User's Manual

NX Interface Module

IM 34M06H29-01E

Applicable Modules:

Model Code	Model Name
F3NX01-0N	NX Interface Module
F3NX01-1N	NX Interface Module

YOKOGAWA
Yokogawa Electric Corporation

IM 34M06H29-01E
2nd Edition

Blank Page

Applicable Product:

● Range-free Multi-controller FA-M3

- Model: F3NX01-0N, F3NX01-1N
- Name: NX Interface Module

The document number and document model code for this manual are given below.

Refer to the document number in all communications; also refer to the document number or the document model code when purchasing additional copies of this manual.

Document No.	:	IM 34M06H29-01E
Document Model Code	:	DOCIM

Media No. IM 34M06H29-01E (CD) 2nd Edition : Feb, 2011 (AR)

Important

■ About This Manual

- This Manual should be passed on to the end user.
- Before using the controller, read this manual thoroughly to have a clear understanding of the controller.
- This manual explains the functions of this product, but there is no guarantee that they will suit the particular purpose of the user.
- Under absolutely no circumstances may the contents of this manual be transcribed or copied, in part or in whole, without permission.
- The contents of this manual are subject to change without prior notice.
- Every effort has been made to ensure accuracy in the preparation of this manual. However, should any errors or omissions come to the attention of the user, please contact the nearest Yokogawa Electric representative or sales office.

■ Safety Precautions when Using/Maintaining the Product

- The following safety symbols are used on the product as well as in this manual.

Danger. This symbol on the product indicates that the operator must follow the instructions laid out in this instruction manual to avoid the risk of personnel injuries, fatalities, or damage to the instrument. The manual describes what special care the operator must exercise to prevent electrical shock or other dangers that may result in injury or the loss of life.

Protective Ground Terminal. Before using the instrument, be sure to ground this terminal.

Function Ground Terminal. Before using the instrument, be sure to ground this terminal.

Alternating current. Indicates alternating current.

Direct current. Indicates direct current.

The following symbols are used only in the instruction manual.

WARNING

- Indicates a “Warning”.
- Draws attention to information essential to prevent hardware damage, software damage or system failure.

CAUTION

- Indicates a “Caution”
- Draws attention to information essential to the understanding of operation and functions.

TIP

- Indicates a “TIP”
- Gives information that complements the present topic.

SEE ALSO

- Indicates a “SEE ALSO” reference.
- Identifies a source to which to refer.

- For the protection and safe use of the product and the system controlled by it, be sure to follow the instructions and precautions on safety stated in this manual whenever handling the product. Take special note that if you handle the product in a manner other than prescribed in these instructions, the protection feature of the product may be damaged or impaired. In such cases, Yokogawa cannot guarantee the quality, performance, function and safety of the product.
- When installing protection and/or safety circuits such as lightning protection devices and equipment for the product and control system as well as designing or installing separate protection and/or safety circuits for fool-proof design and fail-safe design of processes and lines using the product and the system controlled by it, the user should implement it using devices and equipment, additional to this product.
- If component parts or consumable are to be replaced, be sure to use parts specified by the company.
- This product is not designed or manufactured to be used in critical applications which directly affect or threaten human lives and safety — such as nuclear power equipment, devices using radioactivity, railway facilities, aviation equipment, air navigation facilities, aviation facilities or medical equipment. If so used, it is the user’s responsibility to include in the system additional equipment and devices that ensure personnel safety.
- Do not attempt to modify the product.

■ Exemption from Responsibility

- Yokogawa Electric Corporation (hereinafter simply referred to as Yokogawa Electric) makes no warranties regarding the product except those stated in the WARRANTY that is provided separately.
- Yokogawa Electric assumes no liability to any party for any loss or damage, direct or indirect, caused by the use or any unpredictable defect of the product.

■ Software Supplied by the Company

- Yokogawa Electric makes no other warranties expressed or implied except as provided in its warranty clause for software supplied by the company.
- Use the software with one computer only. You must purchase another copy of the software for use with each additional computer.
- Copying the software for any purposes other than backup is strictly prohibited.
- Store the original media, such as floppy disks, that contain the software in a safe place.
- Reverse engineering, such as decompiling of the software, is strictly prohibited.
- No portion of the software supplied by Yokogawa Electric may be transferred, exchanged, or sublet or leased for use by any third party without prior permission by Yokogawa Electric.

■ General Requirements for Using the FA-M3 Controller

● Avoid installing the FA-M3 controller in the following locations:

- Where the instrument will be exposed to direct sunlight, or where the operating temperature exceeds the range 0°C to 55°C (32°F to 131°F).
- Where the relative humidity is outside the range 10 to 90%, or where sudden temperature changes may occur and cause condensation.
- Where corrosive or flammable gases are present.
- Where the instrument will be exposed to direct mechanical vibration or shock.
- Where the instrument may be exposed to extreme levels of radioactivity.

● Select an appropriate field wiring material:

- USE COPPER CONDUCTORS ONLY.

  Use copper conductors having temperature rating of minimum 75°C for the field wiring.

● Securely tighten screws:

- Securely tighten module mounting screws and terminal screws to avoid problems such as faulty operation.
- Tighten terminal block screws with the correct tightening torque as given in this manual.

● Securely lock connecting cables:

- Securely lock the connectors of cables, and check them thoroughly before turning on the power.

● Interlock with emergency-stop circuitry using external relays:

- Equipment incorporating the FA-M3 controller must be furnished with emergency-stop circuitry that uses external relays. This circuitry should be set up to interlock correctly with controller status (stop/run).

● Ground for low impedance:

- For safety reasons, connect the [FG] grounding terminal to a Japanese Industrial Standards (JIS) Class D (earlier called Class 3) Ground[*1]. For compliance to CE Marking, use braided or other wires that can ensure low impedance even at high frequencies for grounding.

  *1 Japanese Industrial Standard (JIS) Class D Ground means grounding resistance of 100 Ω max.

● Configure and route cables with noise control considerations:

- Perform installation and wiring that segregates system parts that may likely become noise sources and system parts that are susceptible to noise. Segregation can be achieved by measures such as segregating by distance, installing a filter or segregating the grounding system.

● Configure for CE Marking Conformance:

- For compliance to CE Marking, perform installation and cable routing according to the description on compliance to CE Marking in the “Hardware Manual” (IM34M06C11-01E).

- **Keep spare parts on hand:**
  - Stock up on maintenance parts including spare modules, in advance.
  - Preventive maintenance (replacement of the module or its battery) is required for using the module beyond 10 years. For enquiries on battery replacement service, contact your nearest Yokogawa Electric representative or sales office. (The module has a built-in lithium battery. Lithium batteries may exhibit decreased voltage, and in rare cases, leakage problems after ten years.)

- **Discharge static electricity before operating the system:**
  - Because static charge can accumulate in dry conditions, first touch grounded metal to discharge any static electricity before touching the system.

- **Never use solvents such as paint thinner for cleaning:**
  - Gently clean the surfaces of the FA-M3 controller with a cloth that has been soaked in water or a neutral detergent and wringed.
  - Do not use volatile solvents such as benzine or paint thinner or chemicals for cleaning, as they may cause deformity, discoloration, or malfunctioning.

- **Avoid storing the FA-M3 controller in places with high temperature or humidity:**
  - Since the CPU module has a built-in battery, avoid storage in places with high temperature or humidity.
  - Since the service life of the battery is drastically reduced by exposure to high temperatures, take special care (storage temperature should be from -20°C to 75°C).
  - There is a built-in lithium battery in a CPU module and temperature control module which serves as backup power supply for programs, device information and configuration information. The service life of this battery is more than 10 years in standby mode at room temperature. Take note that the service life of the battery may be shortened when installed or stored at locations of extreme low or high temperatures. Therefore, we recommend that modules with built-in batteries be stored at room temperature.

- **Always turn off the power before installing or removing modules:**
  - Failing to turn off the power supply when installing or removing modules, may result in damage.

- **Do not touch components in the module:**
  - In some modules you can remove the right-side cover and install ROM packs or change switch settings. While doing this, do not touch any components on the printed-circuit board, otherwise components may be damaged and modules may fail to work.

- **Do not use unused terminals:**
  - Do not connect wires to unused terminals on a terminal block or in a connector. Doing so may adversely affect the functions of the module.

■ Waste Electrical and Electronic Equipment

Waste Electrical and Electronic Equipment (WEEE), Directive 2002/96/EC

(This directive is only valid in the EU.)

This product complies with the WEEE Directive (2002/96/EC) marking requirement.

The following marking indicates that you must not discard this electrical/electronic product in domestic household waste.

Product Category

With reference to the equipment types in the WEEE directive Annex 1, this product is classified as a “Monitoring and Control instrumentation” product.

Do not dispose in domestic household waste.

When disposing products in the EU, contact your local Yokogawa Europe B. V. office.

■ How to Dispose of the Battery Used in This Product

The following description about the new Battery Directive 2006/66/EC is only valid in the EU.

This product uses an embedded battery, which cannot be removed by a customer and should be disposed of together with the product.

Do not dispose in domestic household waste.

When disposing products in the EU, contact your local Yokogawa Europe B. V. office.

Battery category: Lithium battery

Note: With reference to Annex II of the new Battery Directive 2006/66/EC, the above symbol indicates obligatory separate collection.

Introduction

■ Overview of the Manual

This user manual, “NX Interface Module” (IM 34M06H29-01E), explains the specifications of the NX Interface Module (F3NX01-0N, F3NX01-1N) and the NeXUS protocol service for use with autonomous distribution systems.

■ Compatible CPU Modules

For details on the CPU modules which can be used with the NX Interface Module, see Subsection A1.2,3, “Restrictions on I/O Module Installation” of the “Hardware Manual” (IM34M06C11-01E).

■ About NeXUS

NeXUS means a “bond” or a “chained system of [thing – concept]”.

Copyrights and Trademarks

■ Copyrights

Copyrights of the programs and online manual included in this CD-ROM belong to Yokogawa Electric Corporation.

This online manual may be printed but PDF security settings have been made to prevent alteration of its contents.

This online manual may only be printed and used for the sole purpose of operating this product. When using a printed copy of the online manual, pay attention to possible inconsistencies with the latest version of the online manual. Ensure that the edition agrees with the latest CD-ROM version.

Copying, passing, selling or distribution (including transferring over computer networks) of the contents of the online manual, in part or in whole, to any third party, is strictly prohibited. Registering or recording onto videotapes and other media is also prohibited without expressed permission of Yokogawa Electric Corporation.

■ Trademarks

- The trade and company names that are referred to in this document are either trademarks or registered trademarks of their respective companies.

Blank Page

FA-M3

NX Interface Module

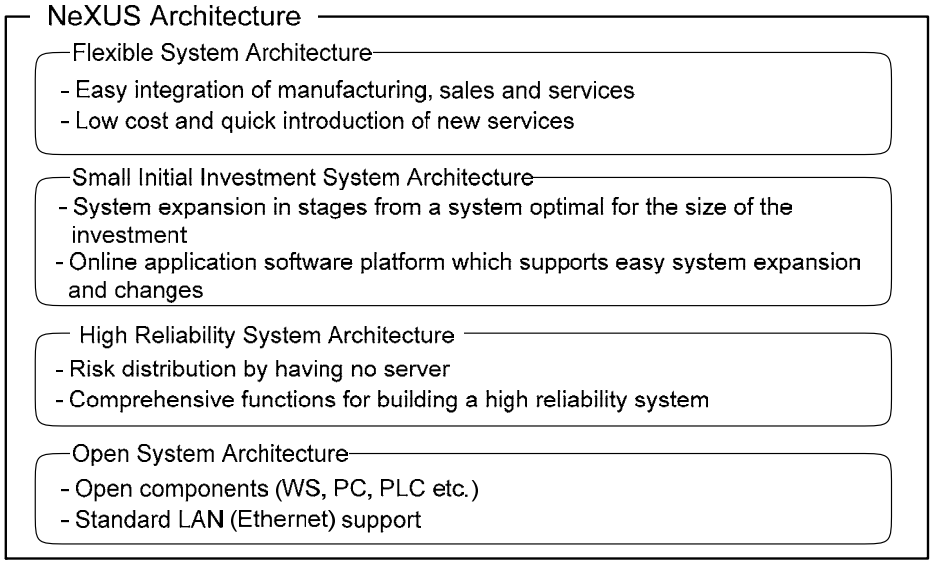
IM 34M06H29-01E 2nd Edition

CONTENTS

1. Overview

1.1 Autonomous Distributed Architecture

Autonomous Distributed Architecture, also called Nexus Architecture, covers small and large distributed systems. As an architecture which integrates the management of workstations (WS), personal computers (PC) and even components of various control equipment (such as controllers) from a real-time server in a system, it achieves the following four main objectives.



1.2 Support for Autonomous Distributed Architecture

The NX Interface Module supports “Category A” Ethernet-based functions of the open autonomous distributed interface, as listed in Table 1.1.

Table 1.1 Category A Communication Function List （✓: Supported, ×: Not supported）

Primary Functions			Secondary Functions		
Class	**Functions**		**Class**	**Functions**	
A-Base-1	Multicast communication	✓	A-Opt-1-a	Redundant LAN control multicast communication	×
A-Base-2	Alive signal transmission	✓	A-Opt-2-a	Fault information transmission	✓
A-Opt-3	One-to-one communication (Max. 16KB)	×	–	–	–
A-Opt-4	Configuration over a network	×	–	–	–

Tip

The level of support provided by the NX Interface Module for the common functions presumed in “Category A” is described below.

- Test nodes are supported.
- Message priority control is not supported. The NX Interface Module always transmits 0 for the PRI (priority level) stored in the NeXUS header.
- Message numbering during message transmission is supported but checking of message number for received messages is not supported.

1.3 System Features

The system features of the NX Interface Module are described below.

● Multicast Communication

A data field is a place where data containing information of a specific type and nature flows. Each node transmits data to the data field to which it belongs by multicast communication; other nodes belonging to the same data field can select to receive this data autonomously. By "autonomously", we mean that there is no need for identification of or synchronization with communication counterparts. By dividing the entire system into several data fields, it increases subsystem independence, system extensibility and maintainability.

● Alive Signal Transmission

By using alive signals, status information of equipment connected to a data field, programs in equipment and hardware can be autonomously collected for configuration management and control of equipment and programs contained in the equipment.

● Fault Monitoring

By adding fault information to alive signals, the location, cause and detailed information of faults that have occurred in the hardware or programs in equipment connected to a data field can be monitored on a PC.

● Test Support

Testing can be performed in an online system without affecting the system. At the end of the test, transition from the test environment to the online environment can be achieved without stopping the system and without changing the user program.

● Extensibility

New node insertion and user program configuration can be done online without stopping the existing online system, allowing easy stepwise system expansions.

1.4 System Configuration

1.4.1 Network Configuration

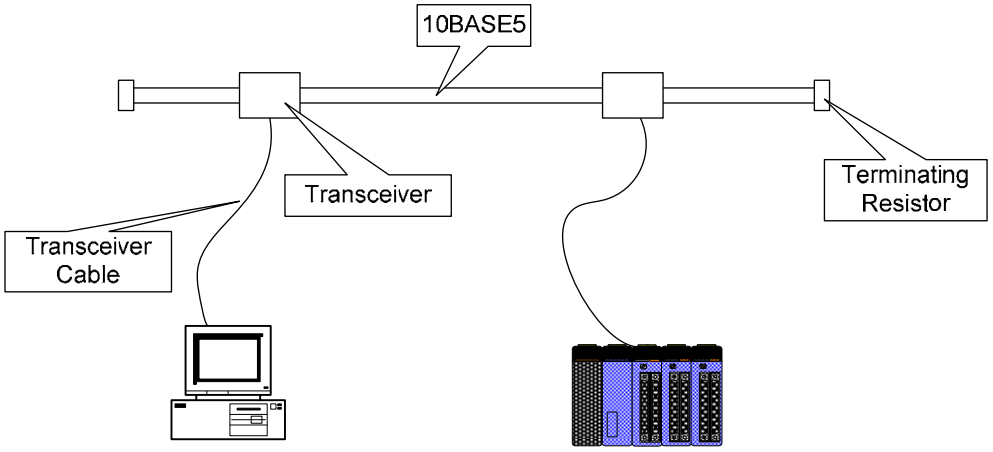
This section describes the hardware components required for configuring a network system.

■ Connection Using 10BASE5

10BASE5 is an Ethernet connection method using coaxial cables.

The transmission rate is 10Mbps.

To connect an instrument such as a PC to the network, connect it to a transceiver using a transceiver cable (AUI cable) and connect the transceiver to the network coaxial cable.



Only the F3NX01-0N can be connected directly to a transceiver cable. A media converter is required for connecting F3NX01-1N to a transceiver cable.

CAUTION

Use 10BASE5 coaxial cables, transceivers, terminating resistors and transceiver cables that conform to the Ethernet (10BASE5) specifications.

The distance between two transceivers should be an integral multiple of 2.5 m.

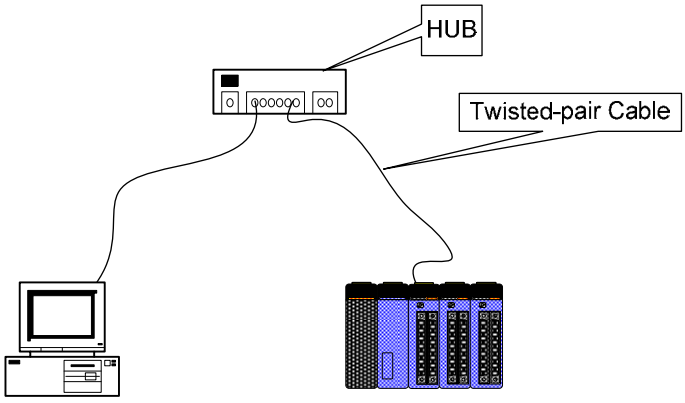
The maximum allowable length for the transceiver cable is 50 m.

■ Connection Using 10BASE-T

10BASE-T is an Ethernet connection method using twisted-pair cables.

The transmission rate is 10Mbps.

In a 10BASE-T network, instruments such as personal computers are connected to a hub using a star topology.



FA2002.VSD

CAUTION

Use hubs and twisted-pair cables, which conform to the Ethernet (10BASE-T) specifications.

Up to 4 segments are allowed for cascade connections to the hub.

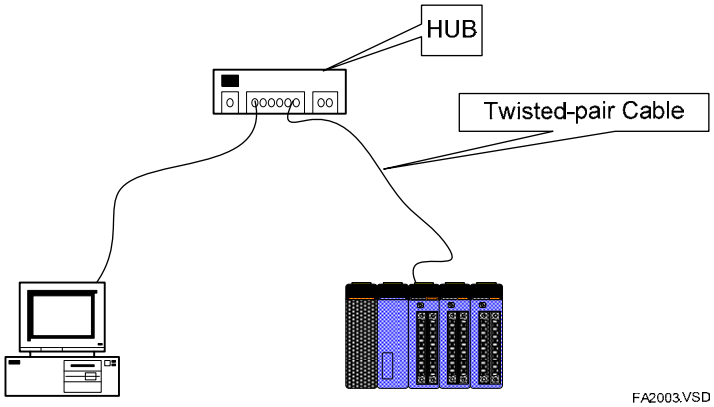
The maximum length allowed for the twisted-pair cables is 100 m.

■ Connection Using 100BASE-TX

100BASE-TX is another Ethernet connection method using twisted-pair cables.

The transmission rate is 100Mbps.

In a 100BASE-TX network, instruments such as personal computers are connected to a hub using a star topology.



Only the F3NX01-1N can be used in a 100BASE-TX connection. The F3NX01-0N cannot be used for a 100BASE-TX connection.

CAUTION

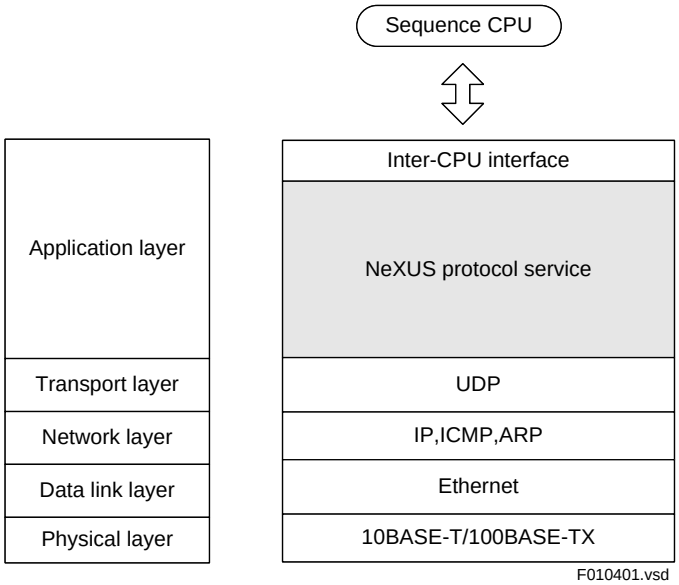
Use hubs and twisted-pair cables, which conform to the Ethernet (100BASE-TX) specifications.

Up to 2 segments are allowed for cascade connections to the hub.

The maximum length allowed for the twisted-pair cables is 100 m.

1.4.2 Software Configuration

The figure below shows the position of the NeXUS protocol services in the network hierarchy



IP : Internet Protocol

UDP : User Datagram Protocol

ICMP : Internet Control Message Protocol

ARP : Address Resolution Protocol

Parts of the software written by The Regents of University of California have been incorporated.

Figure 1.1 Software Configuration

1.4.3 Functional Specifications

■ Model Name and Suffix Codes

Model	Suffix Code	Style Code	Option Code	Description
F3NX01	-0N	……	……	10 Mbps 10BASE5/10BASE-T
	-1N	……	……	10 Mbps/100 Mbps 10BASE-T/100BASE-TX

■ Compatible CPU Modules

The following table lists the CPU modules compatible with the F3NX01-0N and F3NX01-1N modules.

	F3SP05	F3SP08	F3SP21	F3SP22	F3SP25	F3SP28	F3SP35	F3SP38	F3SP53	F3SP58	F3SP59	F3SP66	F3SP67	F3SP71	F3SP76
F3NX01-0N	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
F3NX01-1N	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

■ General Specifications

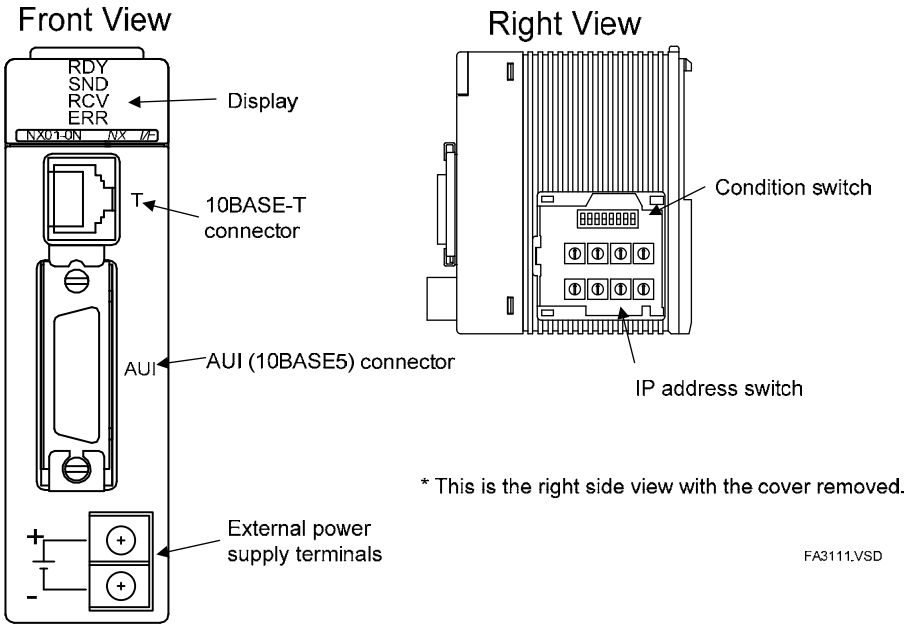
Item		Description			
		F3NX01-0N		F3NX01-1N	
		10BASE5	10BASE-T	10BASE-T	100BASE-TX
Transmission Specifications	Access control	CSMA/CD			
	Transmission rate	10 Mbps			100 Mbps
	Transmission method	Baseband			
	Maximum segment length	500 m	100 m *1		
	Maximum distance between nodes	2500 m	—		
	Largest connection configuration	100 per segment	4 cascade segments max.		2 cascade segments max.
Protocol		UDP/IP, ICMP, ARP			
Number of installed modules		2 max. when used with F3SP05, F3SP08, F3SP21 6 max. when used with F3SP22, F3SP25, F3SP28, F3SP35, F3SP38, F3SP53, F3SP58, F3SP59, F3SP66, F3SP67, F3SP71, F3SP76*3 * Total number including other modules with similar functions (personal computer link modules, FL-net interface modules, etc.)			
Internal current consumption		330 mA		500 mA	
External power supply		12V DC	—		
Fuse		2A time lag fuse (embedded in the power supply terminal and not replaceable.)		—	
External Dimensions		28.98 (W) x 100 (H) x 83.2 (D) mm *2			
Weight		130 g			
Operating ambient temperature		0 to 55°C			
Operating ambient humidity		10 to 90% RH (non-condensing)			
Operating ambient atmosphere		Must be free of corrosive gases, flammable gases and heavy dust			
Storage ambient temperature		-20 to 75°C			
Storage ambient humidity		10 to 90% RH (non-condensing)			

*1: The maximum distance between the module and a hub.
*2: Excluding protruding parts (For details, see “External Dimensions.”)
*3: The NX Interface Module cannot be mounted in a subunit.

1.4.4 Components and Their Functions

1.4.4.1 F3NX01-0N

■ Appearance and Components



■ Component Functions

● LED Display

The LEDs turn on or off to indicate the operation status.

Name (color)	Description	Lit	Flashing	Off
RDY (green)	Internal circuitry status	Normal		Error
SND (green)	Data sending status	Sending data		Not sending data
RCV (green)	Data receiving status	Carrier detected		No carrier detected
ERR (red)	Error status	Error detected		Normal

● AUI Connector

This connector is used for connecting to a transceiver cable.

● 10BASE-T Connector

This connector is used for connecting to a 10BASE-T (UTP/STP) connection.

CAUTION

The 10BASE-T connector and the AUI connector cannot be used simultaneously. Do not connect cables to these two connectors at the same time.

● External Power Supply Terminals

When using the AUI connector, external power source is required to supply power to the transceiver via the AUI connector. No external power is required when using the 10BASE-T connector.

TIP

IEEE802.3 specifies:

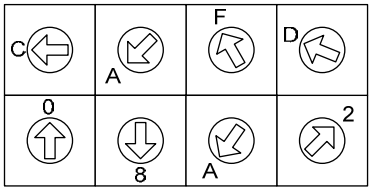
Transceiver input voltage:	12V-6% to 15V+5%
Transceiver DC resistance:	40Ω/km, 50m maximum
Transceiver power consumption:	500mA maximum

Taking into consideration the 0.4 V voltage drop for the NX Interface Module, the target external power supply is 12.68V to 15.75V. However, note that the input voltage of some IEEE802.3-compatible transceivers do not fall within the range of 12V-6% to 15V+5%.

● IP Address Switch

This is a 16-position rotary switch for setting the IP address.

Example: Setting the IP address to 192.168.250.210



Hexadecimal	C0	A8	FA	D2	
	↑	↑	↑	↑	
Decimal	192	. 168	. 250	. 210	FA3112.VSD

CAUTION

Do not touch the components inside the module when setting the IP address switch. Doing so may damage the components or lead to module failure.

● Condition Switch

These are switches for setting the operating conditions.

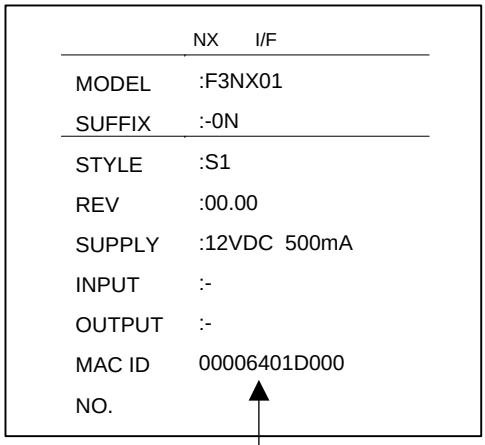
Number	Function	OFF	ON	Factory Setting	
1	Always OFF	–	–	–	OFF
2					
3					
4					
5					
6					
7					
8	Loop-back test	Normal	Test	Normal	OFF

CAUTION

Do not touch the components inside the module when setting the condition switch. Doing so may damage the components or lead to module failure.

● MAC Address

The MAC address is labeled on the left face of the module.

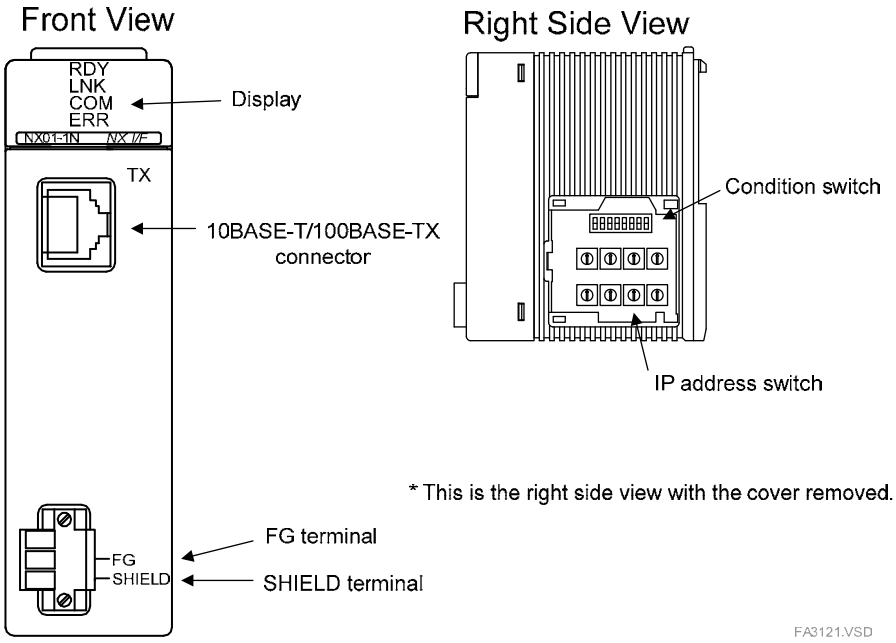


MAC address:
A unique 12-digit hexadecimal number is assigned to every NX interface module.

FA3113.vsd

1.4.4.2 F3NX01-1N

■ Appearance and Components



■ Component Functions

● LED Display

The LEDs turn on or off to indicate the operation status.

Name (Color)	Description	On	Flashing	Off
RDY (Green)	Internal circuit status	Normal		Error
LNK (Green)	Network status	Connected		Not connected
COM (Green)	Data sending/receiving status		Carrier detected	No carrier detected
ERR (Red)	Operation status	Error		Normal

● 10BASE-T/100BASE-TX Connector

This connector is used for connecting to a 10BASE-T/100BASE-TX (UTP/STP) cable.

● SHIELD terminal

The SHIELD terminal is connected to the SHIELD wire of the 10BASE-T/100BASE-TX connector.

It is isolated from the FG terminal of the power module.

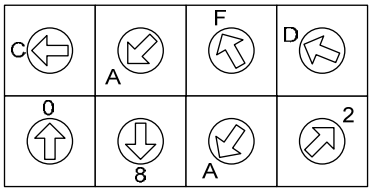
● FG terminal

The FG terminal is connected to the FG terminal of the power supply module.

● IP Address Switch

This is a 16-position rotary switch for setting the IP address.

Example: Setting the IP address to 192.168.250.210



Hexadecimal	C0		A8		FA		D2			
	↑		↑		↑		↑			
Decimal	192	.	168	.	250	.	210			

FA3112.VSD

CAUTION

Do not touch the components inside the module when setting the IP address. Doing so may damage the components or lead to module failure.

● Condition Switch

These are switches for setting the operating conditions.

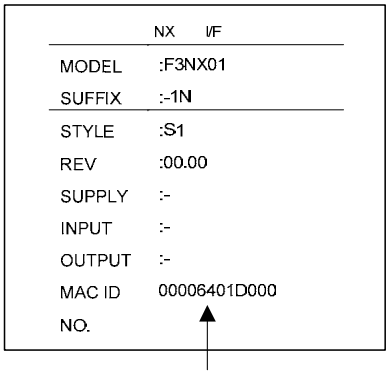
Number	Function	OFF	ON	Factory Setting	
1	Always OFF	—	—	—	OFF
2					
3					
4					
5					
6					
7					
8	Loop-back test	Normal	Test	Normal	OFF

CAUTION

Do not touch the components inside the module when setting the condition switch. Doing so may damage the components or lead to module failure.

● MAC Address

The MAC address of the module is labeled on the left face of the module.



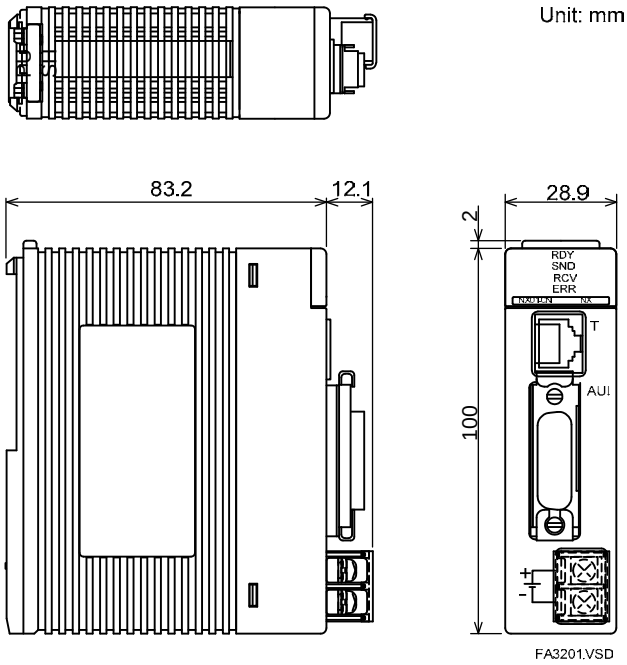
MAC address:
A unique 12-digit hexadecimal number is assigned to every NX interface module.

FA3113.vsd

1.4.4.3 External Dimensions

● F3NX01-0N

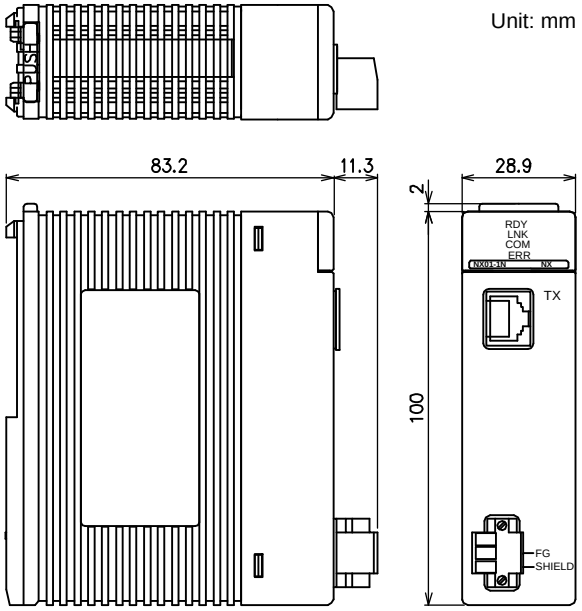
Unit: mm



FA3201.VSD

● F3NX01-1N

Unit: mm



FA3202.VSD

1.4.4.4 Input/Output Relays and Registers

● Input/Output Relays

Input Relay Number	Description
X0□□01	Transmission handle 1 Transmission completed
X0□□02	Transmission handle 2 Transmission completed
X0□□03	Transmission handle 3 Transmission completed
X0□□04	Transmission handle 4 Transmission completed
X0□□05	Transmission handle 5 Transmission completed
X0□□06	Transmission handle 6 Transmission completed
X0□□07	Transmission handle 7 Transmission completed
X0□□08	Transmission handle 8 Transmission completed
X0□□09	Transmission handle 9 Transmission completed
X0□□10	Transmission handle 10 Transmission completed
X0□□11	Transmission handle 11 Transmission completed
X0□□12	Transmission handle 12 Transmission completed
X0□□13	Transmission handle 13 Transmission completed
X0□□14	Transmission handle 14 Transmission completed
X0□□15	Transmission handle 15 Transmission completed
X0□□16	Transmission handle 16 Transmission completed
X0□□17	Receiving handle 1 Request to receive
X0□□18	Receiving handle 2 Request to receive
X0□□19	Receiving handle 3 Request to receive
X0□□20	Receiving handle 4 Request to receive
X0□□21	Receiving handle 5 Request to receive
X0□□22	Receiving handle 6 Request to receive
X0□□23	Receiving handle 7 Request to receive
X0□□24	Receiving handle 8 Request to receive
X0□□25	Receiving handle 9 Request to receive
X0□□26	Receiving handle 10 Request to receive
X0□□27	Receiving handle 11 Request to receive
X0□□28	Receiving handle 12 Request to receive
X0□□29	Optional information transmission completed
X0□□30	Initialization completed
X0□□31	Initialization error exit
X0□□32	Mode status

Output Relay Number	Description
Y0□□33	Transmission handle 1 Request to Transmit
Y0□□34	Transmission handle 2 Request to Transmit
Y0□□35	Transmission handle 3 Request to Transmit
Y0□□36	Transmission handle 4 Request to Transmit
Y0□□37	Transmission handle 5 Request to Transmit
Y0□□38	Transmission handle 6 Request to Transmit
Y0□□39	Transmission handle 7 Request to Transmit
Y0□□40	Transmission handle 8 Request to Transmit
Y0□□41	Transmission handle 9 Request to Transmit
Y0□□42	Transmission handle 10 Request to Transmit
Y0□□43	Transmission handle 11 Request to Transmit
Y0□□44	Transmission handle 12 Request to Transmit
Y0□□45	Transmission handle 13 Request to Transmit
Y0□□46	Transmission handle 14 Request to Transmit
Y0□□47	Transmission handle 15 Request to Transmit
Y0□□48	Transmission handle 16 Request to Transmit
Y0□□49	Receiving handle 1 Receiving completed
Y0□□50	Receiving handle 2 Receiving completed
Y0□□51	Receiving handle 3 Receiving completed
Y0□□52	Receiving handle 4 Receiving completed
Y0□□53	Receiving handle 5 Receiving completed
Y0□□54	Receiving handle 6 Receiving completed
Y0□□55	Receiving handle 7 Receiving completed
Y0□□56	Receiving handle 8 Receiving completed
Y0□□57	Receiving handle 9 Receiving completed
Y0□□58	Receiving handle 10 Receiving completed
Y0□□59	Receiving handle 11 Receiving completed
Y0□□60	Receiving handle 12 Receiving completed
Y0□□61	Request to transmit optional information
Y0□□62	Request to initialize
Y0□□63	
Y0□□64	Request to change mode

□□: FA-M3 slot number where the NX Interface Module is mounted.

● Registers

Channel Number	Item			Description
1 to 7168	Transmission and receiving buffers			Data areas for sending and receiving are allocated according to the specified transmission and receiving buffer case sizes
7169	Receiving Handle 1 receiving data size			Receiving data sizes
7170	Receiving Handle 2 receiving data size			
7171	Receiving Handle 3 receiving data size			
7172	Receiving Handle 4 receiving data size			
7173	Receiving Handle 5 receiving data size			
7174	Receiving Handle 6 receiving data size			
7175	Receiving Handle 7 receiving data size			
7176	Receiving Handle 8 receiving data size			
7177	Receiving Handle 9 receiving data size			
7178	Receiving Handle 10 receiving data size			
7179	Receiving Handle 11 receiving data size			
7180	Receiving Handle 12 receiving data size			
7181 to 7314	Reserved			—
7315 to 7319	Node name	Node information	Initialization data	10 ASCII-code characters
7320	Logical node number			1 to 4095
7321	Local DF number			1 to 255
7322	Node mode at system startup			Specify the mode at system startup
7323	Connected CPU numbers			Specify CPUs using bit data
7324	Reserved			
7325				
7326	Transmission and receiving buffer case sizes			Bit data
7327	Port number for transmitting messages			$0000 to $FFFF
7328	Port number for receiving alive signal			$0000 to $FFFF
7329	Alive signal transmission interval			$0000 to $0E10 (seconds)
7330	Alive signal timeout interval			$0000 to $A8C0 (seconds)
7331	Transmission and receiving handle registration counts			Specify using bit data
7332	Transmission and receiving multicast group registration count for local DF			Specify using bit data
7333	Reserved			
7334				
7335	Remote DF number			1 to 255
7336	Broadcast address for remote DF			Broadcast IP address www.xxx.yyy.zzz Channel no. 7337 7336
7337				
7338	Transmission multicast group registration count for remote DF			0 to 4

Channel no.	Item				Description
7339	Transmission handle 1	Transaction code	Transmission handle information	Initialization data	Transaction code: $0001 to $EA56 Multicast group number: 1 to 255 Transmission data size: 0 to 1024
7340		Multicast group number			
7341		Transmission data size			
7342	Transmission handle 2	Transaction code			
7343		Multicast group number			
7344		Transmission data size			
7345	Transmission handle 3	Transaction code			
7346		Multicast group number			
7347		Transmission data size			
7348	Transmission handle 4	Transaction code			
7349		Multicast group number			
7350		Transmission data size			
7351	Transmission handle 5	Transaction code			
7352		Multicast group number			
7353		Transmission data size			
7354	Transmission handle 6	Transaction code			
7355		Multicast group number			
7356		Transmission data size			
7357	Transmission handle 7	Transaction code			
7358		Multicast group number			
7359		Transmission data size			
7360	Transmission handle 8	Transaction code			
7361		Multicast group number			
7362		Transmission data size			
7363	Transmission handle 9	Transaction code			
7364		Multicast group number			
7365		Transmission data size			
7366	Transmission handle 10	Transaction code			
7367		Multicast group number			
7368		Transmission data size			
7369	Transmission handle 11	Transaction code			
7370		Multicast group number			
7371		Transmission data size			
7372	Transmission handle 12	Transaction code			
7373		Multicast group number			
7374		Transmission data size			
7375	Transmission handle 13	Transaction code			
7376		Multicast group number			
7377		Transmission data size			
7378	Transmission handle 14	Transaction code			
7379		Multicast group number			
7380		Transmission data size			
7381	Transmission handle 15	Transaction code			
7382		Multicast group number			
7383		Transmission data size			
7384	Transmission handle 16	Transaction code			
7385		Multicast group number			
7386		Transmission data size			

Channel no.	Item				Description
7387	Receiving handle 1	Transaction code	Receiving handle information	Initialization data	Transaction code: $0001 to $EA56
7388		Reserved			
7389	Receiving handle 2	Transaction code			
7390		Reserved			
7391	Receiving handle 3	Transaction code			
7392		Reserved			
7393	Receiving handle 4	Transaction code			
7394		Reserved			
7395	Receiving handle 5	Transaction code			
7396		Reserved			
7397	Receiving handle 6	Transaction code			
7398		Reserved			
7399	Receiving handle 7	Transaction code			
7400		Reserved			
7401	Receiving handle 8	Transaction code			
7402		Reserved			
7403	Receiving handle 9	Transaction code			
7404		Reserved			
7405	Receiving handle 10	Transaction code			
7406		Reserved			
7407	Receiving handle 11	Transaction code			
7408		Reserved			
7409	Receiving handle 12	Transaction code			
7410		Reserved			
7411	Transmission multicast group 1 for local DF	Group number	Transmission multicast group information		Group number: 1 to 255 Port number for online mode: $0001 to $FFFF Port number for test mode: $0001 to $FFFF
7412		Port number for online mode			
7413		Port number for test mode			
7414	Transmission multicast group 2 for local DF	Group number			
7415		Port number for online mode			
7416		Port number for test mode			
7417	Transmission multicast group 3 for local DF	Group number			
7418		Port number for online mode			
7419		Port number for test mode			
7420	Transmission multicast group 4 for local DF	Group number			
7421		Port number for online mode			
7422		Port number for test mode			
7423	Transmission multicast group 1 for remote DF	Group number			
7424		Port number for online mode			
7425		Port number for test mode			
7426	Transmission multicast group 2 for remote DF	Group number			
7427		Port number for online mode			
7428		Port number for test mode			
7429	Transmission multicast group 3 for remote DF	Group number			
7430		Port number for online mode			
7431		Port number for test mode			
7432	Transmission multicast group 4 for remote DF	Group number			
7433		Port number for online mode			
7434		Port number for test mode			

Channel no.	Item				Description
7435	Receiving multicast group 1	Group number	Receiving multicast group information	Initialization data	Group number: 1 to 255 Online port number: $0001 to $FFFF Port number for test mode: $0001 to $FFFF
7436		Port number for online mode			
7437		Port number for test mode			
7438	Receiving multicast group 2	Group number			
7439		Port number for online mode			
7440		Port number for test mode			
7441	Receiving multicast group 3	Group number			
7442		Port number for online mode			
7443		Port number for test mode			
7444	Receiving multicast group 4	Group number			
7445		Port number for online mode			
7446		Port number for test mode			
7447	Subnet mask		Routing information		Channel no. 7448 7447 (www.xxx.yyy.zzz)
7448					
7449	Gateway address				Channel no. 7450 7449 (www.xxx.yyy.zzz)
7450					
7451	Hop count				0: Direct connection Others: Via router
7452 to 7506	Reserved				
7507	Protocol violation message transmission source IP address		Protocol violation message		Channel no. 7508 7507 (www.xxx.yyy.zzz)
7508					
7509 to 7540	Protocol violation message header				
7541 to 7606	Reserved				
7607	Error detected module number		Error report area		
7608	Detailed exit code				

1.4.5 NX Interface Module Setup

1.4.5.1 Startup Procedure

The figure below shows a procedure flowchart for setting up an FA-M3 system using an NX Interface Module.

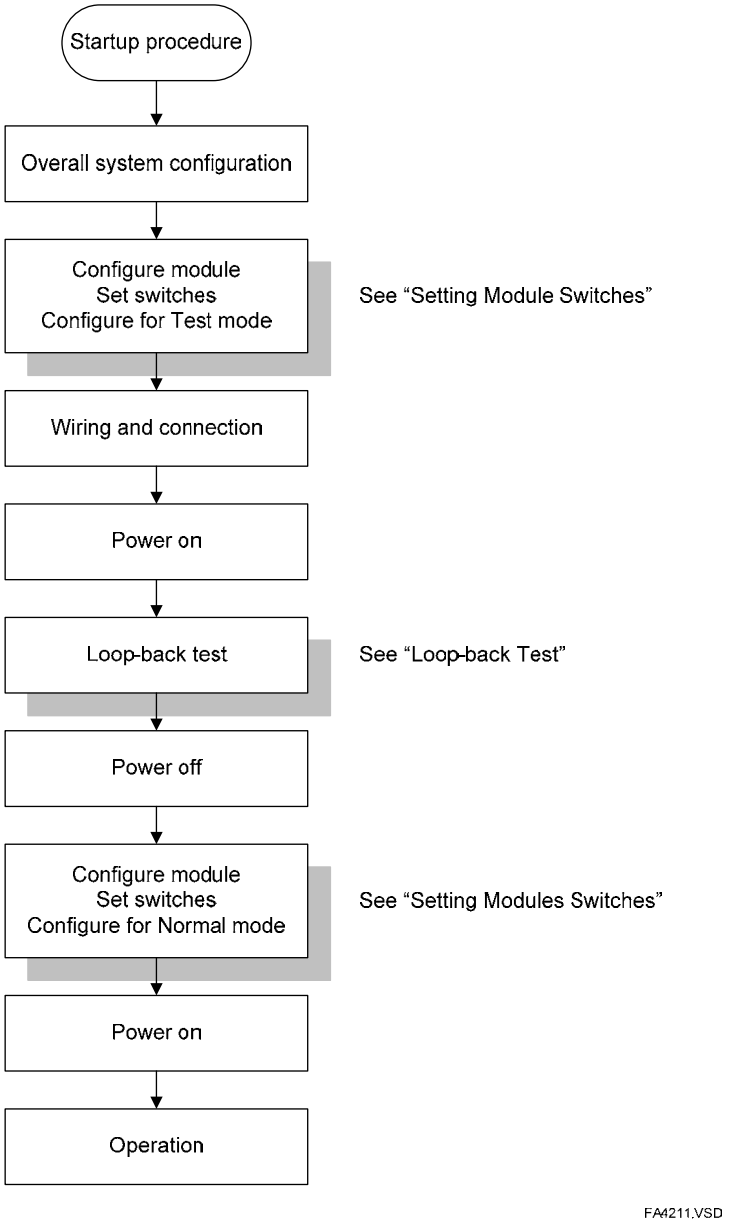


Figure 1.2 Startup Flowchart

1.4.5.2 Setting Module Switches

The NX Interface Module has switches that must be properly set before mounting the module onto the base module. The figure below shows the locations of these switches.

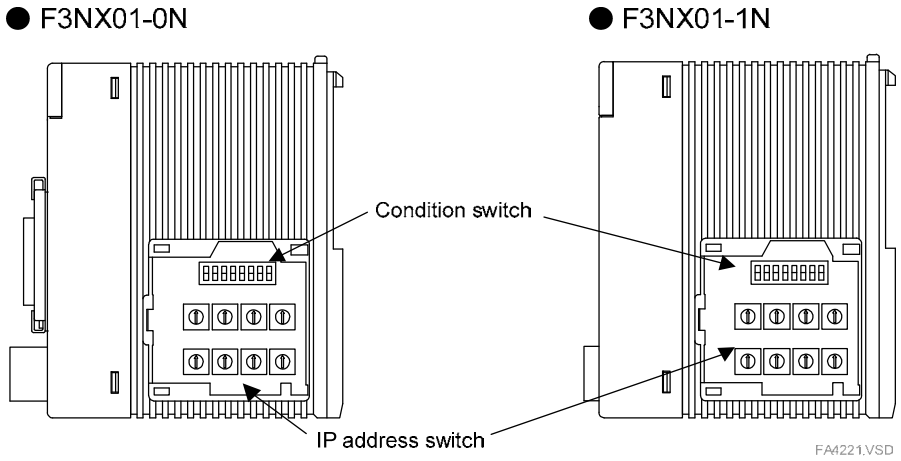


Figure 1.3 Locations of Module Switches

● IP Address Switch

Set the IP address with the 8 hexadecimal rotary switches located on the right side of the module (inside the cover). Set the IP address in hexadecimal representation as shown in the figure below.

The factory setting is 0.0.0.0.

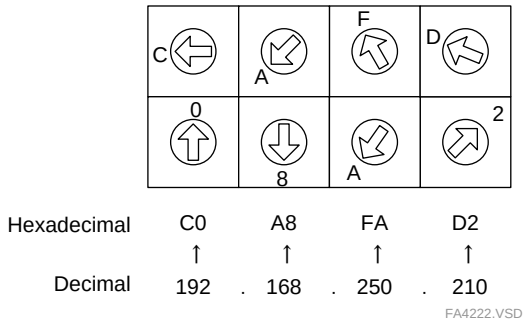


Figure 1.4 How to Set IP Address

● Condition Switch

You can configure operation conditions using the DIP switches located on the right side of the module (inside the cover).

Table 1.2 Condition Switch Settings

Number	Function	OFF	ON
1	System-reserved	Always OFF	Not configurable
2			
3			
4			
5			
6			
7			
8	Loop-back test	Normal	Test

● Loop-back Test

Use condition switch number 8 to define the operation mode of the NX Interface Module after power on.

Off: Normal

On: Test

In test mode, the module performs a loop-back test. The RDY LED lights up if the loop-back test is successful. If the test fails, the ERR LED lights up if the test fails.

At the end of the loop-back test, all functions of the module are disabled. The switch should normally be set to Normal mode (OFF) in runtime operation.

CAUTION

Do not perform a loop-back test with a sequence CPU mounted.

1.4.5.3 External Wiring

■ Connecting Communications Cable

● 10BASE5 Transceiver Cable Connection

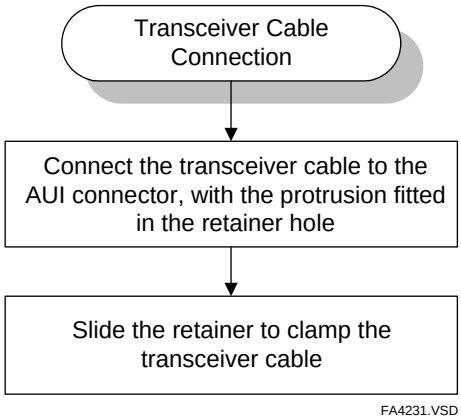


Figure 1.5 Transceiver Cable Connection Flowchart

● 10BASE-T/100BASE-TX Cable Connection

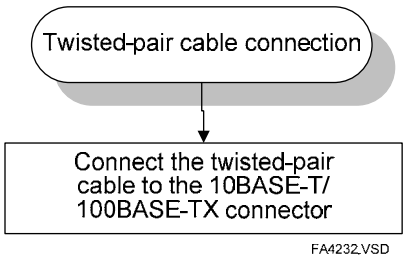


Figure 1.6 10BASE-T/100BASE-TX Cable Connection Flowchart

CAUTION

Do not connect a transceiver cable and a twisted-pair cable to the module simultaneously.

Adequate safety measures should be adopted when laying Ethernet cables. We recommend that you commission a professional contractor to lay the Ethernet cables.

■ Wiring of Power Supply

● F3NX01-0N

Power supply to the transceiver is required for 10BASE5 connections. Apply a 12V DC supply to the external power supply terminals. Some types of transceiver require no power supply. Power supply is also not required for 10BASE-T connections.

Use an AWG28-16 or equivalent cable for the power supply. Tighten the fixing screws for the power cable with an appropriate torque of 0.8 N•m.

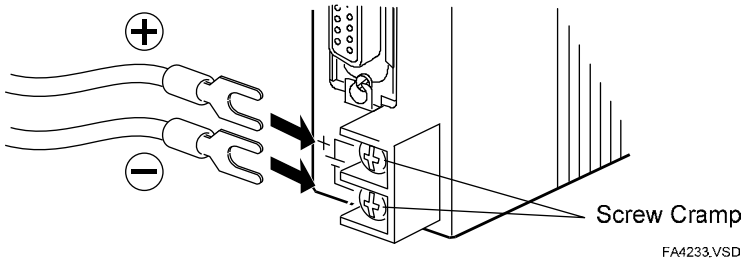


Figure 1.7 Wiring of Power Supply

CAUTION

Turn on the power with the transceiver cable connected. Connecting or disconnecting the transceiver cable with the power on may lead to system failure.

TIP

IEEE802.3 specifies:

Transceiver input voltage:	12V-6% to 15V+5%
Transceiver DC resistance:	40Ω/km, 50m maximum
Transceiver power consumption:	500mA maximum

Taking into consideration the 0.4 V voltage drop for the NX Interface Module, the target external power supply is 12.68V to 15.75V. However, note that the input voltage of some IEEE802.3-compatible transceivers do not fall within the range of 12V-6% to 15V+5%.

● F3NX01-1N

No wiring for power supply is required.

■ Shielding

● F3NX01-0N

When the NX Interface Module is attached to the base module, the shield wire of the transceiver and the connector shell are connected to the aluminum chassis and the FG terminal of the power module.

To enhance shielding effect, perform grounding according to the procedure (for compliance to CE marking) described in subsequent pages.

The following figure shows the internal wiring of the module.

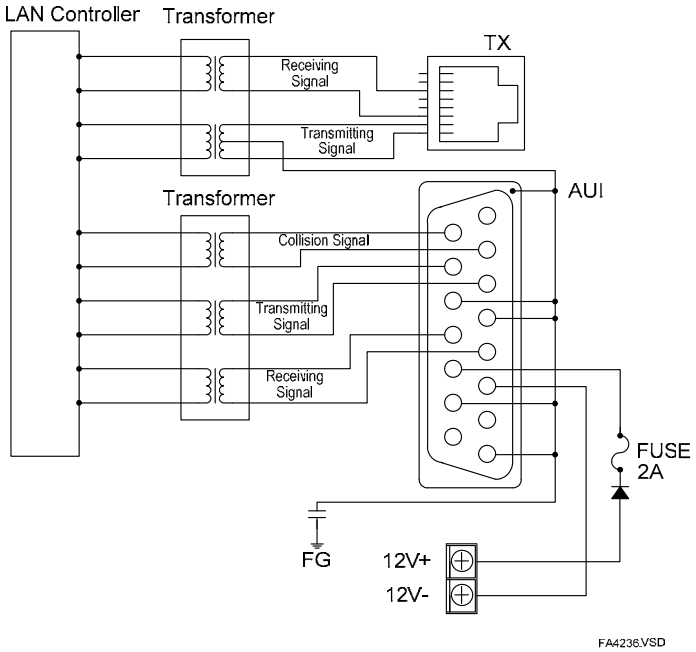


Figure 1.8 F3NX01-0N Connector Internal Wiring

● F3NX01-1N

When using a shielded twisted-pair cable (STP), ground the shield of the wire as required.

The shield of the cable is connected to the shell of the 10BASE-T/100BASE-TX connector, and the shell is connected to the shield terminal located inside the NX Interface Module.

There are two ways to ground the shielded wire.

To enhance the shielding effect, perform grounding according to the procedure (for CE marking compliance) described on the next page.

- **For connection to the aluminum chassis of the base module and the FG terminal of the power supply module via the NX Interface Module:**

  Connect the shield terminal of the NX Interface Module and the FG terminal at the connector.

- **For direct connection not via the NX Interface module:**

  Connect the shield terminal of the NX Interface Module directly to the inner plate of a low-impedance panel or the GND terminal inside a panel enclosure.

  Use an AWG28-16 or equivalent cable for grounding. Tighten the fixing screws for grounding appropriately with a torque of 0.8 N•m.

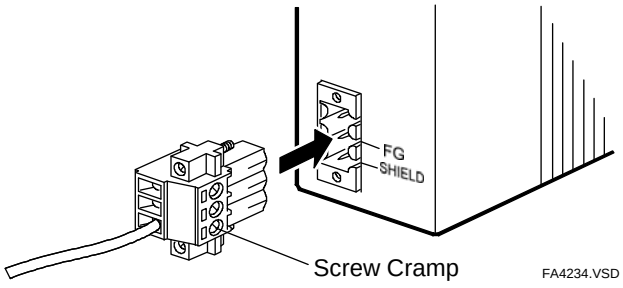


Figure 1.9 Grounding

The following figure shows the internal wiring of the module.

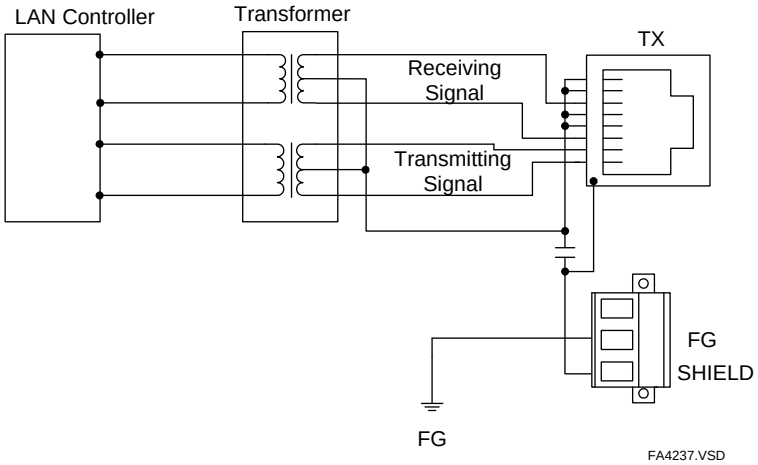
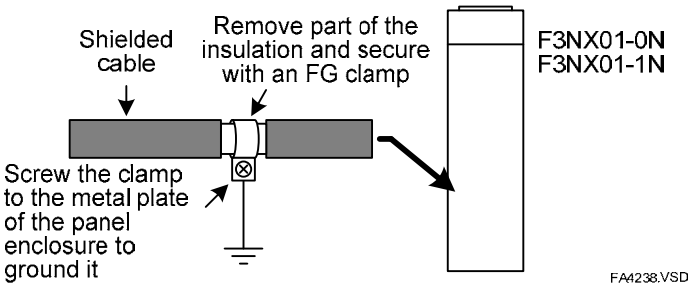


Figure 1.10 F3NX01-1N Connector Internal Wiring

■ CE Marking Conformance

- Use a shielded cable for conforming equipment incorporating the Ethernet Interface Module to CE Marking. Remove part of the cable insulation to expose the shield, ground and secure the shield with an FG clamp.



1.4.5.4 Loop-back Test

The NX Interface Module can perform self-diagnostics (loop-back test) according to the operating condition switches.

In a loop-back test, the module transmits test data over the network and checks if it can receive the same data.

● Test Method

(1) Turn on switch 8 of the condition switches located on the right side of the module.

(2) Connect the module to the network.

(3) Turn on the power.

(4) Testing begins. (The RDY LED blinks.)

● Test Result

- If the module correctly receiving the test data in the loop-back test,
  The RDY LED turns on.
- If the module fails to receive the test data correctly in the loop-back test,
  The ERR LED turns on.

At the end of the loop-back test, all functions of the module are disabled.

After the test, you must set the operation mode to Normal mode. Turn off Switch 8, turn off the power and then turn on the power again.

CAUTION

Do not perform a loop-back test with a sequence CPU mounted.

1.4.5.5 Attaching/Detaching the Module

■ Attaching the Module

The following figure shows how to attach the module to the base module. First hook the anchor slot at the bottom of NX Interface Module onto the anchor pin on the bottom of the base module. Push the top of NX Interface Module toward the base module until the anchor/release button clicks into place.

CAUTION

Always switch off the power before attaching or detaching the module.

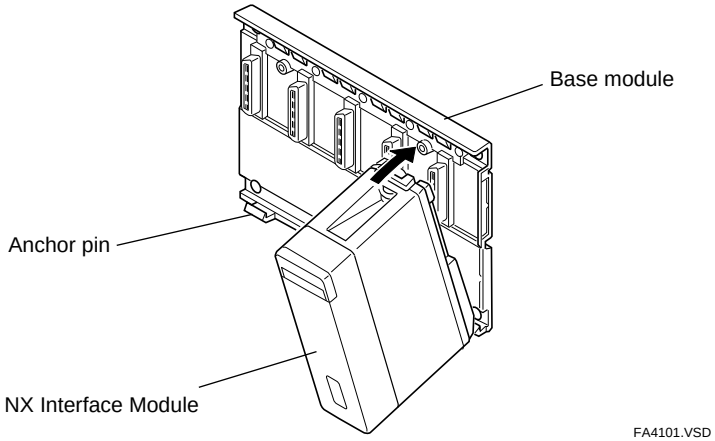


Figure 1.11 Attaching/Detaching the Module

CAUTION

DO NOT bend the connector on the rear of module by force during the above operation. If the module is pushed with improper force, the connector may bend causing an error.

■ Detaching the Module

To remove the module from the base module, reverse the above operation.

Press the anchor/release button on the top of the module to unlock it and tilt the module away from the base module.

■ Attaching Modules in Intense Vibration Environments

If the module is used in intense vibration environments, fasten the module with a screw.

Use screws of type listed in the table below.

Insert these screws into the screw holes on top of the module and tighten them with a Philips screwdriver. A clearance of approximately 80mm between the module and the duct above is necessary to allow the screwdriver to access the screw.

Required Screw
Binding head machine screw M4 of 12 to 15 mm long (washer screw of 14-15 mm long)

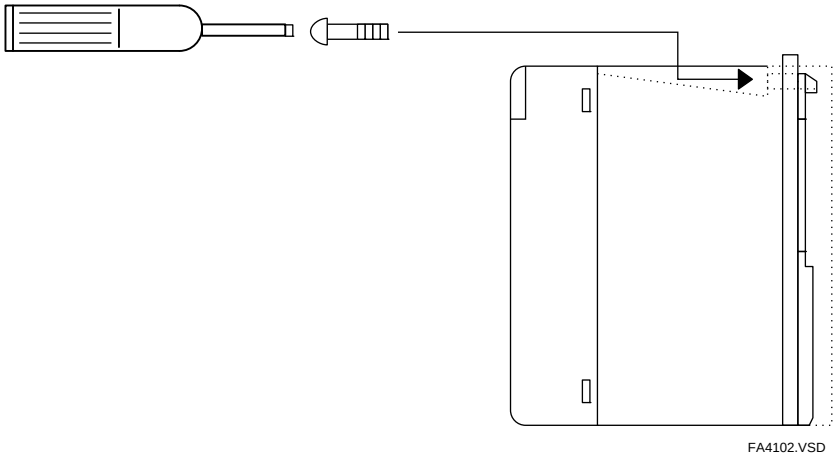


Figure 1.12 Securing Module Using Screws

CAUTION

Do not over-tighten the module fixing screw.

1.5 Glossary

■ Data Field

A data field (DF) is a location where data consisting of information with specific characteristics flows. A data field corresponds to a LAN segment to which equipment belongs. A data field number (DFN) uniquely identifies a data field in a distributed system and is set to a number from 1 through 255.

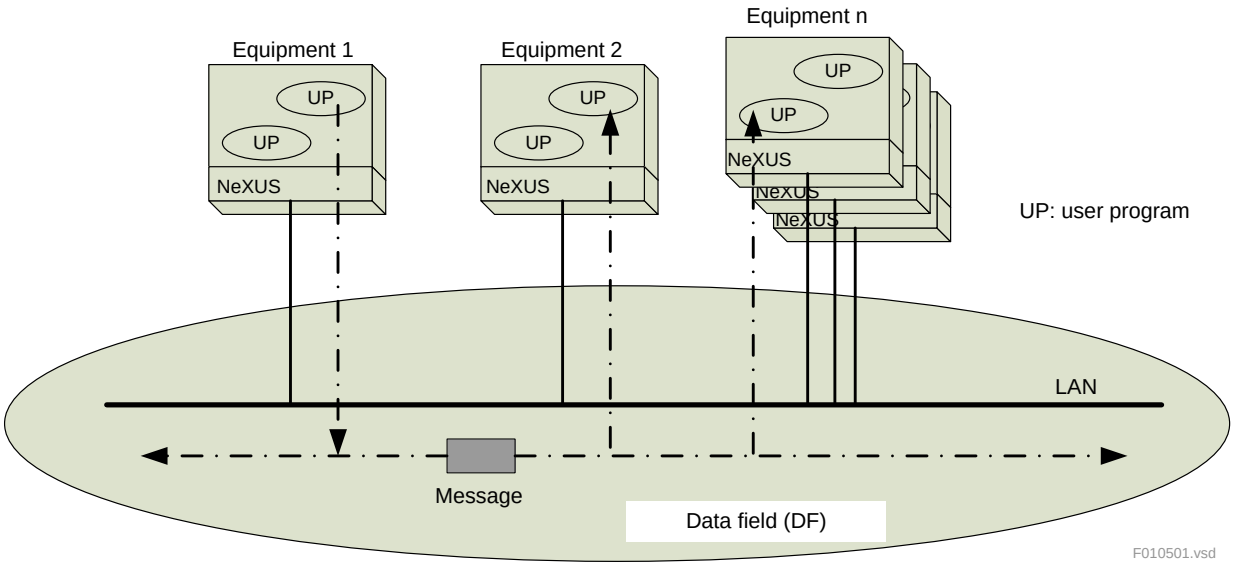


Figure 1.13 Data Field

■ Message

A message is a basic unit of data transmitted between NeXUS and consists of user data and a NeXUS header.

The maximum user data length can be set to 256, 512 or 1024 bytes on the NX Interface Module. Different data lengths can be selected for transmission and receiving.

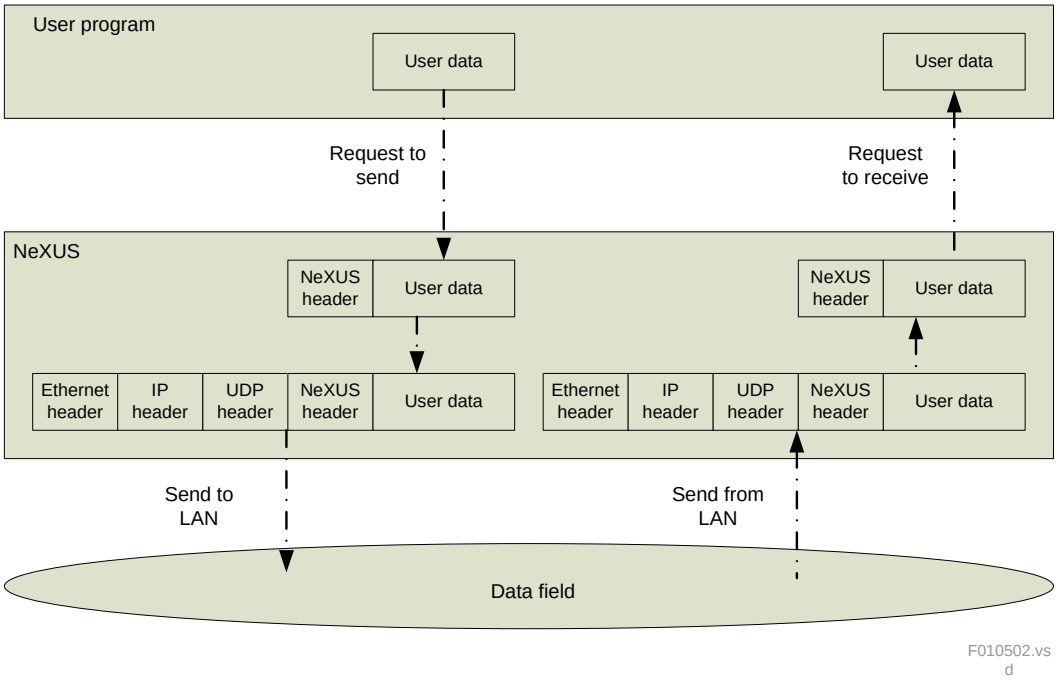


Figure 1.14 Message Structure

CAUTION

The user sets the maximum user data length during initialization. All external devices connected to a data field must be configured with the same length for each transaction code (to be described later).

■ Local/Remote Data Field

Data fields are classified as local data fields and remote data fields.

With the NX Interface Module, one local data field and one remote data field can be defined.

● Local data field

The data field to which the node is directly connected.

● Remote data field

Data fields which are indirectly connected to the module node via routers and gateways.

Example:

When viewed from node 1, which is connected to data field 1 (DFN1), data field 1 is a local data field while data fields 2 and 3 (DFN2 and DFN3) are remote data fields.

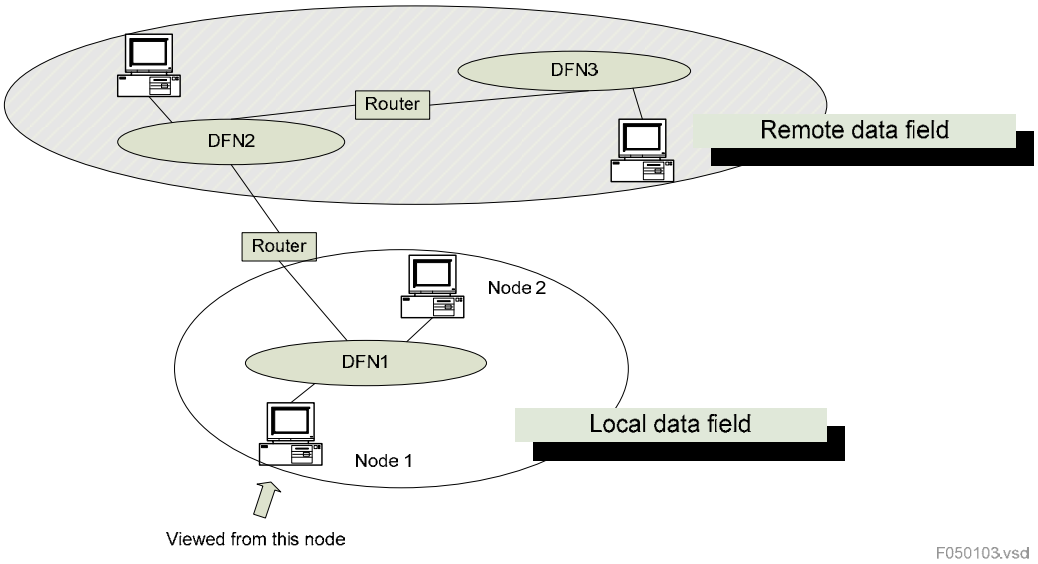


Figure 1.15 Local/Remote Data Fields

■ Logical node

A logical node is a piece of equipment belonging to a data field. A logical node number (LNN), (hereafter known as “node number”, in short), uniquely identifies a logical node in the data field.

Node numbers can be set within the range of 1 through 4095.

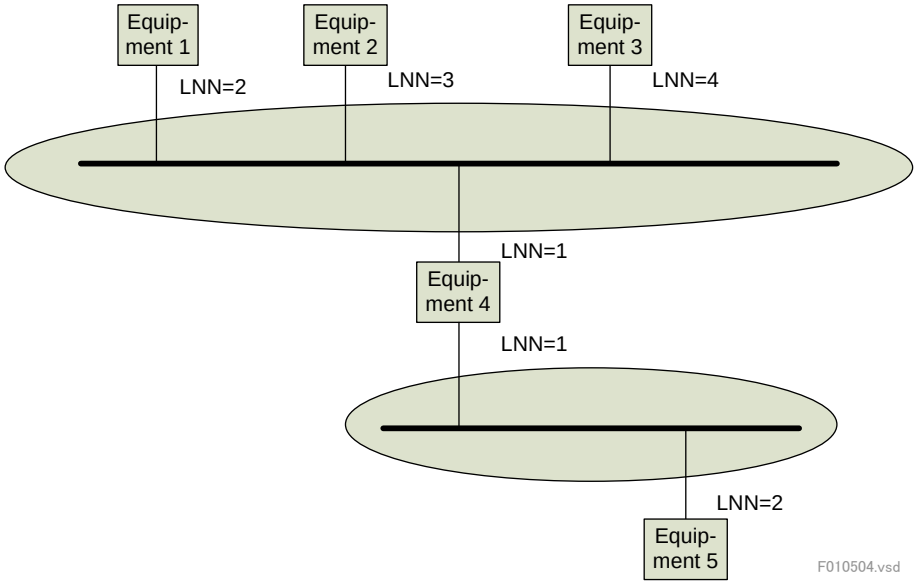


Figure 1.16 Logical Nodes

■ Function Module

A CPU module is called a function module (FM). A function module is the smallest unit in NeXUS control; definition relating to transmission and receiving of messages are performed for each function module. Functions such as alive/dead monitoring and fault monitoring can be used to manage function modules.

The figure below shows the relationship between a logical node and function modules and compares it to NeXUS/Dlink*.

A logical node consists of an NX Interface Module belonging to a data field and the sequence CPU modules connected to the module. The CPU modules (function modules) to be connected to an NX Interface Module are defined at initialization.

* NeXUS/Dlink is a NeXUS-compliant package developed by Hitachi Ltd...

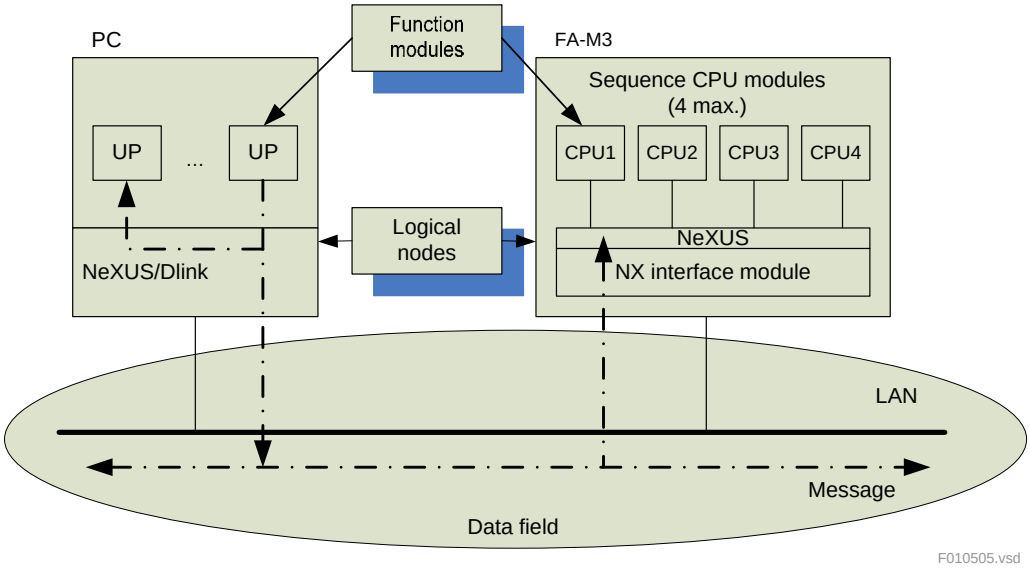


Figure 1.17 Function Module

CPU Modules and NX Interface Modules can be combined in different ways as shown in the figure below. The mapping between the CPU modules and NX Interface Modules are defined at initialization.

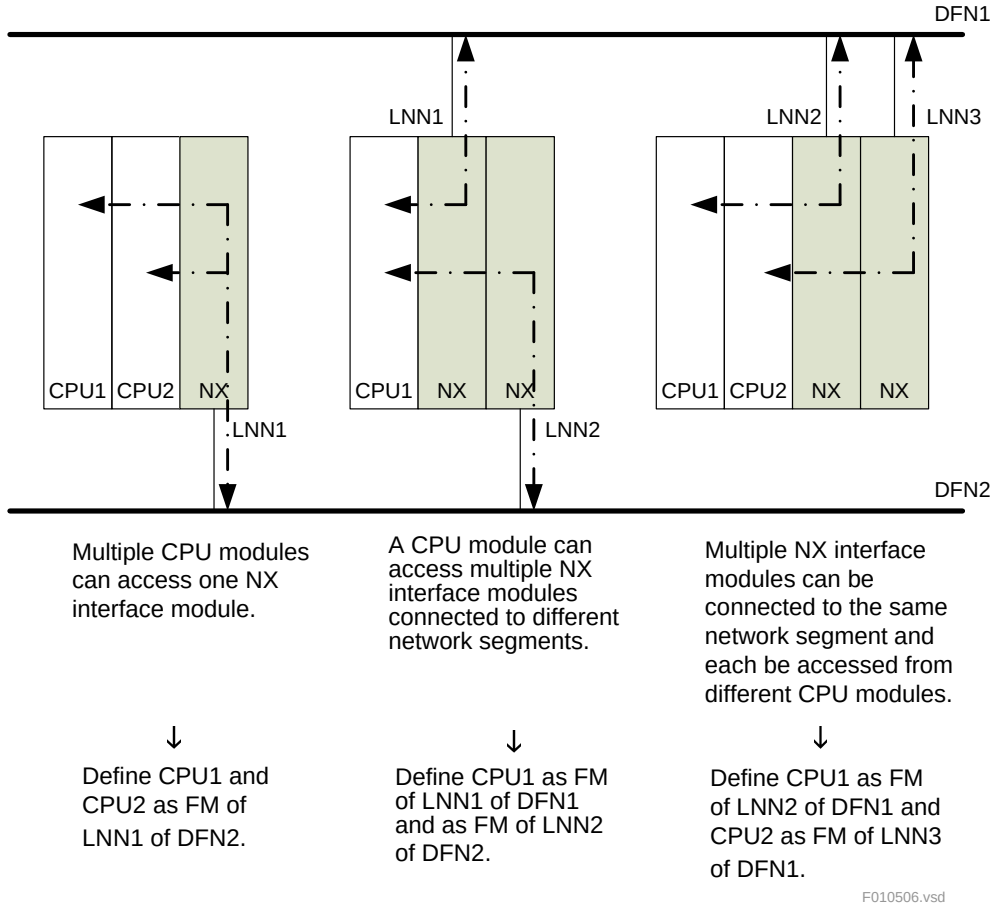


Figure 1.18 Mapping between CPU Modules and NX Interface Modules

■ Intra-Node and Inter-Node Communications

Intra-node communication refers to data communication between a transmitter and a receiver residing within the same node.

Inter-node communication refers to data communication between a transmitter and a receiver residing in different nodes.

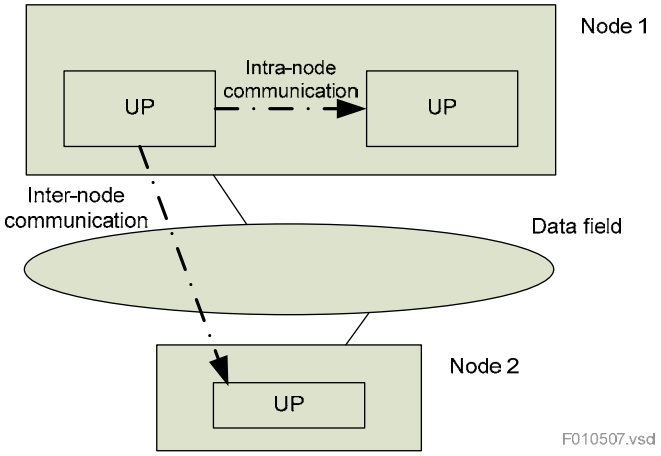


Figure 1.19 Intra-Node Communication and Inter-Node Communication

The NX Interface Module supports only inter-node message exchange. For intra-node communication, use memory shared by CPUs (shared relays, shared registers, extended shared relays and extended shared registers).

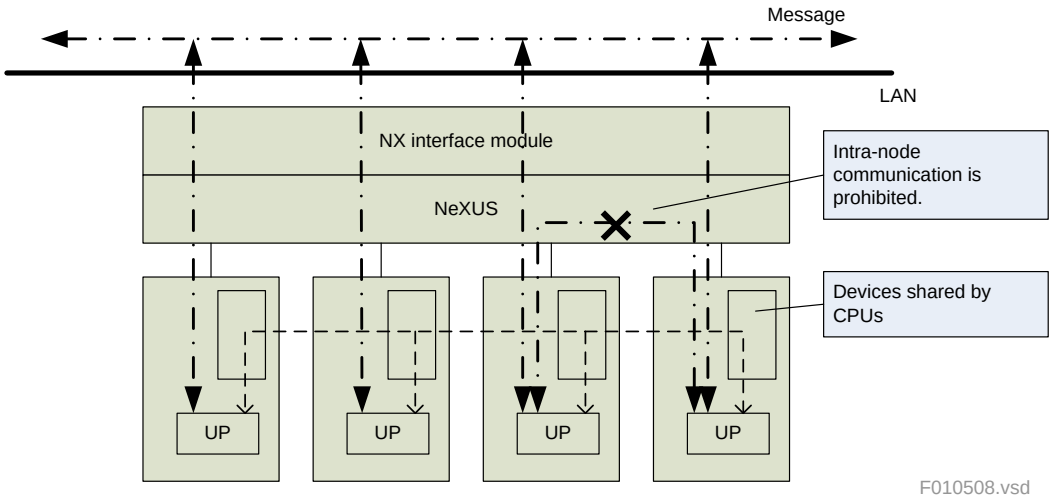


Figure 1.20 Intra-Node Communication

CAUTION

When using a mult-CPU system, note the following points:

The input/output relays of an NX Interface Module can be accessed from only one CPU module; the “Use/Not used” setting in the CPU configuration must be set accordingly. To access the input/output relays from multiple CPUs, use shared relays.

Other data areas (e.g. transmission buffers, receiving buffers and initialization data area) can be accessed from multiple CPUs.

■ Multicast Group

Multicast groups consist of nodes grouped according to whether they should receive data broadcasted from a specific logical node to the data field.

Each data field manages its own multicast groups and multiple multicast groups may be present in one data field. When a task is to be performed cooperatively and autonomously by a group of logical nodes belonging to the data field, you can define a multicast group containing these nodes and assign the task to the multicast group. A logical node belonging to a data field can belong to more than one multicast groups.

A multicast group number (MGN) identifies a multicast group in a data field. With the NX Interface Module, the MGN can be set to any number from 1 through 255 and up to 4 multicast groups (separately for transmitting and receiving) can be defined for each data field.

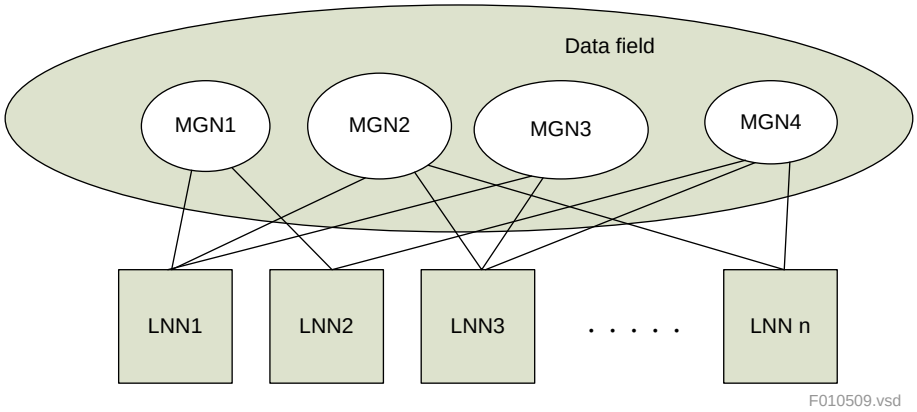


Figure 1.21 Multicast Group Numbers

■ Transaction Code

A transaction refers to the processing associated with a product, service or real-time event. Purchasing something at a store or receiving services at a travel agency can be described as a transaction. Transaction processing involves execution, monitoring and recording of transactions. Database updating and reporting to other divisions performed when an application division receives plant data or share-dealing data are all part of transaction processing.

A transaction code (TCD) is an identifier for processing messages flowing within a data field as transactions.

User programs (UP) specify the TCD and transmit/receive data.

TCDs are defined for each data field.

A user can define any TCD from 1 through 59999.

With the NX Interface module, each node can define up to 16 TCDs for transmission and up to 12 TCDs for receiving.

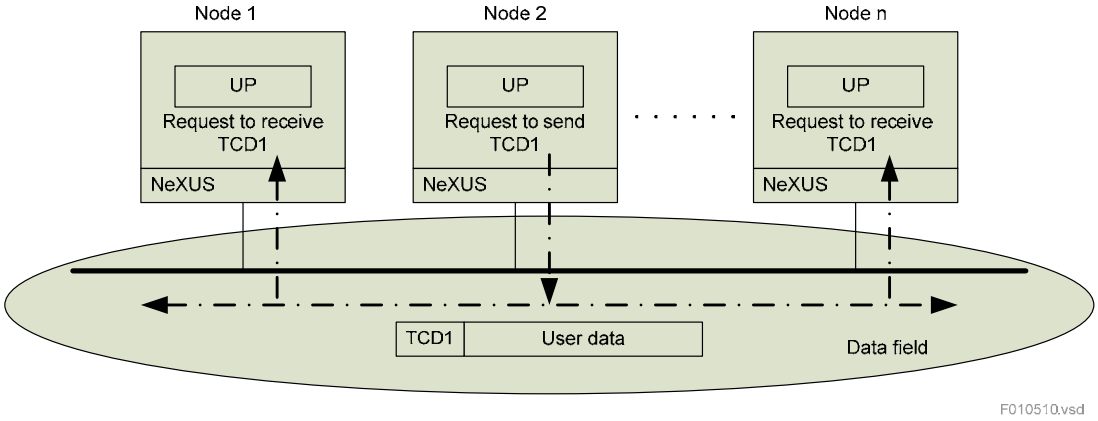


Figure 1.22 Transmission and Receiving using Transaction Codes

■ Node Mode

The node mode refers to the operation mode of a logical node. Nodes in online operation are called online nodes and nodes undergoing testing are called test nodes. By setting test nodes to test mode, testing can be performed without disrupting the online system.

With the test support function, messages transmitted by online nodes are called online mode messages and messages transmitted by test nodes are called test mode messages. The test support function of the NeXUS protocol includes the following features.

- Online nodes receive only online mode messages, allowing testing within an online environment without disrupting the online system.
- Test nodes can also receive online mode messages, allowing testing with online data.
- When testing completes, a test node can be switched to an online node simply by setting the node mode from TEST to ONLINE, without changing network configuration data or user programs.

With the NX Interface Module, the message receiving mode for a test node can be set to any of the following values:

- Receives only online mode messages
- Receives only test mode messages
- Receives both online and test mode messages

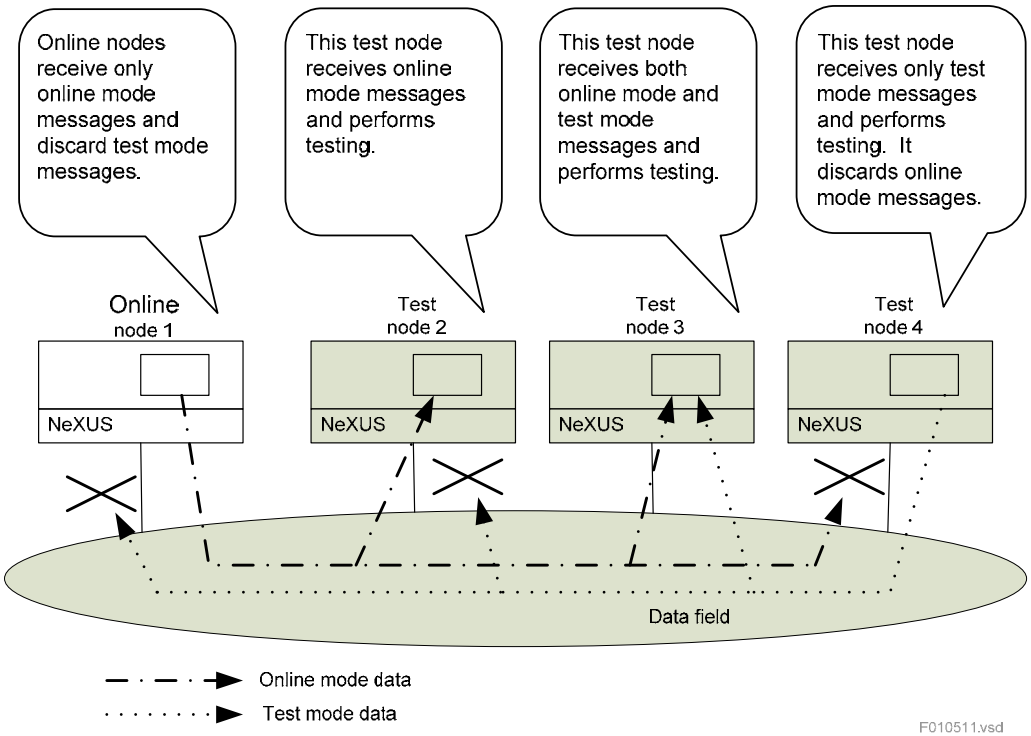


Figure 1.23 Node Modes

■ Communication Handle

By separating user programs and communication attributes, communication attributes for message transmission and receiving may be changed without affecting user programs. This separation of user programs and communication attributes is achieved through communication handles.

Communication attributes are communication control information required for processing transmission and receiving of messages.

For example, communication attributes for transmission include control information such as the destination data field number, the destination multicast group number and the transaction code.

Communication handle numbers are serial numbers assigned to identify a set of communication attributes. During message transmission and receiving, the NX Interface Module designates the input/output relays and logical addresses (channel numbers) of transmitting/receiving buffers, and thus synchronizes communication and passes data between NeXUS and the user. The input/output relay and buffer address map to one communication handle number.

In short, registering communication attributes to a communication handle number allows the logical address to be accessed for the handle number to be determined automatically.

Up to 16 communication handle numbers can be registered for transmission and up to 12 can be registered for receiving.

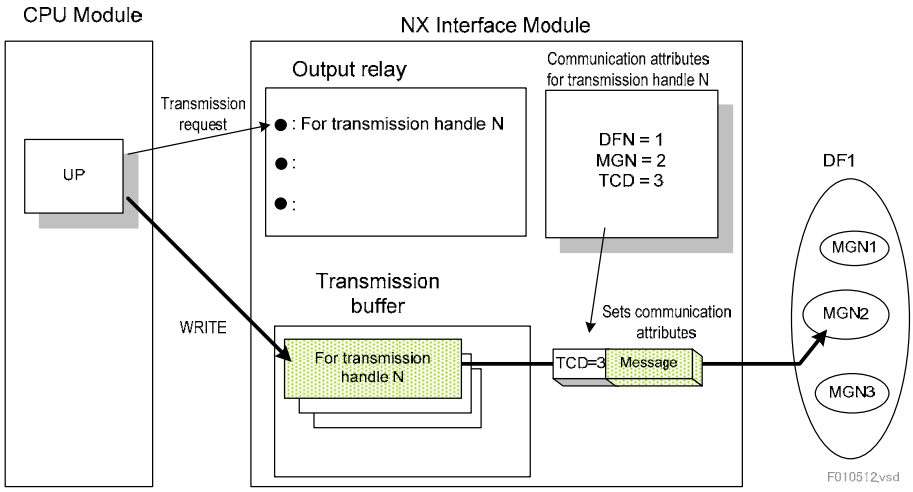


Figure 1.24 Communication Handle for Transmission

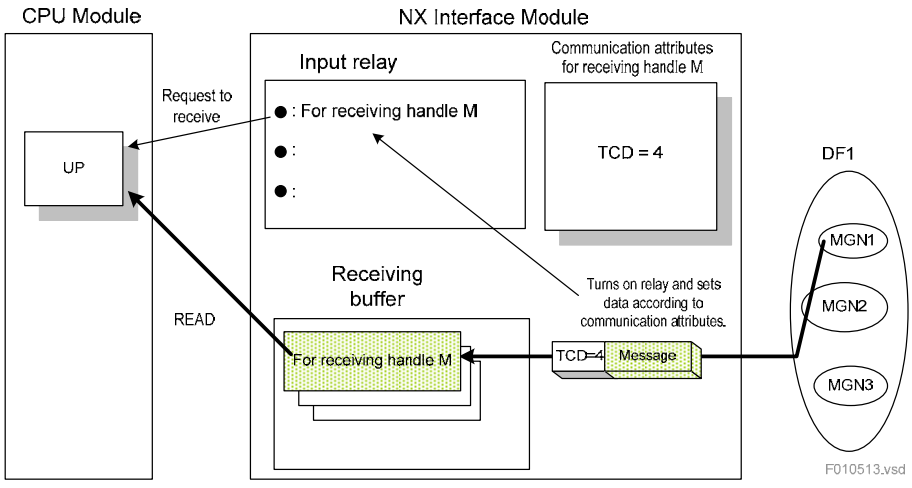


Figure 1.25 Communication Handle for Receiving

2. Multicast Communication

2.1 Overview of Communication Function

The communication function transmits and receives messages with TCD from the CPU module to other nodes via the NT Interface Module using multicast communication.

Multicast communication is based on a cooperative autonomous distributed system and has the following features:

- The same message can be sent to multiple nodes concurrently to achieve efficient transmission.
- It is easy to use with no node dependency relationships in message transmission and receiving and no need to first establish a connection.

2.1.1 Multicast Communication

The transmitting node specifies the destination data field, multicast group and transaction code and transmits the message. On the receiving end, each node retrieves only the required messages autonomously.

Since the transmitting node need not identify or synchronize with the actual communication counterpart, this increases independence of individual equipment and user program and also improves system extensibility.

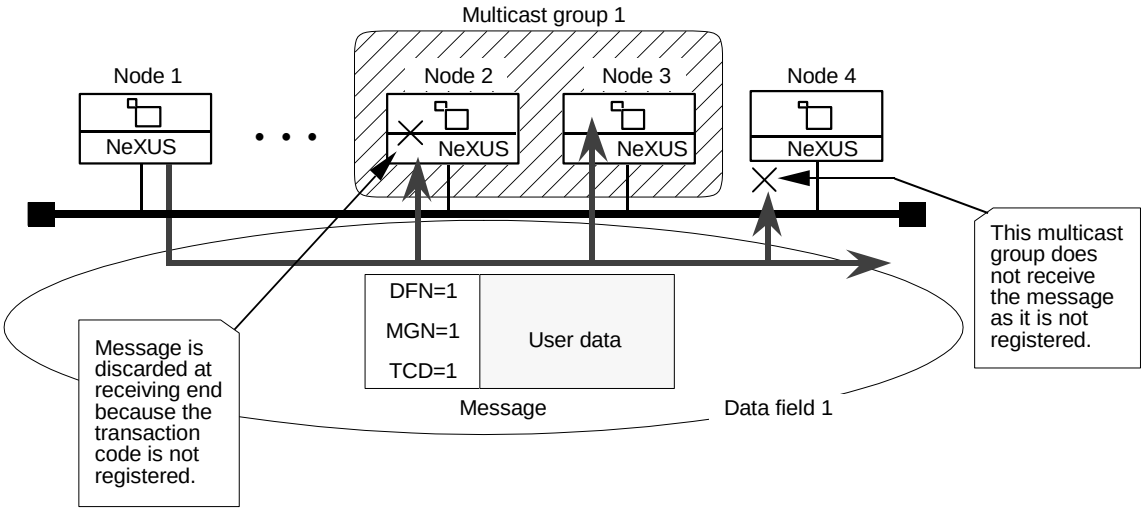


Figure 2.1 Multicast Communication

■ Overview of Communication Specifications

Table 2.1 shows the main communication specifications.

Table 2.1 Communication Specifications

Item	Specifications
Network specifications	Ethernet
Communication protocol	UDP/IP
Message size	1088 bytes max. (including 64 bytes of NeXUS header)
Buffer case size	Selectable for each transmission or receiving: 256, 512, or 1024 bytes
Buffer area size	14 kilobytes max. per node
Number of registered local data fields	1 DF max.
Number of registered remote data fields	1 DF max.
Number of transmission multicast group registrations	4 groups max.
Number of receiving multicast group registrations	4 groups max.
Number of output transaction codes	16 TCDs max.
Number of input transaction codes	12 TCDs max.
Test node setup	Yes

2.1.2 Transaction Management

In NeXUS, data are handled as messages with each message consisting of user data appended with a NeXUS protocol header. The NeXUS protocol header contains an important identifier called the transaction code (TCD) which a user specifies when performing data transmission and receiving.

Transaction management of the NX Interface Module maps one set of relays and one buffer to one TCD.

The relays are switches used by the user to communicate with the NX Interface Module to request for transmission or notify completion of receiving.

The buffers are for transferring to and from memory and are managed separately for transmission and receiving.

When a user writes data to the transmission buffer and turns on the Request to Transmit relay, a message attached with the TCD is transmitted to the network. When a message with a TCD arrives, the data is stored in the receiving buffer, the Request to Receive relay turns on and the user can receive the data from the receiving buffer.

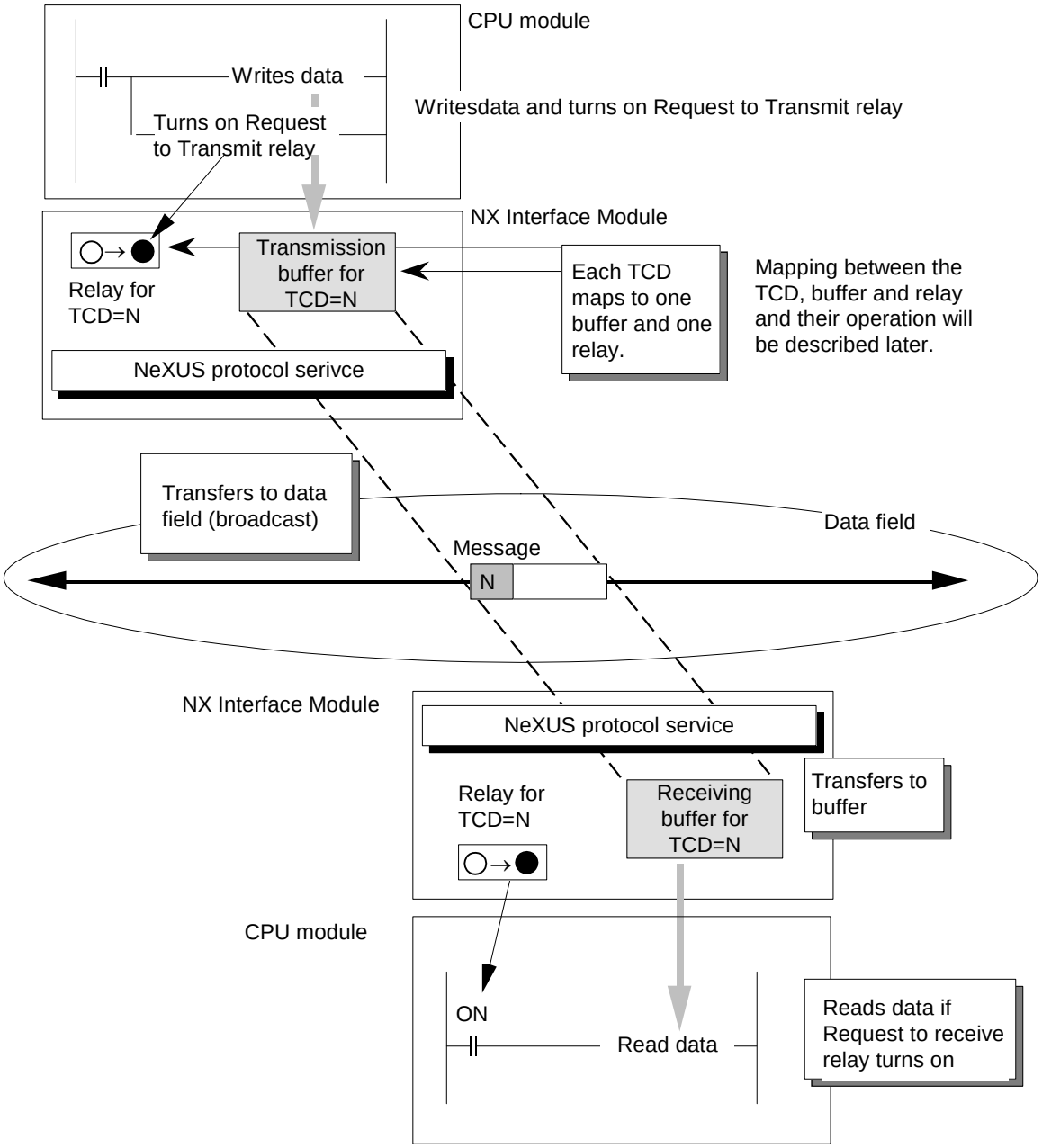


Figure 2.2 Transaction Management

2.2 Input/Output Relays and Transmission/Receiving Buffer

Requests from a CPU module for transmission and receiving are achieved with the following operations, referencing the slot number where the NX Interface Module is attached.

- Turning on/off the input/output relays.
- Reading data from or writing data to the transmission or receiving buffer (channel number) using READ/WRITE instructions.

One communication handle maps to one set of input/output relays and one transmission or receiving buffer, and vice versa.

A communication handle is a concept for managing the communication attributes (TCD, MGN, DFN, etc.) in message transmission and receiving; a communication handle number is a serial number added to a message to identify the communication handle.

Defining the communication attributes of a communication handle specifies where the input/output relay and transmission/receiving buffer required for operation are located.

Example:

When communication handle number 1 is to be defined with communication attributes for transmitting messages with transaction code 3 to multicast group 2 in data field 1:

(1) The user defines the above communication attributes in communication handle 1

(2) The user writes data to transmission buffer for communication handle 1.

(3) The user turns on output relay number for communication handle 1.

(4) NeXUS recognizes communication handle 1 from the output relay number that was turned on, adds the communication attributes (4)-b defined in communication handle 1 to the data in the transmission buffer (4)-a for communication handle 1 and sends the message.

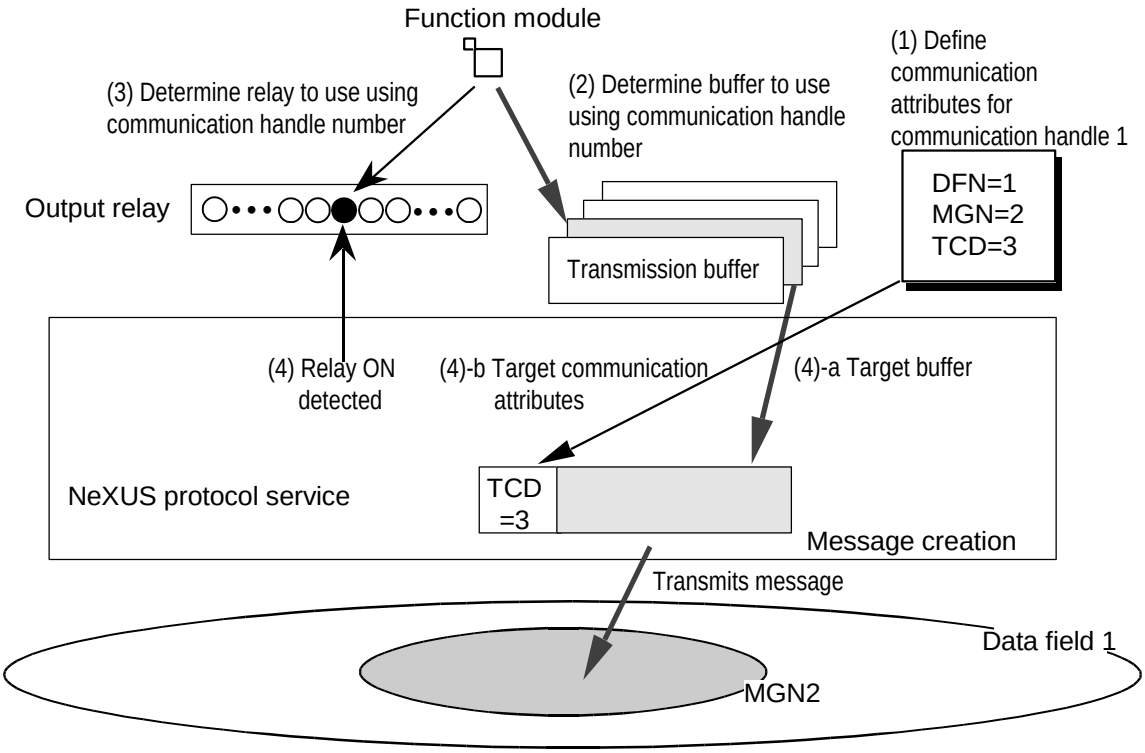


Figure 2.3 Input/Output Relays and Transmission/Receiving Buffer

CAUTION

Take note that if an input/output relay number or buffer address not corresponding to a handle number is incorrectly accessed, irregular data may be transmitted or data may be transmitted with different communication attributes.

2.2.1 Input/Output Relays

There are 32 input relays and 32 output relays.

Input relays are turned on or off by NeXUS and are read-only to the user.

Output relays are turned on or off by the user and are read-only to NeXUS.

Input/output relays are used to synchronize data passing between the user and NeXUS.

The turning on, turning off and accessing of input/output relays, as well as the sequence of such operations are different for transmission and receiving.

For more details on the access methods, see Section 2.3, “Transmission” and Section 2.4 “Receiving”.

Figure 2.4 shows the complete structure diagram for the input and output relays.

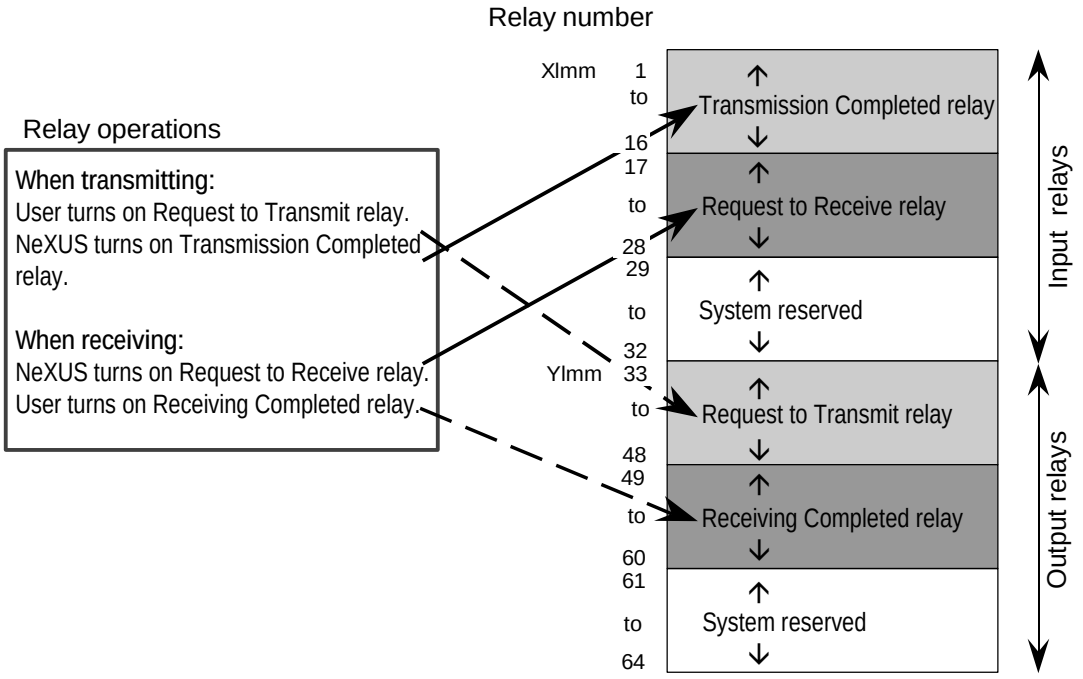


Figure 2.4 Complete Structure Diagram for Input and Output Relays

● Mapping to Communication Handles

As shown in the overall structure diagram of the input and output relays, Request to Transmit relay and Transmission Completed relay are used for transmission. Similarly, Request to Receive relay and Receiving Completed relay are used for receiving. These relays work in pairs and map to the communication handle numbers as follows:

- Transmission handle number 1 maps to relay numbers 1 and 33
- Transmission handle number 16 maps to relay numbers 16 and 48
- Receiving handle number 1 maps to relay numbers 17 and 49
- Receiving handle number 12 maps to relay numbers 28 and 60

To send a message using the communication attributes defined in a transmission handle, you must operate the relays for that handle.

TIP

Up to 16 transmission handles and up to 12 receiving handles can be defined in a NX Interface Module.

However, take note that the maximum number of handles actually allowed depends on the buffer case size described later.

Note also that the input/output relays used for transmission and receiving automatically turn off when the node mode is changed online (to be described later).

CAUTION

When using a multi-CPU system, note the following points:

The input/output relays of an NX Interface Module can be accessed from only one CPU module; the “Use/Not used” setting in the CPU configuration must be set accordingly. To access the input/output relays from multiple CPUs, use shared relays.

Other data areas (e.g. transmission and receiving buffers and initialization data area) can be accessed from multiple CPUs.

2.2.2 Transmission/Receiving Buffer

The logical addresses of transmission or receiving buffers are identified by channel numbers.

Channel numbers are sequential numbers allocated in 2-byte units within a 14 kilobyte transmission/receiving buffer area. Therefore, channel numbers run from 1 through 7168.

The user performs data reading and writing by specifying the channel number for the starting position of each case if cases are uniformly allocated by the case sizes specified for transmission and receiving during initialization.

Figure 2.5 shows the structure of the buffer area and its relation to the channel numbers. The relation between buffer case size and buffer area size is as follows:

- N = buffer case size for transmission × n (n: 16 max.)
- M = buffer case size for receiving × m (m: 12 max.)
- N + M ≤ 14 kilobytes

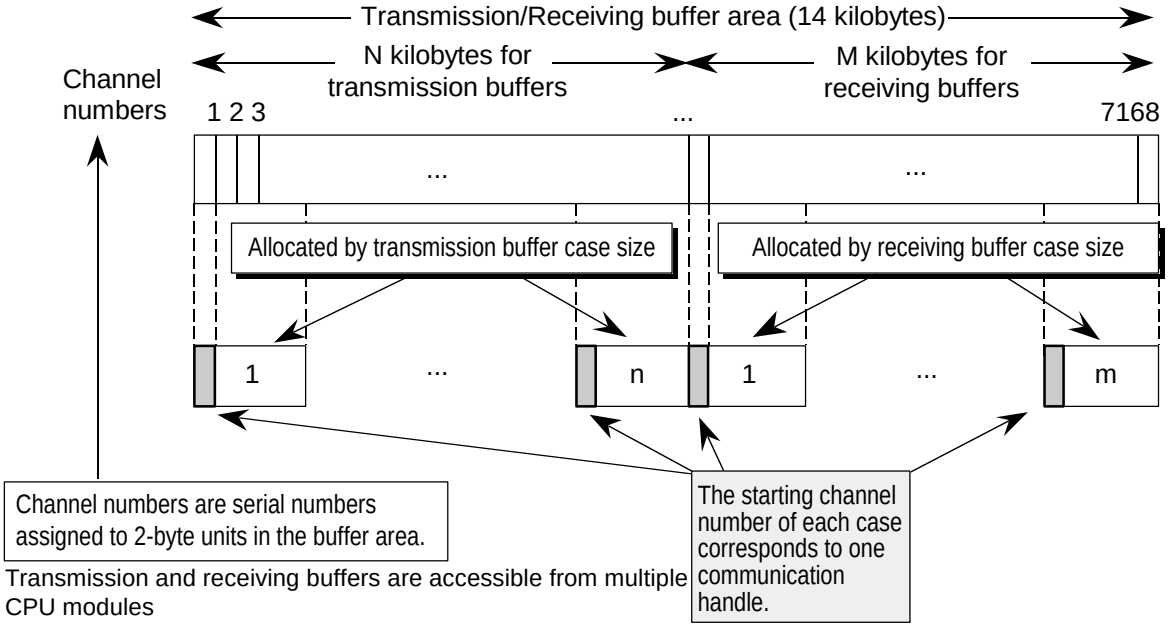


Figure 2.5 Structure of Buffers for Transmitting and Receiving and Channel Numbers

● Mapping to Communication Handles

The channel number of the starting position of each case maps to one communication handle number and hence, one set of input/output relays.

The number of registered communication handles allowed is limited by the transmission and receiving buffer case sizes specified.

Buffer case sizes can be set to 256, 512 or 1024 bytes.

2.2.3 Mapping between Communication Handle Numbers and Channel Numbers

The mapping between communication handle numbers and channel numbers is given in the following formulae.

Channel number for transmission handle N

= ((transmission buffer case size / 2) × (N – 1)) + 1

Channel number for receiving handle M

= ((receiving buffer case size / 2) × (M – 1)) + first channel number for receiving

First channel number for receiving

= ((transmission buffer case size / 2) × number of transmission handles) + 1

The mapping between communication handle numbers and channel numbers is shown in the table below.

The mapping between handle numbers and relay numbers is fixed but the mapping between handle numbers and channel numbers changes according to the buffer case size.

● For Transmission

Request to Transmit relays start from 33; Transmission Completed relays start from 1; channel numbers start from 1.

Table 2.2 Mapping between Communication Handle Numbers and Channel Numbers (for Transmission)

Channel Number	Request to Transmit Relay Number	Transmission Completed Relay Number	Channel Number*1		
			(256)	(512)	(1024)
1	Ylmm33	Xlmm01	1	1	1
2	Ylmm34	Xlmm02	129	257	513
3	Ylmm35	Xlmm03	257	513	1025
4	Ylmm36	Xlmm04	385	769	1537
5	Ylmm37	Xlmm05	513	1025	2049
6	Ylmm38	Xlmm06	641	1281	2561
7	Ylmm39	Xlmm07	769	1537	3073
8	Ylmm40	Xlmm08	897	1793	3585
9	Ylmm41	Xlmm09	1025	2049	4097
10	Ylmm42	Xlmm10	1153	2305	4609
11	Ylmm43	Xlmm11	1281	2561	5121
12	Ylmm44	Xlmm12	1409	2817	5632
13	Ylmm45	Xlmm13	1537	3073	6145
14	Ylmm46	Xlmm14	1665	3329	6657
15	Ylmm47	Xlmm15	1793	3585	*2
16	Ylmm48	Xlmm16	1921	3841	*2

l : unit number
mm : slot number

*1 The numbers enclosed within parenthesis represent the buffer case sizes.
*2 These channel numbers cannot be defined because they exceed the 14 kilobyte buffer area.

● For Receiving

Request to Receive relays start from 17; Receive Completed relays start from 49.

Channel numbers for receiving follow the last channel number for transmission such that:

First channel number for receiving

= ((transmission buffer case size / 2) × number of transmission handles) + 1

When the buffer area exceeds 14 kilobytes, no more definition is allowed.

Table 2.3 Mapping between Communication Handle Numbers and Channel Numbers (for Receiving)

Channel Number	Request to Receive Relay Number	Receive Completed Relay Number	Channel Number*1		
			(256)	(512)	(1024)
1	Ylmm49	Xlmm17	n *2	n *2	n *2
2	Ylmm50	Xlmm18	n + 128	n + 256	n + 512
3	Ylmm51	Xlmm19	n + 256	n + 512	n + 1024
4	Ylmm52	Xlmm20	n + 384	n + 768	n + 1536
5	Ylmm53	Xlmm21	n + 512	n + 1024	n + 2048
6	Ylmm54	Xlmm22	n + 640	n + 1280	n + 2560
7	Ylmm55	Xlmm23	n + 768	n + 1536	n + 3072
8	Ylmm56	Xlmm24	n + 896	n + 1792	n + 3584
9	Ylmm57	Xlmm25	n + 1024	n + 2048	n + 4096
10	Ylmm58	Xlmm26	n + 1152	n + 2304	n + 4608
11	Ylmm59	Xlmm27	n + 1280	n + 2560	n + 5120
12	Ylmm60	Xlmm28	n + 1408	n + 2816	n + 5632

l : unit number
mm : slot number

*1 The numbers enclosed within parenthesis represent the buffer case sizes.
*2 First channel number n = ((transmission buffer case size / 2) × number of transmission buffers) + 1

2.3 Transmission

The transmission function transmits messages attached with TCD from the CPU module to other nodes via the NX Interface Module using multicast communication.

2.3.1 Defining a Transmission Handle

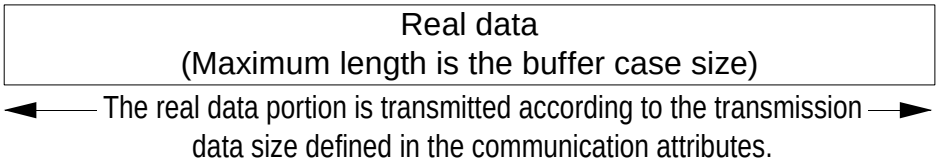
The user must define a communication handle to transmit a message with TCD.

To define a communication handle is to define the communication attributes for the handle number.

- Handle number: This is the transmission control information number and can be set to any number from 1 through 16. It determines the Request to Transmit relay number and the Transmission Completed relay number.
- Communication attributes: defines the following control information for a handle number:
  - Transmission destination data field number (DFN)
  - Multicast group number (MGN)
  - Transmission transaction code (TCD)
  - Transmission data size

2.3.2 Data Format

The data format is shown below.



The data is accessible from multiple CPU modules.

2.3.3 Service Sequence

NeXUS processes transmission requests in ascending order of the handle numbers from 1 through 16. Therefore, if transmission requests are issued simultaneously, the TCD messages for handle number 1 will always be transmitted first.

For details on the definition of transmission handles, see Section 4.2, “Initialization”.

CAUTION

When using a multi-CPU system, note the following points:

The input/output relays of an NX Interface Module can be accessed from only one CPU module; the “Use/Not used” setting in the CPU configuration must be set accordingly. To access the input/output relays from multiple CPUs, use shared relays.

Other data areas (e.g. transmission and receiving buffers and initialization data area) can be accessed from multiple CPUs.

2.3.4 Transmission Procedure

The procedure for transmission from a ladder program is shown below.

The following example assumes transmission is for handle number 1 with buffer size of 256 bytes. Therefore, the Request to Transmit relay number is 33, the Transmission Completed relay number is 1 and the first channel number for the transmission buffer is 1.

(1) The user checks that the Request to Transmit relay and Transmission Completed relay is off, writes the transmission data to the transmission buffer and turns on the Request to Transmit relay. NeXUS attaches the communication attributes corresponding to the designated relay, performs transmission and turns on the Transmission Completed relay when transmission completes.

(2) The user checks that the Transmission Completed relay has turned on and turns off the Request to Transmit relay.

For details on transmission from a ladder program, see Section 5.1, “Transmitting Messages”.

Example:

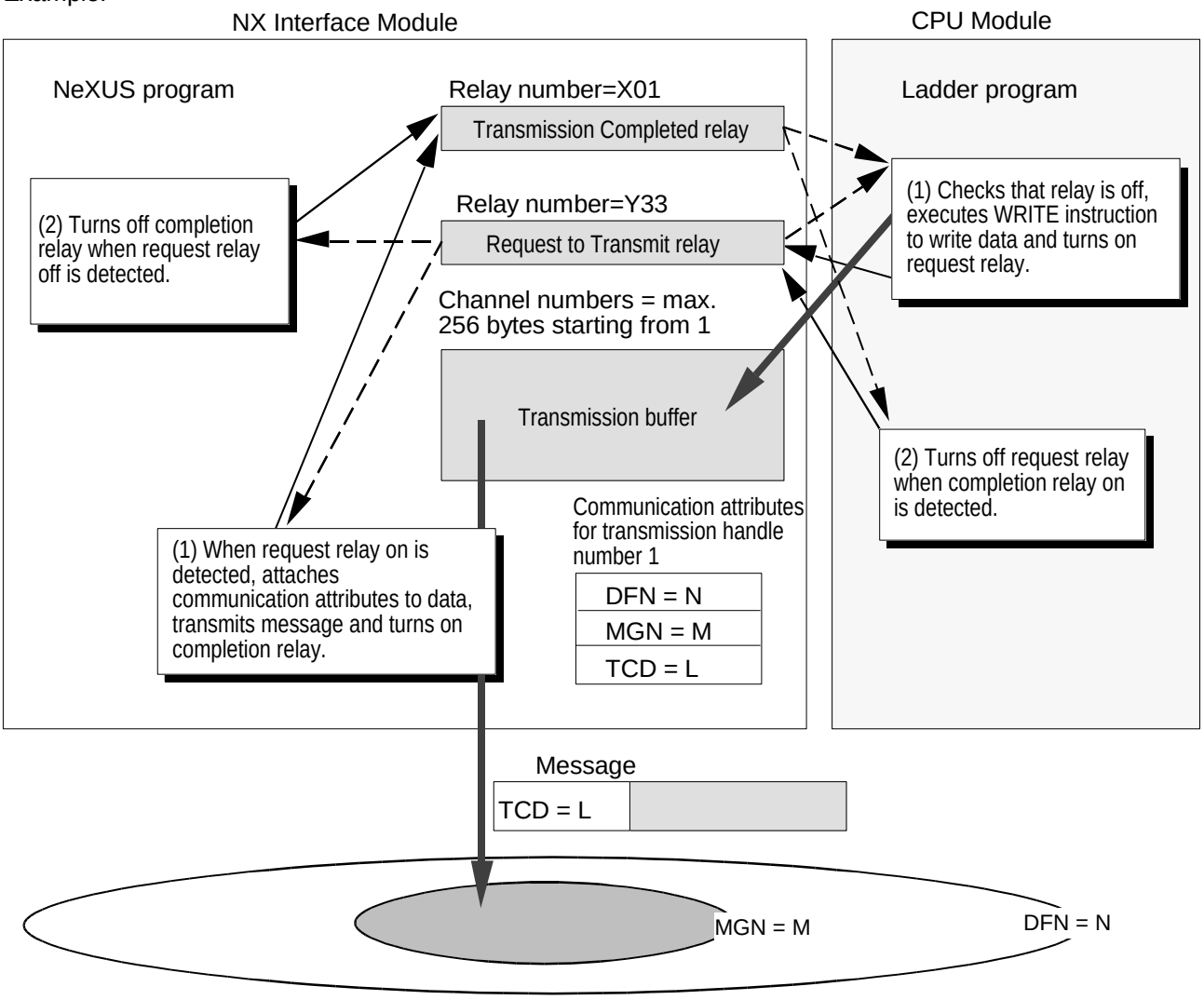


Figure 2.6 Transmission Procedure

2.4 Receiving

The receiving function receives messages attached with TCD sent from other nodes at the CPU module via the NX Interface Module.

2.4.1 Defining a Receiving Handle

NeXUS autonomously retrieves only TCD messages that are required and stores data in the buffer corresponding to the TCD. The user must first define the control information (communication attributes) for this autonomous operation as a communication handle. To define a communication handle is to define the communication attributes for that handle number.

- Handle number: This is the receiving control information number and can be set to any number from 1 through 12. It determines the Request to Receive relay number and the Receiving Completed relay number.
- Communication attributes: Communication attributes: defines the control information for a receiving transaction code (TCD) of a handle number:

For details on defining the communication handle, see Section 4.2, “Initialization”.

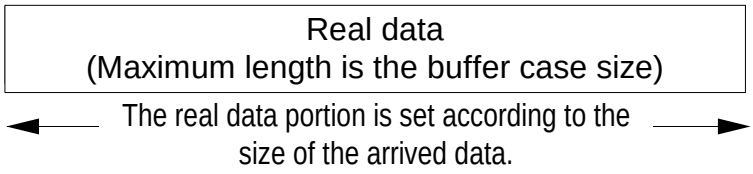
CAUTION

Note that messages that arrive before the user completes receiving processing of an earlier message will be discarded.

If the user did not receive an arrived TCD message and subsequently the same TCD message arrives again, NeXUS will discard the latter message.

2.4.2 Data Format

The data format is shown below.



There is no limit on the number of CPUs receiving one TCD message. Data can be received (READ) by multiple CPU modules.

CAUTION

When using a multi-CPU system, note the following points:

The input/output relays of an NX Interface Module can be accessed from only one CPU module; the “Use/Not used” setting in the CPU configuration must be set accordingly. To access the input/output relays from multiple CPUs, use shared relays.

Other data areas (e.g. transmission and receiving buffers and initialization data area) can be accessed from multiple CPUs.

2.4.3 Receiving Procedure

The procedure for receiving from a ladder program is shown below.

The following example assumes that handle number 1 is defined as a receiving handle with TCD=L and buffer case size=256 (the transmission buffer case size is also 256 bytes and the communication handle is 16). Therefore, the Request to Receive relay number is 17, the Receiving Completed relay number is 49 and the first channel number for the receiving buffer is 2049.

NeXUS receives a message from the network; if the receiving TCD is defined in the communication handle, it stores the data in the receiving buffer for the handle and turns on the Request to Receive relay for the handle.

(1) The user checks that the Request to Receive relay has turned on, reads the valid data portion from the receiving buffer and turns on the Receiving Complete relay.

(2) The user checks that the Request to Receive relay has turned off and turns off the Receiving Complete relay.

For details on receiving from a ladder program, see Section 5.2, “Receiving Messages”.

Example:

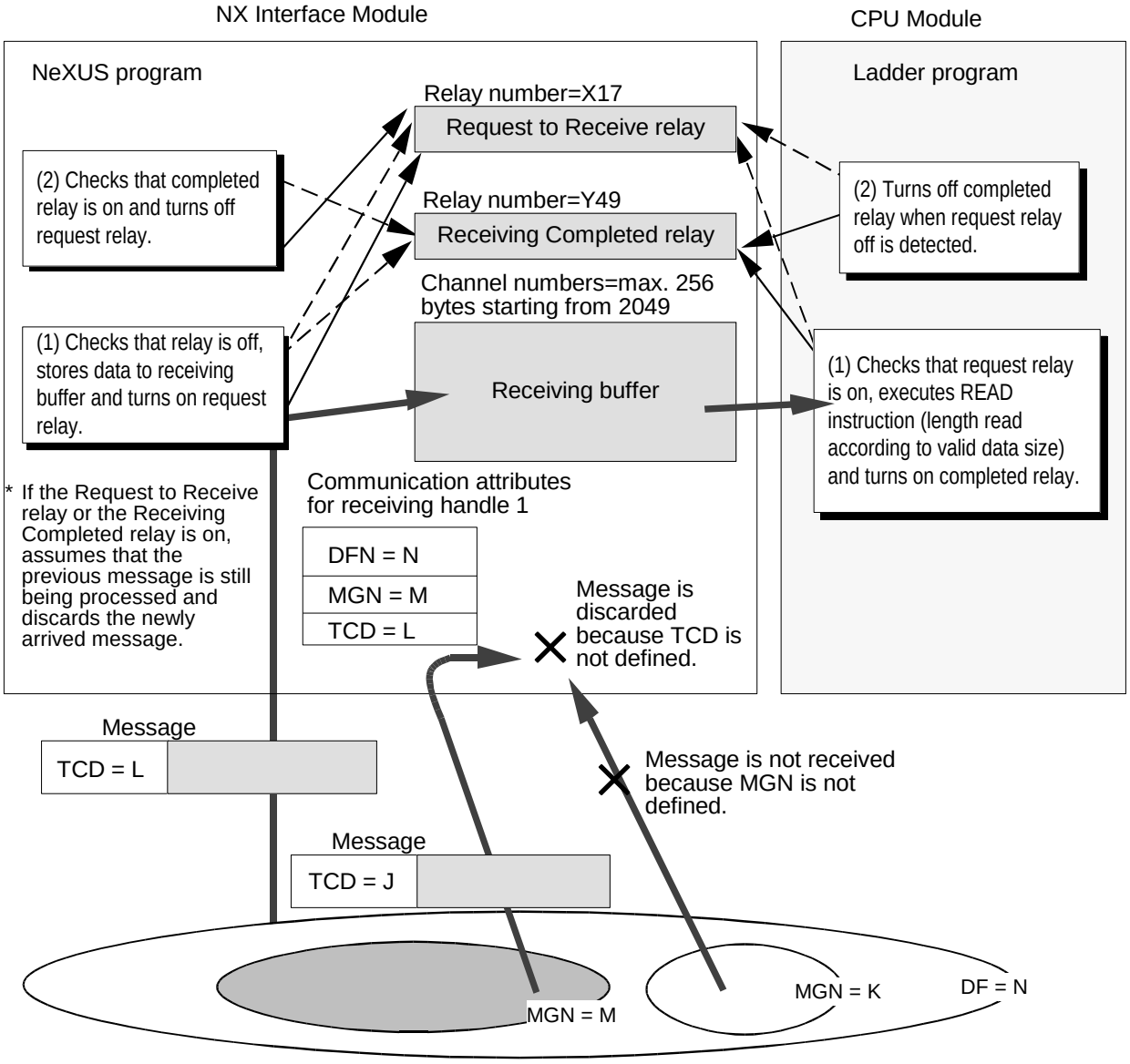


Figure 2.7 Receiving Procedure

2.5 Support for Testing

The NX Interface Module supports various testing modes, which allow easy testing of function modules created by a user.

The NX Interface Module supports 3 types of online simulation test modes:

- Testing with test messages
- Testing with online messages
- Testing with both online and test messages

2.5.1 Mode Setup

Using the NX Interface Module, you can set the mode of a node for implementing the above-mentioned online simulation tests.

There are two types of modes:

- Online mode
- Test mode

The node mode is reflected in the transmitted or received messages with the following relationship.

- Transmission message

Node Mode	Transmission Message Mode
Online	Online
Test	Test

NeXUS stores the node mode unchanged to the mode of the transmission message.

- Received message

Node Mode	Message Modes to Receive
Online	Online only
Test	Test only
	Online only
	Both online and test

Online nodes receive only online mode messages.

Test nodes may be set to selectively receive test mode messages or/and online mode messages as shown on the left. NeXUS will discard messages not to be received.

2.5.2 Port Setup

In NeXUS, port numbers for each multicast group are assigned for each mode. By using different modes, it prevents test messages from affecting online nodes, even within the same multicast group.

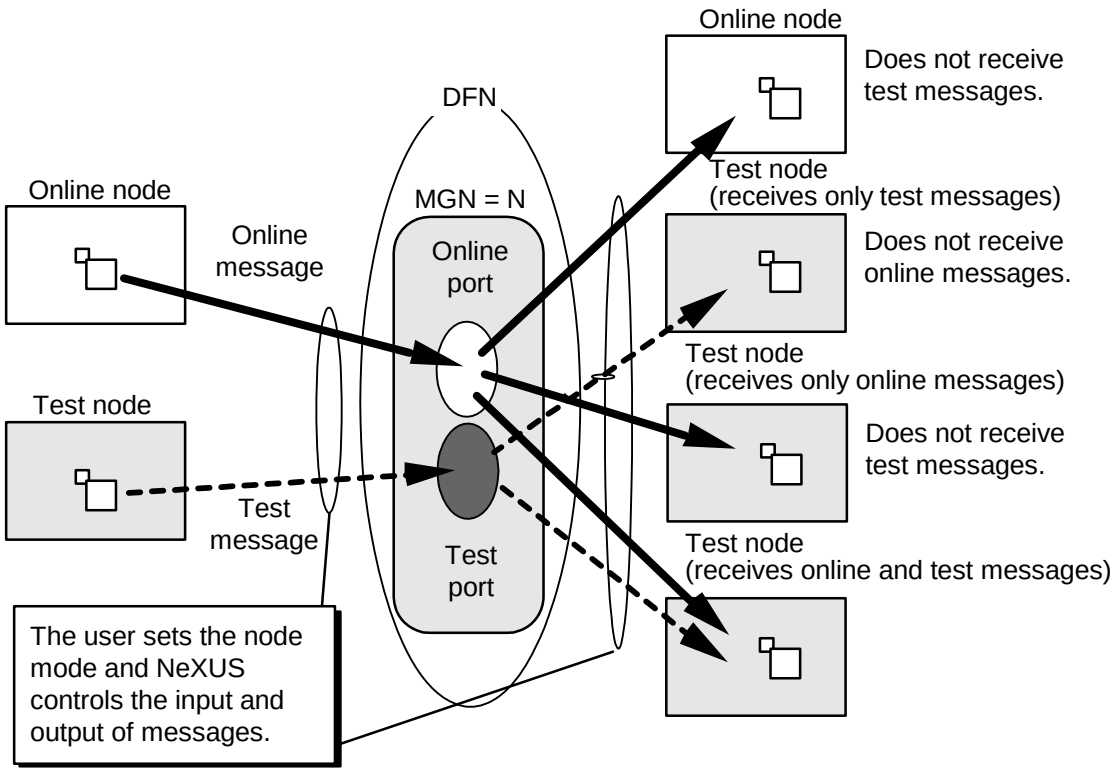


Figure 2.8 Port Setup

2.5.3 Test Mode

Mode setup for the transmitted messages is the same for other NeXUS support equipment and hence, it allows testing within the same network segment as shown below.

Furthermore, when used in combination with the journal function of NeXUS/Dlink, it allows efficient testing.

1. Testing by receiving only online messages

   This test mode is useful for testing whether existing online messages are still handled the same way, say, after a user program on the FA-M3 has been modified. It allows a FA-M3 used for testing to be directly connected to the online network to perform testing without the need to set up a new test environment.

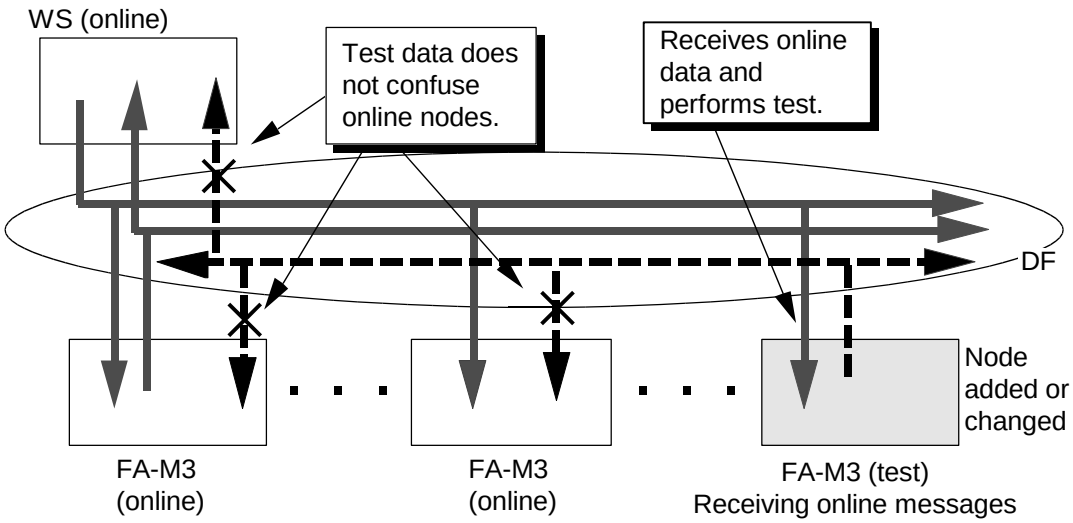


Figure 2.9 Testing by Receiving Online Messages

2. Testing by Receiving Test Messages

   This test mode is useful for testing the receiving of test messages from a WS and returning of a response to the message to the WS, say, after the addition of a new FA-M3 controller and a WS.

   It allows an FA-M3 and WS used for testing to be directly connected to the online network to perform testing without the need to set up a new test environment.

   Furthermore, when used in combination with the journal function of NexUS/Dlink, it allows easy implementation of test message creation and test result comparison.

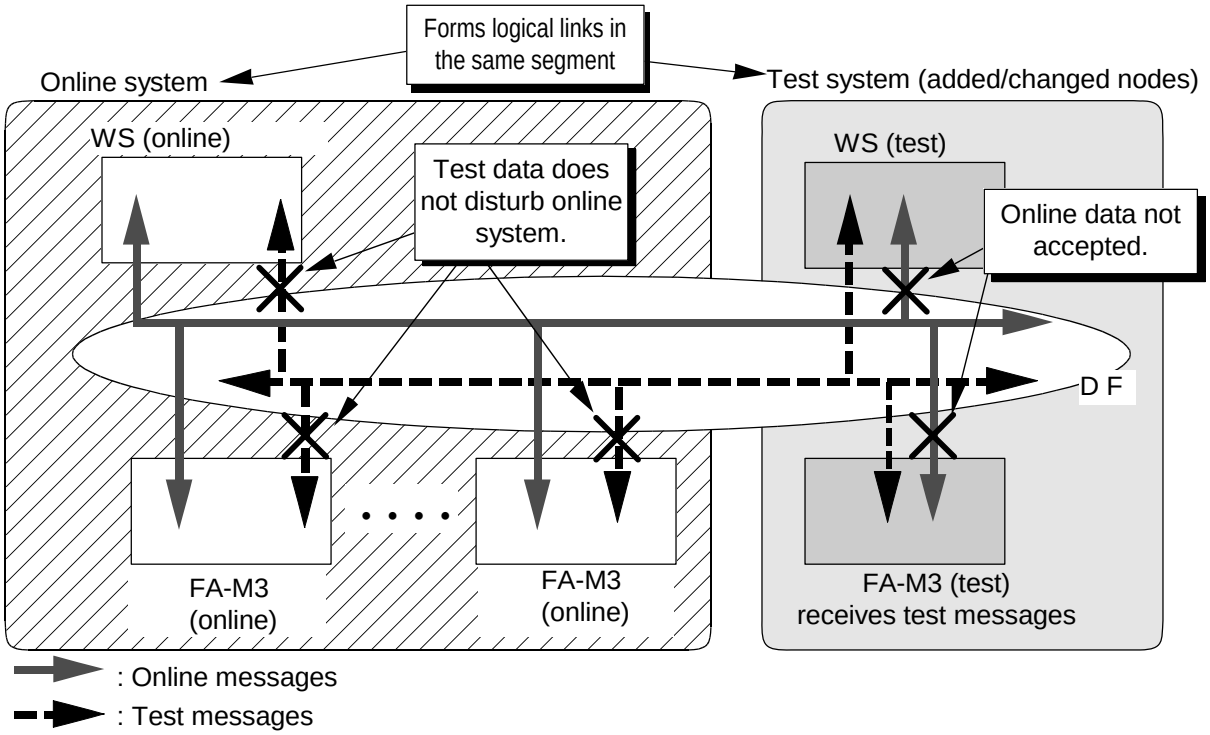


Figure 2.10 Testing by Receiving Test Messages

3. Testing by Receiving Both Test and Online Messages

   This test mode is useful for testing receiving of online messages (e.g. processing instruction data) from an online WS and returning a response to the message (e.g. processing results data) to the test WS, say, after the addition of a new FA-M3 controller and a WS. When used in combination with the journal function of NexUS/Dlink, it allows comparison of the processing result of the FA-M3 for the test system to that of the FA-M3 for the online system to be easily implemented.

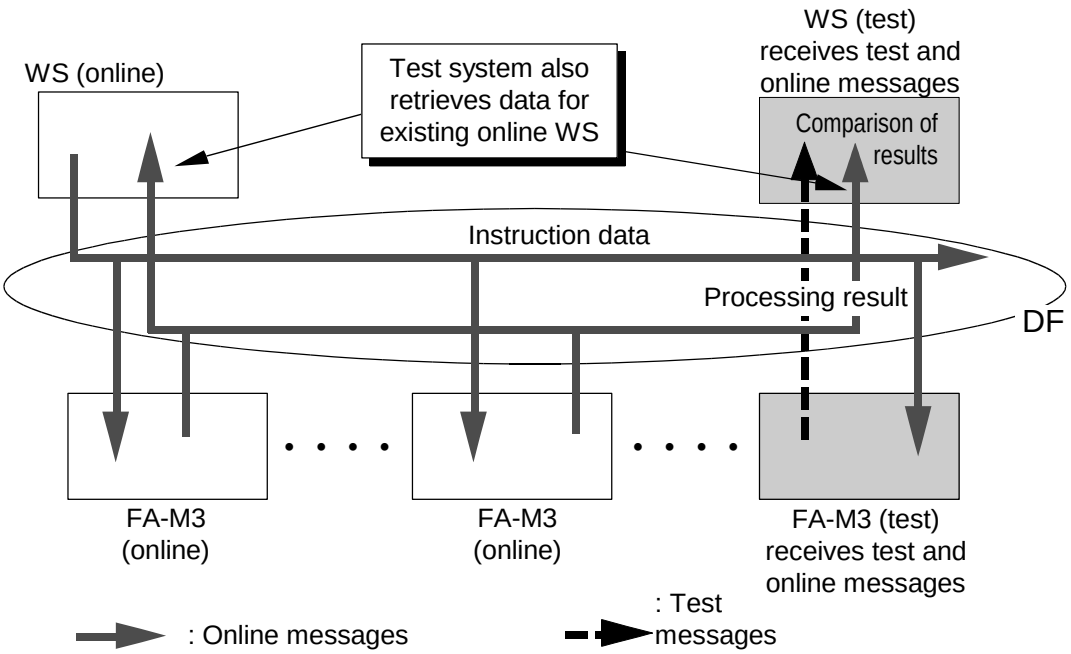


Figure 2.11 Testing by Receiving Both Testing and Online Messages

By suitably combining the node mode and message modes, it allows easy configuration of test environments, over and above those described earlier, that do not adversely affect online processing.

Blank Page

3. Alive Signal Transmission

3.1 Overview of Alive Signal Transmission

The alive signal transmission function of a node transmits periodic notification to other nodes connected to the same data field, informing them that the node is alive. By monitoring such notifications at the other nodes, it allows diagnosis and detection of failures and recoveries of individual nodes and the network. This periodic notification data is known as an alive signal and is transmitted and received in the same way as other messages using multicast communication. However, the NX Interface Module does not support receiving of alive signals and is unable to detect failures and recoveries of other nodes.

Alive signals are transmitted periodically at user-defined intervals, starting from NeXUS startup.

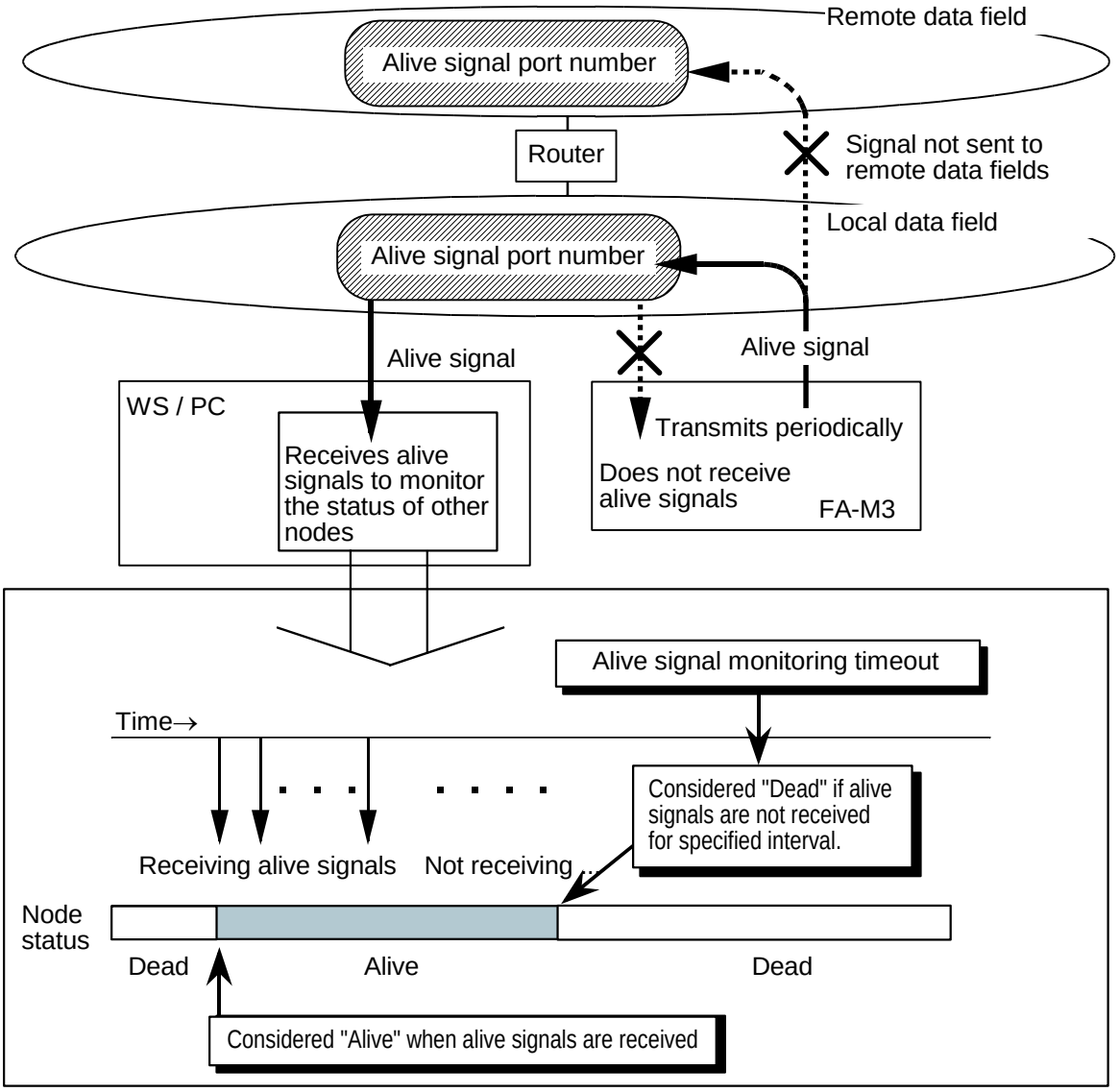


Figure 3.1 Overview of Alive Signal Transmission

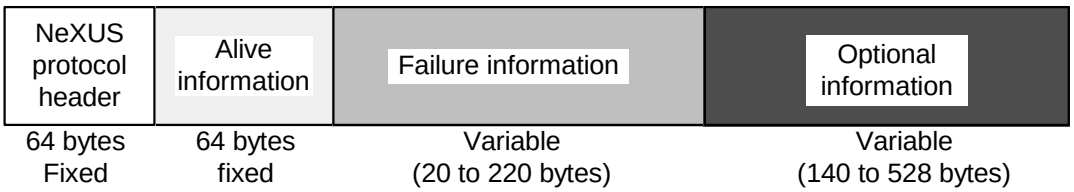
■ Components of Alive Signal

An alive signal consists of the following types of information:

- Alive information
- Failure information
- Optional information

For details on each type of information, see items 3.2 to 3.4.

Alive signal message



3.1.1 Managing Alive Signals

Alive signals are sent by multicast transmission to alive signal receiving port addresses as alive signal messages; Other equipment receiving the alive reports can access the various type of information contained in the alive signals. The signals can be used more effectively together with an NXViewer*.

The online monitoring function of NXViewer* allows the different types of information contained in the alive signal to be displayed graphically in dedicated windows with different colors, complete with automatic logging and other functions. System management of all data fields can be implemented simply by installing an NXViewer* in each data field.

* NXViwer is a NeXUS compliant software package developed by Hitachi Ltd. for personal computers. Contact Hitach Ltd. for details on the use and display format of NXViewer.

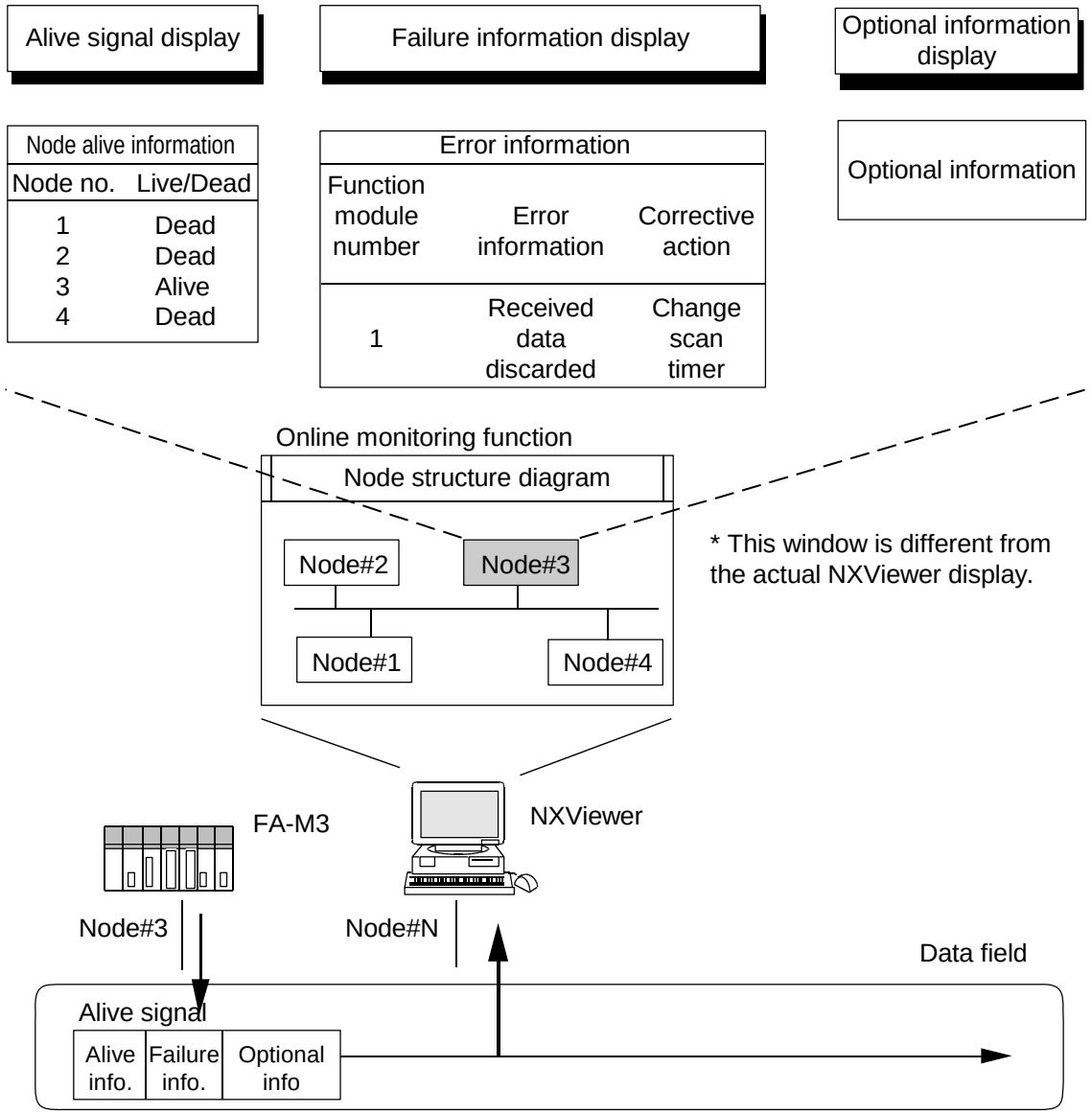


Figure 3.2 Management of Alive Signals

3.2 Managing Alive Information

"Alive Information" refers to information for managing the operation status of nodes and consists of the following information. Transmission of such information is transparent to the user.

- Node name
- Node number
- Node mode (Online or Test)
- Vendor equipment name
- Alive signal monitoring timeout
- System startup time

The NX Interface Module automatically obtains the system startup time using the CPU, which has the smallest number (usually the main CPU module) among the sequence CPU modules defined as function modules.

The vendor equipment name is the constant string "YE_FA_M3".

The node mode stores the current mode of the node.

The remaining information consists of initialization data specified by the user during initialization.

3.3 Managing Failure Information

Failure information consists of the following data. Transmission of failure information is also transparent to the user.

Alive/Dead status of function module

Error code(s) detected by NeXUS (error trace)

A function module is a sequence CPU module registered by the user; the alive/dead status of a function module refers to the operation status of the sequence CPU module. A sequence CPU is considered “dead” if it is in an error state with its “ERR LED” turned on or if its “RDY LED” is turned off, and is considered “alive” otherwise.

NeXUS checks every CPU to determine whether its “ERR” LED is turned on at periodic intervals called the alive signal transmission interval; if then attaches the label “alive”(or respectively “dead”) to the alive signal if the “ERR” LED is off (respectively on) and transmits the signal.

The error information reported includes errors detected for requests (transmission, receiving, etc.) sent from the user and are logged in the trace area of the NX Interface Module. This logged information is transmitted with the alive signal. For details on the error codes, see Appendix A, “Error Codes”.

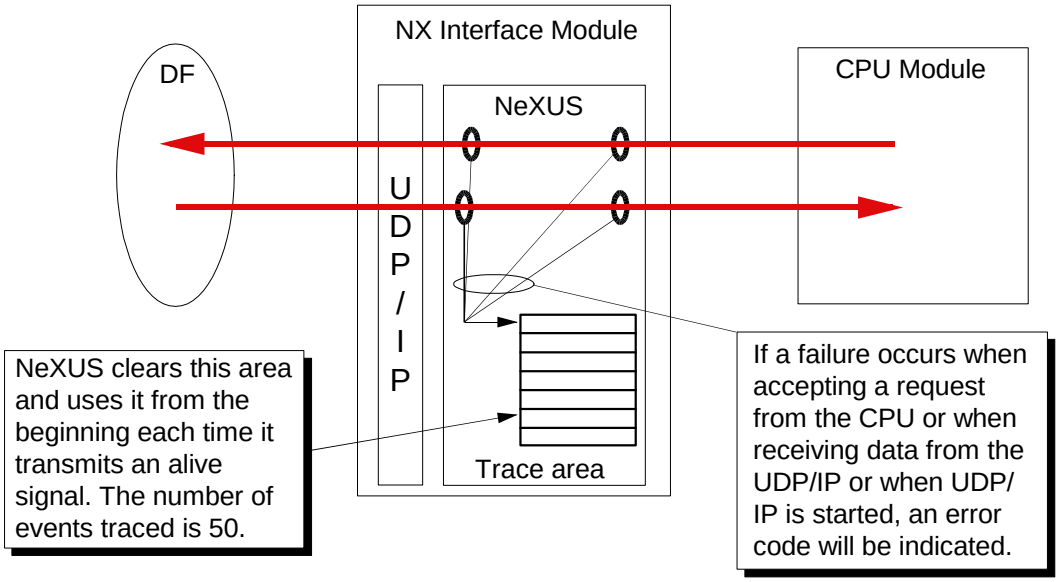


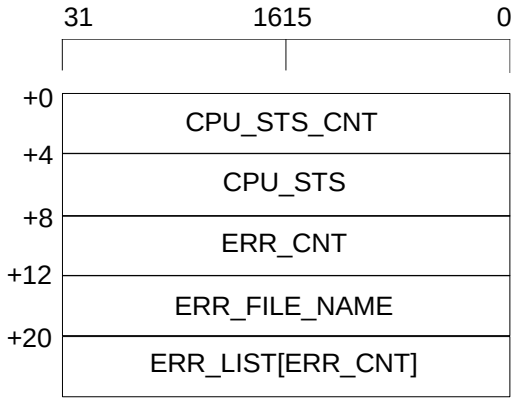
Figure 3.3 Failure Information Management

■ Format of Failure Information

Figure 3.4 shows the format of the failure information.

When creating monitoring programs on other nodes or NXDDE, use the format of the failure information shown in Figure 3.4 to access the data.

For details on the formats of the NeXUS header and alive signal, see Appendix D, “Format of Alive Signal Messages”.



Signal Name	Size	Description
CPU_STS_CNT	4	Number of modules (CPU module) reported
CPU_STS	4	Module alive information (Bitmap of 0s (alive) and 1s (dead) for up to 4 modules)
ERR_CNT	4	Number of error information blocks (Number of valid items in AL_ERR_LIST)
ERR_FILE_NAME	8	Name of error search file (ASCII string constant “NX01”)
ERR_LIST(ERR_CNT)	4×N	Function module number and error code of the error source (2 bytes each). The maximum value for N is 50.

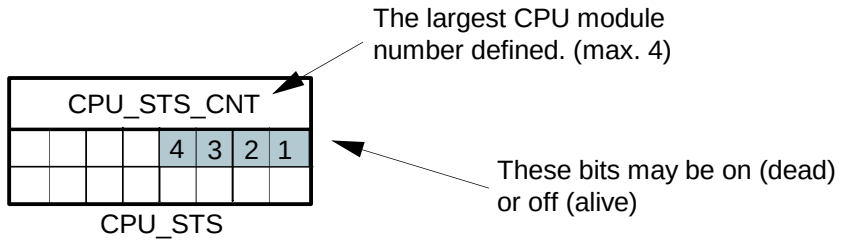


Figure 3.4 Format of Failure Information

3.4 Managing Optional Information

Alive information and failure information within the alive signals are defined by the NeXUS protocol; the NX Interface Module follows these specifications. However, the alive signal also contains optional information which allows equipment-specific formats.

Table 3.1 Contents of Optional Information

Item	Allocation Unit
Number of discarded received messages	Receiving handle
Number of transmitted/received messages	Transmission and receiving handle
Reasons for a lighted “ALM” LED	Function module (CPU)
User-defined information	Node

3.4.1 Number of Discarded Received Messages

The receiving buffer of an NX Interface Module is a memory image map of a data field for each transaction code. When a message for a transaction code is received, data is immediately transferred to the receiving buffer. On a first-come-first-serve basis, whilst the user is receiving a message, all newly-arrived messages with the same TCD will be discarded. These discarded messages are counted as the “number of discarded received messages”

Each handle is allocated two bytes within the storage area for counting the number of discarded received messages from 1 to $7FFF and back to 1 again.

This information can be used to check the suitability of the transmission interval for each transaction code and the scan time of the receiving ladder program.

3.4.2 Number of Transmitted/Received Messages

The number of transmitted/received messages provides statistical information on the traffic of each node. Each receiving or transmission handle is allocated 4 bytes within the storage area for counting the number of transmitted/received messages from1 to $7FFFFFFF and back to 1 again.

3.4.3 Reason for Lighted “ALM” LED

Data for the “reason for lighted “ALM” LED” can also be read from other nodes.

Possible reasons for a lighted “ALM” LED are allocated bits in a 16-word storage area as shown below.

Bit number

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
2	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
3	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
4	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
5	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
6	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
7	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112
8	143	142	141	140	139	138	137	136	135	134	133	132	131	130	129	128
9	159	158	157	156	155	154	153	152	151	150	149	148	147	146	145	144
10	175	174	173	172	171	170	169	168	167	166	165	164	163	162	161	160
11	191	190	189	188	187	186	185	184	183	182	181	180	179	178	177	176
12	207	206	205	204	203	202	201	200	199	198	197	196	195	194	193	192
13	223	222	221	220	219	218	217	216	215	214	213	212	211	210	209	208
14	239	238	237	236	235	234	233	232	231	230	229	228	227	226	225	224
15	255	254	253	252	251	250	249	248	247	246	245	244	243	242	241	240
16																

Bit Number	Cause	Bit Number	Cause
1	Power supply failure (momentary)	101	Configuration information error on FA link 3
9	Scan timeout	102	Duplicate device allocation on FA link 3
17	Input/output mismatch	104	Station number error on FA link 4
18	Input/output mismatch (READ/WRITE instructions)	105	Configuration information error on FA link 4
19	Input/output mismatch (HRD/HWR instructions)	106	Duplicate device allocation on FA link 4
25	Inter-CPU communication error	108	Station number error on FA link 5
33	Parameter error	109	Configuration information error on FA link 5
34	Invalid data	110	Duplicate device allocation on FA link 5
35	BIN/BCD conversion error	112	Station number error on FA link 6
36	Invalid FIFO table pointer	113	Configuration information error on FA link 6
37	Invalid device boundary	114	Duplicate device allocation on FA link 6
38	FOR-NEXT mismatch	116	Station number error on FA link 7
49	Subroutine error (return address not found)	117	Configuration information error on FA link 7
50	Subroutine error (nesting level too deep)	118	Duplicate device allocation on FA link 7
57	Macro instruction error (return address not found)	120	Station number error on FA link 8
58	Macro instruction error (nesting level too deep)	121	Configuration information error on FA link 8
65	Interrupt error (return address not found)	122	Duplicate device allocation on FA link 8
66	Interrupt stack overflow	128 to 143	Error in modules mounted in slots 1 to 16 of main unit
81	Station number error on FA link 1	144 to 159	Error in modules mounted in slots 1 to 16 of subunit 1
82	Configuration information error on FA link 1	160 to 175	Error in modules mounted in slots 1 to 16 of subunit 2
83	Duplicate device allocation on FA link 1	176 to 191	Error in modules mounted in slots 1 to 16 of subunit 3
97	Station number error on FA link 2	192 to 207	Error in modules mounted in slots 1 to 16 of subunit 4
98	Configuration information error on FA link 2	208 to 223	Error in modules mounted in slots 1 to 16 of subunit 5
99	Duplicate device allocation on FA link 2	224 to 239	Error in modules mounted in slots 1 to 16 of subunit 6
100	Station number error on FA link 3	240 to 255	Error in modules mounted in slots 1 to 16 of subunit 7

Each connected CPU is allocated 32 bytes within the area for storing the “reasons for lighted “ALM” LED”.

TIP

Saving the reason for a lighted “ALM” LED

If reading of the “reason for a lighted “ALM” LED” field fails or if a function module is “dead”, the last data read (the last data which can be obtained is saved) is attached to the alive signal and transmitted.

3.4.4 User-defined Information

User-defined information can be added as optional information.

They may include, for example, information on registers in the PLC to be displayed in the NXViewer.

The first 2 bytes (1 channel) of the user-defined information specify the size of the valid data, followed by 2 empty bytes (1 channel) and the data area. For more details, see Section 3.5, “User Optional Information”.

Size of valid data (2 bytes)
System area (2 bytes)
User-defined data (256 bytes max.)

3.4.5 Data Format for Optional Information

The format for the optional information is shown below.

When creating monitoring programs on other nodes or NXDDE, access the data according to the data format of the optional information shown in Figure 3.5.

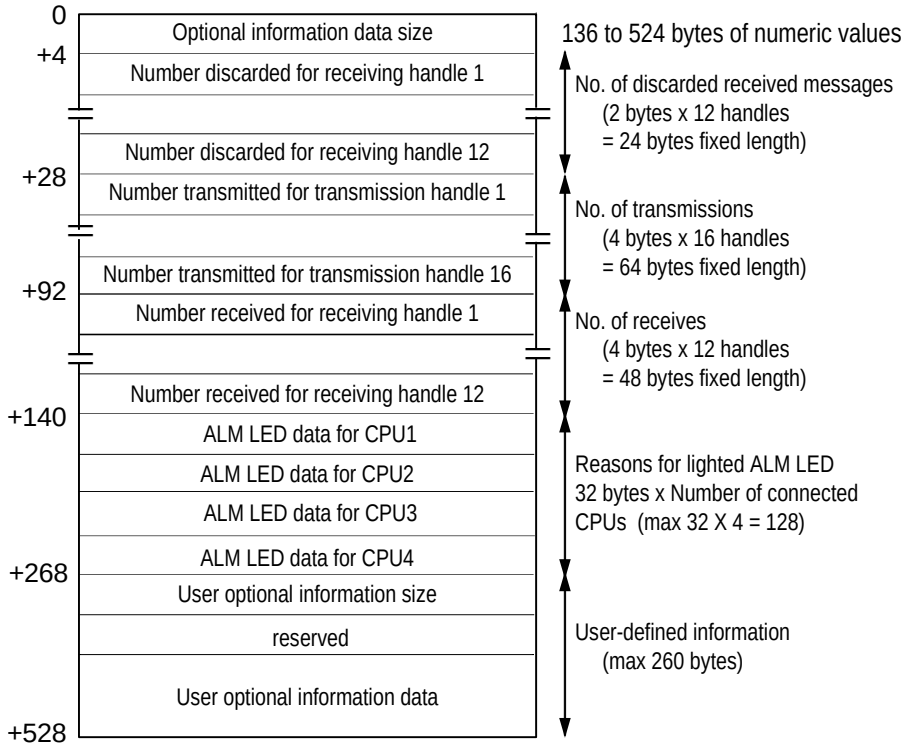


Figure 3.5 Data Format of Optional Information

3.5 Transmitting User Optional Information

The user optional information transmission function allows adding of user-defined information to alive signals for transmission.

3.5.1 Defining a Transmission Handle

There is no need to define a transmission handle and transmission attributes when requesting for transmission.

3.5.2 Data Format

The first 2 bytes (1 channel) of the user optional information area specify the data size, followed by 2 empty bytes (1 channel) and the data area.

NeXUS adds the data specified by the data size to the alive signal message before transmission.

User optional information area

Data size (2 bytes)
Reserved (2 bytes)
User-defined data (up to 256 bytes)

The data may be set from a sequence CPU module or an add-on CPU module.

CAUTION

The maximum data length is 256 bytes. If more than 256 bytes are specified, NeXUS will ignore the bytes beyond 256.

3.5.3 Saving User Optional Information

If a request for transmission of user-defined information has been made previously but there is no new transmission request at a subsequent alive signal transmission cycle, NeXUS attaches the data last specified (saved in the system area) to the alive signal and transmits the signal.

3.5.4 Transmission Procedure for User Optional Information

The procedure for optional information transmission from a ladder program is shown below.

For more details, see Section 5.3, “Transmitting Optional Information”

(1) The user checks that the Request to Transmit Optional Information relay and Optional Information Transmission Completed relay is off, sets the transmission data and turns on the Request to Transmit Optional Information relay.
NeXUS attaches the data to the alive signal, performs transmission and turns on the Optional Information Transmission Completed relay when transmission completes.

(2) The user checks that the Optional Information Transmission Completed relay has turned on and turns off the Request to Transmit Optional Information relay.
NeXUS checks that the Request to Transmit Optional Information relay is off and turns off the Optional Information Transmission Completed relay.

Example:

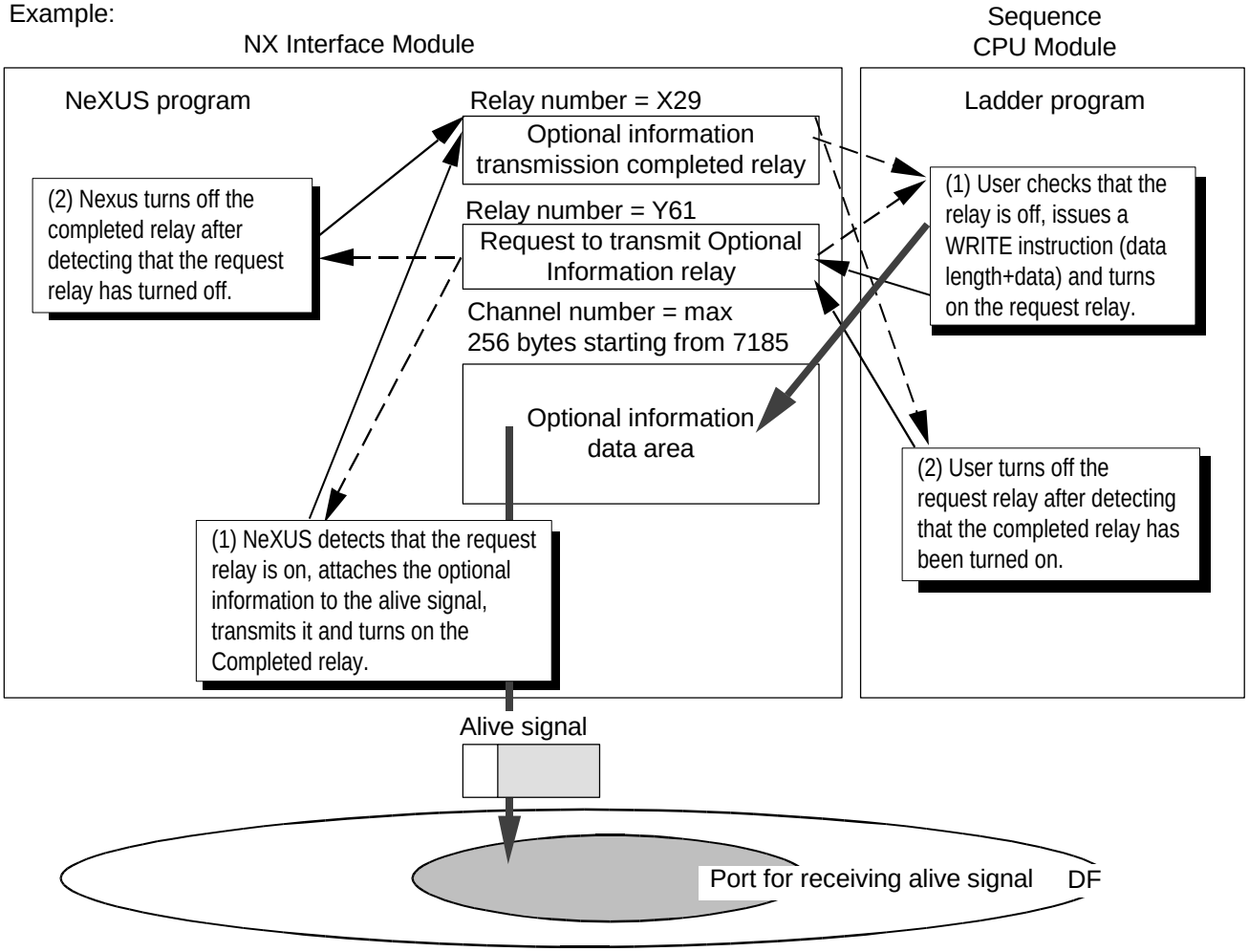


Figure 3.6 Optional Information Transmission Procedure

CAUTION

When using a multi-CPU system, note the following points:

The input/output relays of an NX Interface Module can be accessed from only one CPU module; the “Use/Not used” setting in the CPU configuration must be set accordingly. To access the input/output relays from multiple CPUs, use shared relays.

Other data areas (e.g. transmission and receiving buffers and initialization data area) can be accessed from multiple CPUs.

Blank Page

4. Starting and Stopping the System

4.1 Function Overview

The system startup and stop functions allow the user to request the NX Interface Module to startup or stop NeXUS from a sequence CPU module. To issue a startup request, the user should first set up the initialization data.

● System Operation Procedure

The workflow for building a system until system operation is shown below.

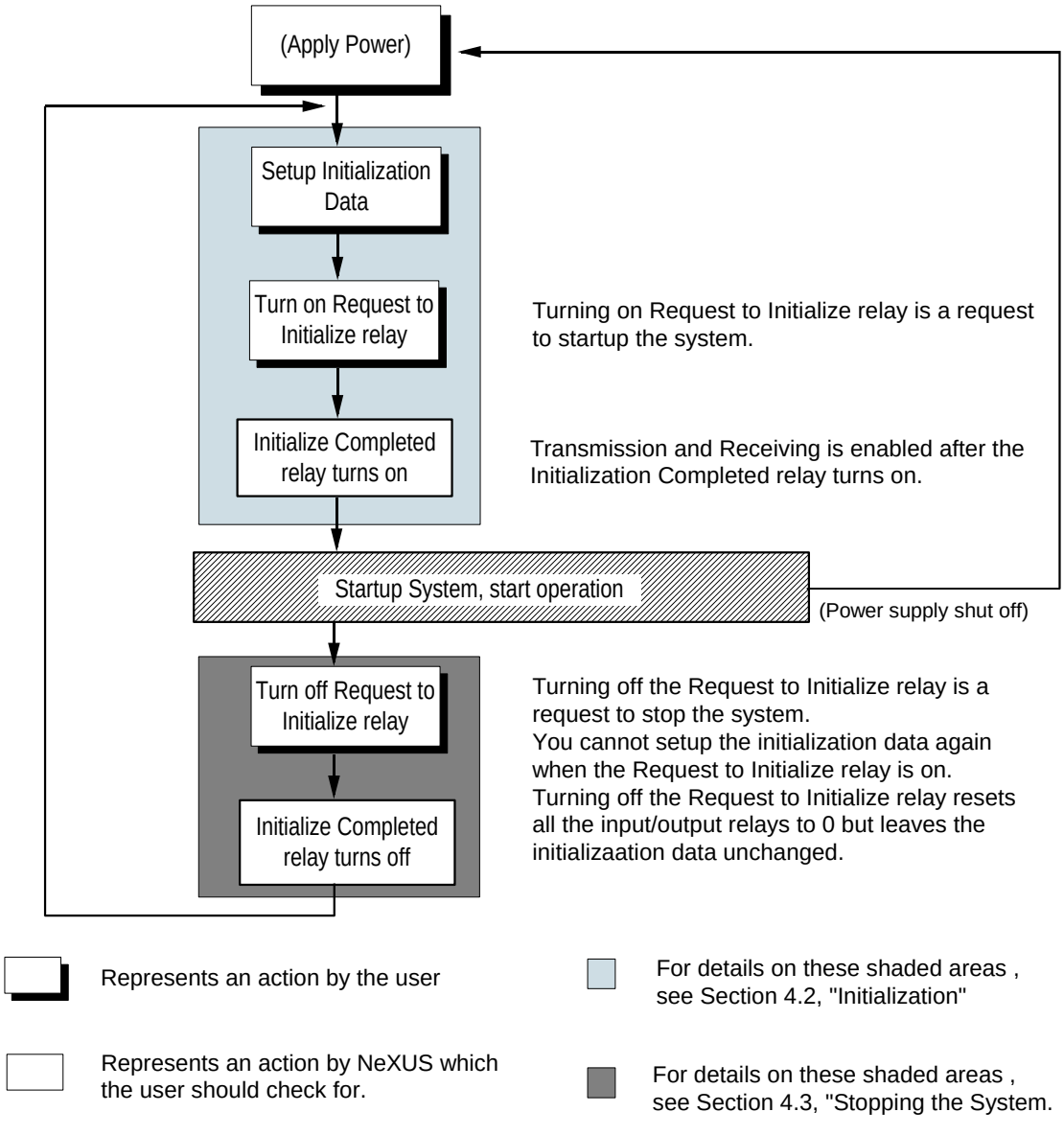


Figure 4.1 System Operation Procedure

If the Request to Initialize relay is turned off, NeXUS resets the input/output relays to zero.

Therefore, do not turn off the Request to Initialize relay during transmission processing or during receiving processing.

The initialization data is preserved even after the Request to Initialize relay has been turned off. To update the initialization data subsequently, only the modified values need to be set again.

4.2 Initialization

The user must setup the initialization data before issuing an initialization request.

The initialization data is stored at channel numbers, starting from 7315 for up to 136 words.

This data area is cleared to zeroes at powering on.

To use a default value, simply do not set the corresponding control information since NeXUS takes a zero value to mean “default”. The initialization data area is preserved even after a stop request has been issued; hence, you do not need to set the initialization data again when restarting with the same values.

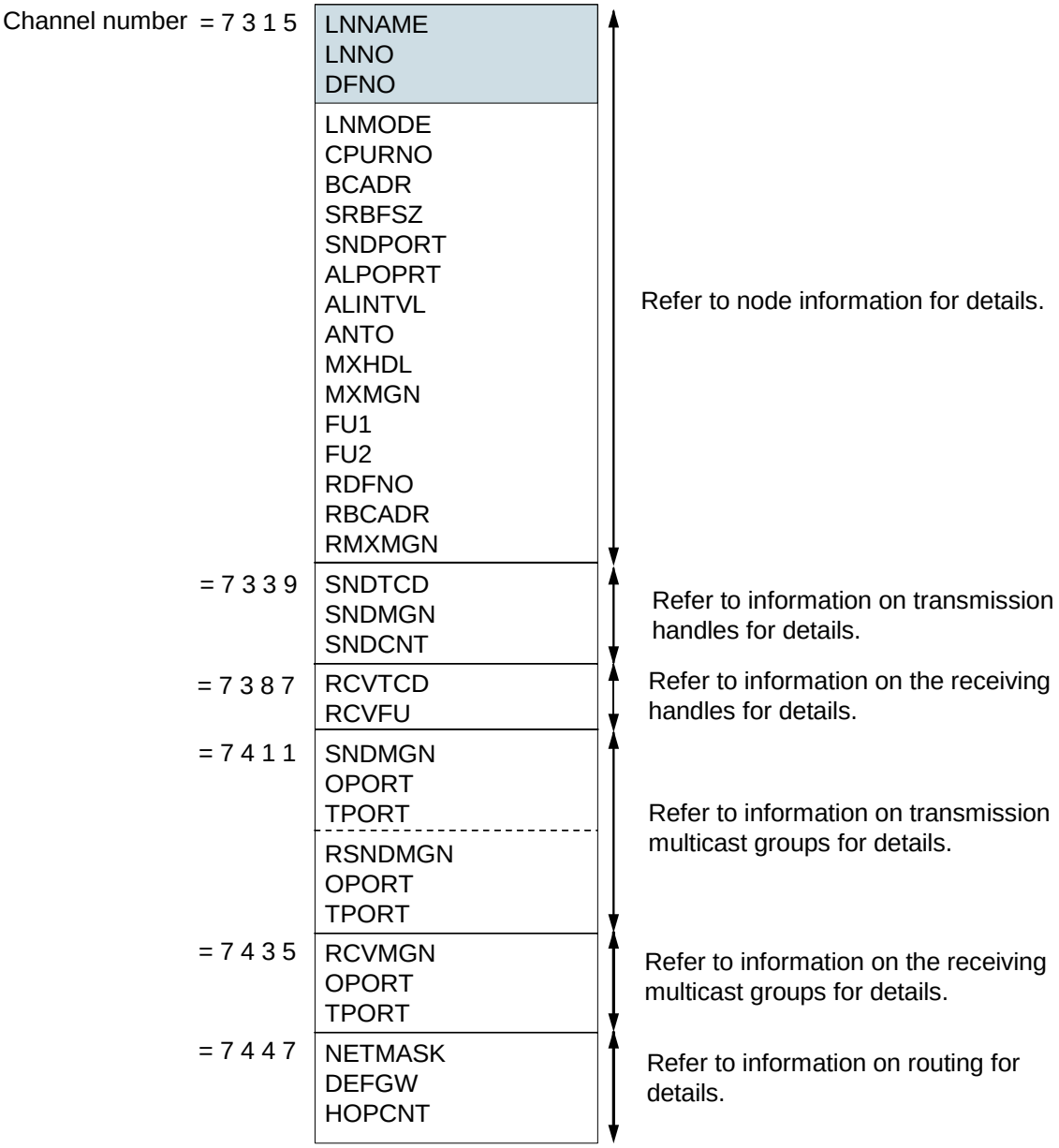


Figure 4.2 Structure Overview of Initialization data

Items in the shaded area cannot use the default values and must be set.

However, if the system is started with only items in the shaded area specified, only the alive signal transmission function is available.

● Mapping between System Configuration and Initialization data

Figure 4.3 illustrates the meaning of each initialization data item. Each symbol within parenthesis maps to an item in the previous section and is accompanied by its name, which is self-explanatory.

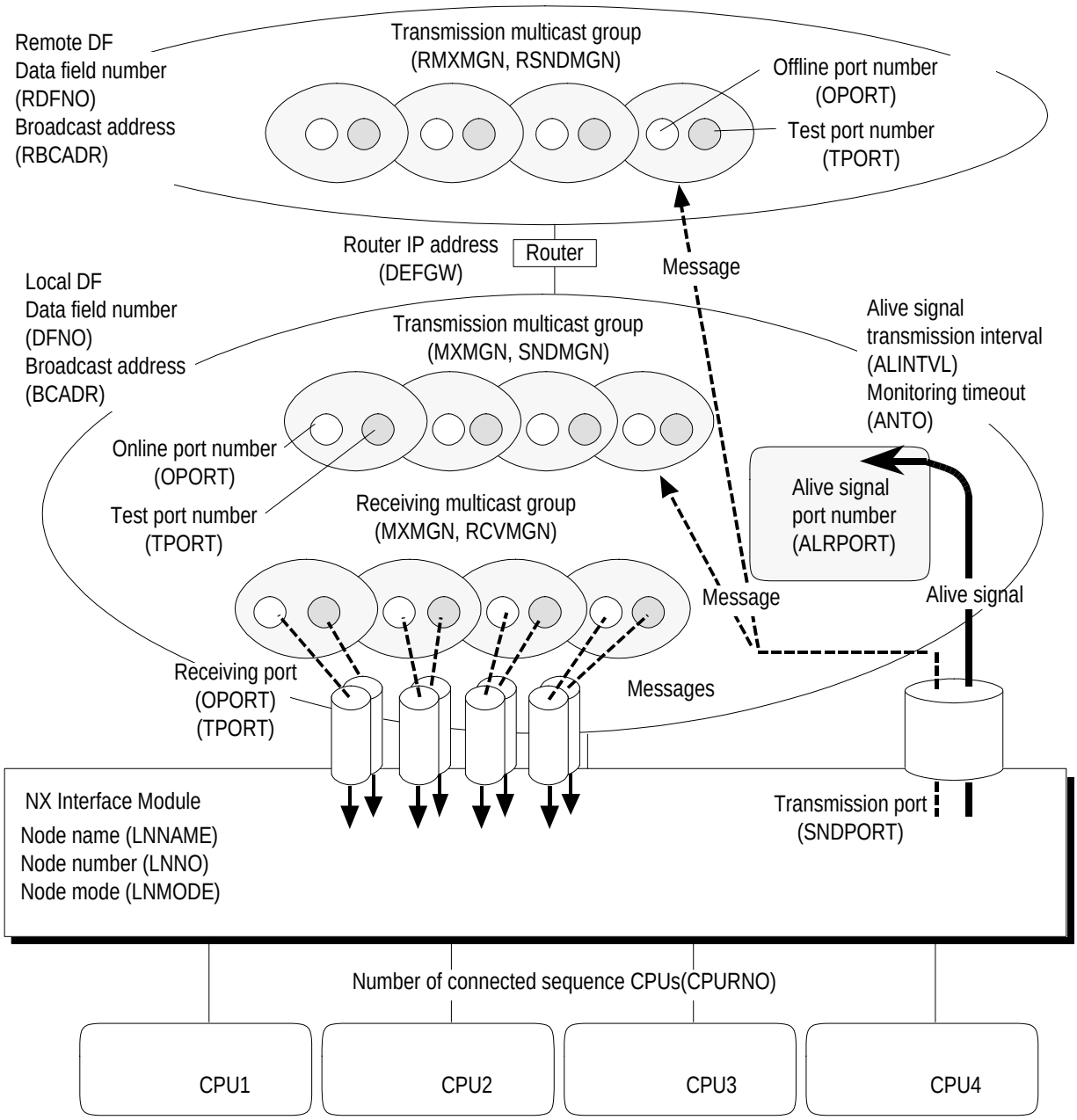


Figure 4.3 Mapping between System Configuration and Initialization data

In addition to the above data items, the NX Interface Module also holds the following control information internally.

- Transmission handle count (MXHDL)
- Transmission TCD (SNDTCD), transmission MGN (SNDMGN) and transmission data size (SNDCNT) for each transmission handle
- Receiving handle count (RCVHDL)
- Receiving TCD for each receiving handle (RCVTCD)
- Transmission and receiving buffer sizes (SRBFSZ)

- The input and output relays used by this function are listed in Table 4.1. See the description for each function for more details.

Table 4.1 List of Input/Output Relays

Input/Output Relay Number	Relay Name	Description
Xlmm30	Initialization Completed relay	Shows whether NeXUS is running. (on: running, off: not running)
Xlmm31	Initialization Error relay	Shows the outcome of the initialization. (on: an error has occurred, off: no error)
Xlmm32	Mode Status relay	Shows the operation mode of NeXUS. (on: test, off: online)
Xlmm62	Request to Initialize relay	Relay used to issue a request to startup NeXUS.
Xlmm64	Request to Change Mode relay	Relay used to issue a request to change the operation mode of NeXUS.

l: slot number; mm: slot number

CAUTION

When using a multi-CPU system, note the following points:

The input/output relays of an NX Interface Module can be accessed from only one CPU module; the “Use/Not used” setting in the CPU configuration must be set accordingly. To access the input/output relays from multiple CPUs, use shared relays.

Other data areas (e.g. transmission and receiving buffers and initialization data area) can be accessed from multiple CPUs.

4.2.1 Initialization data

● Node Information

Node information consists of information for controlling the node and information for controlling the data field.

Data field information consists of local data field information and remote data field information.

The default value is the value used when zero is specified.

At powering on, the initialization data area is set to all zeroes. Hence, to use the default value for a data item, simply do not set the data item.

Table 4.2 Node Information (1/2)

Channel Number	Mnemonic	Description	Setting Range	Default Value
7315	LNNAME	Specify a 10-character ASCII string with a ASCII NULL character at the end. This data item must be specified.	10 ASCII characters	Required
7320	LNNO	Logical node number. This data item must be specified.	1 to 4095	Required
7321	DFNO	Local DF number. This data item must be specified.	1 to 255	Required
7322	LNMODE	Select the node mode at system startup. $0000 : online mode $8000 : test mode (receiving online mode messages) $8001 : test mode (receiving test mode messages) $8002 : test mode (receiving all messages)	As shown on left	Online
7323	CPURNO	Specify the CPU number of connected CPUs with the following bit pattern. 15 … 3 2 1 0 0 | Add on CPU3, | Add on CPU2, | Add on CPU1, | Sequence CPU When 0 is specified, it is assumed that only the sequence CPU is connected.	As shown on left	Only sequence CPU is connected
7324	FUB	Always set this to 0.	0	–
7325	FUA	Always set this to 0.	0	–
7326	SRBFSZ	Select the transmission/receiving buffer case sizes (bytes). $0000 : 256 bytes for both transmission and receiving $0001 : 256 bytes for transmission, 512 bytes for receiving $0002 : 256 bytes for transmission, 1024 bytes for receiving $0100 : 512 bytes for transmission, 256 bytes for receiving $0101 : 512 bytes for both transmission and receiving $0102 : 512 bytes for transmission, 1024 bytes for receiving $0200 : 1024 bytes for transmission, 256 bytes for receiving $0201 : 1024 bytes for transmission, 512 bytes for receiving $0202 : 1024 bytes for both transmission and receiving	As shown on left	256 bytes for both transmission and receiving

* Set each word in ASCII.

Table 4.2 Node Information (2/2)

Channel Number	Mnemonic	Description	Setting Range	Default Value
7327	SNDPORT	Message transmission port number When 0 is specified, port 1025 is used as the transmission port.	0 to 65535	1025
7328	ALRPORT	Receiving port number for alive reporting When 0 is specified, port 600 is used as the receiving port for alive reporting	0 to 65535	600
7329	ALINTVL	Alive signal transmission interval (seconds) When 0 is specified, 10 seconds is assumed.	0 to 3600	10 s
7330	ALTO	Alive signal monitoring timeout (seconds) When 0 is specified, 30 seconds is assumed.	1 to 43200	30 s
7331	MXHDL	Specify the number of transmission and receiving handles to be registered in hexadecimal as follows: (e.g. $XXYY) 15 … 87 … 0 Transmission handle count | Receiving handle count	Receiving: 0 to 12 Transmission: 0 to 16	Transmission and receiving not defined
7332	MXMGN	Specify the number of transmission and receiving multicast groups to be registered for the local DF in hexadecimal as follows: 15 … 87 … 0 Transmission MGN count | Receiving MGN count	Receiving: 0 to 4 Transmission: 0 to 4	Transmission and receiving not defined
7333	FU1	Reserved	–	Do not specify
7334	FU2	Reserved	–	Do not specify
7335	RDFNO	Remote DF number. Set to 0 if remote DF is not used.	1 to 255	Not used
7336 7337	RBCADR	Broadcast address for remote DF in hexadecimal (e.g. $XXXXXXXX)	–	Not used
7338	RMXMGN	Number of transmission multicast group registrations for the remote DF	0 to 4	Transmission and receiving not defined

● Information on the Transmission Handles

Information on the transmission handles consists of the transmission transaction code and transmission multicast group number for each transmission handle.

Up to 16 transmission handles can be registered. Define the maximum handle number to be registered in the node information (MXHDL) and performs handle definition sequentially starting from the first handle.

Table 4.3 Information on the Transmission Handles

Channel Number	Mnemonic	Description	Setting Range	Default Value
7339	SNDTCD1	Transmission transaction code	1 to 59999	Not defined
7340	SNDMGN1	Transmission multicast group number*	1 to 255	Not defined
7341	SNDCNT1	Transmission data size (excluding NeXUS header)	0 to 1024	Not defined
7342	SNDTCD2	Transmission transaction code	1 to 59999	Not defined
7343	SNDMGN2	Transmission multicast group number*	1 to 255	Not defined
7344	SNDCNT2	Transmission data size (excluding NeXUS header)	0 to 1024	Not defined
7345	SNDTCD3	Transmission transaction code	1 to 59999	Not defined
7346	SNDMGN3	Transmission multicast group number*	1 to 255	Not defined
7347	SNDCNT3	Transmission data size (excluding NeXUS header)	0 to 1024	Not defined
7348	SNDTCD4	Transmission transaction code	1 to 59999	Not defined
7349	SNDMGN4	Transmission multicast group number*	1 to 255	Not defined
7350	SNDCNT4	Transmission data size (excluding NeXUS header)	0 to 1024	Not defined
• • •	• • •	• • •	• • •	• • •
7375	SNDTCD13	Transmission transaction code	1 to 59999	Not defined
7376	SNDMGN13	Transmission multicast group number*	1 to 255	Not defined
7377	SNDCNT13	Transmission data size (excluding NeXUS header)	0 to 1024	Not defined
7378	SNDTCD14	Transmission transaction code	1 to 59999	Not defined
7379	SNDMGN14	Transmission multicast group number*	1 to 255	Not defined
7380	SNDCNT14	Transmission data size (excluding NeXUS header)	0 to 1024	Not defined
7381	SNDTCD15	Transmission transaction code	1 to 59999	Not defined
7382	SNDMGN15	Transmission multicast group number*	1 to 255	Not defined
7383	SNDCNT15	Transmission data size (excluding NeXUS header)	0 to 1024	Not defined
7384	SNDTCD16	Transmission transaction code	1 to 59999	Not defined
7385	SNDMGN16	Transmission multicast group number*	1 to 255	Not defined
7386	SNDCNT16	Transmission data size (excluding NeXUS header)	0 to 1024	Not defined

* When transmitting to the remote DF, turn on the first bit of the transmission multicast group number (e.g. for transmitting to MGN=255, use $80FF)

● Information on the Receiving Handles

Information on the receiving handles consists of the receiving transaction code for each receiving handle.

Up to 12 receiving handles can be registered. Define the maximum handle number to be registered in the node information (MXHDL) and performs the definition sequentially starting from the first handle.

Table 4.4 Information on the Receiving Handles

Channel Number	Mnemonic	Description	Setting Range	Default Value
7387	RCVTCD1	Receiving transaction code	1 to 59999	Not defined
7388	RCVFU1	Reserved	–	–
7389	RCVTCD2	Receiving transaction code	1 to 59999	Not defined
7390	RCVFU2	Reserved	–	–
7391	RCVTCD3	Receiving transaction code	1 to 59999	Not defined
7392	RCVFU3	Reserved	–	–
7393	RCVTCD4	Receiving transaction code	1 to 59999	Not defined
7394	RCVFU4	Reserved	–	–
7395	RCVTCD5	Receiving transaction code	1 to 59999	Not defined
7396	RCVFU5	Reserved	–	–
7397	RCVTCD6	Receiving transaction code	1 to 59999	Not defined
7398	RCVFU6	Reserved	–	–
7399	RCVTCD7	Receiving transaction code	1 to 59999	Not defined
7400	RCVFU7	Reserved	–	–
7401	RCVTCD8	Receiving transaction code	1 to 59999	Not defined
7402	RCVFU8	Reserved	–	–
7403	RCVTCD9	Receiving transaction code	1 to 59999	Not defined
7404	RCVFU9	Reserved	–	–
7405	RCVTCD10	Receiving transaction code	1 to 59999	Not defined
7406	RCVFU10	Reserved	–	–
7407	RCVTCD11	Receiving transaction code	1 to 59999	Not defined
7408	RCVFU11	Reserved	–	–
7409	RCVTCD12	Receiving transaction code	1 to 59999	Not defined
7410	RCVFU12	Reserved	–	–

● Information on the Transmission Multicast Groups

Information on the transmission multicast groups consists of the port numbers for each transmission multicast group number.

Up to 4 transmission multicast groups each for the local data field and remote data field can be registered sequentially, starting from the first registration.

Port numbers for each group consist of an online port number and a test port number. For online nodes, define only the online port number. For test nodes, define the test port and if required, the online port as well.

Table 4.5 Information on the Transmission Multicast Groups

Channel Number	Mnemonic	Description	Setting Range	Default Value
7411	SNDMGN1	Transmission multicast group number for local data field.	1 to 255	Not defined
7412	OPORT1	Port number for online mode	1 to 65535	Not defined
7413	TPORT1	Port number for test mode	1 to 65535	Not defined
7414	SNDMGN2	Transmission multicast group number for local data field.	1 to 255	Not defined
7415	OPORT2	Port number for online mode	1 to 65535	Not defined
7416	TPORT2	Port number for test mode	1 to 65535	Not defined
7417	SNDMGN3	Transmission multicast group number for local data field.	1 to 255	Not defined
7418	OPORT3	Port number for online mode	1 to 65535	Not defined
7419	TPORT3	Port number for test mode	1 to 65535	Not defined
7420	SNDMGN4	Transmission multicast group number for local data field.	1 to 255	Not defined
7421	OPORT4	Port number for online mode	1 to 65535	Not defined
7422	TPORT4	Port number for test mode	1 to 65535	Not defined
7423	RSNDMGN1	Transmission multicast group number for remote data field.	1 to 255	Not defined
7424	ROPORT1	Port number for online mode	1 to 65535	Not defined
7425	RTPORT1	Port number for test mode	1 to 65535	Not defined
7426	RSNDMGN2	Transmission multicast group number for remote data field.	1 to 255	Not defined
7427	ROPORT2	Port number for online mode	1 to 65535	Not defined
7428	RTPORT2	Port number for test mode	1 to 65535	Not defined
7429	RSNDMGN3	Transmission multicast group number for remote data field.	1 to 255	Not defined
7430	ROPORT3	Port number for online mode	1 to 65535	Not defined
7431	RTPORT3	Port number for test mode	1 to 65535	Not defined
7432	RSNDMGN4	Transmission multicast group number for remote data field.	1 to 255	Not defined
7433	ROPORT4	Port number for online mode	1 to 65535	Not defined
7434	RTPORT4	Port number for test mode	1 to 65535	Not defined

To change the node mode online, you must define both the online port and the test port in advance.

For details, see “TIP” in item “Information on the Receiving Multicast Groups”.

● Information on the Receiving Multicast Groups

Information on the receiving multicast groups consists of the port numbers for each receiving multicast group number.

Up to 4 receiving multicast groups for the local data field can be registered sequentially, starting from the first registration.

Port numbers for each group consist of an online port number and a test port number. For online nodes, define only the online port number. For test nodes, define the test port and if required, the online port as well.

Table 4.6 Information for the Receiving Multicast Groups

Channel Number	Mnemonic	Description	Setting Range	Default Value
7435	RCVMGN1	Receiving multicast group number for local data field.	1 to 255	Not defined
7436	OPORT1	Port number for online mode	1 to 65535	Not defined
7437	TPORT1	Port number for test mode	1 to 65535	Not defined
7438	RCVMGN2	Receiving multicast group number for local data field.	1 to 255	Not defined
7439	OPORT2	Port number for online mode	1 to 65535	Not defined
7440	TPORT2	Port number for test mode	1 to 65535	Not defined
7441	RCVMGN3	Receiving multicast group number for local data field.	1 to 255	Not defined
7442	OPORT3	Port number for online mode	1 to 65535	Not defined
7443	TPORT3	Port number for test mode	1 to 65535	Not defined
7444	RCVMGN4	Receiving multicast group number for local data field.	1 to 255	Not defined
7445	OPORT4	Port number for online mode	1 to 65535	Not defined
7446	TPORT4	Port number for test mode	1 to 65535	Not defined

TIP

To change the node mode online, you must define both the online port and the test port in advance.

For details on changing the node mode, see Section 4.4 “Changing to Online Mode”.

The following table lists whether specification of the online port number and the test port number is required for each node mode.

LNMODE	Node Mode (Message Receiving Mode)	Transmission Port		Receiving Port	
		Online	Test	Online	Test
$0000	Online	Required	No	Required	No
$8000	Test (online)	No	Required	Required	No
$8001	Test (test)	No	Required	No	Required
$8002	Test (both)	Required*	Required	Required	Required

* Set this to 0 if you do not intend to change the node mode.

● Routing Information

Routing information defines network information for the network segment connected using routers or other equipment.

One network address can be "subnetted" using a network mask.

The routing information determines the IP address of the gateway connected to the remote DF.

Table 4.7 Routing Information

Channel Number	Mnemonic	Description	Setting Range	Default Value
7447	NETMASK	Netmask with value 0 or $XXXXXXXX (hexadecimal value)	–	Not defined
7448				
7449	DEFGW	Router IP address with value 0 or $XXXXXXXX (hexadecimal value)	–	Not defined
7450				
7451	HOPCNT	Hop count*	–	Not defined

* For the hop count, specify 0 for a direct connection and a non-zero value for an indirect connection.

4.2.2 Startup Procedure

The procedure for system startup from a ladder program is given below.

(1) The user writes the initialization data and turns on the Request to Initialize relay. NeXUS turns on the Initialization Completed relay after performing initialization.

(2) The user confirms that the Initialization Completed relay has been turned on before issuing a transmission request or a receiving request. Note that in this case, the Request to Initialize relay should not be turned off since turning off the Request to Initialize relay will indicate to stop the system.

(3) If an initialization data error is detected, the Initialization Error relay will turn on. If the Initialization Error relay turns on, the user should correct the initialization data and turn off the Request to Initialize relay again.
When NeXUS detects that the Request to Initialize relay has been turned off, it will turn off the Initialization Error relay.

(4) The user checks that the Initialization Error relay has been turned off, writes the initialization data and turns on the Request to Initialize relay again.

For details on initialization from a ladder program, see Section 5.5, "System Startup".

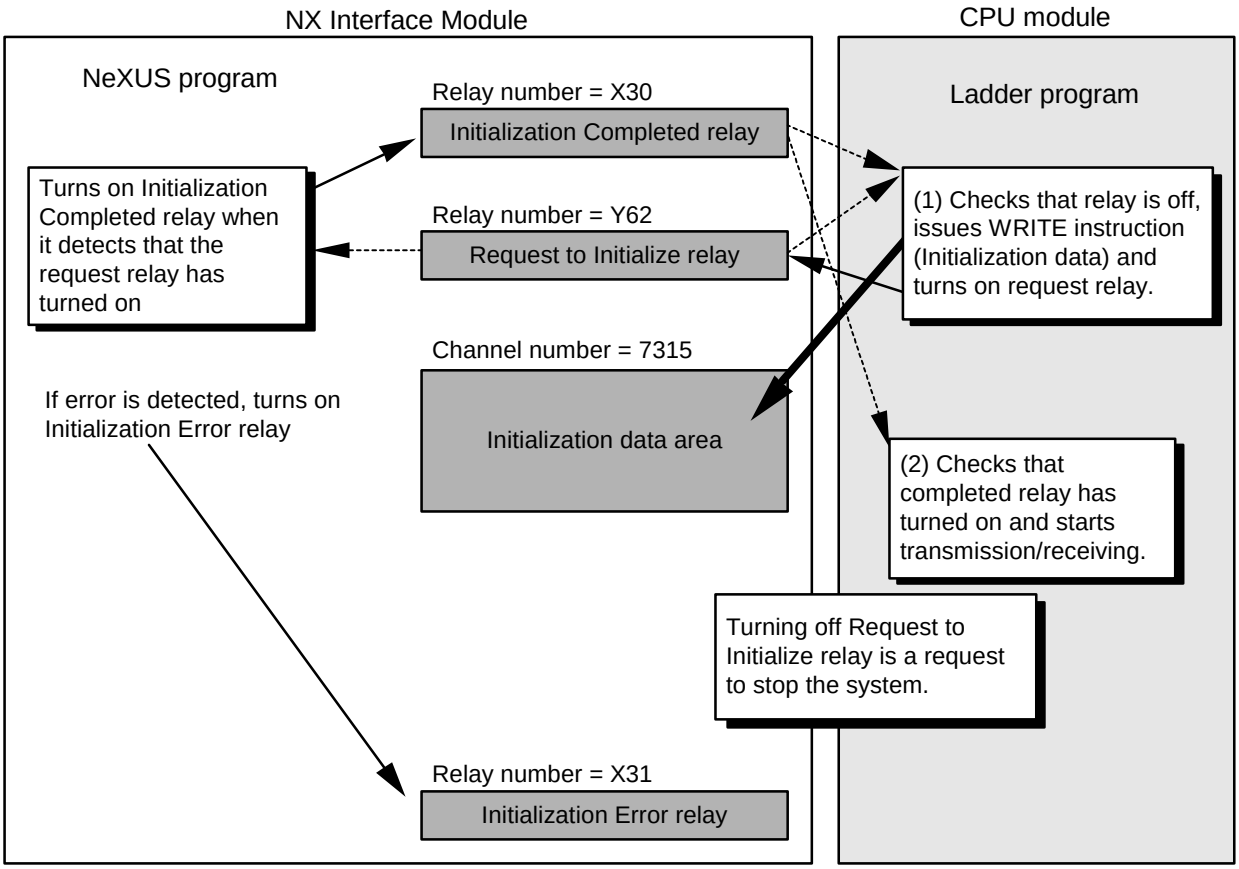


Figure 4.4 Startup Procedure

4.3 Stop Function

To change the initialization data, you do not need to turn off the power but you need to temporarily stop NeXUS.
A stop request can be issued by turning off the Request to Initialize relay but the following precautions must be noted:

- When NeXUS detects that the Request to Initialize relay has been turned off, it will reset the input relays used for transmission and receiving to zero. Therefore, do not turn off the Request to Initialize relay when a user program for transmission or receiving is running.
- Turning off the Request to Initialize relay resets the input relay to zero but does not reset the initialization data area to zeroes. Therefore, if only part of the initialization data is to be changed, you need to change only the affected data in the initialization data area.
- When NeXUS detects that the Request to Initialize relay has been turned off, it also turns off the Initialization Completed relay after performing stop processing. To restart the system, the user can turn on the Request to Initialize relay after confirming that the Initialization Completed relay has been turned off.

.

4.3.1 Procedure for Stopping the System

The procedure for stopping the system from a ladder program is given below.

The user turns off the Request to Initialize relay.

NeXUS turns off the Initialization Completed relay after performing stop processing.

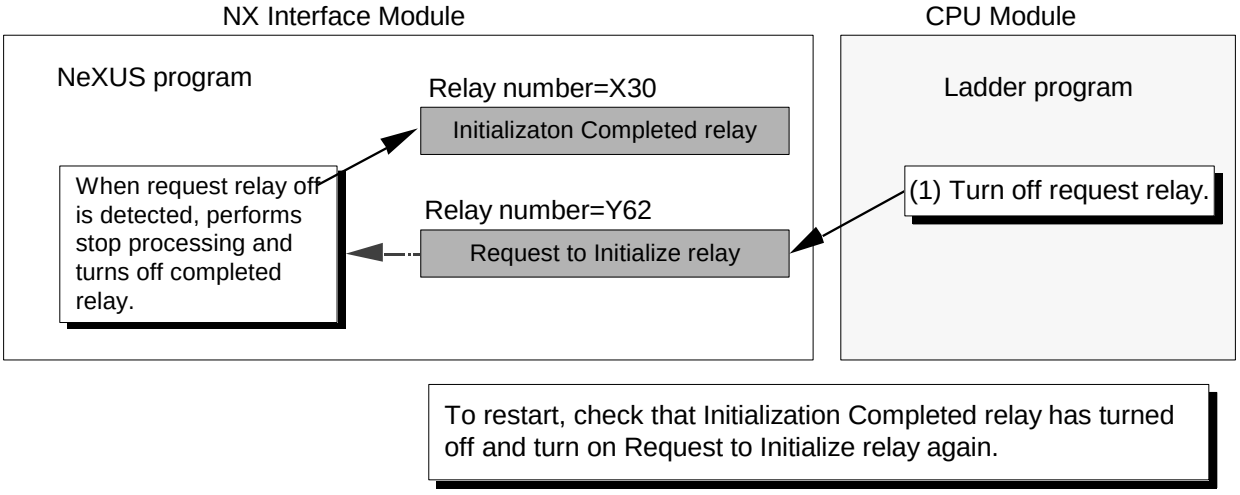


Figure 4.5 Procedure for Stopping the System

TIP

Alive signal reporting to all equipment connected to the data field is different depending on whether the Request to Initialize relay is turned off when stopping the system. If power is turned off after turning off the Request to Initialize relay, an equipment stop notification alive signal will be reported so that other equipment connected to the data field will know that it is a scheduled system stop.

If power is turned off without turning off the Request to Initialize relay, an equipment stop notification alive signal will not be reported. Therefore, it is recommended that you turn off the Request to Initialize relay before stopping the system in systems that monitors the operation status of the node.

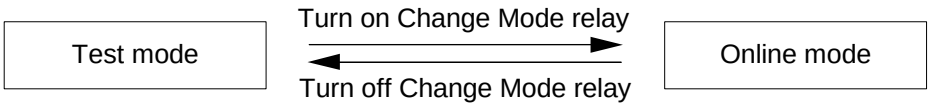
4.4 Changing to Online Mode

When changing the initialization data, you need to temporarily stop NeXUS except in the following situation.

● Making a Transition from Test Mode to Online Mode

By operating the Change Mode relay, you can make a transition from test mode to online mode without stopping NeXUS. This allows you to implement a test in an online environment and switch over to the online environment quickly after the test is completed. It also minimizes problems associated with the environment migration since there is absolutely no need to change the user program or configuration information.

Further, if a problem occurs during the migration to the online environment, you can quickly return to testing by simply using the Change Mode relay.



The Change Mode relay is enabled only if the system is started with the node mode at system startup set to "test mode". If the system is started in "online mode", any operation of the Change Mode relay will be ignored.

4.4.1 Restrictions

Since a test node can make a transition to an online node, the configuration information for online operation must be set up in advance from the test phase.

The configuration information for online operation refers to the transmission and receiving port numbers to be used for the transmission and receiving multicast groups. This information must be specified in the initialization data.

3 types of test nodes can be specified using configuration information, as listed below. Among these, mode change is only allowed for type 3.

(1) Receiving only test mode messages

(2) Receiving only online mode messages

(3) Receiving both test mode messages and online mode messages

4.4.2 Operation After Mode Change

When a test node is changed to an online node, the following attributes are changed.

Transmission message: changes from test mode messages to online mode messages. The transmission message serial number (including the version number) is initialized.

Receiving message: only online mode messages can be received; receiving of test messages is prohibited.

System startup time: remains unchanged.

Relays for transmission and receiving: all input relays are set to 0.

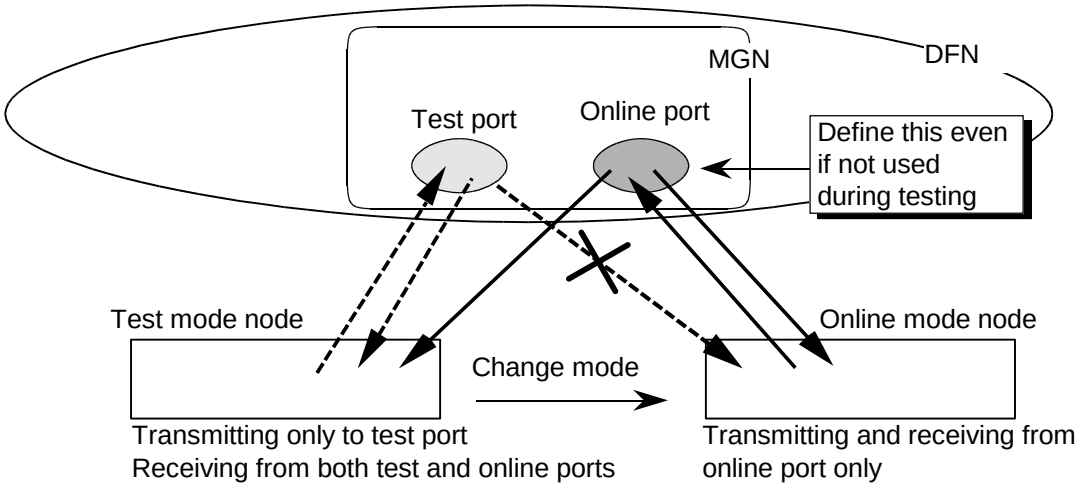


Figure 4.6 Operation After Mode Change

4.4.3 Procedure for Mode Change

The procedure for changing the mode from a ladder program is given below.

(1) The user turns on the Change Mode relay.

(2) When NeXUS detects that the Change Mode relay is on, it changes the node mode to online and turns off the Mode Status relay.
The Mode Status relay is on if the system was started in test mode.
Note that turning off the Change Mode relay after turning it on returns the node mode to test mode.

For details on changing the mode from a ladder program, see Section 5.4, "Changing the Node Mode".

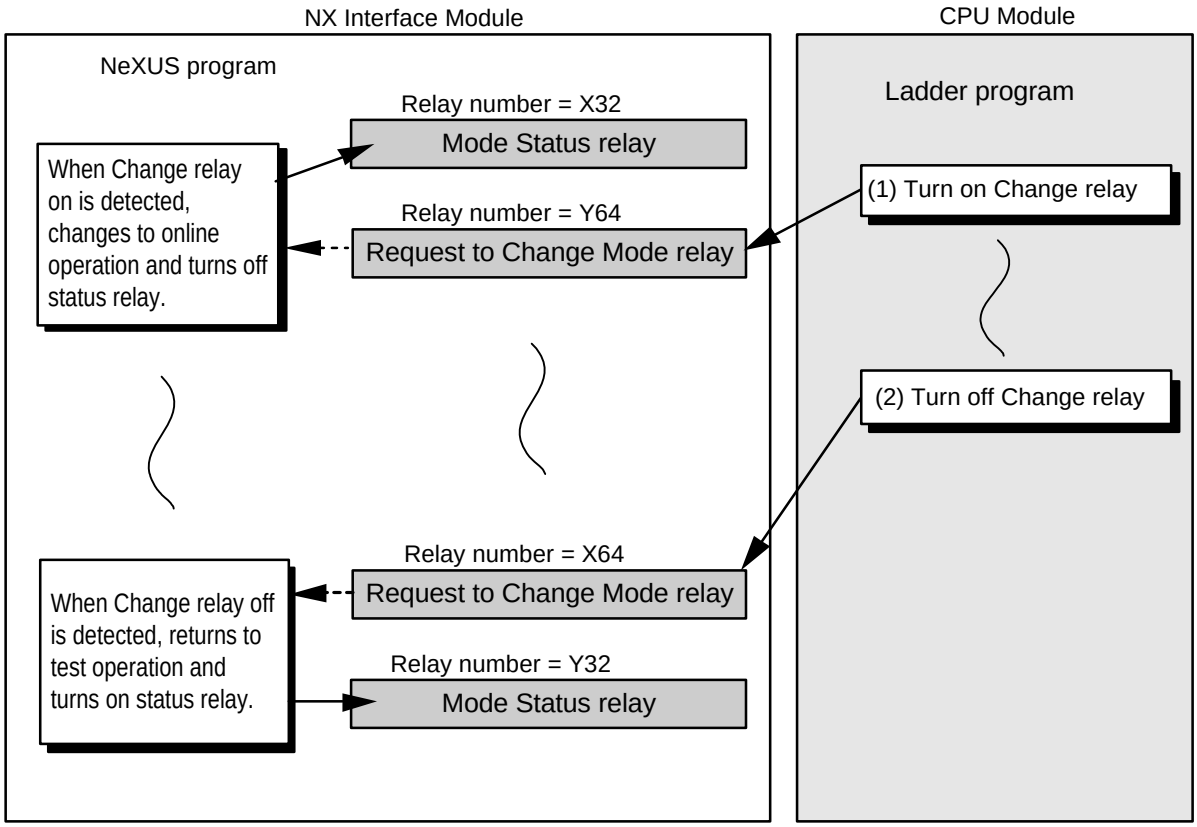


Figure 4.7 Procedure for Mode Change

Blank Page

5. Programmer's Reference

This chapter describes the ladder program interface.

For details on the buffer size, relay number and definition of the communication attributes, see Chapter 2, "Multicast Communication"

5.1 Transmitting Messages

5.1.1 Transmission Procedure

To transmit a message, set the transmission data and turn on the Request to Transmit relay. When transmission processing completes, the Transmission Completed relay will turn on.

Since transmission is performed according to the communication attributes and message size specified in the initialization data, it always ends normally.

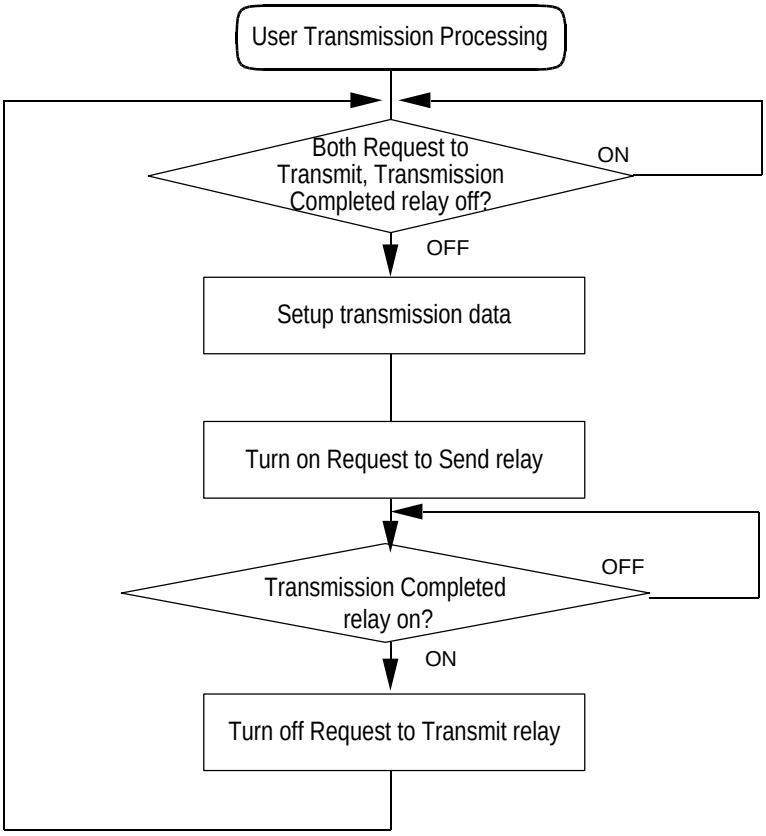
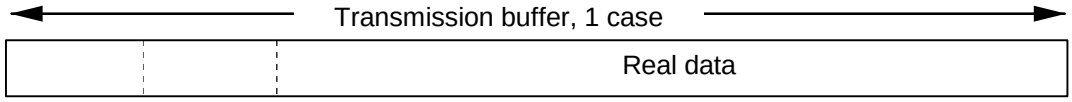


Figure 5.1 Procedure for Transmitting Messages

5.1.2 Data Format

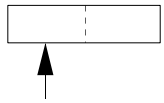
Set the data according to the following format.



Channel N N+1 . . .

Real data: Write the data to be transmitted using a WRITE instruction, specifying the channel number corresponding to the handle number containing the required communication attributes.
Set the real data such that its length is equal to the transmission data size (SNDCNT) specified in the transmission handle information of the initialization data.

Left-justify the data.
When transmitting an odd number of bytes, the last byte should be in the high-order position.
If the last data byte is $31 when transmitting an odd number of bytes, Set the value to $3100.



5.1.3 Relay Number

A request for message transmission is issued by turning on an output relay of the NX Interface Module. Completion of the transmission is indicated by an input relay.

Table 5.1 Relay Numbers for Message Transmission

Handle Number	Output Relay (Request to Transmit)	Input Relay (Completion Notification)
1	Ylmm33	Xlmm01
2	Ylmm34	Xlmm02
3	Ylmm35	Xlmm03
4	Ylmm36	Xlmm04
5	Ylmm37	Xlmm05
6	Ylmm38	Xlmm06
7	Ylmm39	Xlmm07
8	Ylmm40	Xlmm08
9	Ylmm41	Xlmm09
10	Ylmm42	Xlmm10
11	Ylmm43	Xlmm11
12	Ylmm44	Xlmm12
13	Ylmm45	Xlmm13
14	Ylmm46	Xlmm14
15	Ylmm47	Xlmm15
16	Ylmm48	Xlmm16

l : unit number
mm : slot number

5.1.4 Exit Code

There is no exit code.

5.1.5 Transmission Example

An example for transmitting 4 bytes of data ($12345678) using handle number 1 (with NX Interface Module mounted in slot number 4 and channel number =1) is given below.

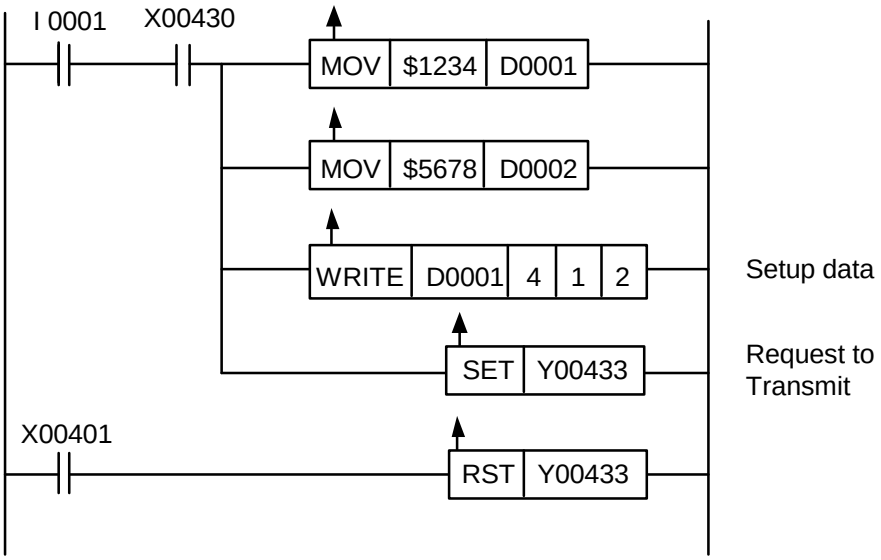


Figure 5.2 Sample Program for Message Transmission

CAUTION

When using a multi-CPU system, note the following points:

The input/output relays of an NX Interface Module can be accessed from only one CPU module; the “Use/Not used” setting in the CPU configuration must be set accordingly. To access the input/output relays from multiple CPUs, use shared relays.

Other data areas (e.g. transmission and receiving buffers and initialization data area) can be accessed from multiple CPUs.

5.2 Receiving Messages

5.2.1 Procedure for Receiving

Message receiving starts when the user detects that the Request to Receive relay is on.

When a message with TCD arrives, NeXUS stores the data in the channel for the TCD and turns on the Request to Receive relay.

When the user detects that the Request to Receive relay is on, he reads the data stored in the channel and turns on the Receiving Completed relay.

When NeXUS checks that the Receiving Completed relay has turned on, it turns off the Request to Receive relay.

When the user detects that the Request to Receive relay has turned off, he turns off the Receiving Completed relay.

Always turn off the Receiving Completed relay; failing or forgetting to do so will cause NeXUS to discard newly arrived TCD messages.

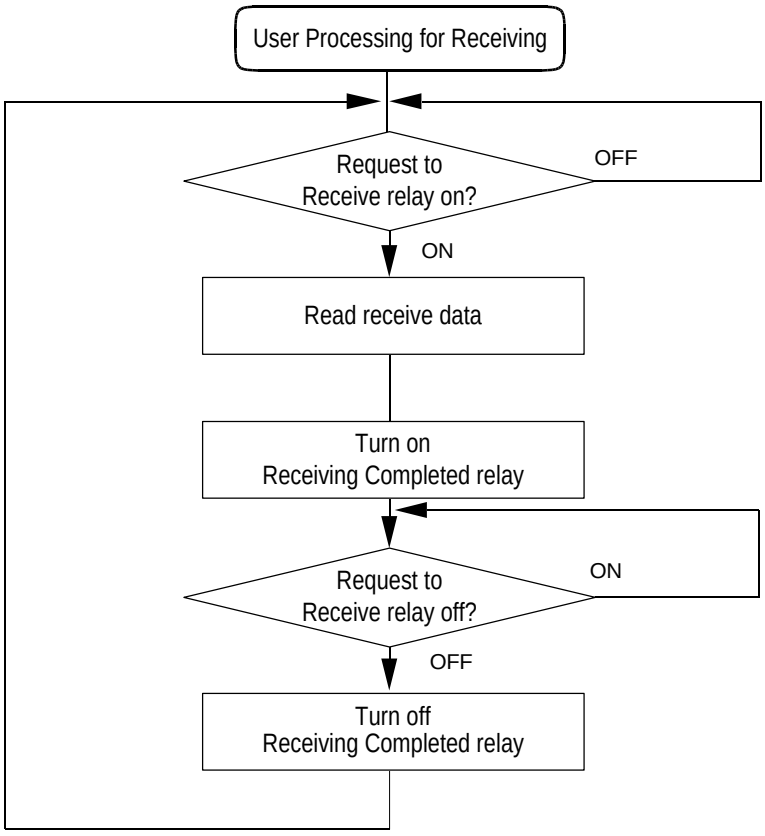
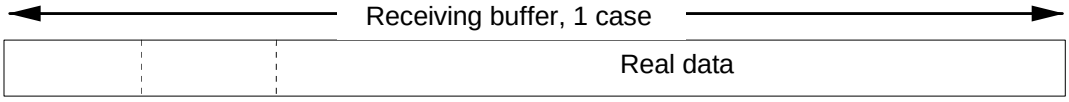


Figure 5.3 Procedure for Receiving Messages

5.2.2 Data Format

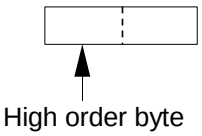
Data received is stored with the following format.



Channel N N+1 • • •

Real data: Read the data to receive using a READ instruction.
In the READ instruction, specify the channel number corresponding to the handle number with the desired communication attributes.
The size of the stored data will never exceed the receiving buffer size.
If data longer than the receiving buffer size is received, it is discarded by NeXUS.

The data will be left-justified.
When an odd number of bytes is received, the last byte will be in the high-order position.
If the last data byte is $31 when an odd number of bytes is received, It is stored as $3100.



To determine the size of the receive data, read the following channel number.

Table 5.2 Channel Number Storing the Size of the Receive Data

Handle Number	Channel Number Storing the Size of the Receive Data
1	7169
2	7170
3	7171
4	7172
5	7173
6	7174
7	7175
8	7176
9	7177
10	7178
11	7179
12	7180

5.2.3 Relay Number

NeXUS notifies the user that a message has arrived by turning on an input relay of the NX Interface Module.

The user notifies NeXUS that he has completed receiving by turning on an output relay.

Table 5.3 Relay Numbers for Message Receiving

Handle Number	Output Relay (Completion Notification)	Input Relay (Arrival Notification)
1	Ylmm49	Xlmm17
2	Ylmm50	Xlmm18
3	Ylmm51	Xlmm19
4	Ylmm52	Xlmm20
5	Ylmm53	Xlmm21
6	Ylmm54	Xlmm22
7	Ylmm55	Xlmm23
8	Ylmm56	Xlmm24
9	Ylmm57	Xlmm25
10	Ylmm58	Xlmm26
11	Ylmm59	Xlmm27
12	Ylmm60	Xlmm28

l : unit number
mm : slot number

5.2.4 Exit Code

There is no exit code.

5.2.5 An Example for Receiving

An example for receiving 4 bytes of data using handle number 1 (with slot number of NX Interface Module = 4 and channel number = 2049) is given below.

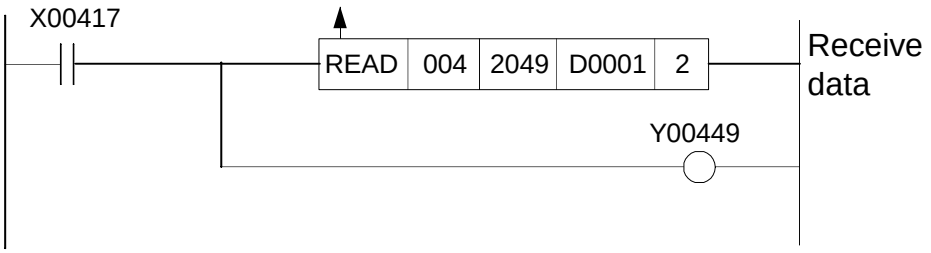


Figure 5.4 Sample Program for Receiving Messages

CAUTION

When using a multi-CPU system, note the following points:

The input/output relays of an NX Interface Module can be accessed from only one CPU module; the “Use/Not used” setting in the CPU configuration must be set accordingly. To access the input/output relays from multiple CPUs, use shared relays.

Other data areas (e.g. transmission and receiving buffers and initialization data area) can be accessed from multiple CPUs.

5.3 Transmitting Optional Information

5.3.1 Transmission Procedure

To transmit optional information, set the transmission data size and transmission data and turn on the Request to Transmit Optional Information relay. The channel number for the transmission data size and transmission data is fixed at 7185.

When transmission processing completes, the Optional Information Transmission Completed relay will turn on.

The maximum transmission data size is 256 bytes; if more than 256 bytes of data are specified, only 256 bytes will be transmitted but no error will be signaled.

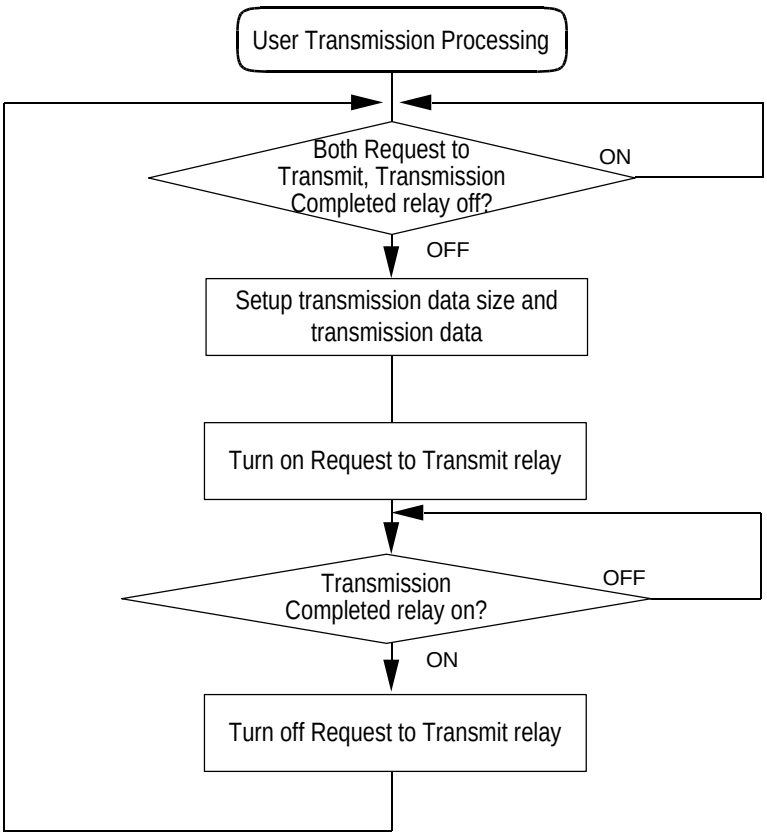
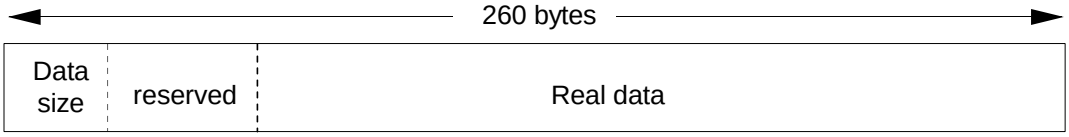


Figure 5.5 Procedure for Transmitting Optional Information

5.3.2 Data Format

Set the data according to the following format.



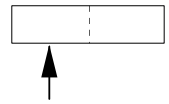
Channel number 7185 7186 7187

Data Size : specifies the length of the data to transmit. Up to 256 bytes of data can be specified. If more than 256 bytes are specified, only 256 bytes will be transmitted.

Reserved : Any data set in this field will be ignored.

Real data : Write the data to transmit using a WRITE instruction, specifying the channel number as 7185. Set the real data such that its length is equal to the "Data Size" specified above.

Left-justify the data.
When transmitting an odd number of bytes, the last byte should be in the high-order position.



If the last data byte is $31 when transmitting an odd number of bytes, set the value to $3100.

5.3.3 Relay Number

A request for optional information transmission is issued by turning on an output relay of the NX Interface Module. Completion of the transmission of the optional information is indicated by an input relay.

Output Relay (Request to Transmit)	Input Relay (Completion Notification)
Ylmm61	Xlmm29

l : unit number
mm : slot number

5.3.4 Exit Code

There is no exit code.

5.3.5 Transmission Example

An example for transmitting 4 bytes of data ($12345678) (with NX Interface Module mounted in slot number 4) is given below.

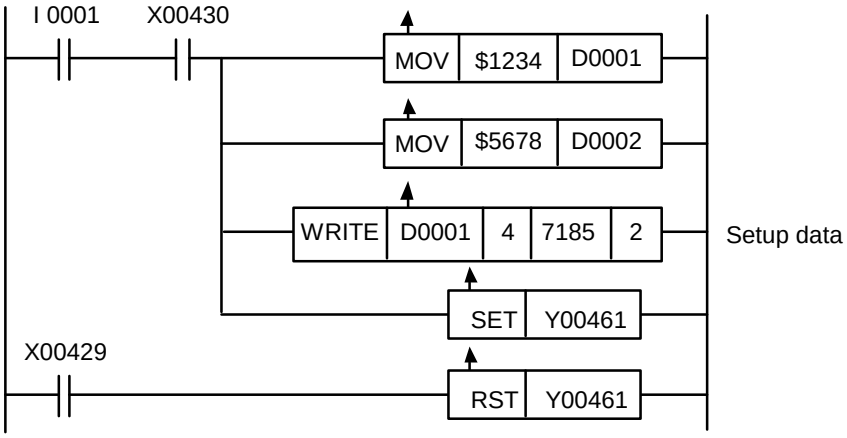


Figure 5.6 Sample Program for Optional Information Transmission

CAUTION

When using a multi-CPU system, note the following points:

The input/output relays of an NX Interface Module can be accessed from only one CPU module; the “Use/Not used” setting in the CPU configuration must be set accordingly. To access the input/output relays from multiple CPUs, use shared relays.

Other data areas (e.g. transmission and receiving buffers and initialization data area) can be accessed from multiple CPUs.

5.4 Changing Node Mode

5.4.1 Procedure for Changing Node Mode

Changing the node mode to online mode is enabled only when the node is a test node and set to receive online/test messages at startup. In this case, the Status relay will be in “test” (on) state at system startup.

To request for changing to online mode, turn on the Change Mode relay.

Note that changing to online mode will reset all input relays for message transmission or receiving to 0, invalidating all (if any) requests for transmission and receiving which are being processed. Check that the Status relay has been turned off before start online operations.

Note that turning off the Change Mode relay will start revert processing described in Section 5.4.2 “Procedure for Reverting”.

5.4.2 Procedure for Reverting

After online operation is started using the procedure described in the above subsection, you may again revert to test mode, if required.

To do that, turn off the Change Mode relay.

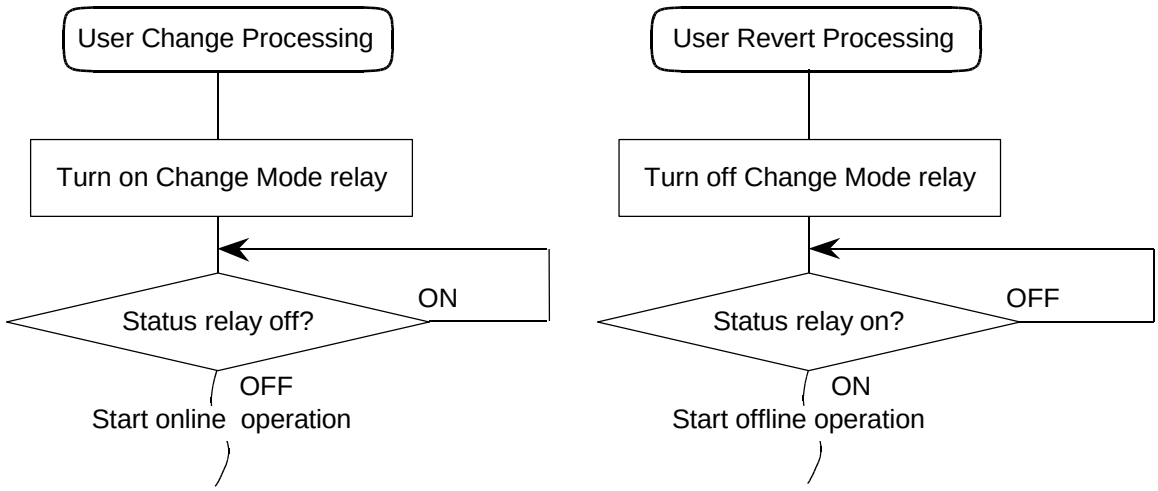


Figure 5.7 User Revert Processing

5.4.3 Relay Number

A request to change to online mode is issued by turning on an output relay of the NX Interface Module. The status after the change is indicated in an input relay.

Output Relay (change request)	Input Relay (status)
Ylmm64	Xlmm32

l : unit number
mm : slot number

5.4.4 Exit Code

There is no exit code.

5.4.5 Example for Changing Mode

(NX Interface Module is mounted in slot number 4)

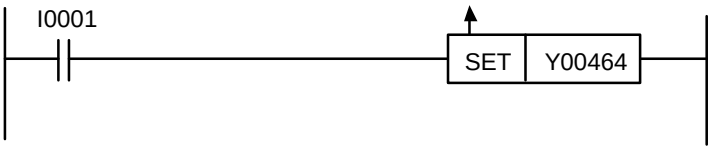


Figure 5.8 Sample Program for Changing Mode

CAUTION

When using a multi-CPU system, note the following points:

The input/output relays of an NX Interface Module can be accessed from only one CPU module; the “Use/Not used” setting in the CPU configuration must be set accordingly. To access the input/output relays from multiple CPUs, use shared relays.

Other data areas (e.g. transmission and receiving buffers and initialization data area) can be accessed from multiple CPUs.

5.5 Starting and Stopping the System

5.5.1 Startup Procedure

To startup NeXUS, set up the initialization data and turn on the Request to Initialize relay. The channel numbers for setting the initialization data start from 7315.

If system startup completes normally, the Initialization Completed relay turns on. Check that Initialization Completed relay has turned on before starting online operations.

If the initialization data is invalid or a system error is detected, the Initialization Error Exit relay turns on and the error reason detected is stored in the error report area. If this happens, remove the error reason and perform initialization again. To request for re-initialization, temporarily turn off the Request to Initialize relay turned on earlier, check that the Initialization Error Exit relay also turns off before performing initialization again.

5.5.2 Procedure for Stopping and Restarting the System

Turning off the Request to Initialize relay which has been turned on earlier is equivalent to a request to stop the system. As described in the earlier section, you can also perform re-initialization even after an error exit when the Request to Initialize relay is not turned off by NeXUS. To restart the system after the Request to Initialize relay has turned off, check that the Initialization Completed relay has also turned off before turning on the Request to Initialize relay.

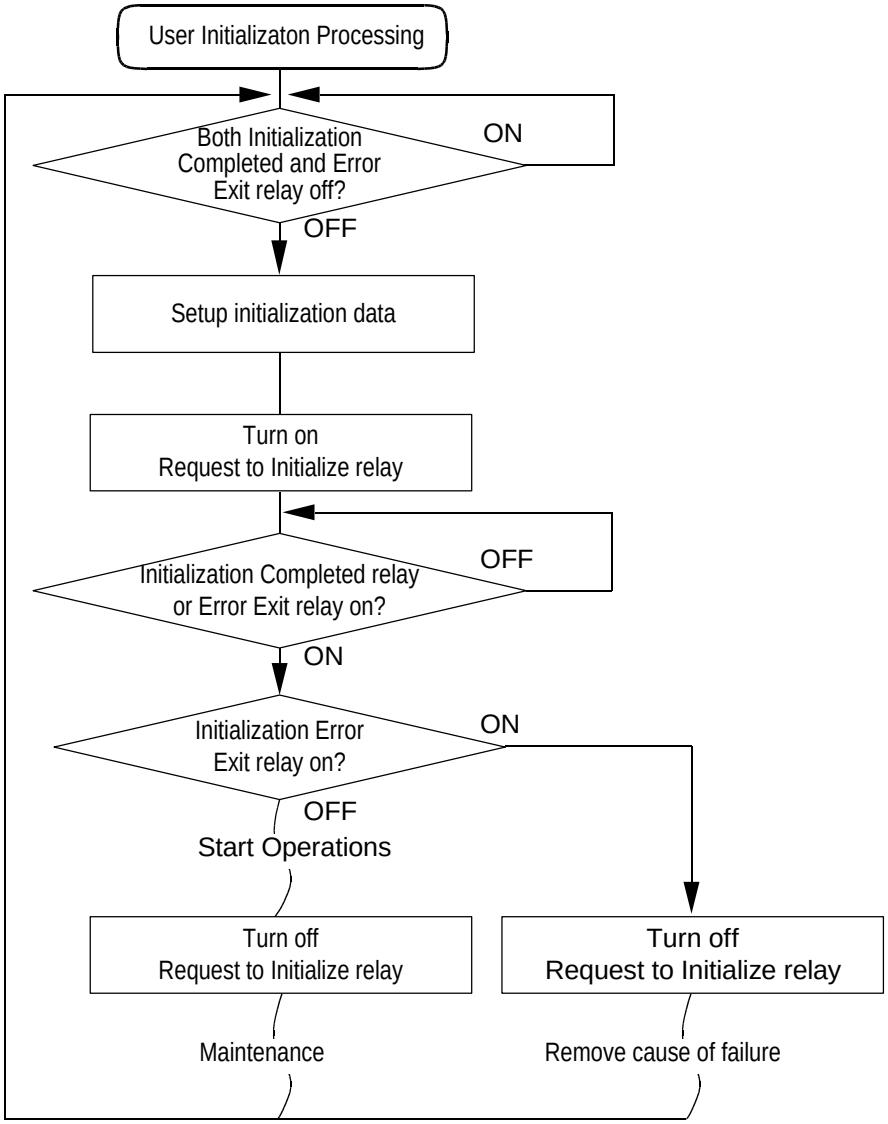


Figure 5.9 Procedure for Stopping and Restarting the System

5.5.3 Data Format

For details on the format of the initialization data, see Section 4.2, “Initialization”.

This section describes the minimal initialization data for the following system.

> Only the sequence CPU is connected.
> Active signals are transmitted at 10-second intervals.
> Transmission is via transmission port number 1025, destination port number is 600.
> Message transmission and receiving is not enabled.

The minimal initialization data format defines only 3 data items as shown below and uses default values for all the remaining data items. By “default”, we mean specifying a value of zero for the data item, the semantics of which depends on each control information. (For details, see “Initialization”)

At powering on, NeXUS resets the initialization data area to 0 and hence, you can operate with the minimal configuration shown above by setting only the following information.

Channel Number	Description	Setting Data
7315	Node name	10-character ASCII string
7320	Node number	1 to 4095
7321	Local data field number	1 to 255

5.5.4 Relay Numbers

An initialization request is issued by turning on an output relay of the NX Interface Module.

Completion of the initialization is indicated in an input relay.

Output relay (request to Initialize)	Input Relay (completion notification)
Ylmm62	Xlmm30 (when normal)
	Xlmm31 (on error)

l : unit number
mm : slot number

5.5.5 Exit Code

An exit code is stored in the error report area when the Initialization Error Exit relay is turned on.

The structure of the error report area is shown below. For details on the exit code, see Appendix A, “Error Code System”

Channel Number	Description
7607	Module number where error was detected (For error detected during initialization, this is always 0x0100)
7608	Detailed exit code (See Appendix A, “Error Code System”)

5.5.6 Initialization Example

Node name = “MITAKA ($4D4954414B41)”

Node number = 150

Local data field number = 2

(NX Interface Module is mounted in slot number 4)

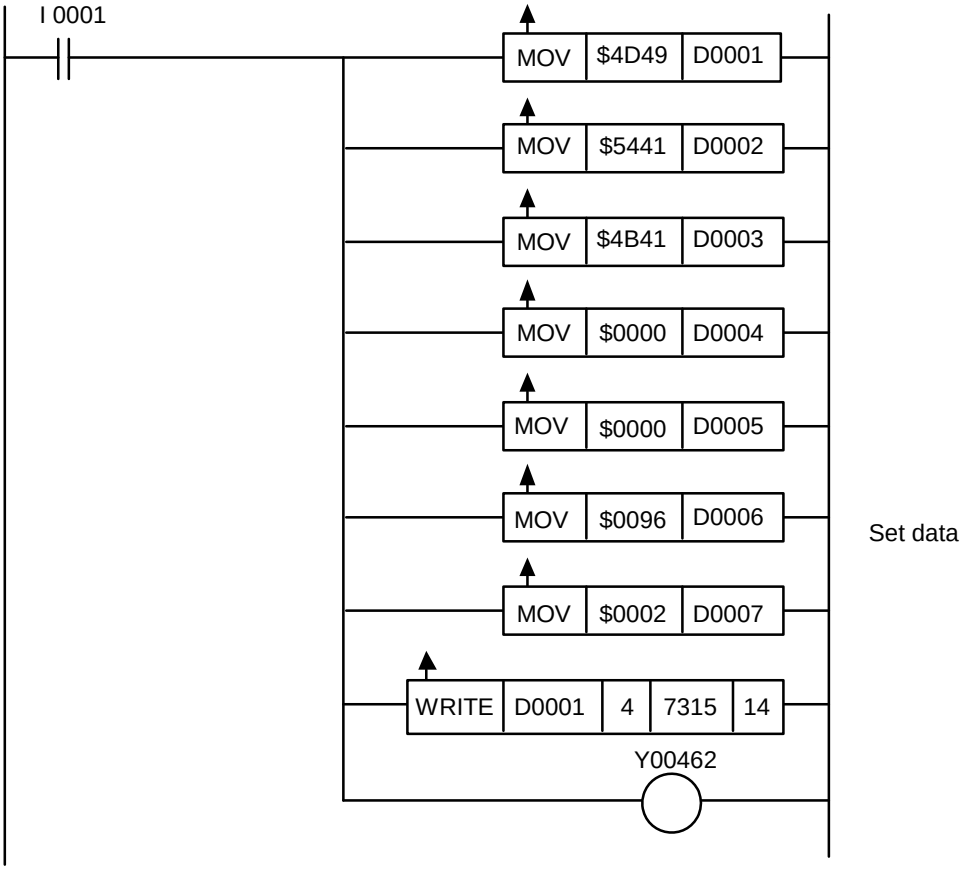


Figure 5.10 Sample Program for Setting the Initialization Data

CAUTION

When using a multi-CPU system, note the following points:

The input/output relays of an NX Interface Module can be accessed from only one CPU module; the “Use/Not used” setting in the CPU configuration must be set accordingly. To access the input/output relays from multiple CPUs, use shared relays.

Other data areas (e.g. transmission and receiving buffers and initialization data area) can be accessed from multiple CPUs

Blank Page

Appendix A. Error Code System

NeXUS detects two types of failure information: failure due to incompatibility with the communication counterpart and failure due to errors in the user initialization data.

When NeXUS detects a failure of the first type, it logs an error code in the system trace area, attaches the error code to an alive signal and transmits it. In this case, it does not send a notification to the ladder program in the sequence CPU and processing continues.

When NeXUS detects a failure of the second type, it stores a detailed error code in the error report area and stops processing (waiting for the user to turn off the Request to Initialize relay). In this case, NeXUS cannot transmit an alive signal so it notifies the ladder program in the sequence CPU module that initialization has exited with error instead (by turning on the Initialization Error Exit relay).

The detailed error codes for these two types of failure are shown in the following pages under separate sections for alive signal reported failure and initialization failure.

■ Alive Signal Reported Failure

The meaning of the error codes generated when an error occurs is described below. The format of the error codes is the format of “ERR_LIST” shown in the format of failure information in Section 3.3, “Managing Failure Information”.

ERR_LIST Format

0 15 23 31 (bit)

Failure source Function Module No.	Error type	Detailed code

Details for each type of error is described below.

● Received message discarded due to invalid header (Error type=$02)

This error type means that a message with an invalid header has been received and discarded.

- Failure source function module number : always 0
- Detailed code:

Detailed Code	Description	Contents
$01	Invalid received message size	Receiving data size is out of the valid range (0 to RBUFSZ*)
$02	Invalid Type in header	Type is not equal to ASCII string “NUXM”
$03	Invalid ML in header	Received message size and ML value are different
$04	Invalid DA in header	The transmission destination DF and DFNO* are different
$05	Invalid M_CTL in header	M_CTL ≠ $80000000
$06	Invalid MODE in header	MODE and port type (ONLINE/TEST) are different
$07	Invalid PVER in header	PVER ≠ 1
$08	Invalid CBN in header	CBN ≠ 1
$09	Invalid TBN in header	TBN ≠ 1
$0A	Invalid BSIZE in header	BSIZE ≠ ML
$0B	Invalid DA in header	Transmission destination MCNO and RCVMGN* are different

* These are the mnemonic names at initialization.

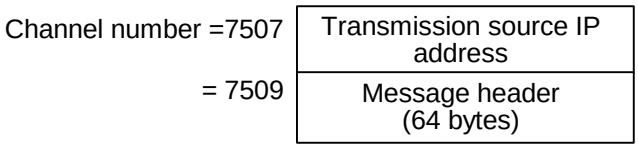
This type of failure may result because data that should be consistent throughout the data field is inconsistent or that data violates the protocol of the transmission source.

In the case of protocol violation, both the transmission source address and the transaction code are uncertain and only an error code indicating that a message is discarded due to a protocol violation is attached to an alive signal and transmitted.

However, the transmission source IP address and the message header (64 bytes) of the message for which protocol violation was first detected are stored in channel number 7507 onwards and can be read and checked by the user.

● Storage Area for the Protocol Violation Message

The message format is shown below.



The above information is always stored in the storage area when an error is detected but the first 4 bytes, once written, are never overwritten until they are cleared to zeroes. The user can reset the first 4 bytes of the storage area to 0, as required, by using any of the following methods:

- Procedure for clearing storage area to zeroes -

1. Restart the system
2. Change the node mode of the system
3. Clear channel numbers 7507 and 7508 to zeroes.

● Function Module Exit Error (Error Type = $03)

This error type indicates that a failure has occurred in a function module (= CPU module)

- Failure source function module number: function module number (=CPU module number)
- Detailed code:

Detailed Code	Description	Contents
$01	CPU access timeout	CPU access did not exit within the stipulated time
$02	CPU access error	CPU access exited with error
$03	CPU error	ERR LED is lit

■ Initialization Failure

If an error is detected during initialization, NeXUS turns on the Initialization Error Exit relay and stores the error code listed below in the detailed exit code (channel number 7608) of the error report area.

● Node Information Errors

Error Number	Description	Contents	Corrective Action
$0101	Invalid node name	Node name is NULL or is not terminated by the NULL character.	Correct the node name
$0102	Invalid logical node number	Logical node number is out of the valid range (1 to 4095)	Correct the logical node number
$0103	Invalid local DF number	Local DF number is out of the valid range (1 to 255)	Correct the local DF number
$0104	Invalid CPU number	CPU number is out of the valid range (1: CPU1, 2: CPU2, 4: CPU3, 8: CPU4)	Correct the CPU number
$0105	Invalid Transmission handle count	Transmission handle count is out of the valid range (0 to 16)	Correct the transmission handle count.
$0106	Invalid receiving handle count	The receiving handle count is out of the valid range (0 to 12)	Correct the receiving handle count
$0107	Invalid alive signal transmission interval.	Alive signal transmission interval is out of the valid range of 1 to 3600 seconds (1 hour) or is shorter than the alive signal timeout period.	Correct the alive signal transmission interval.
$0108	Invalid alive signal timeout monitoring interval.	Alive signal timeout monitoring interval is out of the valid range of 1 to 43200 seconds (12 hours).	Correct the alive signal timeout monitoring interval.
$0109	Invalid high-order byte of node mode	The high-order byte of the node mode is out of the valid range ($00: online, $80: test).	Correct the node mode.
$010A	Invalid transmission buffer case size	The transmission buffer case size is out of the valid range (00: 256 bytes, $01: 512 bytes, $02: 1024 bytes).	Correct the transmission buffer case size.
$010B	Invalid local DF transmission MGN count.	The local DF transmission MGN count is out of the valid range (0 to 4).	Correct the local DF transmission MGN count.
$010C	Invalid remote DF number.	The remote DF number is out of the valid range (0 to 255).	Correct the remote DF number.
$010D	Invalid remote DF transmission multicast registration count.	The remote DF transmission multicast registration count is out of the valid range (0 to 4).	Correct the remote DF transmission multicast registration count.
$010E	Invalid low-order byte of node mode	The low-order byte of the node mode is out of the valid range ($00: online, $01: test, $02: online/test).	Correct the low-order byte of the node mode.
$010F	Invalid data receiving buffer size.	The data receiving buffer size is out of the valid range ($00: 256 bytes, $01: 512 bytes, $02: 1024 bytes).	Correct the data receiving buffer size.
$0110	Invalid local DF receiving MGN count.	The local DF receiving MGN count is out of the valid range (0 to 4).	Correct the receiving MGN count for the local DF.
$0111	Invalid total receiving and transmission buffer size.	The transmission and receiving buffer case sizes and the transmission and receiving handle registration counts, on calculation, produce a total receiving and transmission buffer size exceeding 14 kilobytes.	Correct the transmission and receiving buffer case sizes and the transmission and receiving handle registration counts.
$0113	Invalid broadcast address for the local DF.	The broadcast address for the local DF is specified as $FFFFFFFF or the network address of the broadcast address and the IP address are different.	Correct the broadcast address.
$0114	Invalid broadcast address for the remote DF.	The broadcast address for the remote DF is specified as $00000000 or $FFFFFFFF.	Correct the broadcast address.

● Transmission Handle Information Errors

Error Number	Description	Contents	Corrective Action
$0121	Invalid transmission transaction code.	The transmission transaction code is out of the valid range (1 to 59999).	Correct the transmission transaction code.
$0122	Invalid transmission multicast group number. (undefined)	The transmission multicast group number does not match that defined in the transmission multicast group information for the local/remote DF.	Correct the transmission multicast group number.
$0123	Invalid transmission multicast group number. (out of range)	The transmission multicast group number is out of the valid range (1 to 255).	Correct the transmission multicast group number.
$0124	Invalid DF identifier.	The DF identifier is out of the valid range ($00: local DF, $80: remote DF).	Correct the DF identifier.
$0125	Invalid transmission data size.	The transmission data size is out of the valid range (0 to the transmission buffer case size).	Correct the transmission data length.

● The Receiving Handle Information Errors

Error Number	Description	Contents	Corrective Action
$0131	Invalid receiving transaction code.	The receiving transaction code is out of the valid range (1 to 59999).	Correct the receiving transaction code.
$0115	Invalid receiving transaction code.	Duplicate definition of the receiving transaction code.	Correct the receiving transaction code.

● The Transmission Multicast Group Information Error (for local data field)

Error Number	Description	Contents	Corrective Action
$0141	Invalid transmission MCG number	The transmission MCG number is out of the valid range (1 to 255).	Correct the transmission MCG number.
$0142	Invalid port number for online mode	The port number for online mode is undefined.	Define the port number.
$0143	Invalid port number for test mode	The port number for test mode is undefined.	Define the port number.

● The Transmission Multicast Group Information Error (for remote data field)

Error Number	Description	Contents	Corrective Action
$0151	Invalid transmission MCG number	The transmission MCG number is out of the valid range (1 to 255).	Correct the transmission MCG number.
$0152	Invalid port number for online mode.	The port number for online mode is undefined.	Define the port number.
$0153	Invalid port number for test mode.	The port number for test mode is undefined.	Define the port number.

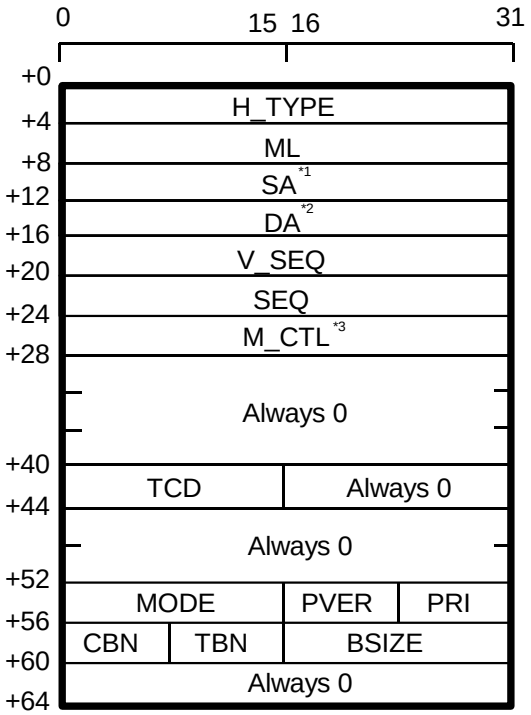
● The Receiving Multicast Group Information Error

Error Number	Description	Contents	Corrective Action
$0161	Invalid receiving MCG number	The receiving MCG number is out of the valid range (1 to 255).	Correct the receiving MCG number.
$0162	Invalid port number for online mode.	The port number for online mode is undefined.	Define the port number.
$0163	Invalid port number for test mode.	The port number for test mode is undefined.	Define the port number.
$0112	Invalid receiving port number.	Duplicate definition of the port number.	Define the port number.

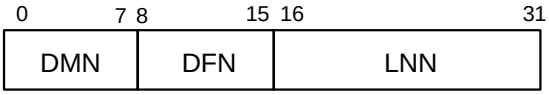
● Routing Information Error

Error Number	Description	Contents	Corrective Action
$0171	Invalid own IP address	The class of the own IP address is neither A, nor B, nor C.	Correct the own IP address.
$0172	Invalid net mask.	The class of the net mask is neither A, nor B, nor C.	Correct the net mask.
$0173	Invalid default gateway IP address.	The class of the default gateway IP address is different from the network address of the own IP address.	Correct the default gateway IP address.

Appendix B. Format of NeXUS Protocol Header

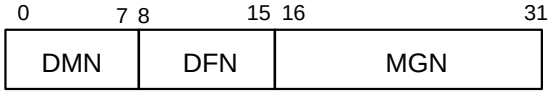


*1: SA



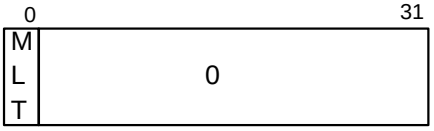
DMN : Domain number (Currently always 0)
DFN : Data field number
LNN : Logical node number

*2: DA



DMN : Domain number (Currently always 0)
DFN : Data field number
MGN : Logical node number

*3: M_CTL



MLT : Multicast flag

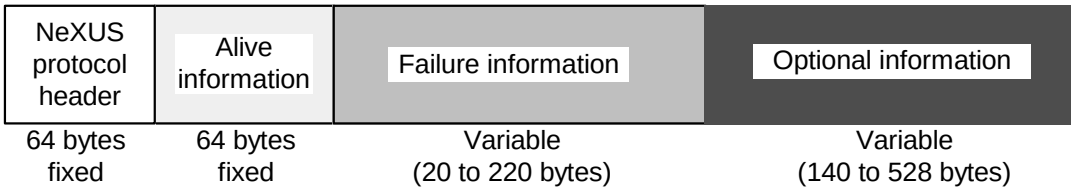
No.	Mnemonic	Size	Description
1	H_TYPE	4	Header type. Set to ASCII string “NUXM”. Used to differentiate from other protocol messages.
2	ML	4	Message length. Combined length of message header and user data in bytes.
3	SA	4	Message transmission source address.
4	DA	4	Message transmission destination address.
5	V_SEQ	4	Version number of the message transmission serial number. Set to time when the message transmission serial number is initialized.
6	SEQ	4	Message transmission serial number. Valid values run from $00000001 to $7FFFFFFF and back to $00000001
7	M_CTL	4	Message transmission control information.
8	TCD	2	Transaction code
9	MODE	2	Message mode. (0: online mode, 1: test mode)
10	PVER	1	NeXUS protocol version number (Always 1).
11	PRI	1	Priority level of message (Always 0)
12	CBN	1	Current fragment block number. (Always 1)
13	TBN	1	Total number of fragment blocks (Always 1)
14	BSIZE	2	Allocation block size (this is the allocation size at fragmentation, including the header size)

Blank Page

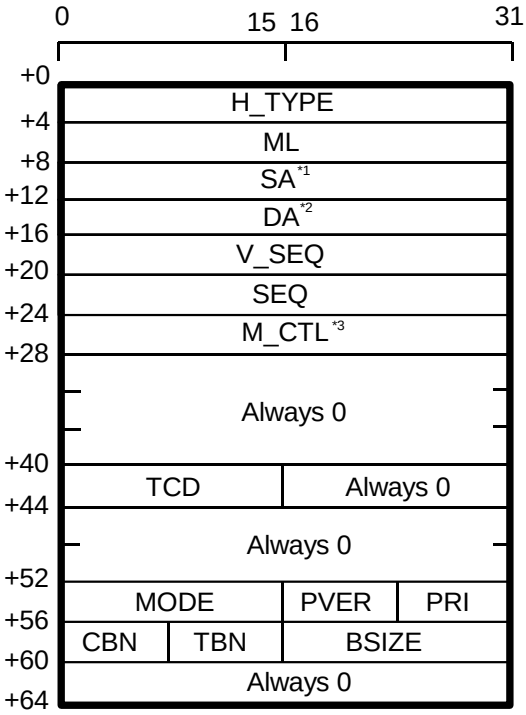
Appendix C. Format of Alive Signal Messages

An alive signal consists of alive information, failure information and optional information, as shown in the figure below. Details for each component are shown separately in the following pages.

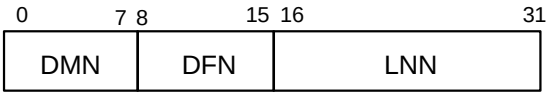
Alive signal message



■ NeXUS Protocol Header

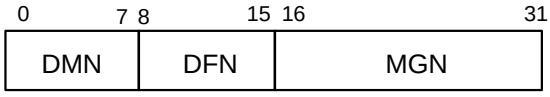


*1: SA



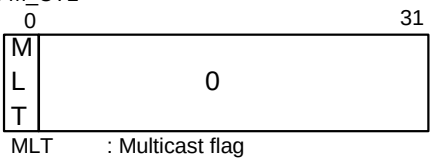
DMN : Domain number (Currently always 0)
DFN : Data field number
LNN : Logical node number

*2: DA



DMN : Domain number (Currently always 0)
DFN : Data field number
MGN : Logical node number

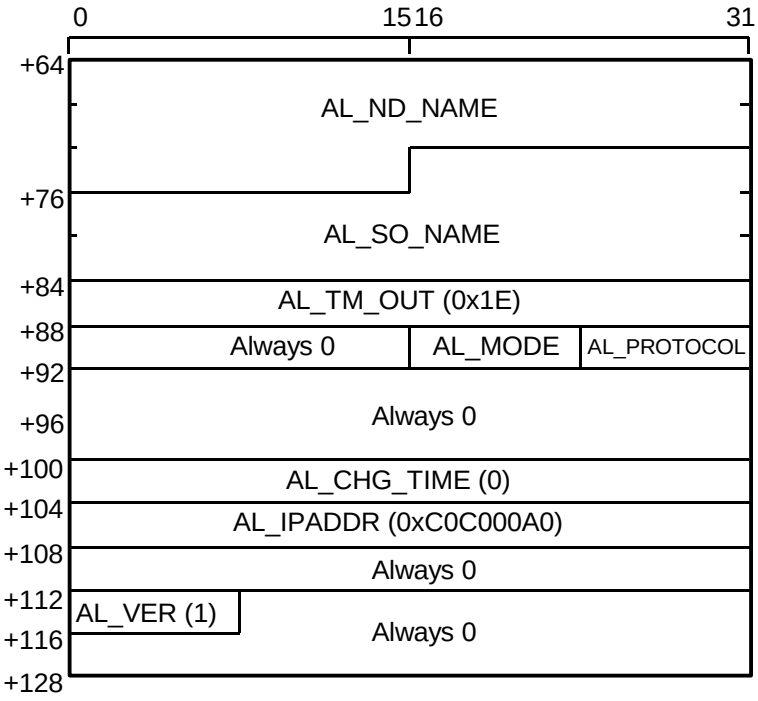
*3: M_CTL



MLT : Multicast flag

No.	Mnemonic	Size	Description
1	H_TYPE	4	Header type. Set to ASCII string “NUXM”. Used to differentiate from other protocol messages.
2	ML	4	Message length. Combined length of the message header and other data in bytes.
3	SA	4	Message transmission source address.
4	DA	4	Message transmission destination address.
5	V_SEQ	4	Message transmission sequence version number.
6	SEQ	4	Message transmission sequence number. Valid values run from $00000001 to $7FFFFFFF and back to $00000001
7	M_CTL	4	Message transmission control information.
8	TCD	2	Transaction code (Always TCD=60003)
9	MODE	2	Message mode. (0: online mode, 1: test mode)
10	PVER	1	NeXUS protocol version number (Always 1).
11	PRI	1	Priority level of message (Always 1)
12	CBN	1	Current fragment block number. (Always 1)
13	TBN	1	Total number of fragment blocks (Always 1)
14	BSIZE	2	Allocation block size (this is the allocation size at fragmentation, including the header size)

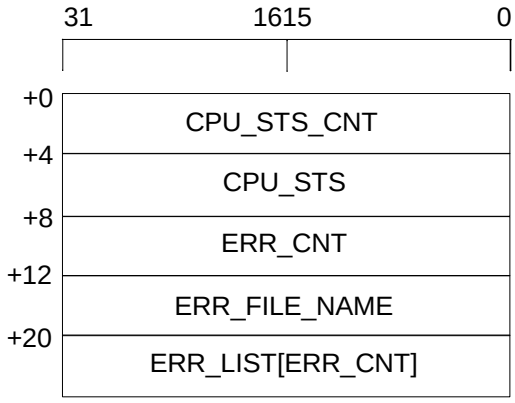
■ Alive Information



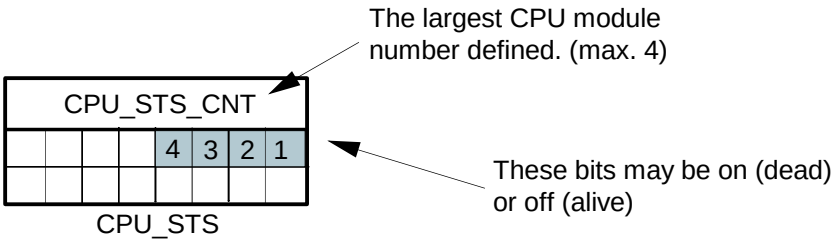
No.	Mnemonic	Size	Description
1	AL_ND_NAME	10	Node name (10 ASCII characters)
2	AL_SO_NAME	10	Vendor equipment name (YE_FA_M3)
3	AL_TM_OUT	4	Alive signal monitoring timeout (seconds)
4	AL_MODE	2	Alive signal mode*1
5	AL_PROTOCOL	1	Protocol type (Always 4)
6	AL_CHG_TIME	4	Time node mode was changed *2
7	AL_IPADDR	4	IP address
8	AL_VER	1	Alive signal message version number (Always 1)

*1 AL_MODE = 0: normal mode
= 1: shutdown notification mode
*2 AL_CHG_TIME ≠ 0: System startup time. If AL_MODE=0, the value is always non-zero.
= 0: Always transmitted with value 0 if AL_MODE=1.

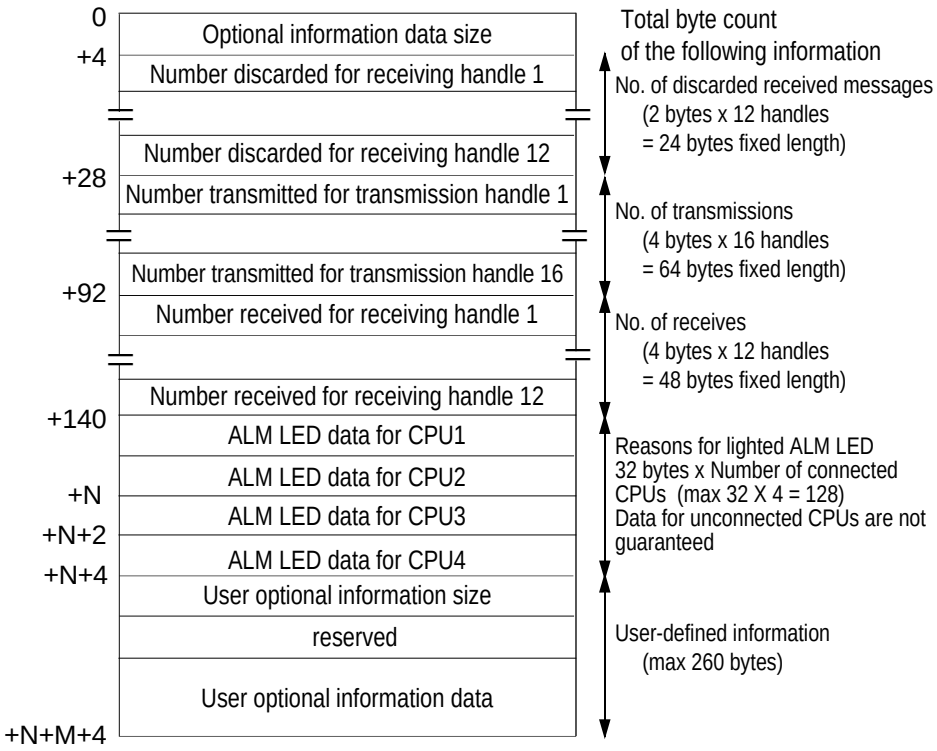
■ Failure Information



Signal Name	Size	Description
CPU_STS_CNT	4	Number of modules (CPU module) reported
CPU_STS	4	Module alive information (Bitmap of 0s (alive) and 1s (dead) for up to 4 modules)
ERR_CNT	4	Number of error information blocks (Number of valid items in AL_ERR_LIST)
ERR_FILE_NAME	4	Name of error search file (ASCII string constant “NX01”)
ERR_LIST(ERR_CNT)	4×N	Function module number and error code of the error source (2 bytes each). The maximum value for N is 50.



■ Optional Information



N: 172 when only CPU 1 is connected;
204 when CPU1 and CPU2 are connected;
236 when CPU1, CPU2 and CPU3 are connected;
268 when CPU1, CPU2, CPU3 and CPU4 are connected.
M: User optional information size (256 bytes max.)

Blank Page

Appendix D. Saving Error Message File (with NXViewer installed)

By registering an error message file in NXViewer, failure information (only for alive signal reported failure information, not for initialization failure information. See Appendix A for details) transmitted from FA-M3 can be displayed as error messages, instead of error numbers.

The name of the error message file is “NX01.MSG”.

The directory where the file is stored is given below.

Home directory of NeXUS\MSG\YE_FA_M3 \NX01.MSG

The home directory of NeXUS is defined in the “HomeDir” entry of the [Home Directory] section in the “WINDOWS\NXDDE.INI” file on the system disk (drive). It is usually set to “C:\NEXUS”.

Contents of Error Message File “NX01.MSG”

No: 257 E E NODENAME
Msg: Error in node name
Cause: Node name is NULL or is not terminated by the NULL character.
Action: Correct the node name
No: 258 E E LNN
Msg: Error in logical node number
Cause: Logical node number is out of the valid range (1 to 4095).
Action: Correct the logical node number.
No: 259 E E LOCAL DFNO
Msg: Error in local DF number
Cause: Local DF number is out of the valid range (1 to 255).
Action: Correct the local DF number.
No: 260 E E CPUNO
Msg: Error in CPU number
Cause: CPU number is out of the valid range (0, 1: CPU1, 2: CPU2, 4: CPU3, 8: CPU4).
Action: Correct the CPU number.
No: 261 E E SNDHD
Msg: Error in Transmission handle count.
Cause: Transmission handle count is out of the valid range (0 to 16).
Action: Correct the transmission handle count.
No: 262 E E RCVHD
Msg: Error in the receiving handle count.
Cause: The receiving handle count is out of the valid range (0 to 12).
Action: Correct the receiving handle count.
No: 263 E E ALIVE SND TIM
Msg: Error in the alive signal transmission interval.
Cause: Alive signal transmission interval is out of the valid range (0 to 3600) or is shorter than the alive signal timeout period.
Action: Correct the alive signal transmission interval.
No: 264 E E TIMOUT
Msg: Error in the alive signal timeout interval.
Cause: Alive signal timeout interval is out of the valid range (0 to 43200)
Action: Correct the alive signal timeout interval.
No: 265 E E NODMD
Msg: Error in node mode.
Cause: Node mode is out of the valid range (0: online, $80: test).
Action: Correct the node mode.
No: 266 E E SND BUF SIZE
Msg: Error in the message transmission buffer size.
Cause: The message transmission buffer size is out of the valid range (0: 256, 1: 512, 2: 1024 bytes).
Action: Correct the message transmission buffer size.

No: 267 E E SND LOCAL DF MCGNO
Msg: Error in the local DF transmission MCG count.
Cause: The local DF transmission MCG count is out of the valid range (0 to 4).
Action: Correct the local DF transmission MCG count.
No: 268 E E REMOTE DFNO
Msg: Error in the remote DF number.
Cause: The remote DF number is out of the valid range (1 to 255).
Action: Correct the remote DF number.
No: 269 E E SND REMOTE DF MCGNO
Msg: Error in the remote DF transmission MCG count.
Cause: The remote DF transmission MCG count is out of the valid range (1 to 4).
Action: Correct the remote DF transmission MCG count.
No: 270 E E RCV MD
Msg: Error in the message receiving mode.
Cause: The message receiving mode is out of the valid range (0: online, 1: test, 2: both online and test)
Action: Correct the message receiving mode.
No: 271 E E RCV BUF SIZE
Msg: Error in the message receiving buffer size.
Cause: The message receiving buffer size is out of the valid range (0: 256, 1: 512, 2: 1024 bytes).
Action: Correct the message receiving buffer size.
No: 272 E E RCV LOCAL DF MCGNO
Msg: Error in the local DF receiving MCG count.
Cause: The local DF receiving MCG count is out of the valid range (1 to 4).
Action: Correct the local DF receiving MCG count.
No: 273 E E SND RCV BUFSZ
Msg: Error in the total receiving and transmission buffer size.
Cause: The calculated total receiving and transmission buffer size exceeds 14 kilobytes.
Action: Correct the data so that the calculated total receiving and transmission buffer size does not exceed 14 kilobytes.
No: 274 E E RCVPTNO MUT
Msg: Error in the message receiving port number.
Cause: Duplicate definition of the message receiving port number.
Action: Correct the message receiving port number.
No: 275 E E LOCAL DF BC
Msg: Error in the BC address for the local DF.
Cause: The BC address for the local DF is specified as $FFFFFFFF or the class of the specified address differs from that of its own IP address.
Action: Correct the BC address for the local DF.
No: 276 E E REMOTE DF BC
Msg: Error in the BC address for the remote DF.
Cause: The BC address for the remote DF is specified as $00000000 or $FFFFFFFF.
Action: Correct the BC address for the remote DF.

No: 277 E E RCVTCDNO MUT
Msg: Error in the message receiving TCD number.
Cause: Duplicate definition of the message receiving TCD number.
Action: Correct the message receiving TCD number.
No: 289 E E SM TCD
Msg: Error in the transmission TCD number.
Cause: The transmission TCD number is out of the valid range (1 to 65399).
Action: Correct the transmission TCD number.
No: 290 E E SM MCG UNK
Msg: Error in the transmission MCG number.
Cause: The transmission MCG number is undefined (in transmission MCG information).
Action: Correct the transmission MCG number.
No: 291 E E SM MCG OVR
Msg: Error in the transmission MCG number.
Cause: The transmission MCG number is out of the valid range (1 to 255).
Action: Correct the transmission MCG number.
No: 292 E E SM DFNO
Msg: Error in the transmission DF type.
Cause: The transmission DF type is out of the valid range (0: local, $80: remote).
Action: Correct the transmission DF type.
No: 293 E E SM SND MSGLEN
Msg: Error in the transmission message length.
Cause: The transmission message length is out of the valid range (0 to the specified transmission buffer size).
Action: Correct the transmission message length.
No: 305 E E RM TCD
Msg: Error in the receiving TCD number.
Cause: The receiving TCD number is out of the valid range (1 to 65399).
Action: Correct the receiving TCD number.
No: 306 E E RM CPUNO
Msg: Error in the CPU number.
Cause: The CPU number is out of the valid range (1 to 4).
Action: Correct the CPU number.
No: 321 E E SLB MCG
Msg: Error in the local DF transmission MCG number.
Cause: The local DF transmission MCG number is out of the valid range (1 to 255).
Action: Correct the local DF transmission MCG number.
No: 322 E E SLB ONLINE PTNO
Msg: Error in the local DF transmission online port number.
Cause: The local DF transmission online port number is undefined.
Action: Correct the local DF transmission online port number.

No: 323 E E SLB TEST PTNO
Msg: Error in the local DF transmission test port number.
Cause: The local DF transmission test port number is undefined.
Action: Correct the local DF transmission test port number.
No: 337 E E SRB MCG
Msg: Error in the remote DF transmission MCG number.
Cause: The remote DF transmission MCG number is out of the valid range (1 to 255).
Action: Correct the remote DF transmission MCG number.
No: 338 E E SRB ONLINE PTNO
Msg: Error in the remote DF transmission online port number.
Cause: The remote DF transmission online port number is undefined.
Action: Correct the remote DF transmission online port number.
No: 339 E E SRB TEST PTNO
Msg: Error in the remote DF transmission test port number.
Cause: The remote DF transmission test port number is undefined.
Action: Correct the remote DF transmission test port number.
No: 353 E E RLB MCG
Msg: Error in the local DF receiving MCG number.
Cause: The local DF receiving MCG number is out of the valid range (1 to 255).
Action: Correct the local DF receiving MCG number.
No: 354 E E RLB ONLINE PTNO
Msg: Error in the local DF receiving online port number.
Cause: The local DF receiving online port number is undefined.
Action: Correct the local DF receiving online port number.
No: 355 E E RLB TEST PTNO
Msg: Error in the local DF receiving test port number.
Cause: The local DF receiving test port number is undefined.
Action: Correct the local DF receiving test port number.
No: 369 E E RT CLASS
Msg: Error in the own IP address.
Cause: The class of the own IP address is neither A, nor B, nor C.
Action: Correct the own IP address.
No: 370 E E RT NETMSK
Msg: Error in the network mask.
Cause: The network mask does not match the class of the own IP address.
Action: Correct the network mask.
No: 371 E E RT DEFIP
Msg: Error in the default IP address.
Cause: The class of the default IP address does not match that of the own IP address.
Action: Correct the default IP address.

No: 372 E E RT DES
Msg: Error in the destination IP address.
Cause: The destination IP address is specified as $00000000 or $FFFFFFFF.
Action: Correct the destination IP address.
No: 373 E E RT GATWAT
Msg: Error in the gateway IP address.
Cause: The gateway IP address does not match the class of the own IP address.
Action: Correct the gateway IP address.
No: 513 E E MSG LEN PDU
Msg: Received message with invalid message length.
Cause: A message with an invalid message length is received.
Action: Contact and check with the message transmission source.
No: 514 E E HTYPE PDU
Msg: Message with invalid H TYPE received.
Cause: Message with invalid H TYPE received.
Action: Contact and check with the message transmission source.
No: 515 E E ML PDU
Msg: Message with invalid ML received.
Cause: Message with invalid ML received.
Action: Contact and check with the message transmission source.
No: 516 E E DF PDU
Msg: Message with invalid DF number received.
Cause: Message with invalid DF number received.
Action: Contact and check with the message transmission source.
No: 517 E E MCTL PDU
Msg: Message with invalid M CTL received.
Cause: Message with invalid M CTL received.
Action: Contact and check with the message transmission source.
No: 518 E E MSGMDL PDU
Msg: Message with invalid message mode received.
Cause: Message with invalid message mode received.
Action: Contact and check with the message transmission source.
No: 519 E E PVER PDU
Msg: Message with invalid PVER received.
Cause: Message with invalid PVER received.
Action: Contact and check with the message transmission source.
No: 520 E E CBN PDU
Msg: Message with invalid CBN received.
Cause: Message with invalid CBN received.
Action: Contact and check with the message transmission source.

No: 521 E E TBN PDU
Msg: Message with invalid TBN received.
Cause: Message with invalid TBN received.
Action: Contact and check with the message transmission source.
No: 522 E E BSIZE PDU
Msg: Message with invalid BSIZE received.
Cause: Message with invalid BSIZE received.
Action: Contact and check with the message transmission source.
No: 523 E E DAMCG PDU
Msg: Message with invalid MCG received.
Cause: Message with invalid MCG within DA received.
Action: Contact and check with the message transmission source.
No: 769 E E LIF TMOUT
Msg: Link IF timeout occurred.
Cause: Timeout occurred in link IF.
Action: Contact and check with the system administrator.
No: 770 E E LIF ABN
Msg: Link IF error exit.
Cause: Link IF exited with error.
Action: Contact and check with the system administrator.
No: 771 E E LIF ERRLED
Msg: ERR LED is lit.
Cause: ERR LED is lit.
Action: Contact and check with the system administrator.
No: 2509 E E RSND ERR
Msg: Remote DF message transmission error
Cause: Message transmission to remote DF exited with error.
Action: Correct the routing information.

Blank Page

Revision Information

Document Name : NX Interface Module
Document No. : IM 34M06H29-01E

Edition	Date	Revised Item
1st	Sep. 2001	New Publication
2nd	Feb. 2011	Added descriptions for F3NX01-1N

Written by PLC Solution Department
Marketing Center
Industrial Automation Business Headquarters
Yokogawa Electric Corporation
Published by Yokogawa Electric Corporation
2-9-32 Nakacho, Musashino-shi, Tokyo, 180-8750, JAPAN
Printed by Kohoku Publishing & Printing Inc.

Blank Page