIM is the point of connection between an Advantys STB island bus and the DeviceNet network. The interface is a male 5-pin open style connector on the face of the NIM.

Fieldbus Port Connections

The fieldbus interface is located on the front of the DeviceNet NIM at the top:



The table shows the pinout for the 5-pin open style connector:

Pin	Signal	Description	Color Code
1	V-	0 V supply	black
2	CAN_L	CAN-low bus line	blue
3	shield	shield	grey
4	CAN_H	CAN-high bus line	white
5	V+	11 . . . 25 V supply	red
Note: Pin numbers correspond to callouts in the figure above.			

DeviceNet Network Connectors

Any female network cable you connect to the Advantys STB CANopen NIM must observe the above pin assignment scheme (meeting ODVA specifications). Use either:

- STBXTS 1111 screw-type connector
- STBXTS 2111 spring connector

Baud

The DeviceNet NIM is not equipped with switches for setting the device’s baud. Instead, the baud will be set automatically by the device, itself.

NOTE: To obtain a new baud, cycle the power to the NIM.

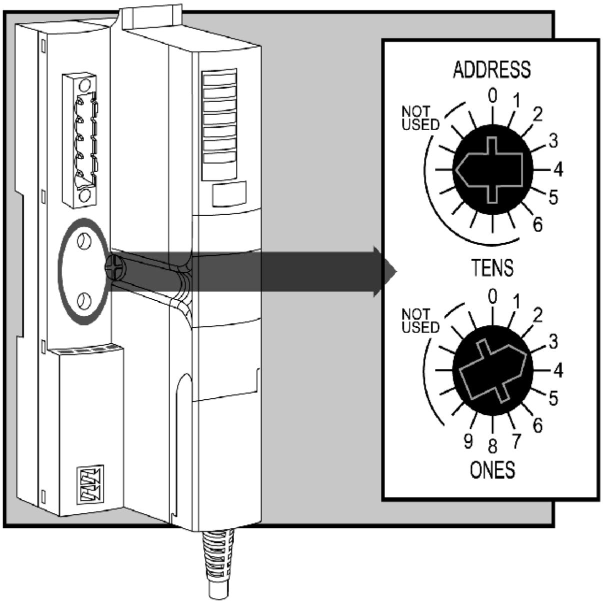
Rotary Switches: Setting the Network Node Address

Summary

As a single node on a DeviceNet network, the Advantys STB island requires a network address. The address can be any numeric from 0 to 63 that is unique with respect to other nodes on the network. The node address is set with a pair of rotary switches on the NIM module. The fieldbus master and the island bus can communicate over the DeviceNet network only while the rotary switches are set to a valid address *(see page 29)*.

Physical Description

The two rotary switches are located on the front of the DeviceNet NIM, below the fieldbus connection port. Each switch has sixteen positions.



The Node Address

Because the DeviceNet fieldbus master sees the Advantys STB island as *one* network node, the island has a single fieldbus network address. The NIM reads the node address from the rotary switches each time the island powers up. (It is not stored in Flash memory.)

Setting the Node Address

Instructions for setting the node address are in the table.

Step	Action	Comment
1	Bring the power down on the island.	The changes you are about to make will be detected only at the next power up.
2	Select a node address that is currently available on your fieldbus network.	Your list of active fieldbus nodes indicates whether a particular address is available.
3	With a small screwdriver, set the lower rotary switch to the position that represents the digit in the ones position of your selected node address.	For example, for a node address of 43, set the lower switch to 3.
4	With a small screwdriver, set the upper rotary switch to the position that represents the two digits in the tens and hundreds position of your selected node address.	For example, for a node address of 43, set the upper switch to 4.
5	Power up Advantys.	The NIM reads the rotary switch settings only during power up.

Using the Node Address

The node address is not stored in Flash memory. Instead, the NIM reads the node address from the rotary switches each time the island powers up. For this reason, it is best to leave the rotary switches set to the same address. This way, the fieldbus master identifies the island at the same node address at each power up.

Valid DeviceNet Node Addresses

Each rotary switch position that you can use to set the node address for your island is marked incrementally on the NIM housing. The available positions on each rotary switch are:

- upper switch—0 to 6 (tens digit)
- lower switch—0 to 9 (ones digit)

For example, in the figure *(see page 28)* at the beginning of this topic, an address of 43 is represented by the selection of 3 on the lower switch and 4 on the upper switch.

Note that it is *mechanically* possible to set any node address from 00 to 69, however, addresses 64 through 69 are not available because DeviceNet supports only 64 node addresses (0 to 63).

Communicating on the Fieldbus

The NIM will only communicate with the fieldbus network while the rotary switches are set to a valid DeviceNet node address *(see page 29)*. If the combination of the switch settings represents an invalid DeviceNet address, the NIM will wait for you to set a node address before it begins to communicate on the fieldbus.

If the island has an invalid node address, it cannot communicate with the master. To establish communication, set the switches to a valid address and cycle power on the island.

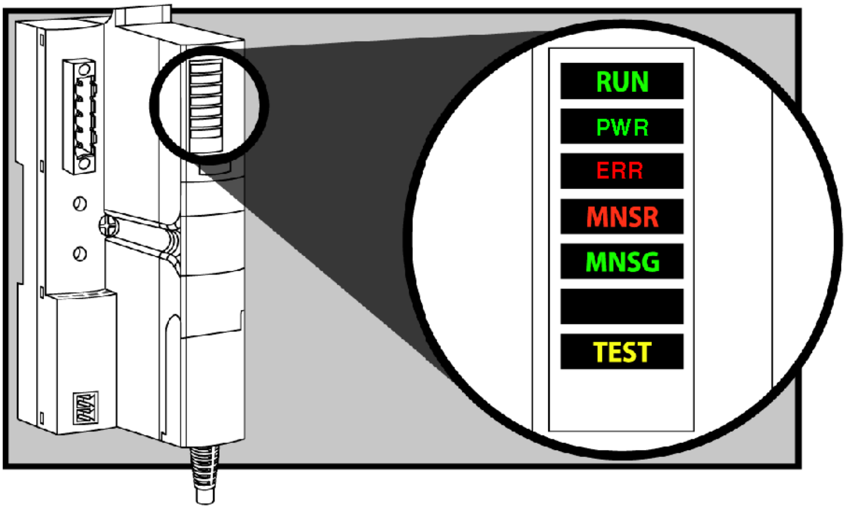
LED Indicators

LED Location

The six LEDs on the STB NDN 2212 DeviceNet NIM visually indicate the operating status of the island bus on a DeviceNet network. The LED array is located at the top of the NIM front bezel:

- LED 4 (MNSR) and LED 5 (MNSG) *(see page 32)* indicate the status of data exchange between the DeviceNet fieldbus master and the Advantys STB island bus.
- LEDs 1, 2, 3, and 7 indicate activity or events on the NIM. *(see page 33)*
- LED 6 is not used.

The figure shows the six LEDs used by the Advantys STB DeviceNet NIM:



Blink Patterns

Individual blinks are approximately 200 ms. There is a 1-second interval between blink sequences. For example:

- 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.

NOTE: It is assumed that the *PWR* LED is on continuously, indicating that the NIM is receiving adequate power. *(see page 33)* If the *PWR* LED is off, logic power *(see page 41)* to the NIM is off or insufficient.

DeviceNet Communications LEDs

The following table describes the indicated condition(s) and the colors and blink patterns that the MNSR and MNSG LEDs use to show normal operations and error conditions for an Advantys STB DeviceNet NIM on a DeviceNet fieldbus.

Label	Pattern	Meaning
MNSR (red)	blinking	Recoverable fault or one or more I/O connections are in the time-out state.
	on	The device has experienced an unrecoverable fault (for example, wrong baud, duplicate MAC ID, wiring problem), rendering it incapable of communicating on the network.
MNSG (green)	off	Device is not online: ● The device may not have completed the duplicate MAC ID test. ● The device may not be powered up.
	blinking	Device is operating in a normal condition and one of the following is true: ● The device is online with no connections in the established state. ● Configuration is missing, incomplete, or incorrect.
	on	Device is operating normally and the device is on-line with connections in the established state.

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 41)* 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 57)*.
		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 49)* or auto-configuring *(see page 53)* 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 detected a module assignment error; 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 50)* ● mandatory module is incorrectly configured *(see page 113)* ● process image is not valid ● device is incorrectly configured *(see page 53)* ● 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.			

Power LED

The PWR (power) LED indicates whether or not the STB NIC 2212's internal power supplies are operating at the correct voltages. The PWR LED is directly driven by the STB NIC 2212's reset circuitry.

The following table summarizes the PWR LED states:

Label	Pattern	Meaning
PWR	Steady on	The STB NIC 2212 internal voltages are all at or above their minimum voltage level.
PWR	Steady off	One or more of the STB NIC 2212 internal voltages is below minimum voltage level.

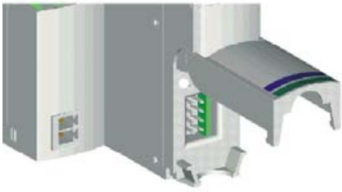
The CFG Interface

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, you must 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	RTU

NOTE: To restore all of the CFG port’s communication parameters to their factory default settings, push the RST button *(see page 60)* on the NIM. Be aware, however, that this action overwrites all of the island’s current configuration values with factory default values.

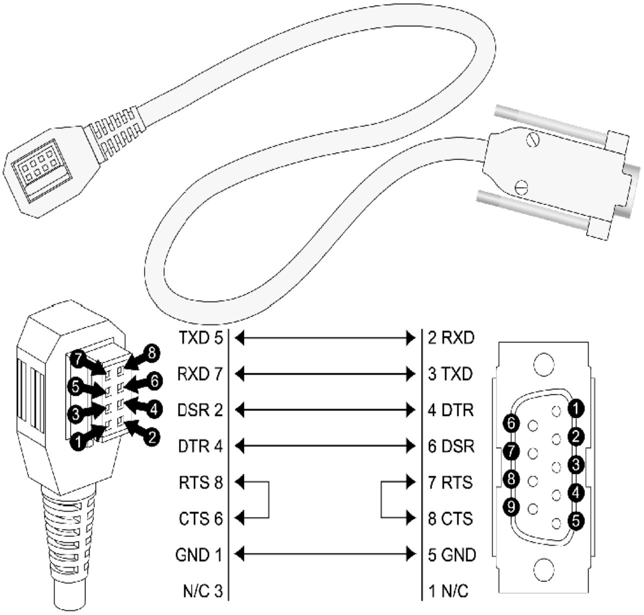
If you want to retain your configuration and still use the RST button to reset your port parameters, write the configuration to an STB XMP 4440 removable memory card *(see page 54)* and insert the card in its drawer in the NIM.

You can also password-protect a configuration *(see page 123)*. If you do this, however, the RST button is disabled and you are unable to use it to reset the port parameters.

Connections

An STB XCA 4002 programming cable must be used 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

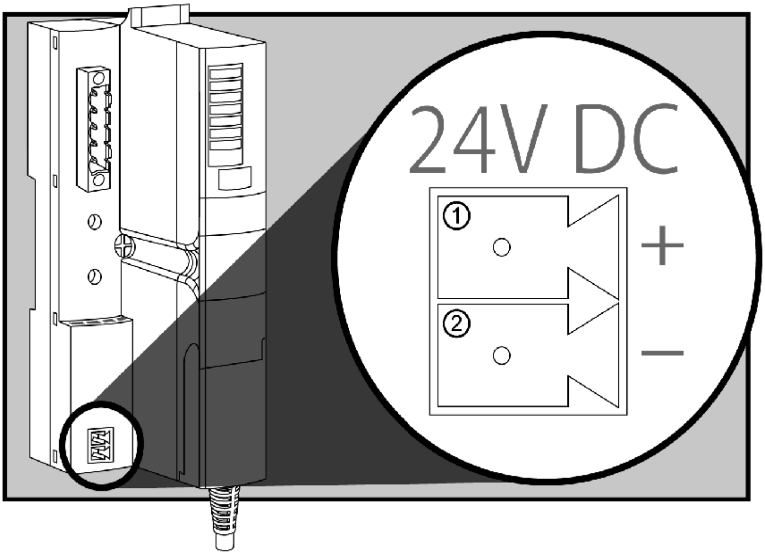
Power Supply Interface

Introduction

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 Advantys STB island is the two-receptacle connector illustrated below.

Physical Description

Power from the external 24 VDC supply comes in to the NIM through a two-receptacle connector located at the bottom left of the module:



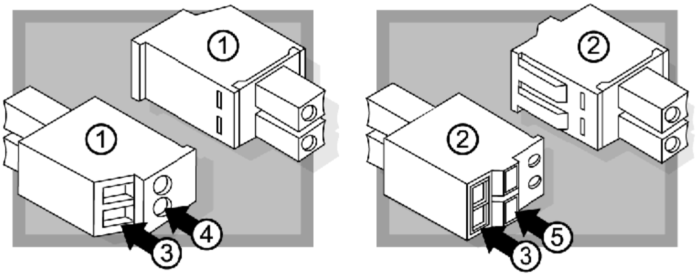
1 receptacle 1—24 VDC
2 receptacle 2—common voltage

Connectors

Use either:

- a *screw type* power connector, available in a kit of 10 (model STB XTS 1120)
- a *spring clamp* power connector, available in a kit of 10 (model STB XTS 2120)

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 STBXTS 1120 screw-type power connector
2 STBXTS 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 to1.5 mm^2 (28 to 16 AWG). Each connector has a 3.8 mm (0.15 in) pitch between the receptacles.

We recommend that you trim 10 mm from the wire’s jacket to make this connection.

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.

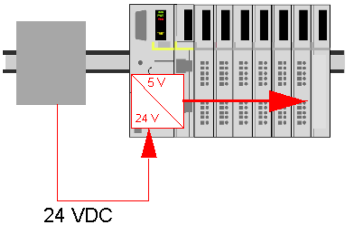
External Source Power

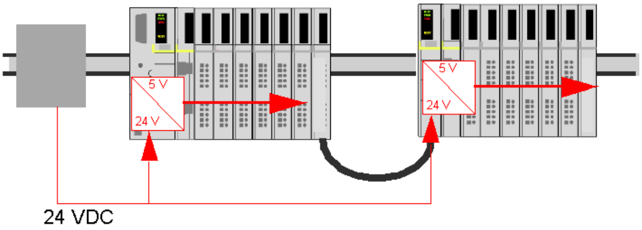
> ⚠ **CAUTION**
>
> **IMPROPER GALVANIC ISOLATION**
>
> **Failure to follow these instructions can result in injury or equipment damage.**

Input from an external 24 VDC power supply *(see page 43)* 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 must 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:





Island Bus Loads

Advantys STB System

Planning and Installation Guide

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

Logic Power Requirements

Characteristics of the External Power Supply

⚠ CAUTION

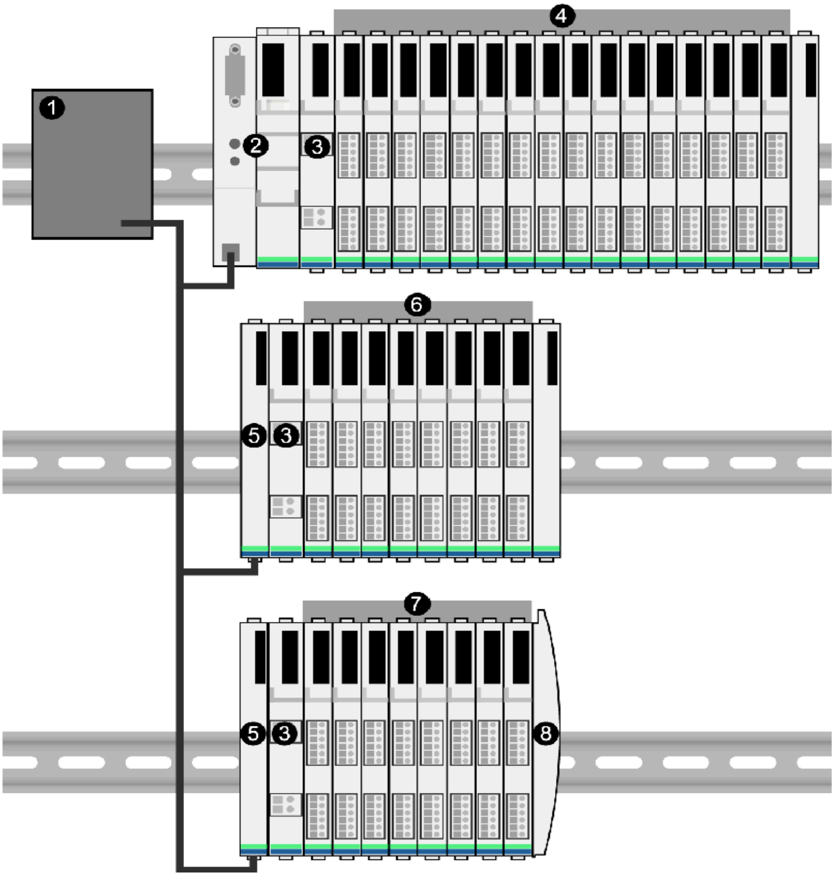
IMPROPER GALVANIC ISOLATION

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

safety extra low voltage

Calculating the Wattage Requirement

(see page 41)



1

2

3

4

5

6

7

8

-
-

NOTE:

amps *volts* = *watts*.

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.

The following information describes the general specifications for the NIM.

The following table lists the system specifications for the STB NDN 2212 DeviceNet NIM:

dimensions	width	40.5 mm (1.59 in)
	height	130 mm (5.12 in)
	depth	70 mm (3.15 in)
interface connectors	to the DeviceNet network	5-pin open style connector (male)
	RS-232 port for configuration software or HMI panel	8-receptacle HE-13
	to the external 24 VDC power supply	2-receptacle
built-in power supply	input voltage	24 VDC nominal
	input power range	19.2 ... 30 VDC
	input current	400 mA @ 24 VDC
	output voltage to the island bus	5 VDC @ 1.2 A
	output current rating	5 VDC @ 1.2 A
	isolation	no internal isolation (isolation must be provided by a SELV-rated external 24 VDC source power supply.)
	noise immunity (EMC)	EN 61131-2
DeviceNet power	input voltage	24 VDC nominal
	input power range	11 . . . 25 VDC
	input current	10 mA (maximum) / 4.5 mA (typical) @ 24 VDC
addressable I/O modules supported	per segment	16 maximum
	per island	32 maximum
segments supported	primary (required)	one
	extension (optional)	six maximum
standards	DeviceNet conformance	Open DeviceNet Vendors Assoc.
	MTBF	200,000 hours GB (ground benign)
storage temperature		-40 to 85° C
operating temperature range*		0 to 60° C

agency certifications	refer to the *Advantys STB System Planning and Installation Guide, 890 USE 171 00*
*This product supports operation at normal and extended temperature ranges. Refer to the *Advantys STB System Planning and Installation Guide, 890 USE 171 00* for a complete summary of cabalities and limitations.	

How to Configure the Island

3

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's 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

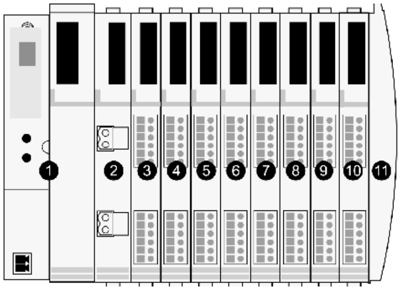
Advantys STB I/O modules and preferred devices are auto-addressable. Enhanced CANopen modules are not auto-addressable. They require manual address settings.

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 .
- 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
- 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

⚠ **CAUTION**

LOSS OF CONFIGURATION: MEMORY CARD DAMAGE OR CONTAMINATION

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

- Use care when handling the card.
- Inspect for contamination, physical damage, and scratches before installing the card in the NIM drawer.
- If the card does get dirty, clean it with a soft dry cloth.

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

The STB XMP 4440 removable memory card is a 32-kbyte subscriber identification module (SIM) that lets you store , 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.

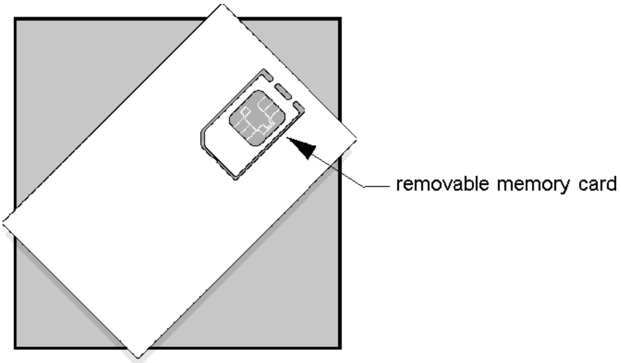
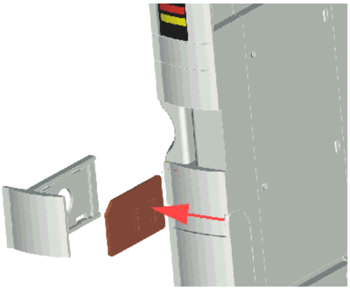
The removable memory card is an optional Advantys STB feature.

Remember:

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

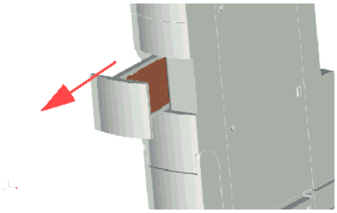
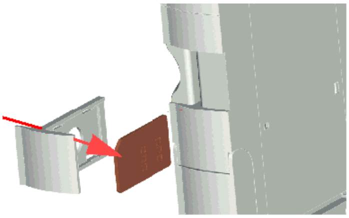
Installing the Card

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.  Make sure 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 must 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. As a handling precaution, 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 only if an island is in mode. If an island is in protected mode , 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 must have the same part number
- configuring multiple islands with the same configuration data

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

Edit Mode

Your island bus must 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.

Initial Configuration and Reconfiguration Scenarios

Use the following procedure to set up an island bus with configuration data that was previously saved to a removable memory card. You can use this procedure to configure a new island or to overwrite an existing configuration. (**NOTE:** 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 54)*.	
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 123)*, 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 . *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 must all have the same part number.

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 57)*.

Physical Description

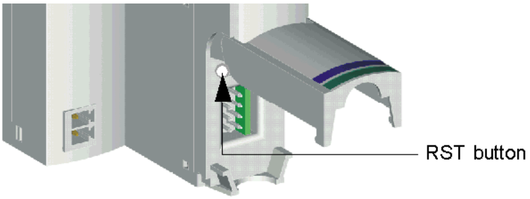
⚠ **CAUTION**

UNINTENDED EQUIPMENT OPERATION/CONFIGURATION OVERWRITTEN—RST BUTTON

Do not attempt to restart the island with the RST button. Pushing the RST button reconfigures the island with default settings (no custom parameters).

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

The RST button is located immediately above the CFG port *(see page 36)*, 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

⚠ CAUTION

UNINTENDED EQUIPMENT OPERATION/CONFIGURATION DATA OVERWRITTEN—RST BUTTON

Do not attempt to restart the island by pushing the RST button. Pushing the RST button *(see page 60)* causes the island bus to reconfigure itself with factory default operating parameters.

Failure to follow these instructions can result in 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 60)* for at least two seconds. The RST button is enabled only in edit mode. In protected mode *(see page 123)*, 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 36)*.
- Add a new I/O module to a previously auto-configured *(see page 53)* 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 56)*.
2	Place the island in edit mode *(see page 57)*.
3	Hold the RST button *(see page 60)* 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 50)* 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 36)*.
4	It re-initializes the island bus and brings it into operational mode.

Fieldbus Communications Support

4

Introduction

This chapter describes how an Advantys STB island node can be accessed from other devices on a DeviceNet fieldbus network.

What's in this Chapter?

This chapter contains the following sections:

4.1 Object Model

Introduction

This section describes the object model for the DeviceNet NIM.

For general information about the object model for a particular DeviceNet device, refer to ODVA specifications.

What's in this Section?

This section contains the following topics:

Introduction to the Object Model

Introduction

A DeviceNet node is modeled as a collection of objects. Each object provides an abstract representation of a particular component within a product. Detailed descriptions of all supported classes and instances (and their attributes) are presented elsewhere in this section.

Addressing Object Attributes

Objects provide *services* and implement *behaviors*. 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

The table below lists the DeviceNet objects supported by the Advantys STB island:

Object Class	Class ID	Instance ID	Messages	Description
identity object	1	1	explicit	device type, vendor ID, serial number, etc.
DeviceNet object	3	1	explicit	maintains physical connection to DeviceNet; allocates/de-allocates the master/slave connection set
assembly object *(see page 70)*	4	100–103	explicit, I/O	provides collection of other object's attributes (frequently used for I/O messaging)
connection object *(see page 73)*	5	1–4, 5–14	explicit	allows explicit messages to be conducted
island bus object	101 (65h)	1	explicit	provides error/diagnostic data and I/O data to/from the DeviceNet NIM

Identity Object (Class ID 1)

Introduction

The identity object provides the configuration and status of the physical attachment of the Advantys STB DeviceNet NIM to the DeviceNet network.

Class Attributes

The following attributes are supported by the identity object class:

Attr. ID	Name	Data Type	Description	Value
1	revision	UINT	revision of identity object class definition	1

Class Services

The following class services are supported by the identity object class:

Service Code	Service Name	Description
0Eh	get_attribute_single	reads identity object class attribute value

Object Instance Attributes

The following table lists the attributes supported by the identity object:

Attr. ID	Name	Services	Data Type	Description
1	vendor ID	get	UINT	Schneider Electric’s ODVA-assigned vendor ID (243)
2	device type	get	UINT	identification of general product type, in the case of the Advantys STB island, distributed I/O (value = 12 [0Ch])
3	product code	get	UINT	product code (2212) for the Advantys STB DeviceNet NIM
4	revision	get	STRUCT of USINT USINT	revision of the Advantys STB DeviceNet NIM
	major revision			
	minor revision			
5	status	get	word	status of the Advantys STB DeviceNet NIM
6	serial number	get	UDINT	serial number of the Advantys STB DeviceNet NIM
7	product name	get	short string	*human readable identification*—number of bytes transferred in polled mode, formatted as *STB NDN 2212 IN<XX> OUT<YY>* where <XX> = number of input bytes and <YY> = number of output bytes
10	heartbeat interval	get/set	USINT	nominal interval between heartbeat messages in seconds (0, the default, disables the heartbeat)

Instance Services

The following instance services are supported by the identity object class:

Service Code	Service Name	Description
05h	reset	reset the NIM (similar to power up)
0Eh	get_attribute_single	reads identity object instance attribute value
10h	set_attribute_single	modifies identity object instance attribute value

DeviceNet Object (Class ID 3)

Introduction

The DeviceNet object sends the configuration and status data for the physical connection of the Advantys STB island's DeviceNet NIM to the fieldbus. By accessing the DeviceNet object, users can identify network information like the island node's baud and MAC ID.

Class Attributes

The following table lists the attributes supported by the DeviceNet object class:

Attr. ID	Name	Data Type	Description	Value
1	revision	UINT	revision of the DeviceNet object class definition	2

Class Services

The following class services are supported by the DeviceNet object class:

Service Code	Service Name	Description
0Eh	get_attribute_single	reads DeviceNet object class attribute value

Object Instance Attributes

The following table lists the attributes supported by the DeviceNet object:

Attr. ID	Name	Services	Data Type	Description
1	MAC ID	get	USINT	node address (0–63)
2	baud	get	USINT	device baud (0 = 125 k, 1 = 250 k, 2 = 500 k)
3	BOI	get/set	BOOl	bus-off interrupt (value = 0)
4	bus-off counter	get/set	USINT	diagnostic counter (0–255)
5	allocation information	get	structure of byte & USINT	slave allocation information—allocation choice (value = 18) and the master's MAC ID (either 0–63 or 255)

Instance Services

The following instance services are supported by the DeviceNet object class:

Service Code	Service Name	Description
0Eh	get_attribute_single	reads DeviceNet object instance attribute value
10h	set_attribute_single	modifies DeviceNet object instance attribute value
4Bh	allocate_master_slave_ connection_set	requests the use of a predefined master/slave connection
4Ch	release_master_slave_ connection_set	indicates that specified connections within the predefined master/slave connection are no longer desired (these connections are to be released)

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 DeviceNet NIM.

Assembly objects can be used to bind input data or output data, as defined from the network's perspective; an input will produce data *on* the network and an output will consume data *from* the network.

Class Attributes

The following table lists the attributes supported by the assembly object class:

Attr. ID	Name	Data Type	Description	Value
1	revision	UINT	revision of the assembly object class definition	2

Class Services

The following class services are supported by the assembly object class:

Service Code	Service Name	Description
0Eh	get_attribute_single	reads an assembly object class attribute value

Instances of Assembly Object

The Advantys STB DeviceNet NIM provides four instances of the assembly object class:

Instance ID	Data Type	Description
100	static input	diagnostic and error data from the Advantys STB system
101	static input	input process image data from the Advantys STB system
102	static output	output process image data from the Advantys STB system
103	static output	reserved

Object Instance Attributes

The following table lists the attributes supported by the assembly object:

Attr. ID	Name	Services	Data Type
3	member data	get/set	array of byte
100	extended member list	get	array of USINT
101	number of members in list	get	array of USINT
102	member list	get	array of STRUCT
	member data description		UINT
	member path size		UINT
	member path		EPATH

Instance Services

The following instance services are supported by the assembly object class:

Service Code	Service Name	Description
0Eh	get_attribute_single	reads assembly object instance attribute value
10h	set_attribute_single	modifies an assembly object instance attribute value
18h	get_member	reads one member of an assembly object instance attribute value

Instance ID 100: Diagnostic and Error Data from the Island Bus

Instance 100 of the assembly object class binds the diagnostic and error data from island bus object class ID 101 of the DeviceNet NIM to an input assembly.

The following table shows the island bus object (class ID 101) mapping for instance 100 (instance ID 1) to attribute 3:

Linked Object Class		Attribute		
ID	Name	ID	Name	Data Type
101	island bus object class	1	island bus state	word
101	island bus object class	2	global diagnostics	word
101	island bus object class	3	node configured	array of word
101	island bus object class	4	node assembly fault	array of word
101	island bus object class	5	node error	array of word
101	island bus object class	6	node operational	array of word

Instance ID 101: Input Process Image Data from the Island Bus

Instance 101 of the assembly object class binds the input process image data from island bus object class ID 101 of the DeviceNet NIM to an input assembly.

The following table shows the island bus object (class ID 101) mapping for instance 101 (instance ID 1) to attribute 3:

Linked Object Class		Attribute		
ID	**Name**	**ID**	**Name**	**Data Type**
101	island bus object class	21	NIM status	word
101	island bus object class		packed input data	array of word
		14	HMI-to-PLC data	array of word

Instance ID 102: Output Process Image Data from the Island Bus

Instance 102 of the assembly object class binds the output process image data from island bus object class ID 101 of the DeviceNet NIM to an input assembly.

The following table shows the island bus object (class ID 101) mapping for instance 102 (instance ID 1) to attribute 3:

Linked Object Class		Attribute		
ID	**Name**	**ID**	**Name**	**Data Type**
101	island bus object class		packed output data	array of word
		16	PLC-to-HMI data	array of word

Connection Object (Class ID 5)

Introduction

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

Supported Instances

The following table lists the assembly object instances supported by the connection object:

Instance ID	Type	Instance Name
1	predefined connection set	explicit messaging connection object instance
2	predefined connection set	poll connection I/O messaging object instance
3*	predefined connection set	bit-strobe connection I/O messaging object instance
4	predefined connection set	COS/cyclic connection I/O messaging object instance
5–14	UCMM	dynamic explicit and I/O messaging connection object instances
*The Advantys STB DeviceNet NIM does not support bit-strobe connection messaging.		

NOTE: The format and characteristics for the following instances are specified by ODVA.

Instance ID 1: Explicit Messaging Connection Object Instance

This instance provides a point-to-point, explicit messaging connection between two nodes on a DeviceNet network. These connections are usually used to configure nodes, get diagnostic information, and provide network management.

Instance ID 2: Poll Connection I/O Messaging Object Instance

The poll connection I/O object messaging instance provides the communication characteristics for an I/O connection that processes I/O poll command and response messages.

These messages move any amount of I/O data between a master and its polled slaves.

In this point-to-point poll connection, a DeviceNet master and slave act as client and server, respectively. The client sends application data to the server with a poll command and receives application data from the server with a poll response.

Default values for consumed and produced poll connection data are described in the following table:

Attr. ID	Name	Semantic of Value	Description
7	produced_connection_size	depends on attribute ID 14	maximum number of bytes transmitted across this connection
8	consumed_connection_size	depends on attribute ID 16	maximum number of bytes consumed across this connection
14	produced_connection_path	assembly object class 4, instance ID 101, attribute ID 3	specifies the application object(s) whose data is to be produced through this connection
16	consumed_connection_path	assembly object class 4, instance ID 102, attribute ID 3	specifies the application object(s) whose data is to be consumed through this connection

Instance attribute 14 (produced_connection_path) links to the assembly object class ID 4, instance 101 (input process image data from the island), while instance attribute 16 (consumed_connection_path) links to the assembly object class ID 4, instance ID 102 (output process image data to the island).

Therefore, a poll connection is used by a PLC on the DeviceNet fieldbus to read the process image input data from the island bus and to write the process image output data to the island. By default, no diagnostic data is supported here.

Because the process image value is limited, the maximum amount of output or input data transmitted across this connection is 4096 bytes for the produced and consumed path.

Instance ID 4: COS/Cyclic Connection I/O Messaging Object Instance

The COS/cyclic connection messaging object instance provides the communication characteristics for an I/O connection that processes I/O change of state/cyclic messages.

In a point-to-point change of state/cyclic connection, a DeviceNet master and slave act as a server and client, respectively. The client sends application data to the server with a COS/cyclic message. The master configures the message to be triggered cyclically or when a change in the data occurs.

Default values for data consumed and produced through a COS/cyclic connection are described in the following table:

Attr. ID	Name	Semantic of Value	Description
7	produced_connection_size	depends on attribute ID 14	maximum number of bytes transmitted across this connection
8	consumed_connection_size	depends on attribute ID 16, default value = 0	maximum number of bytes consumed across this connection
14	produced_connection_path	assembly object class 4, instance ID 100, attribute ID 3	specifies the application object(s) whose data is to be produced through this connection
16	consumed_connection_path	acknowledgement handler object, class ID 43, instance ID 1	specifies the application object(s) whose data is to be consumed through this connection

Instance attribute 14 (produced_connection_path) links to the assembly object class 4, instance ID 100 (diagnostic/error data from the island), while instance attribute 16 (consumed_connection_path) links to the acknowledgement handler object. Therefore, a change of state/cyclic connection is used by the island on the DeviceNet fieldbus to send the diagnostic/error data from the island either on a change of state or cyclically.

Instance ID 5–14: Dynamic Explicit and I/O Messaging Connection Object Instances

With the UCMM port, the island bus allows you to establish up to five dynamic explicit and five dynamic I/O message connections.

Island Bus Object (Class ID 101)

Introduction

The island bus object is an application object that provides the diagnostic and error data, as well as input and output data from all modules on the island.

Class Attributes

The following table lists the attributes supported by the island bus object class:

Attr. ID	Name	Data Type	Description	Value
1	revision	UINT	revision of the island bus object class definition	1

Class Services

The following class services are supported by the island bus object class:

Service Code	Service Name	Description
0Eh	get_attribute_single	reads island bus object class attribute value

Object Instance Attributes

The following table lists the attributes supported by the island bus object:

Attr. ID	Name	Services	Data Type	Description	Value (from NIM)
1	island bus state	get	word	communication status	diagnostic data
2	global diagnostics	get	word	global errors	
3	node configured	get	array of word	indicates configured modules	
4	node assembly fault	get	array of word	indicates incorrectly assembled modules	
5	node error	get	array of word	indicates modules with errors	
6	node operational	get	array of word	indicates operational modules	
7	input data size	get	UINT	size of input data in words	unpacked input process image
8	input data	get	array of word	unpacked input data from island modules	
9	output data size	get	UINT	size of output data in words	unpacked output process image
10	output data	get/set	array of word	unpacked output data to island modules	
13	HMI-to-PLC data size	get	UINT	size of HMI-to-PLC input data in words	HMI-to-PLC input data table
14	HMI-to-PLC data	get	array of word	HMI-to-PLC output data	

Attr. ID	Name	Services	Data Type	Description	Value (from NIM)
15	PLC-to-HMI data size	get	UINT	size of PLC-to-HMI output data in words	PLC-to-HMI output data table
16	PLC-to-HMI data	get/set	array of word	PLC-to-HMI output data	
21	NIM status	get	word	NIM status word	status word

Instance Services

The following class services are supported by the island bus object class:

Service Code	Service Name	Description
0Eh	get_attribute_single	reads island bus object instance attribute value
10h	set_attribute_single	modifies island bus object instance attribute value

4.2 Diagnostic 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's in this Section?

This section contains the following topics:

Diagnostic Data

Introduction

This topic discusses the diagnostic data for the Advantys STB NDN 2212 DeviceNet NIM.

Diagnostic Data Structure

The diagnostic and error 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 68 bytes (34 words):

Diagnostic Information	Data Type	Description
island bus state	word	shows the communication state and diagnostics of the island bus
global diagnostics	word	indicates the occurrence of a fatal error or the detection of a network error (also reports local island bus errors)
node configured	word array (8)	characterizes every node as configured or not configured
node assembly fault	word array (8)	characterizes every node as deviating from its configured and expected state
node error	word array (8)	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 (8)	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.

Each bit in the *island bus state* low byte array indicates a specific error or 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.

Byte Value	Meaning
64h	The process image is being set up.
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.

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

Communication Diagnostic	Meaning of Value
D8*	1 = low-priority receive queue software overrun error
D9*	1 = NIM overrun error
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 error
D14*	1 = high-priority receive queue software overrun error
D15*	1 = high-priority transfer queue software overrun error
*fatal NIM errors	

Island bus state diagnostics can also be accessed through the DeviceNet 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.

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

Bit	Meaning
D0*	*fatal error*—Because of the severity, no further communications are possible on the island bus.
D1*	*module ID error*—A standard CANopen device is using a module ID reserved for the Advantys STB modules.
D2*	Auto-addressing has failed.
D3*	Mandatory module configuration error.
D4*	*process image error*—Either the process image configuration is inconsistent or it could not be set during auto-configuration.
D5*	*auto-configuration error*—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*	*assignment error*—The initialization process in the NIM has detected a module assignment error.
*fatal NIM errors	

Each bit in the *global diagnostics* high byte array indicates a specific error or 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
*fatal NIM errors	

NOTE: Errors marked with an asterisk (*) in the *global diagnostics* tables are fatal NIM errors. They are caused by internal errors related to either the NIM or a failure in the island configuration software or hardware.

The detection of these errors will result 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.

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

Node Configured

Node configured is an array of 8 words (16 bytes, 128 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 DeviceNet 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
. . .										
7	15	120	119	118	117	116	115	114	113	word offset 7, low byte
	16		127	126	125	124	123	122	121	word offset 7, high byte
*assigned word offset inside the PLC										

The STB NDN 2212 DeviceNet NIM supports a maximum of 32 modules. The first 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 DeviceNet explicit connection by following the path: class 101\instance 1\attribute 3.

Node Assembly Fault

Node assembly fault is an array of 8 words (16 bytes, 128 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 DeviceNet 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

Word*	Byte	Bit								Status Data
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
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
. . .										
7	15	120	119	118	117	116	115	114	113	word offset 7, low byte
	16		127	126	125	124	123	122	121	word offset 7, high byte
*assigned word offset inside the PLC										

The STB NDN 2212 DeviceNet NIM supports a maximum of 32 modules. The first 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 DeviceNet explicit connection by following the path: class 101\instance 1\attribute 4.

Node Error

Node error is an array of 8 words (16 bytes, 128 bits). Each bit represents one specific addressable I/O module on the island bus. After the master receives an emergency message (not error-free) from a module, the corresponding bit is set:

- A value of 1 in a bit position indicates the presence of a newly received emergency 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 DeviceNet 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
. . .										
7	15	120	119	118	117	116	115	114	113	word offset 7, low byte
	16		127	126	125	124	123	122	121	word offset 7, high byte
*assigned word offset inside the PLC										

The STB NDN 2212 DeviceNet NIM supports a maximum of 32 modules. The first 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 DeviceNet explicit connection by following the path: class 101\instance 1\attribute 5.

Node Operational

Node operational is an array of 8 words (16 bytes, 128 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 faults 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 DeviceNet 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
. . .										
7	15	120	119	118	117	116	115	114	113	word offset 7, low byte
	16		127	126	125	124	123	122	121	word offset 7, high byte
*assigned word offset inside the PLC										

The *node operational* diagnostics can also be accessed through the DeviceNet 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 NDN 2212 DeviceNet NIM. The status data (along with the I/O data) is transmitted through the polled I/O connection.

NIM Status Data

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

Bit	Meaning of Value
D0 & D1	initialization status of bit packing unit indicated by the combined value of these two bits
	3 = initialization not started
	2 = initialization in progress
	1 = initialization aborted due to error
	0 = initialization finished
D2*	1 = NIM’s DeviceNet handler has finished initialization and is waiting for island bus to start
	0 = NIM’s DeviceNet handler prohibits island bus from starting, or island bus is already running
D3*	1 = initialization of NIM’s DeviceNet handler is not finished or is no longer valid
	0 = initialization of NIM’s DeviceNet handler is valid
D4*	1 = NIM’s DeviceNet handler did not start and is not running
	0 = NIM’s DeviceNet handler running or has already run
D5*	1 = DeviceNet LEDs are not initialized or initialization is no longer valid
	0 = DeviceNet LEDs are initialized and operating as defined by DeviceNet
D6	1 = NIM’s CAN controller is bus-off
	0 = NIM’s DeviceNet handler’s CAN controller is not bus-off
D7*	0 = the value of this reserved bit
*fatal NIM errors	

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

Bit	Meaning of Value
D8*	device failure
	1 = any module on the island bus has failed
	0 = no module failure

Bit	Meaning of Value
D9*	internal failure
	1 = any global diagnostic bit (except *RESET*) is set
	0 = all global diagnostic bits are set to 0
D10*	external failure
	1 = fieldbus problem
	0 = fieldbus operating normally
D11	protected mode
	1 = NIM in protected mode—RST button is disabled and the island configuration requires a password to write to it
	0 = NIM not in protected mode—RST button is enabled and the island configuration is not password-protected
D12	removable memory card validity
	1 = configuration on the card is invalid
	0 = 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 = fieldbus master is controlling the output data of the island's process image
*fatal NIM errors	

The *NIM status* can also be accessed through the DeviceNet explicit connection by following the path: class 101\instance 1\attribute 21.

4.3 Data Exchange

DeviceNet Data Exchange

Introduction

This topic discusses the manner in which bit packed process image data is exchanged between the STB NDN 2212 DeviceNet 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 DeviceNet fieldbus master involves three types of objects:

- *data objects*—operating values the DeviceNet master either reads from the input modules or writes to the output modules
- *status objects*—module health records sent by I/O modules and read by the DeviceNet master
- *echo output data objects*—sent by digital output modules to the DeviceNet 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	1 byte or less	does not apply	
		status*	1 byte or less	does not apply	
digital output		echo output data	1 byte or less	data	1 byte max
		status*	1 byte 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	

Module Type		Objects in the Input Data Image		Objects in the Output Data Image	
		Objects	Size	Objects	Size
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 00) for relevant modules.					

The Internal Process Image

The STB NDN 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.
- *output data to the island bus*—The island bus scanner handles output data and places it in the process image's output buffer.

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

The internal input process image can be accessed through the DeviceNet explicit messaging connection by following this path: class 101, instance number 1, attribute number 8. The path for the internal output process image is: class 101, instance number 1, attribute number 10.

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, resulting 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 NDN 2212 NIM observes the following rules for the bit packing of 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 may 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 will be 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 DeviceNet's bit packing rules to the sample island *(see page 92)* assembly (in the *Application Example* chapter) will 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 output 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 *(see page 92)*. 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):

Word*	**Byte**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
1	1	NIM status (low byte)							
	2	NIM status (high byte)							
2	3	N2 output status		N2 output echo		N1 input status		N1 input data	
	4	N3 input status				N3 input data			
3	5	N4 output status				N4 echo output data			
	6	empty (set to 0)		N5 input data					
4	7	empty (set to 0)		N5 input status					
	8	empty (set to 0)		N6 echo output data					
5	9	empty (set to 0)		N6 output status					
	10	empty (set to 0)							
6	11	N7 (channel 1) analog input data (low byte)							
	12	N7 (channel 1) analog input data (high byte)							
7	13	N7 (channel 1) analog input status							
	14	empty (set to 0)							
8	15	N7 (channel 2) analog input data (low byte)							
	16	N7 (channel 2) analog input data (high byte)							
9	17	N7 (channel 2) analog input status							
	18	N8 (channel 1) analog output status							
10	19	N8 (channel 2) analog output status							
*assigned word offset inside PLC									

Application Examples

5

Introduction

This chapter presents two examples for configuring the Advantys STB island on a DeviceNet network. Each example implements the same sample island assembly with an Advantys STB NDN 2212 DeviceNet NIM at the head.

What's in this Chapter?

This chapter contains the following topics:

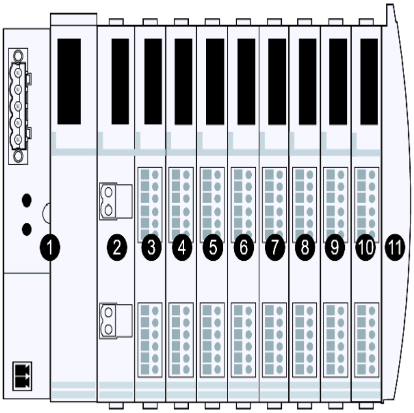
Sample Island Assembly

Introduction

To understand the configuration example(s) and bit packing for the NIM, you will need to implement a particular Advantys STB island assembly. Your island assembly is independent of the network's master scanner because the island is represented by the NIM as a single node on the fieldbus network.

Sample Island

The sample I/O system implements a variety of analog and digital modules. The following Advantys STB island modules are used in the example(s):



1 STB NDN 2212, DeviceNet NIM
2 STB PDT 3100, 24 VDC Power Distribution Module
3 STB DDI 3230, 24 VDC 2-channel digital input module (2 bits of data, 2 bits of status)
4 STB DDO 3200, 24 VDC 2-channel digital output module (2 bits of data, 2 bits of echo output data, 2 bits of status)
5 STB DDI 3420, 24 VDC 4-channel digital input module (4 bits of data, 4 bits of status)
6 STB DDO 3410, 24 VDC 4-channel digital output module (4 bits of data, 4 bits of echo output data, 4 bits of status)
7 STB DDI 3610, 24 VDC 6-channel digital input module (6 bits of data, 6 bits of status)
8 STB DDO 3600, 24 VDC 6-channel digital output module (6 bits of data, 6 bits of echo output data, 6 bits of status)
9 STB AVI 1270, +/-10 VDC 2-channel analog input module (16 bits of data [channel 1], 16 bits of data [channel 2], 8 bits of status [channel 1], 8 bits of status [channel 2])
10 STB AVO 1250, +/-10 VDC 2-channel analog output module (8 bits of status [channel 1], 8 bits of status [channel 2], 16 bits of data [channel 1], 16 bits of data [channel 2])
11 STB XMP 1100 termination plate

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

I/O Model	Module Type	Island Bus Address	Island Node Number
STB DDI 3230	two-channel digital input	1	N1
STB DDO 3200	two-channel digital output	2	N2
STB DDI 3420	four-channel digital input	3	N3
STB DDO 3410	four-channel digital output	4	N4
STB DDI 3610	six-channel digital input	5	N5
STB DDO 3600	six-channel digital output	6	N6
STB AVI 1270	two-channel analog input	7	N7
STB AVO 1250	two-channel analog output	8	N8

The NIM, the PDM, and the termination plate do not consume island bus addresses, and they do not exchange data or status objects with the fieldbus master.

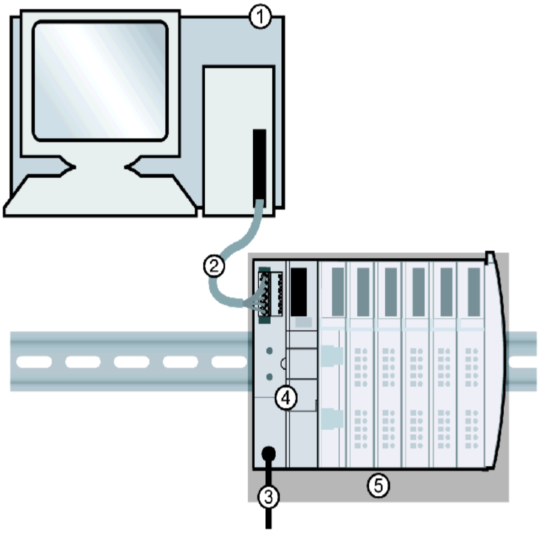
Configuring a Hilscher PC-based DeviceNet Master with SyCon

Introduction

These instructions are for configuring a Hilscher PCI master card (part SMS-CIF50-DNM) for use with a DeviceNet NIM at the head of an Advantys STB island node. The stages of this process are described in the following table:

Stage	Description
1	add a master to your network configuration
2	import the NIM's EDS file to the SyCon database
3	add the NIM as a device in your network configuration
4	configuring device parameters
5	download the configuration
6	verify and save the configuration

The following figure shows the connection between a Hilscher PCI master card and an STB NDN 2212 NIM over a DeviceNet network:



1 Hilscher PCI master card in a standard PC
2 DeviceNet network cable (not supplied)
3 external power supply interface
4 STB NDN 2212 DeviceNet NIM
5 Advantys STB island assembly

Before You Begin

To use this application example, you should have a working familiarity with both the DeviceNet fieldbus protocol and Hilscher's SyCon configuration software.

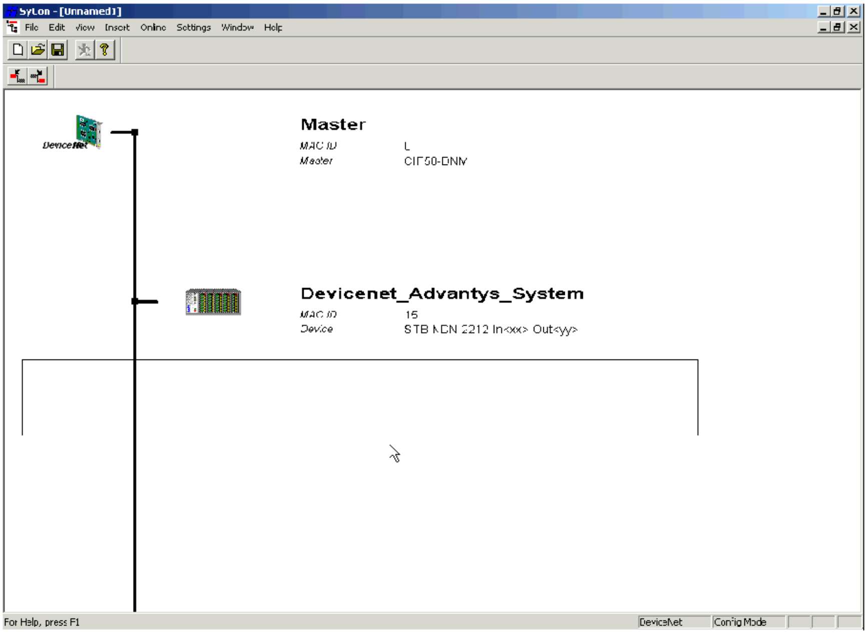
Before you begin, make sure:

- your Advantys modules are fully assembled, installed, and powered according to your particular system, application, and network requirements
- you have properly set the node address *(see page 29)* of the DeviceNet NIM
- you have the basic EDS file and corresponding bitmap files that were supplied with the STB NDN 2212 DeviceNet NIM (also available at *www.schneiderautomation.com*), or you have generated an EDS that is specific to the sample island assembly *(see page 92)* with the Advantys configuration software

The SyCon Workspace

In this configuration example, you will add a master device and an Advantys STB island slave to your configuration using SyCon.

The SyCon workspace should resemble the following figure after you've added the CIF50-DNM master and DeviceNet NIM slave to your network configuration with the following instructions:



Master

MAC ID 0

Master CIF50-DNM

Devicenet_Advantys_System

MAC ID 15

Device STB NDN 2212 In<xx> Out<yy>

Add a Master to the Configuration

Use the steps in the following table to add a DeviceNet master to your configuration:

Step	Action	Comment
1	From SyCon's Insert menu, select Master.	A list of DeviceNet masters appears in the Insert Master dialogue box.
2	Select a master appropriate to your application and press Add.	For this example, select CIF50-DNM.
3	Enter the MAC ID and Description of the selected master.	For the purposes of this example, you can simply accept the defaults.
4	Press OK.	A graphic that represents the selected master appears in SyCon's workspace.

Import the NIM's EDS

During the procedure in the following table, the STB NDN 2212 DeviceNet NIM's EDS is saved to your SyCon database even if you don't save your actual network configuration, making the NIM's EDS available for any configuration that implements that device. To import the EDS file:

Step	Action	Comment
1	From SyCon's File menu, select Copy EDS.	The Copy EDS dialogue box appears.
2	Navigate to the location of the EDS file you wish to import and open it.	
3	If you are asked, "Do you want to import the corresponding bitmap field?" answer Yes or No.	Answer according to your system requirements. The Import 'Configuration Bitmap' window appears.
4	Press OK when the Comment dialogue box appears.	The Comment dialogue box verifies that the EDS has been imported into the SyCon database.

Add the NIM to the Configuration

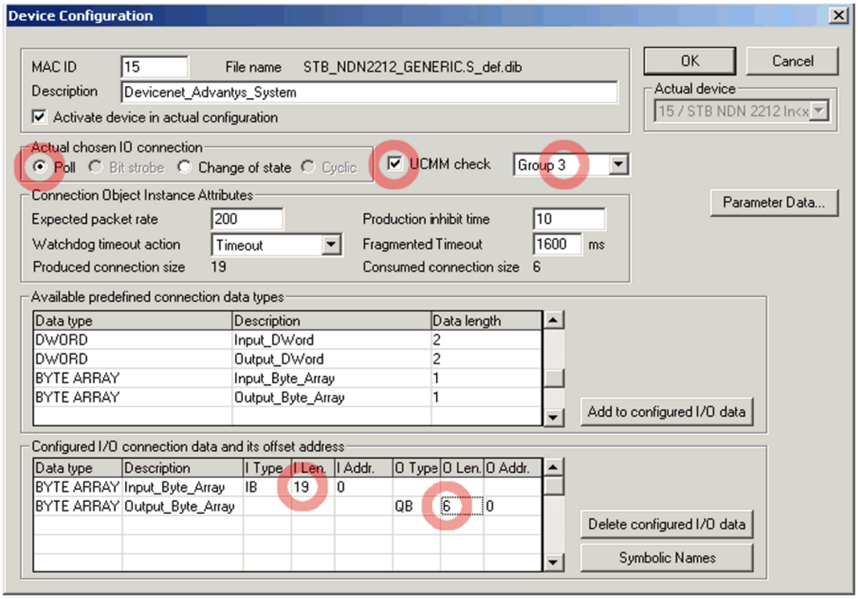
You must import the NIM's EDS before you configure it as a network device. To add the NIM to the network configuration:

Step	Action	Comment
1	From the Insert menu, select *Device*.	The cursor becomes a large *D*.
2	Click the mouse in the area below the CIF50-DNM master and to the right of the black vertical line.	The Insert Device dialogue box appears.
3	Select *STB NDN 2212* from the Available Devices list and press Add.	The Available Devices list represents EDS files in the SyCon database. *STB NDN 2212* now appears in the Selected devices list.
4	In the MAC ID field, enter the MAC ID of the selected device.	The MAC ID should match node address *(see page 29)* that was set with the NIM's rotary switches. Use *15* for this example.

Step	Action	Comment
5	Enter a description in the Description field.	The Description will appear as the name of your device in SyCon’s workspace. Use *Devicenet_Advantys_System* for this example.
6	Press OK.	A graphic that represents the selected device appears in SyCon’s workspace.

SyCon’s Device Configuration Dialogue Box

To complete this sample configuration, you should set up the text fields in the Device Configuration dialogue box to resemble the following figure:



Device Configuration

MAC ID 15 File name STB_NDN2212_GENERIC.S_def.dib

Description Devicenet_Advantys_System

☑ Activate device in actual configuration

Actual chosen IO connection

● Poll ○ Bit strobe ○ Change of state ○ Cyclic

☑ UCMM check Group 3

Actual device 15 / STB NDN 2212 In<x

Connection Object Instance Attributes

Expected packet rate 200

Watchdog timeout action Timeout

Produced connection size 19

Production inhibit time 10

Fragmented Timeout 1600 ms

Consumed connection size 6

Available predefined connection data types

Data type	Description	Data length
DWORD	Input_DWord	2
DWORD	Output_DWord	2
BYTE ARRAY	Input_Byte_Array	1
BYTE ARRAY	Output_Byte_Array	1

Configured I/O connection data and its offset address

Data type	Description	I Type	I Len.	I Addr.	O Type	O Len.	O Addr.
BYTE ARRAY	Input_Byte_Array	IB	19	0			
BYTE ARRAY	Output_Byte_Array				QB	6	0

Instructions for making changes to the Device Configuration dialogue box follow.

NOTE: You can customize information in the *Connection Object Instance Attributes* fields for your particular applications.

Device Configuration Parameters

Use the following instructions to open the Device Configuration dialogue box and enter the appropriate information in the text fields:

Step	Action	Comment
1	In the SyCon workspace, double-click on the Advantys NIM device.	The Device Configuration dialogue box appears.
2	Select Poll in the Actual chosen IO connection list.	
3	Check the UCMM check box for Group 3.	If necessary, scroll to Group 3.
4	Double-click on *Input_Byte_Array* in the Available Predefined Connection Data Types window.	*Input_Byte_Array* will appear in the Configured I/O Connection data and its offset address list.
5	Double-click on *Output_Byte_Array*.	*Output_Byte_Array* will appear in the Configured I/O Connection data and its offset address list.
6	Change the input length (I Len.) of the *Input_Byte_Array* to *19*.	See the next paragraph.
7	Change the output length (O Len.) of the *Output_Byte_Array* to *6* and press OK.	

About the NIM's Input and Output Data Lengths

In the above procedure, you were required to enter the number of input and output bytes the NIM produces. The master device needs this information to allocate data space for each network node.

The number of input and output bytes the NIM produces can be determined in either an *offline* or *online* manner:

- *offline calculation*—You must calculate these data sizes using the NIM's bit packing rules *(see page 89)*.
- *online determination*—These data sizes can be read from the NIM directly by using the Get Attribute command (from SyCon's Options menu) for class 1, instance 1, attribute 7. From the product name string, *STB NDN 2212 IN19 OUT6*, in the Value text box, you can deduce that the NIM produces 19 bytes of input and 6 bytes of output data.

NOTE: The STB NDN 2212 DeviceNet NIM always provides 68 bytes of diagnostic data through a COS/cyclic connection.

Download the Configuration

The following table provides instructions for downloading the DeviceNet NIM's configuration to your master:

Step	Action	Comment
1	In the SyCon workspace, select the CIF50-DNM master.	
2	In the Online menu, select Download.	The Download dialogue box appears.
3	Wait for the download to finish.	The NIM's configuration has been downloaded to the master device.
4	Press OK.	

Verify and Save the NIM Configuration

The following table provides instructions for verifying and saving the DeviceNet NIM's configuration to your master:

Step	Action	Comment
1	From SyCon's Online menu, select Live List.	The Live List is a list of all available DeviceNet MAC IDs (0–63).
2	Find the NIM's MAC ID	A MAC ID of *15* was used in this example.
3	Verify that the NIM's MAC ID appears in black.	The MAC ID of every device known to the master appears in black (not grey).
4	Save your configuration by selecting Save from SyCon's File menu.	This is a standard *Windows* command.

Configuring a SLC-500 DeviceNet Master with RSNetWorx

Introduction

Use these sample instructions to configure an Allen Bradley SLC-500 PLC (1747-SDN) with a DeviceNet NIM at the head of an Advantys STB island node. The configuration software is Rockwell's RSNetWorx for DeviceNet configuration software. The stages of this process are described in the following table:

Stage	Description
1	assemble the DeviceNet network *(see page 102)*
2	register the NIM's EDS *(see page 102)*
3	connect devices to your network *(see page 103)*
4	upload the NIM configuration *(see page 104)*
5	add the NIM to the Scanlist *(see page 106)*
6	create an EDS for the NIM *(see page 108)*

Before You Begin

Before you begin, make sure:

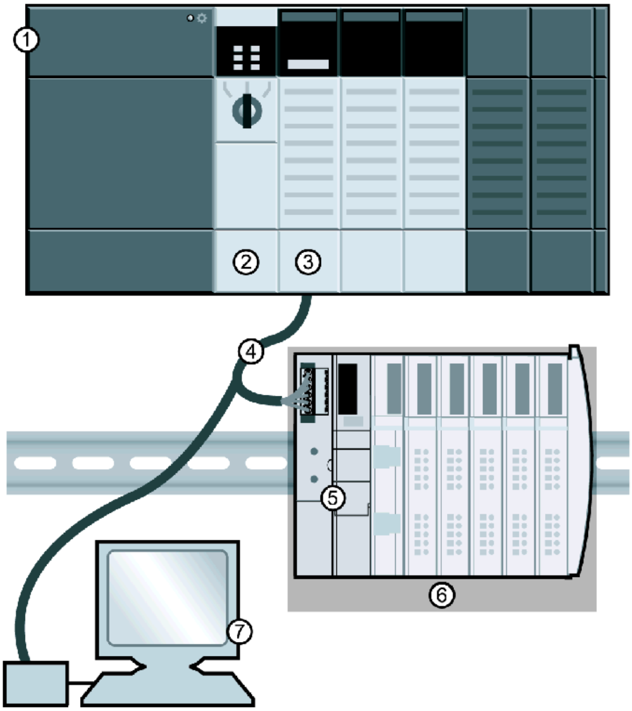
- your Advantys modules are fully assembled, installed, and powered according to your particular system, application, and network requirements
- you have properly set the node address *(see page 29)* of the DeviceNet NIM
- you have the basic EDS *(see page 22)* file and corresponding bitmap files that were supplied with the STB NDN 2212 DeviceNet NIM (also available at *www.schneiderautomation.com*), or you have generated an EDS that is specific to the sample island assembly

NOTE: With the RSNetWorx configuration software, you can import only one EDS per product family. For maximum flexibility, it is therefore suggested that you use the basic EDS with any Advantys STB island that you place on your DeviceNet network. EDS files other than the basic one (those that you create with the Advantys configuration software) are configuration-specific and will probably not work with multiple Advantys STB islands.

To use this application example, you should have a working familiarity with both the DeviceNet fieldbus protocol and RSNetWorx for DeviceNet, version 3.21.00. (The described procedures can not practically anticipate every RSNetWorx prompt or option you may encounter during configuration.)

Connection Figure

Before assembling the network, look at the required hardware connections. The following figure shows the DeviceNet network connections between an Allen-Bradley PLC, an STB NDN 2212 NIM, and RSNetWorx:



1 Allen-Bradley SLC-500 PLC
2 PLC processor module
3 1747-SDN DeviceNet scanner module
4 DeviceNet network cable
5 STB NDN 2212 DeviceNet NIM
6 Advantys STB island
7 PC running RSNetWorx (properly connected to your network)

The scanner module is the control mechanism for all network traffic. It reads and writes every piece of I/O data that is moved on the network.

Assemble the Physical Network

The following procedure describes the connections required to construct a physical DeviceNet network.

⚠ **CAUTION**

EQUIPMENT DAMAGE IF VOLTAGE IS PRESENT

Read and understand this manual and the Allen-Bradley PLC user's manual before installing or operating this equipment. Only qualified personnel should install, adjust, repair, and maintain this equipment.

- Disconnect all power to the PLC before making the network connection.
- Place a DO NOT TURN ON sign on the system power disconnect.
- Lock the disconnect in the open position.

You are responsible for conforming to all applicable code requirements with respect to grounding all equipment.

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

Step	Action	Comment
1	Install the DeviceNet scanner module in the desired PLC slot.	The connection figure *(see page 101)* above shows the scanner in slot 2 of the PLC.
2	Using the rotary switches on the STB NDN 2212 NIM, set the island to the desired DeviceNet network node address *(see page 28)*.	Use an address of *15* for this example.
3	Make connections with DeviceNet network cable and end connectors, manufactured in accordance with ODVA specifications.	The cable and end connectors are not supplied.
4	Place the island on the network by connecting the PLC to the STB NDN 2212 NIM with the DeviceNet cable.	
5	Place the RSNetWorx PC on the network through the DeviceNet cable.	

Register the NIM's EDS

To register the NIM's EDS *(see page 22)* in RSNetWorx's EDS library, follow the procedure in the following table:

Step	Action	Comment
1	From the RSNetWorx Tools menu, select EDS Wizard.	The Wizard's welcome screen appears.
2	Click Next.	The Options screen appears.
3	Select Register an EDS file(s) and click Next.	The Registration screen appears.

Step	Action	Comment
4	Select Register a single file and Browse to the NIM's EDS file.	You must already know the location of the EDS file.
5	Click Next.	The EDS File Installation Test Results screen appears.
6	Click Next.	The Change Graphic Image screen appears. The NIM should be listed in the Product Types field as a Communication Adapter.
7	Click Next.	The Final Task Summary screen appears.
8	Verify that the NIM is to be registered and click Next.	The completion screen appears.
9	Click Finish.	The EDS Wizard closes.

Connect Devices to Your Network

This example requires you to add two devices to your project view:

- the NIM—at the head of an Advantys STB island
- the master scanner—in PLC slot 2

Connection with RSNetWorx can be achieved in either an *offline* or *online* manner:

- *offline connection*—Connection between the configuration tool and a physical network is not required for this type of connection.
- *online connection*—Connect and build the network with parameters uploaded from devices on the physical network.

Make network connections using either the offline or online procedures in the tables that follow. (These are standard RSNetWorx procedures.)

Offline Device Connection

Use this offline procedure for adding devices to your network when you are not connected:

Step	Action	Comment
1	From the Hardware list, double-click on the NIM's EDS under Schneider Automation, Inc.\Communication Adapter.	The new device appears in the project view. The lowest available MAC ID has been assigned to it, even if that ID is inappropriate.
2	Double-click on the NIM graphic.	The NIM's properties window appears.
3	Change the MAC ID in the Address text field to *15*.	*15* is the MAC ID used throughout this example.
4	Click OK.	Note that the MAC ID of the NIM is now *15* in the project view.
5	Repeat steps 1 to 4 to add the 1747-SDN Scanner Module to the network with MAC ID *00*.	The scanner's EDS is in the *Hardware* list at Rockwell Automation - Allen Bradley/Communication Adapter.
6	Save your configuration by choosing Online from the Network menu.	Save offline configurations for later use.

Online Device Connection

Use this online procedure for adding devices to your network when your DeviceNet network is already assembled:

Step	Action	Comment
1	From the Network menu, select Online.	The Browse for network screen appears.
2	Set a communication path that is in accordance with your system and application requirements.	When the Browsing network screen finishes, the physically connected devices will appear in the project view.
3	Click OK, indicating that you will upload or download the required device information.	

Upload and Download Device Configurations

After the online connection of devices, you must upload or download the required device information.

Using these selections from the Device menu, only individual (selected) devices will have their configurations reconciled:

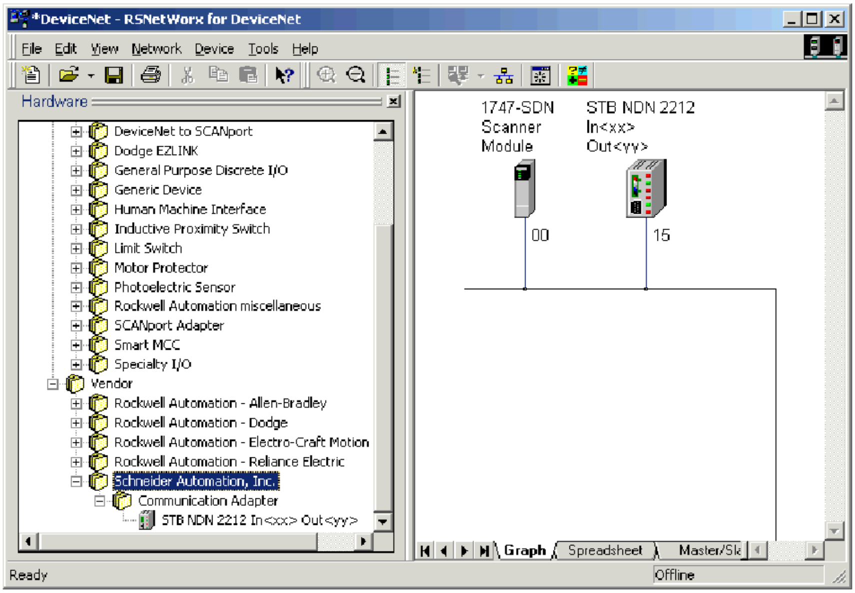
- Download to Device—Download the offline configuration to the device.
- Upload from Device—Upload the configuration from the device.

Use the following selections from the Network menu to upload or download configurations of all online devices in the project view:

- Download to Network—Download the offline configurations to the devices.
- Upload from Network—Upload the configurations of all online devices.

The RSNetWorx Project View

With either the online or offline connection procedure, the RSNetWorx project view should resemble the following figure after you've added the NIM and the master scanner to your network configuration:


*DeviceNet - RSNetWorx for DeviceNet
File Edit View Network Device Tools Help
Hardware
DeviceNet to SCANport
Dodge EZLINK
General Purpose Discrete I/O
Generic Device
Human Machine Interface
Inductive Proximity Switch
Limit Switch
Motor Protector
Photoelectric Sensor
Rockwell Automation miscellaneous
SCANport Adapter
Smart MCC
Specialty I/O
Vendor
Rockwell Automation - Allen-Bradley
Rockwell Automation - Dodge
Rockwell Automation - Electro-Craft Motion
Rockwell Automation - Reliance Electric
Schneider Automation, Inc.
Communication Adapter
STB NDN 2212 In<xx> Out<yy>
1747-SDN Scanner Module
STB NDN 2212 In<xx> Out<yy>
00
15
Graph
Spreadsheet
Master/Sla
Ready
Offline

Add the NIM to the Scanlist

For recognition on the network, the NIM must be uploaded to the master scanner's Scanlist using the online procedure in the following table:

Step	Action	Comment
1	From the project view, double-click on the scanner icon.	The scanner configuration screen appears.
2	Select the Scanlist tab.	The Scanner Configuration Applet screen appears.
3	Select Upload.	Wait for the Uploading from Scanner timer to finish.
4	At the Scanlist tab, highlight the NIM (at MAC ID *15*) in the Available Devices list and click the right arrow.	The NIM now appears in the Scanlist.
5	With the NIM selected, click the Edit I/O Parameters button.	The Edit I/O Parameters window appears.
6	Check Polled and enter *19* in the Rx Size text field and *6* in the Tx Size text field.	These are the data sizes for the sample island. (Determination of the NIM's input and output data lengths is described in the next paragraph.)
7	Check Change of State/Cyclic and enter *68* in the Rx Size text field.	The DeviceNet NIM always provides 68 bytes of diagnostic data through a COS/cyclic connection.
8	Click OK.	The Edit I/O Parameters window is closed.
9	Click Download to scanner.	The Downloading Scanlist from Scanner widow appears.
10	Click Download.	Wait for the Downloading to Scanner timer to finish.
11	Click OK.	The scanner properties widow closes.

About the NIM's Input and Output Data Lengths

In the above procedure, you were required to enter the number of input and output bytes produced by the NIM. The master device needs this information to allocate data space for each network node.

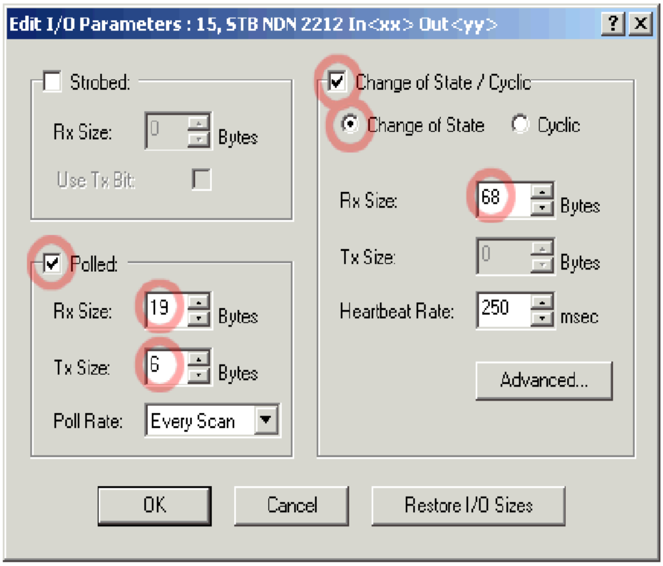
The number of input and output bytes the NIM produces can be determined in either an *offline* or *online* manner:

- *offline calculation*—You must calculate these data sizes using the NIM's bit packing rules *(see page 89)*.
- *online determination*—These data sizes can be read from the NIM directly by using the Class Instance Editor command (from RSNetWorx's Device menu) for class 1, instance 1, attribute 7. From the product name string, *STB NDN 2212 IN19 OUT6*, you can deduce that the NIM produces 19 bytes of input and 6 bytes of output data.

NOTE: The STB NDN 2212 DeviceNet NIM always provides 68 bytes of diagnostic data through a COS/cyclic connection.

The Edit I/O Parameters Screen

The NIM’s Edit I/O Parameters screen should resemble the following figure after you’ve customized it as described above:



Edit I/O Parameters : 15, STB NDN 2212 In<xx> Out<yy>

□ Strobed:

Rx Size: 0 Bytes

Use Tx Bit: □

☑ Polled:

Rx Size: 19 Bytes

Tx Size: 6 Bytes

Poll Rate: Every Scan

☑ Change of State / Cyclic

● Change of State ○ Cyclic

Rx Size: 68 Bytes

Tx Size: 0 Bytes

Heartbeat Rate: 250 msec

Create an EDS for the NIM

Devices that do not correspond with specific EDS *(see page 22)* files during online network browsing will appear in the project view as Unrecognized Devices. If your NIM is not recognized, you must create an EDS using the following procedure:

Step	Action	Comment
1	In the project view, double-click on the NIM.	You will be asked if you want to register the NIM with the EDS Wizard.
2	Click Yes.	The Wizard's welcome screen appears.
3	Click Next.	The Options screen appears.
4	Select Create an EDS file and click Next.	RSNetWorx will upload the NIM's identity information, displayed in the Device Description screen.
5	Record the product name string, *STB NDN 2212 IN19 OUT6*, and click Next.	The Input/Output screen appears.
6	Check Polled and enter the appropriate values for input and output sizes. Also check COS and enter an input size value of 68. Click Next.	From the product name string, you can deduce that the NIM produces 19 bytes of input and 6 bytes of output data.
7	Change the icon, if you wish, at the Change Graphic Image and click Next.	The Final Task Summary screen appears.
8	Verify that the NIM is to be registered and click Next.	The completion screen appears.
9	Click Finish.	The EDS Wizard closes. Now that you've created an EDS, you should add the NIM to the Scanlist using the directions above.

Saving the Configuration

Save your configuration by selecting Save from the RSNetworx File menu. This is a standard Windows command.

Advanced Configuration Features

6

Introduction

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

What's in this Chapter?

This chapter contains the following topics:

STB NDN 2212 Configurable Parameters

Introduction

This topic discusses the configuration of DeviceNet NIM parameters using the Advantys configuration software.

The following operating parameters are user configurable:

- 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 CANopen device

General Information

To get general information about the NIM module (model name, version number, vendor code, etc.):

Step	Action	Comment
1	Open your configuration with the Advantys configuration software.	The NIM is the leftmost module in your island assembly.
2	Double-click on the NIM in the configuration workspace.	The *module editor* window appears.
3	Select the *General* tab.	The *General* tab gives general information about the NIM.

Accessing Configurable Parameters

To access the NIM values that are configurable:

Step	Action	Comment
1	Double-click the NIM module in the configuration workspace.	The *module editor* window appears.
2	Select the *Parameters* tab.	Configurable parameters are on this tab.
3	In the *Parameter name* column, expand the *Additional Info Store list* by clicking on the plus (+) sign.	The configurable NIM parameters become visible.

Reserved Sizes (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) value is selected, the range of available data sizes (in words) appears in the window (see the above figure). The maximum size includes both the input data produced by the island modules and the HMI to PLC data. Therefore, space that you reserve for the HMI to PLC data—plus the input data from the island bus modules—must not exceed the maximum value shown. For example, if your input modules produce 10 words of input data, you can reserve only the remaining 117 words (out of 127 maximum) of the input data table for the HMI to PLC data.

Reserved Sizes (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) value is selected, the range of available data sizes (in words) appears in the window (see the above figure). The maximum size includes both the data sent to the island modules and the PLC to HMI data. Therefore, space that you reserve for PLC to HMI data—plus the output data for the island bus modules—must not exceed the maximum value. For example, if your output modules consume three words of output data, you can reserve only the remaining 125 words (out of 128 maximum) of the output data table for the PLC to HMI data.

Reserving Data Sizes

To transfer data to the PLC from a Modbus HMI you must reserve sizes for that data. To reserve these data sizes:

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 become visible.
3	Double-click in the *Value* column next to the *Reserved Size (Words) of HMI to PLC table*.	The value is highlighted.
4	Enter a value that represents the data size that will be reserved for data sent from the HMI panel to the PLC.	The value you enter *plus* the data size of your island can not exceed the maximum value. If you accept the default (0), no space will be reserved in the HMI table in the process image.

Step	Action	Result
5	Repeat the above steps to select a *Value* for the *Reserved Size (Words) of PLC to HMI table* row.	
6	Press *OK* when you have entered the desired data sizes.	

CANopen Device Node IDs

On the Parameters tab, you can set the maximum node ID of the last module on the island bus. Standard CANopen devices follow the last segment of STB I/O modules. CANopen modules are addressed by counting backwards from the value you enter 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 will result 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) will result in node IDs of 1 through 5 for STB I/O modules and 30 through 32 for standard CANopen devices. Those unnecessarily high addresses are not desirable if any of your standard CANopen devices have a limited address range.

Assigning the Max. Node ID (CANopen Devices)

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

Step	Action	Comment
1	In the *module editor* window, select the *Parameters* tab.	Configurable parameters are on 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 you consider the module or device critical to 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 120)*. 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 must 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

⚠ **WARNING**

UNINTENDED EQUIPMENT OPERATION/LOSS OF CONFIGURATION—RST BUTTON WHILE RECOVERING FROM MANDATORY STOP

Pushing the RST button *(see page 60)* causes the island bus to reconfigure itself with factory-default operating parameters, which do not support mandatory I/O status.

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

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

Pushing the RST button *(see page 60)* while recovering from a mandatory stop loads the island's default configuration data.

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

⚠ **WARNING**

UNEXPECTED OUTPUT OPERATION

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.

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.

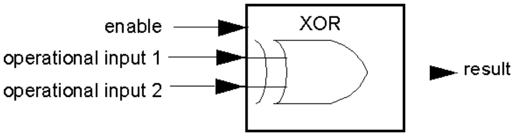
Configuring a Reflex Action

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

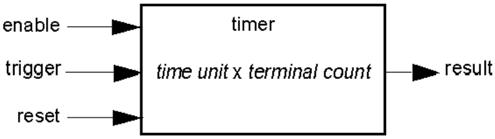
Each block must be assigned a set of inputs and a result. Some blocks also require that you specify one or more user-preset values—a compare block, for example, requires that you preset threshold values and a delta value for hysteresis.

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 must always be a Boolean, whereas the operational input to an analog latch must always 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.

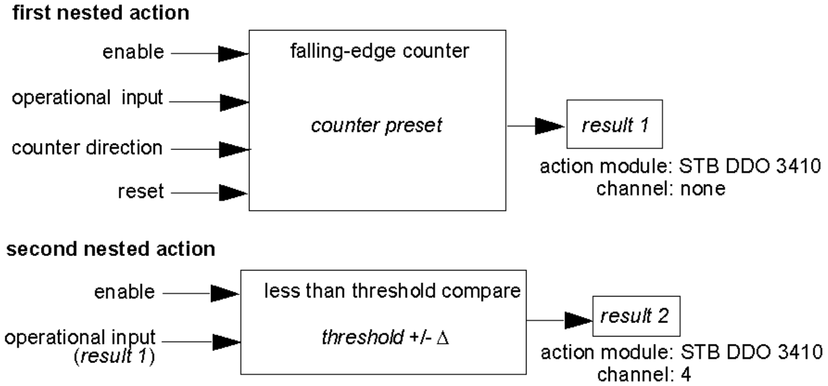
The exception is when a reflex block is the first of two actions in a nested reflex action.

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). The best way to preserve processing resources is to:

- 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:

- loss of fieldbus communications: Communications with the PLC are lost.
- 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 all of 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 116)* 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 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 always 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 54)*. 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 36)* 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 saves the configuration data to the NIM's flash memory.
4	Install the card *(see page 55)* 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 60)* is disabled, and pushing it has no effect on island bus operations.
- The presence of a removable memory card *(see page 54)* is ignored. The configuration data currently stored in Flash cannot be overwritten by data on the card.

NOTE: The STB NIP 2311 NIM never ignores the removable memory card.

Password Characteristics

A password must meet the following criteria:

- It must 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:

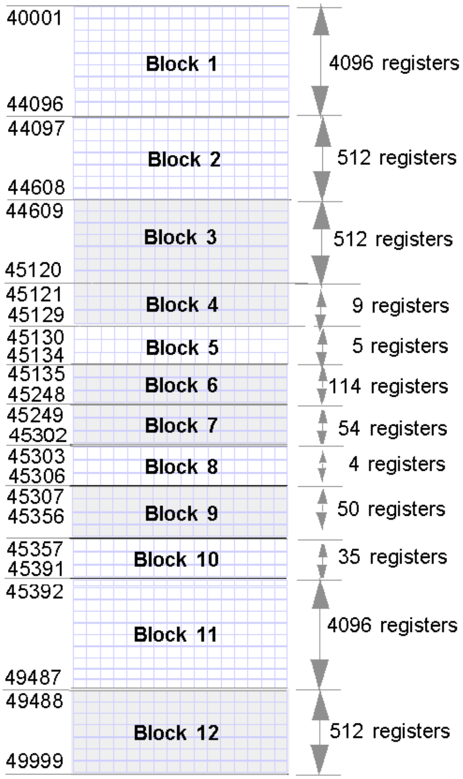
MSB															LSB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

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 all 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 36)*. The Advantys configuration software reads all 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 148)*.

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 must be the island bus master in order for it to write to the data image—i.e., the island must be in *test* mode.

The Island's Process Image Blocks

Summary

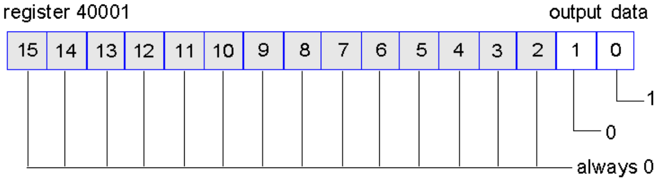
Two blocks of registers in the island's data image *(see page 125)* are the focus for this discussion. The first block is the output data process image, which starts at register 40001 and goes to register 44096. The other block is the input data and I/O status process image, which also consumes 4096 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 always 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.

Predefined Diagnostics Registers in the Data Image

Summary

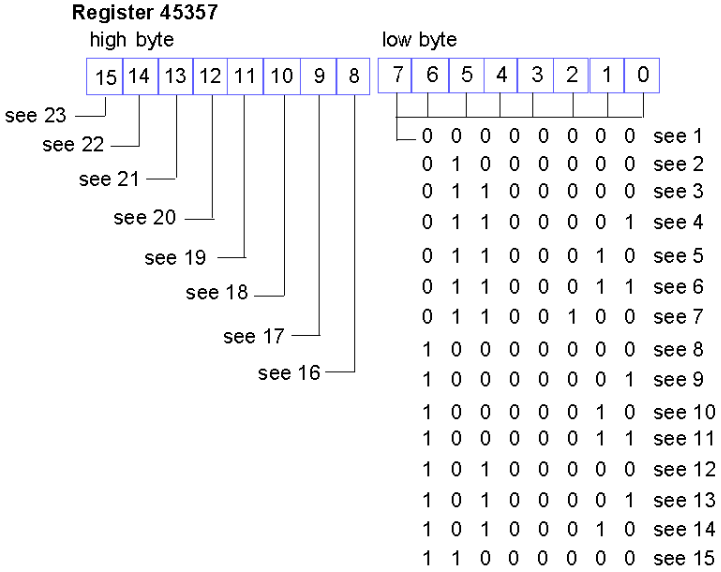
Thirty-five contiguous registers (45357 through 45391) in the island bus data image *(see page 125)* are provided for reporting diagnostic information. These registers have predefined meanings that are described below.

These registers can be accessed and monitored through an HMI panel, by the Advantys configuration software, or by the fieldbus master using the Run-time Parameters (RTP) feature. Refer to the Run-time Parameters topic for information on RTP.

NOTE: RTP is supported by standard NIMs with firmware version 2.0 or greater. RTP is not available in Basic NIMs.

Island Communications Status

Register 45357 describes the state of communications across the island bus. The low byte (bits 7 through 0) displays one of 15 possible eight-bit patterns that indicates the current state of communication. Each bit in the high byte (bits 15 through 8) is used to signal the presence or absence of a specific error condition:



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

1 The island is initializing.
2 The island has been put in the pre-operational state, for example, with the reset function in the Advantys STB configuration software.
3 The NIM is configuring or auto-configuring—communication to all modules is reset.

4 The NIM is configuring or auto-configuring—checking for any modules that are not auto-addressed.
5 The NIM is configuring or auto-configuring—Advantys STB and preferred modules are being auto-addressed.
6 The NIM is configuring or auto-configuring—boot-up is in progress.
7 The process image is being set up.
8 Initialization is complete, the island bus is configured, the configuration matches, and the island bus is not started.
9 Configuration mismatch—non-mandatory or unexpected modules in the configuration do not match, and the island bus is not started.
10 Configuration mismatch—at least one mandatory module does not match, and the island bus is not started.
11 Serious configuration mismatch—the island bus has been set to pre-operational mode, and initialization is aborted.
12 The configuration matches, and the island bus is operational.
13 Island is operational with a configuration mismatch. At least one standard module does not match, but all the mandatory modules are present and operating.
14 Serious configuration mismatch—the island bus was started but is now in pre-operational mode because of one or more mismatched mandatory module(s).
15 Island has been set to pre-operational mode, for example, with the stop function in the Advantys STB configuration software.
16 A value of 1 in bit 8 is a fatal error. It indicates a low-priority receive queue software overrun error.
17 A value of 1 in bit 9 is a fatal error. It indicates a NIM overrun error.
18 A value of 1 in bit 10 indicates an island bus-off error.
19 A value of 1 in bit 11 is a fatal error. It indicates that the error counter in the NIM has reached the warning level and the error status bit has been set.
20 A value of 1 in bit 12 indicates that the NIM’s error status bit has been reset.
21 A value of 1 in bit 13 is a fatal error. It indicates a low-priority transfer queue software overrun error.
22 A value of 1 in bit 14 is a fatal error. It indicates a high-priority receive queue software overrun error.
23 A value of 1 in bit 15 is a fatal error. It indicates a high-priority transfer queue software overrun error.

The low byte of the Island Communications Status can be accessed via RTP by writing the following values to the RTP request block:

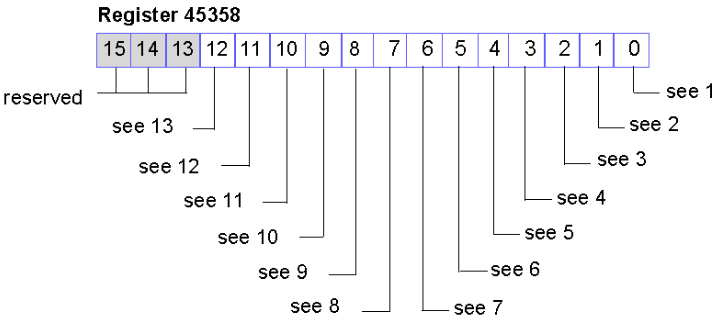
Length	1
Index (low byte)	0x00
Index (high byte)	0x41
Sub-Index	2
Node ID	127

The high byte of the Island Communications Status can be accessed via RTP by writing the following values to the RTP request block:

Length	1
Index (low byte)	0x00
Index (high byte)	0x41
Sub-Index	3
Node ID	127

Error Reporting

Each bit in register 45358 is used to report a global error condition. A value of 1 in the bit indicates that a specific global error has been detected:



1 Fatal error. Because of the severity of the error, no further communications are possible on the island bus.
2 Module ID error—a standard CANopen device is using a module ID reserved for the Advantys STB modules.
3 Auto-addressing has failed.
4 Mandatory module configuration error.
5 Process image error—either the process image configuration is inconsistent, or it could not be set up during auto-configuration.
6 Auto-configuration error—a module is not in its configured location, and the NIM cannot complete auto-configuration.
7 Island bus management error detected by the NIM.
8 Assignment error—the initialization process in the NIM has detected a module assignment error, possibly the result of at least one application parameter mismatch.
9 Internal triggering protocol error.
10 Module data length error.
11 Module configuration error.
12 Application parameter error.
13 Application parameter services or timeout error.

The Error Reporting register can be accessed via RTP by writing the following values to the RTP request block:

Length	2
Index (low byte)	0x00
Index (high byte)	0x41
Sub-Index	1
Node ID	127

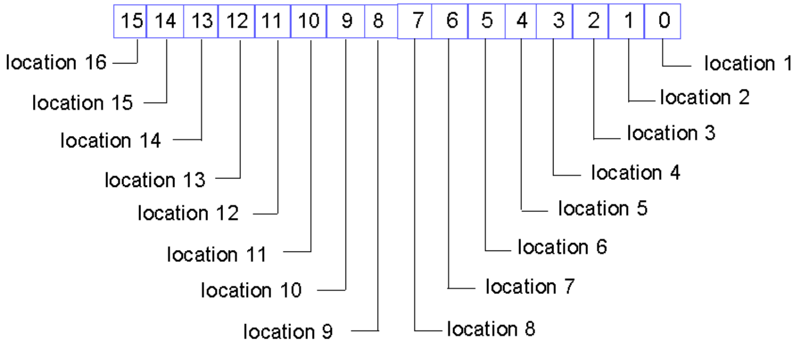
Node Configuration

The next eight contiguous registers (registers 45359 through 45366) display locations where modules have been configured on the island bus. This information is stored in Flash memory. At start up, the actual locations of the modules on the island are validated by comparing them to the configured locations stored in memory. Each bit represents a configured location:

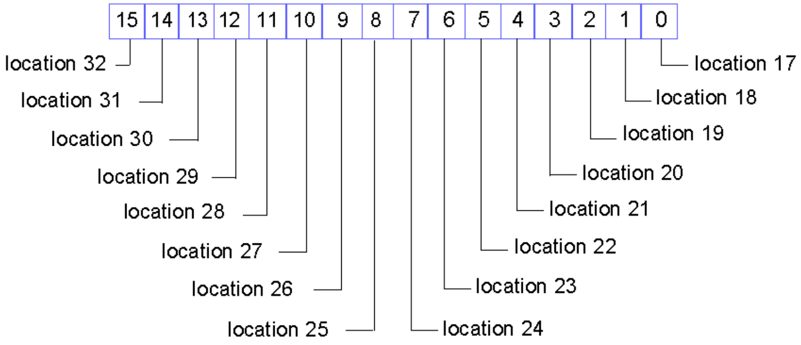
- A value of 1 in a bit indicates that a module has been configured for the associated location.
- A value of 0 in a bit indicates that a module has not been configured for the associated location.

The first two registers, shown below, provide the 32 bits that represent the module locations in a typical island configuration. The remaining six registers (45361 through 45366) are available to support island expansion capabilities.

Register 45359



Register 45360



The Node Configuration registers can be accessed via RTP by writing the following values to the RTP request block:

Length	1
Index (low byte)	0x01
Index (high byte)	0x41
Sub-Index	1 for the low byte of register 45359 (locations 1 to 8) 2 for the high byte of register 45359 (locations 9 to 16) 3 for the low byte of register 45360 (locations 17 to 24) 4 for the high byte of register 45360 (locations 25 to 32)
Node ID	127

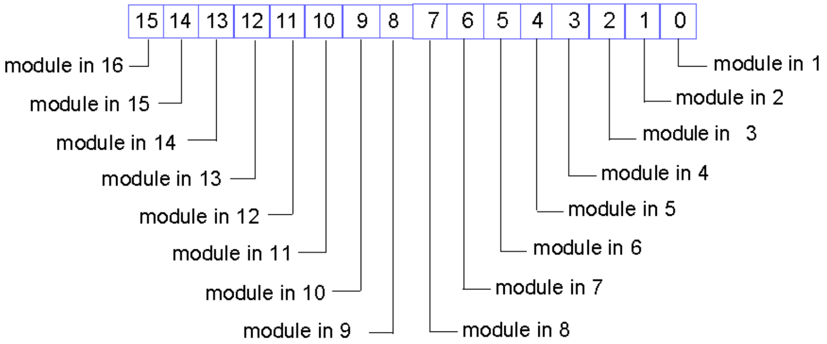
Node Assembly

The next eight contiguous registers (registers 45367 through 45374) indicate the presence or absence of configured modules in locations on the island bus. This information is stored in Flash memory. At start up, the actual locations of the modules on the island are validated by comparing them to the configured locations stored in memory. Each bit represents a module:

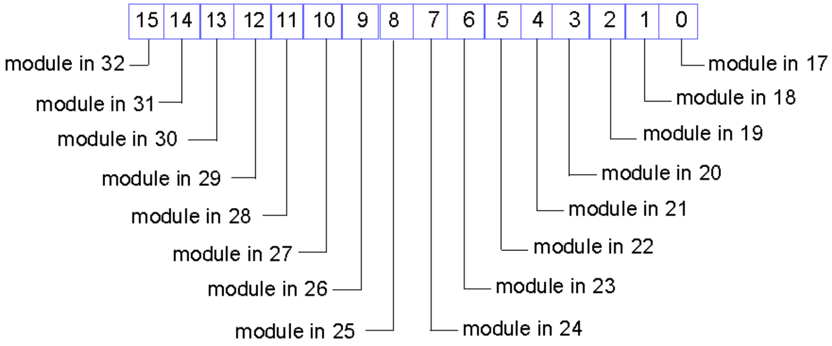
- A value of 1 in a given bit indicates either that the configured module is not present or that the location has not been configured.
- A value of 0 indicates that the correct module is present in its configured location.

The first two registers, shown below, provide the 32 bits that represent the module locations in a typical island configuration. The remaining six registers (45369 through 45374) are available to support island expansion capabilities.

Register 45367



Register 45368



The Node Assembly registers can be accessed via RTP by writing the following values to the RTP request block:

Length	1
Index (low byte)	0x02
Index (high byte)	0x41
Sub-Index	1 for the low byte of register 45367 (locations 1 to 8) 2 for the high byte of register 45367 (locations 9 to 16) 3 for the low byte of register 45368 (locations 17 to 24) 4 for the high byte of register 45368 (locations 25 to 32)
Node ID	127

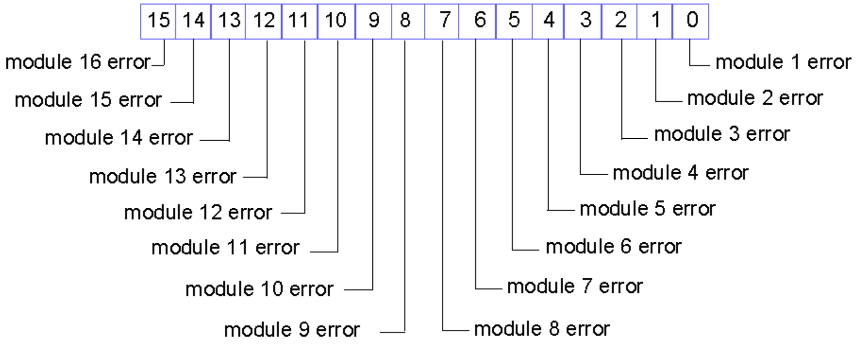
Emergency Messages

The next eight contiguous registers (registers 45375 through 45382) indicate the presence or absence of newly received emergency messages for individual modules on the island. Each bit represents a module:

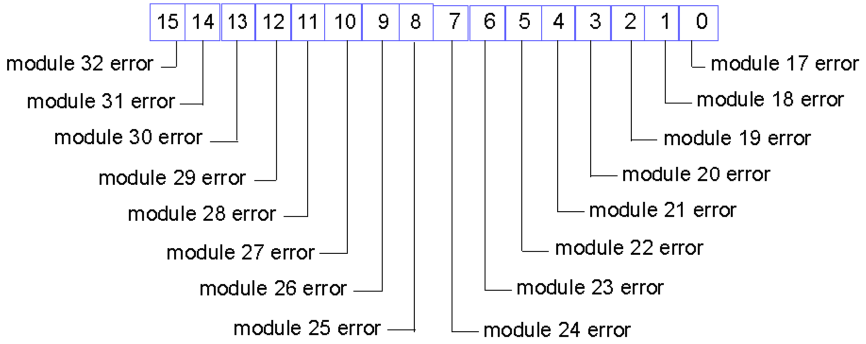
- A value of 1 in a given bit indicates that a new emergency message has been queued for the associated module.
- A value of 0 in a given bit indicates that no new emergency messages have been received for the associated module since the last time the diagnostic buffer was read.

The first two registers, shown below, provide the 32 bits that represent the module locations in a typical island configuration. The remaining six registers (45377 through 45382) are available to support island expansion capabilities.

Register 45375



Register 45376



The Emergency Messages registers can be accessed via RTP by writing the following values to the RTP request block:

Length	1
Index (low byte)	0x04
Index (high byte)	0x41
Sub-Index	1 for the low byte of register 45375 (locations 1 to 8) 2 for the high byte of register 45375 (locations 9 to 16) 3 for the low byte of register 45376 (locations 17 to 24) 4 for the high byte of register 45376 (locations 25 to 32)
Node ID	127

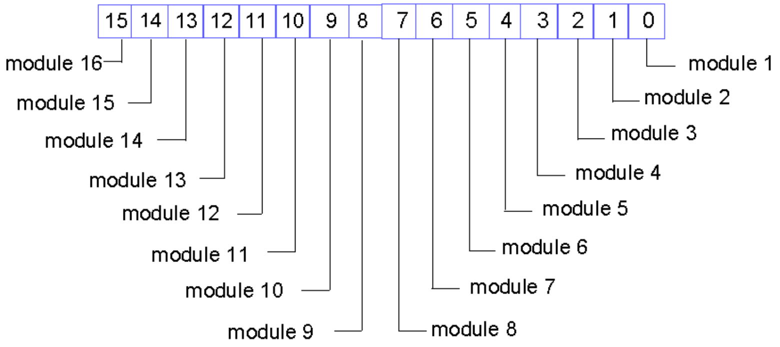
Fault Detection

The next eight contiguous registers (registers 45383 through 45390) indicate the presence or absence of operational faults detected on the island bus modules. Each bit represents a module:

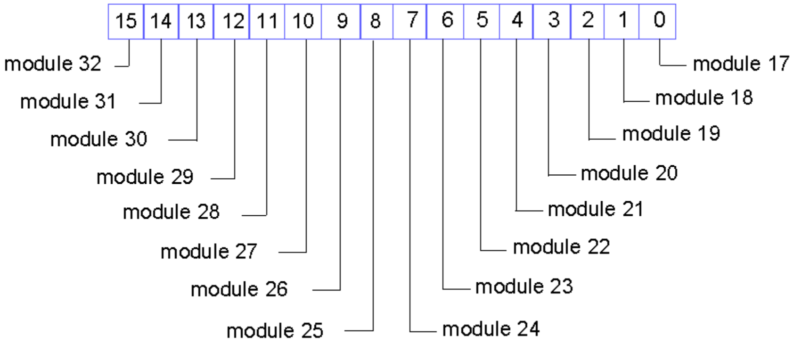
- A value of 1 in a bit indicates that the associated module is operating and that no faults were detected.
- A value of 0 in a bit indicates that the associated module is not operating either because it has a fault or because it has not been configured.

The first two registers, shown below, provide the 32 bits that represent the module locations in a typical island configuration. The remaining six registers (45385 through 45390) are available to support island expansion capabilities.

Register 45383



Register 45384



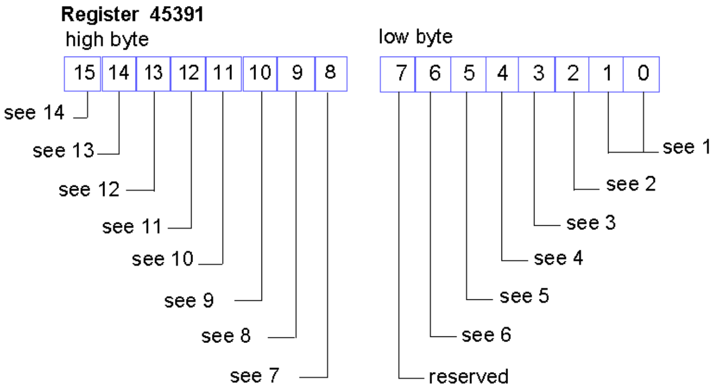
The Fault Detection registers can be accessed via RTP by writing the following values to the RTP request block:

Length	1
Index (low byte)	0x03
Index (high byte)	0x41

Sub-Index	1 for the low byte of register 45383 (locations 1 to 8) 2 for the high byte of register 45383 (locations 9 to 16) 3 for the low byte of register 45384 (locations 17 to 24) 4 for the high byte of register 45384 (locations 25 to 32)
Node ID	127

NIM Status

The high and low bytes in register 45391 report the status of the DeviceNet NIM:



1 The combined value of bits 0 and 1 indicates the initialization status of the bit packing unit. A value of 3 indicates that initialization has not started. A value of 2 indicates that initialization is in progress. A value of 1 indicates that initialization was aborted due to an error. A value of 0 indicates that initialization is finished.
2 A value of 1 in bit 2 indicates that the NIM’s DeviceNet handler has finished initialization and is waiting for the island bus to start. A value of 0 indicates that the NIM’s DeviceNet handler currently prohibits the island bus from starting or that the island bus is already running.
3 A value of 1 in bit 3 indicates that the initialization of the NIM’s DeviceNet handler is not finished or is no longer valid. A value of 0 indicates that initialization of the NIM’s DeviceNet handler is valid.
4 A value of 1 in bit 4 indicates that the NIM’s DeviceNet handler did not start and is not running. A value of 0 indicates that the NIM’s DeviceNet handler is running or has already run.
5 A value of 1 in bit 5 indicates that the DeviceNet LEDs are not initialized or the initialization is no longer valid. A value of 0 indicates that the DeviceNet LEDs are initialized and operating as defined by DeviceNet.
6 A value of 1 in bit 6 indicates that the NIM’s CAN controller is bus-off. A bit value of 0 indicates that the NIM’s CAN controller is not bus-off.
7 Module failure—A value of 1 in bit 8 indicates the failure of an island bus module.
8 A value of 1 in bit 9 indicates an internal failure—at least one global bit is set.
9 A bit value of 1 in bit 10 indicates an external failure—the problem is on the fieldbus.

10 A value of 1 in bit 11 indicates that the configuration is protected—the RST button is disabled and the configuration software requires a password before you can write. A bit value of 0 indicates that the configuration is standard—the RST button is enabled and the configuration software is not password-protected.
11 A value of 1 in bit 12 indicates that the configuration on the removable memory card is invalid.
12 A value of 1 in bit 13 indicates that reflex action functionality has been configured. (For NIMs with firmware version 2.0 or greater.)
13 A value of 1 in bit 14 indicates that one or more island modules have been hot-swapped. (For NIMs with firmware version 2.0 or greater.)
14 Island bus output data master—A value of 0 in bit 15 indicates that the fieldbus master device is controlling the output data of the island's process image. A bit value of 1 indicates that the Advantys configuration software is controlling the output data of the island's process image.

The NIM Status register can be accessed via RTP by writing the following values to the RTP request block:

Length	2
Index (low byte)	0x05
Index (high byte)	0x41
Sub-Index	0
Node ID	127

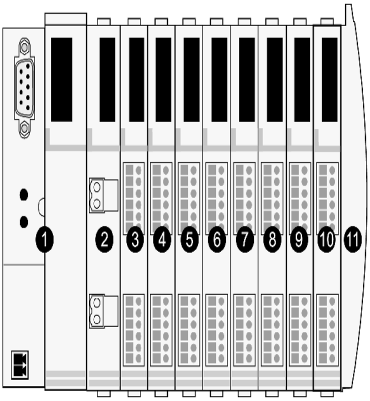
An Example of a Modbus View of the Process Image

Summary

The following example shows what the output data process image and the input data and I/O status process image might look like when it represents a specific island bus configuration.

The Sample Configuration

The sample island comprises the following 10 modules and a termination plate:



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

The I/O modules have the following island bus addresses *(see page 50)*:

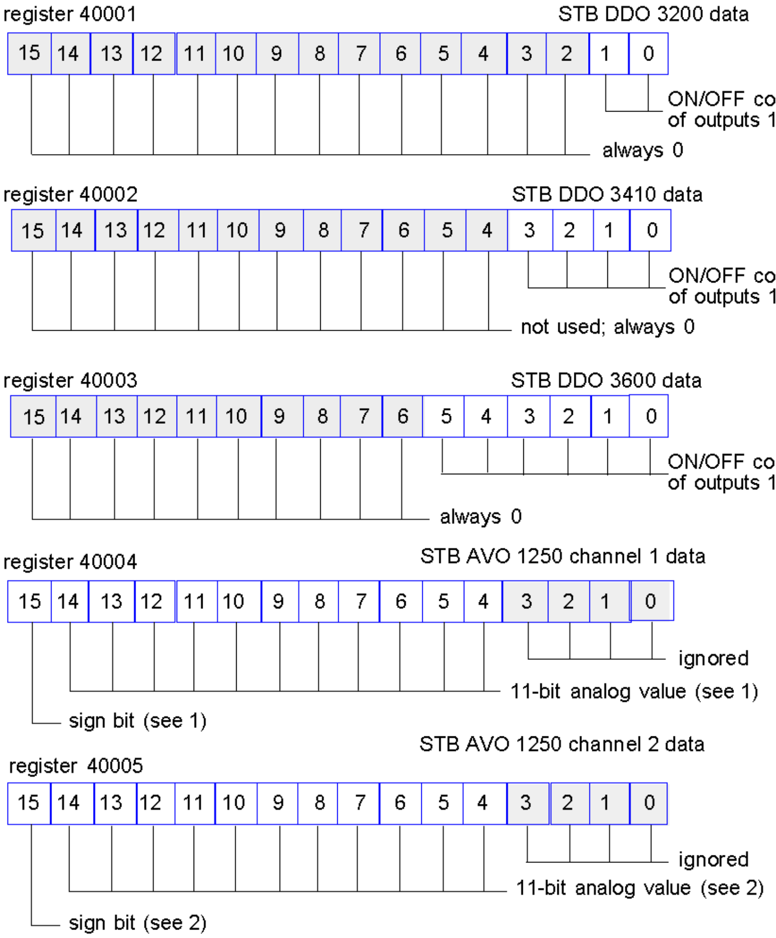
I/O Model	Module Type	Island Bus Address
STB DDI 3230	two-channel digital input	1
STB DDO 3200	two-channel digital output	2
STB DDI 3420	four-channel digital input	3
STB DDO 3410	four-channel digital output	4
STB DDI 3610	six-channel digital input	5
STB DDO 3600	six-channel digital output	6
STB AVI 1270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

The PDM and the termination plate do not consume island bus addresses and are not represented in the process image.

The Output Data Process Image

Let's look first at the register allocation required to support the output data process image *(see page 127)*. This is the data written to the island from the fieldbus master to update the output modules on the island bus. The four output modules are affected—the three digital output modules at addresses 2, 4, and 6 and the one analog output module at address 8.

The three digital output modules utilize one Modbus register apiece for data. The analog output module requires two registers, one for each output channel. A total of five registers (registers 40001 through 40005) are used for this configuration:



1 The value represented in register 40004 is in the range +10 to -10 V, with 11-bit resolution plus a sign bit in bit 15.

2 The value represented in register 40005 is in the range +10 to -10 V, with 11-bit resolution plus a sign bit in bit 15.

The digital modules use the LSBs to hold and display their output data. The analog module uses the MSBs to hold and display its output data.

The Input Data and I/O Status Process Image

Now let’s look at the register allocation required to support the input data and I/O status process image *(see page 128)*. This is the information that the NIM collects from the island modules so that it can be read by the fieldbus master or by some other monitoring device.

All eight I/O modules are represented in this process image block. The modules are assigned registers in the order of their island bus addresses, starting at register 45392.

Each digital I/O module uses two contiguous registers:

- Digital input modules use one register to report data and the next to report status.
- Digital output modules use one register to echo output data and the next to report status.

NOTE: The value in an *echo output data* register is basically a copy of the value written to the corresponding register in the output data process image. Generally, this is the value written to the NIM by the fieldbus master, and its echo is of not much interest. When an output channel is configured to perform a reflex action *(see page 116)*, however, the echo register provides a location where the fieldbus master can look to see the current value of the output.

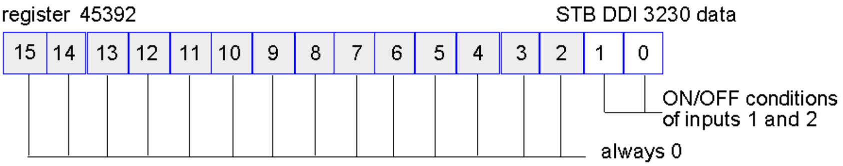
The analog input module uses four contiguous registers:

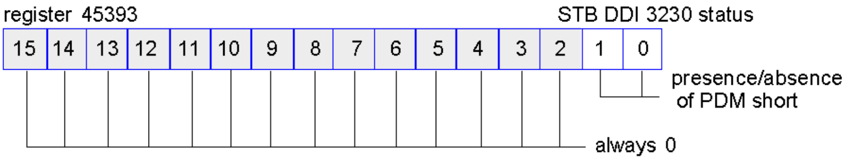
- the first register to report the data for channel 1
- the second register to report status for channel 1
- the third register to report the data for channel 2
- the fourth register to report status for channel 2

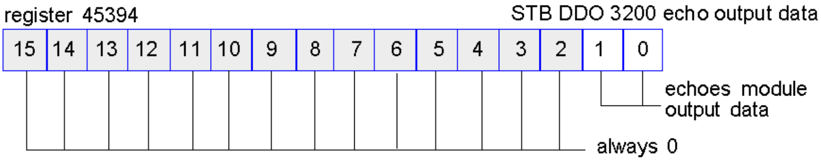
The analog output module uses two contiguous registers:

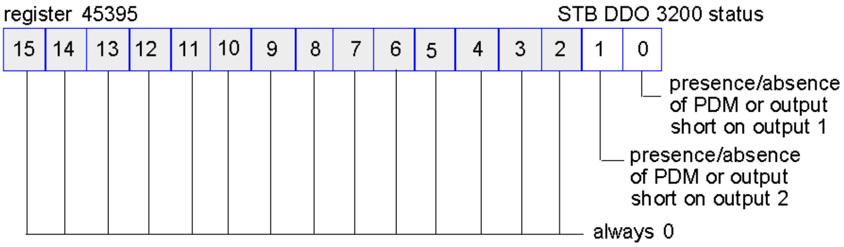
- the first register to report status for channel 1
- the second register to report status for channel 2

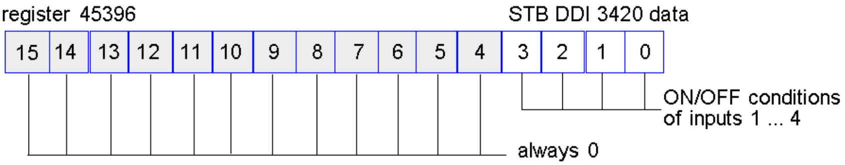
In total, 18 registers (registers 45392 through 45409) are used to support our configuration:

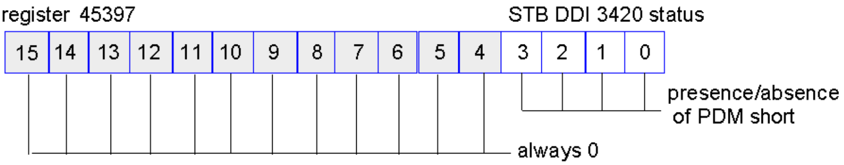


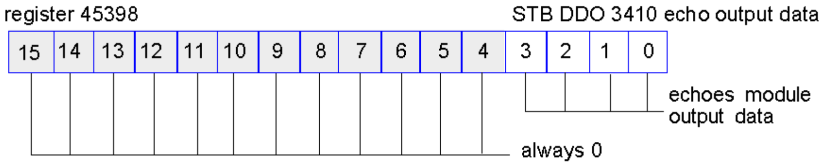


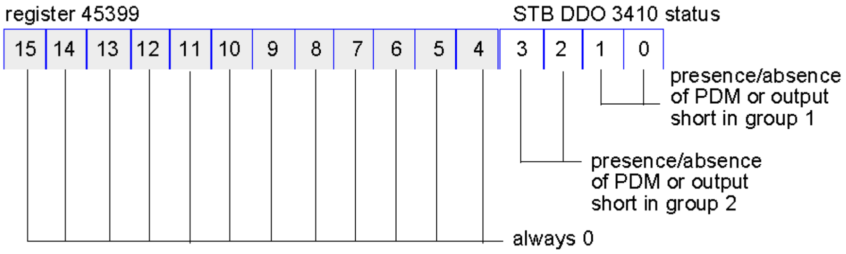


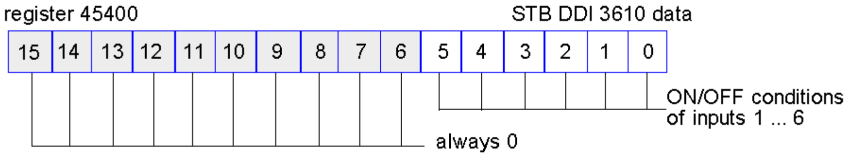

register 45395
STB DDO 3200 status
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
presence/absence of PDM or output short on output 1
presence/absence of PDM or output short on output 2
always 0

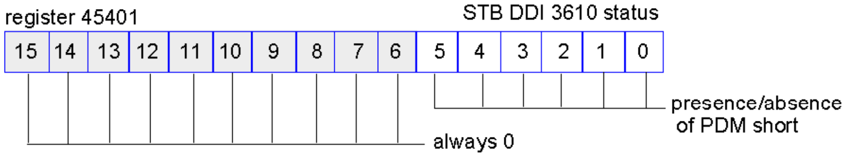


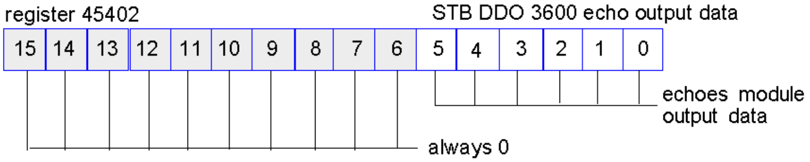


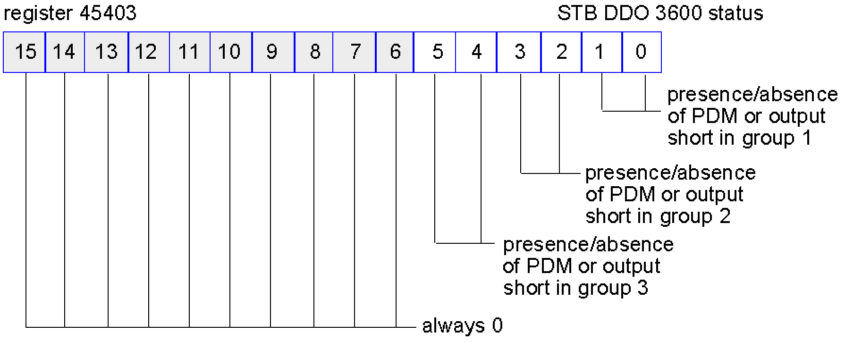

register 45398
STB DDO 3410 echo output data
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
echoes module output data
always 0

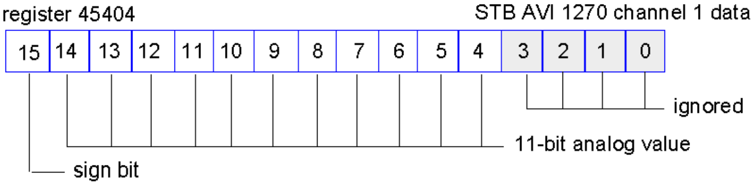

register 45399
STB DDO 3410 status
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
presence/absence of PDM or output short in group 1
presence/absence of PDM or output short in group 2
always 0

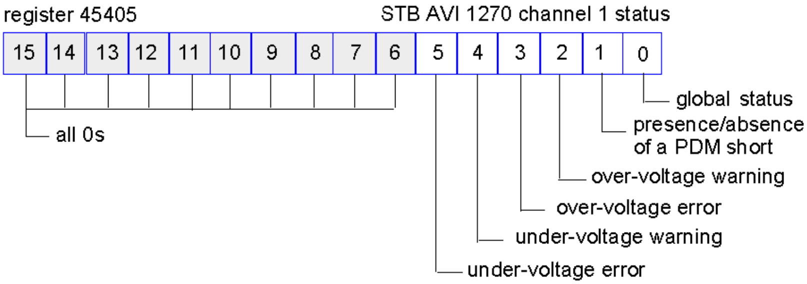

register 45400
STB DDI 3610 data
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ON/OFF conditions of inputs 1 ... 6
always 0

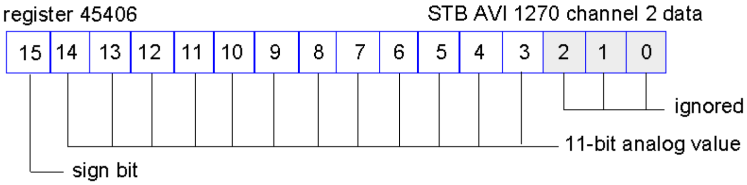

register 45401
STB DDI 3610 status
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
presence/absence of PDM short
always 0


register 45402
STB DDO 3600 echo output data
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
echoes module output data
always 0


register 45403
STB DDO 3600 status
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
presence/absence of PDM or output short in group 1
presence/absence of PDM or output short in group 2
presence/absence of PDM or output short in group 3
always 0

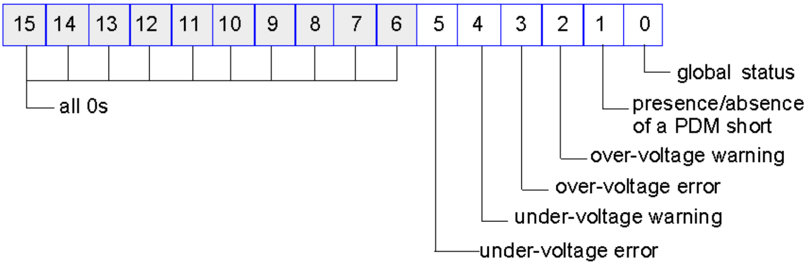

register 45404
STB AVI 1270 channel 1 data
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ignored
11-bit analog value
sign bit


register 45405
STB AVI 1270 channel 1 status
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
global status
presence/absence of a PDM short
all 0s
over-voltage warning
over-voltage error
under-voltage warning
under-voltage error


register 45406
STB AVI 1270 channel 2 data
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ignored
11-bit analog value
sign bit

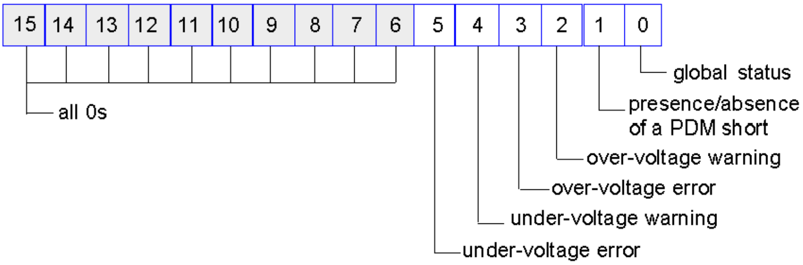
register 45407

STB AVI 1270 channel 2 status



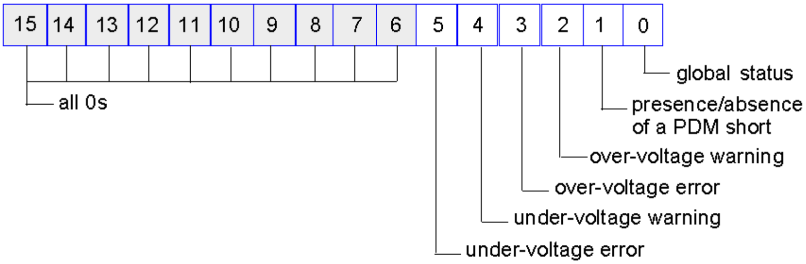
register 45408

STB AVO 1250 channel 1 status



register 45409

STB AVO 1250 channel 2 status



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 36)* on the NIM. Using the Advantys configuration software, you can reserve one or two blocks of registers in the data image *(see page 124)* 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 125)* and specify the number of registers in this block that you want to use for HMI-to-fieldbus master data transfers. You must 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 128)* 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 125)* 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 127)* 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: Loss of Modbus communications do 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 you input must 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 issue 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: Password Test Mode must be activated only 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 must 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

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.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Write all bytes in the RTP request 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.

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. You must 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 an SDO protocol error is detected, 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.

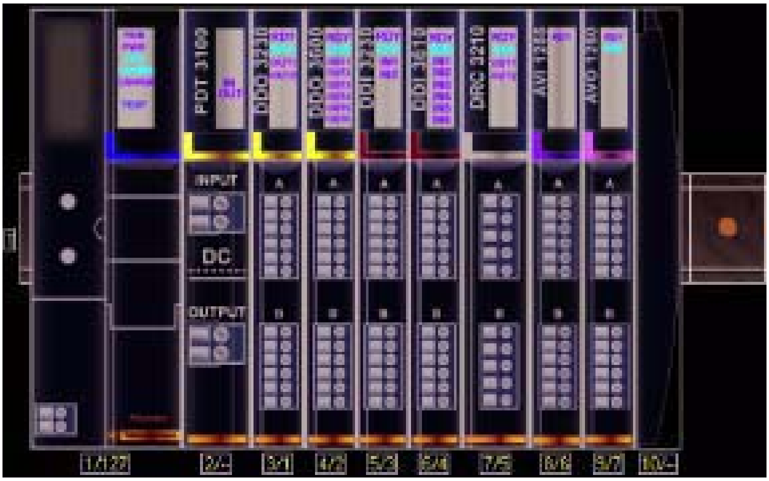
NOTE: Advantys configuration software is required to set Virtual Placeholder status.

Setting Virtual Placeholder Status

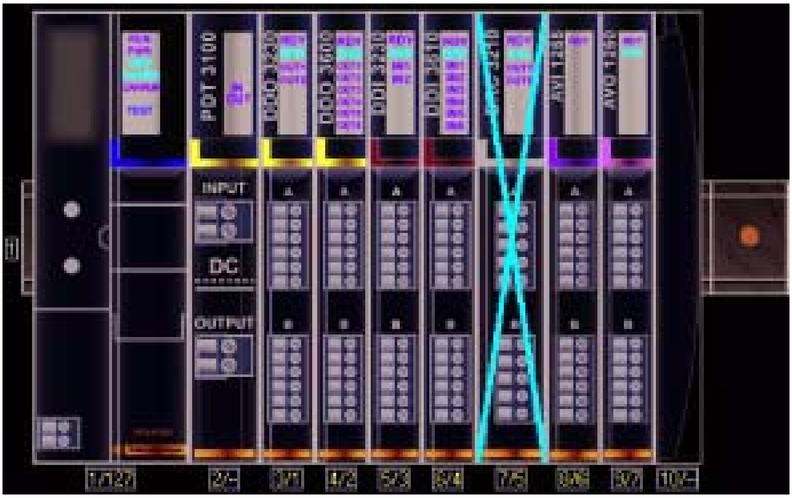
To set Virtual Placeholder status:

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:



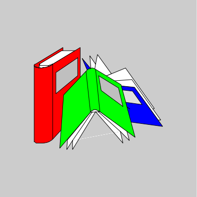
After you assign Virtual Placeholder status to the DRC 3210 digital relay output module (by selecting **Not Present** in its Options tab), the Advantys STB configuration software marks the virtual placeholder module with a red "X" as shown below:



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.

Glossary



0-9

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 usually 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 usually 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 requires a 5 A fuse to protect the I/O.

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 must 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 ensure 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 error checking 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. Differential design solves the problem of ground differences found in single-ended connections, and it also reduces the cross-channel noise problem.

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, malfunction, 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 must 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, usually 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, hang 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.

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, usually 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 must 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.

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 usually 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 must 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 must 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 always 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 all 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 clustered 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 earth. A return line across the bus for fault currents generated at a sensor or actuator device in 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 ATV31, 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 must 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.

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 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.

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 must 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 and protected 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 must be grounded, and the signal ground and data acquisition interface ground (the PDM lead) must 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, hang 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, hang 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, hang 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 (usually 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; must 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 to protect the input modules and an 8 A fuse to protect 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 reliable 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. Never mix modules with different voltage requirements in the same voltage group.

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 generates a fault.

Index