PROFIBUS-DP Master
NX5001
User Manual

MU214601 Rev. L

January 16, 2024

www.altus.com.br

According to current legislation in Brazil, the Consumer Defense Code, we are giving the following information to clients who use our products, regarding personal safety and premises.

The industrial automation equipment, manufactured by Altus, is strong and reliable due to the stringent quality control it is subjected to. However, any electronic industrial control equipment (programmable controllers, numerical commands, etc.) can damage machines or processes controlled by them when there are defective components and/or when a programming or installation error occurs. This can even put human lives at risk. The user should consider the possible consequences of the defects and should provide additional external installations for safety reasons. This concern is higher when in initial commissioning and testing.

The equipment manufactured by Altus does not directly expose the environment to hazards, since they do not issue any kind of pollutant during their use. However, concerning the disposal of equipment, it is important to point out that built-in electronics may contain materials which are harmful to nature when improperly discarded. Therefore, it is recommended that whenever discarding this type of product, it should be forwarded to recycling plants, which guarantee proper waste management.

It is essential to read and understand the product documentation, such as manuals and technical characteristics before its installation or use. The examples and figures presented in this document are solely for illustrative purposes. Due to possible upgrades and improvements that the products may present, Altus assumes no responsibility for the use of these examples and figures in real applications. They should only be used to assist user trainings and improve experience with the products and their features.

Altus warrants its equipment as described in General Conditions of Supply, attached to the commercial proposals.

Altus guarantees that their equipment works in accordance with the clear instructions contained in their manuals and/or technical characteristics, not guaranteeing the success of any particular type of application of the equipment.

Altus does not acknowledge any other guarantee, directly or implied, mainly when end customers are dealing with third-party suppliers. The requests for additional information about the supply, equipment features and/or any other Altus services must be made in writing form. Altus is not responsible for supplying information about its equipment without formal request. These products can use EtherCAT® technology (www.ethercat.org).

COPYRIGHTS

OPEN SOURCE SOFTWARE NOTICE

To obtain the source code under GPL, LGPL, MPL and other open source licenses, that is contained in this product, please contact opensource@altus.com.br. In addition to the source code, all referred license terms, warranty disclaimers and copyright notices may be disclosed under request.

Contents

1. Introduction

This manual aims to guide the user during products selecting stages for certain application, as well as at installation, programming and maintenance related to the product Master PROFIBUS-DP NX5001, which is part of the Nexto Series of programmable controllers.

1.1. Nexto Series

Nexto Series is a powerful and complete series of Programmable Controllers (PLC) directed to satisfy the necessities of small and middle-sized applications, as well as high level applications. The series has many features and brings the best cost-benefit both for great process automations and machinery automations. Nexto Series is based in a modular system which allows flexibility between performance and cost.

Nexto Series architecture has a great variety of input and output modules. These modules combined with a powerful 32 bits processor and a high speed bus based on Ethernet, fit to several application kinds as high speed control for small machines, complex distributed processes, redundant applications and systems with a great number of I/O as building automation. Furthermore, Nexto Series has modules for motion control, communication modules encompassing the most popular field networks among other features.

Nexto Series uses an advanced technology in its bus, which is based on a high speed Ethernet interface, allowing input and output information and data to be shared between several controllers inside the same system. The system can be easily distributed throughout the whole field, allowing the use of bus expansion with the same performance of a local module, turning possible the use of every module in the local frame or in the expansion frames with no restrictions. For interconnection between frames expansions a simple standard Ethernet cable is used.

Figure 1: Nexto Series – Overview

1.2. Innovative Features

Nexto Series brings to the user many innovations regarding utilization, supervision and system maintenance. These features were developed focusing a new concept in industrial automation.

One Touch Diag: One Touch Diag is an exclusive feature that Nexto Series brings to PLCs. With this new concept, the user can check diagnostic information of any module present in the system directly on CPU's graphic display with one single press in the diagnostic switch of the respective module. OTD is a powerful diagnostic tool that can be used offline (without supervisor or programmer), reducing maintenance and commissioning times.

ETD – Electronic Tag on Display: Another exclusive feature that Nexto Series brings to PLCs is the Electronic Tag on Display. This new functionality brings the process of checking the tag names of any I/O pin or module used in the system directly to the CPU's graphic display. Along with this information, the user can check the description, as well. This feature is extremely useful during maintenance and troubleshooting procedures.

DHW – Double Hardware Width: Nexto Series modules were designed to save space in user cabinets or machines. For this reason, Nexto Series delivers two different module widths: Double Width (two backplane rack slots are required) and Single Width (only one backplane rack slot is required). This concept allows the use of compact I/O modules with a high-density of I/O points along with complex modules, like CPUs, fieldbus masters and power supply modules.

iF Product Design Award 2012: Nexto Series was the winner of iF Product Design Award 2012 in industry + skilled trades group. This award is recognized internationally as a seal of quality and excellence, considered the Oscars of the design in Europe..

1.3. PROFIBUS

Fieldbus networks usage as communication link between Central Processing Units of automation and field devices grow popular every day. Experience has shown that usage of network technologies brings many benefits to installation, configuration and maintenance of wirings in comparison with previous technology. In field networks, only one pair of wires is needed to transmit information, such as input or output data, parameters, diagnostics, programs or power for field devices.

Field networks have been used for quite some time, however the first ones were proprietary and incompatible with each other, with high costs for configuration or interfacing between equipments of different manufacturers. New networks offer open standards, which dismisses projects of complex interfaces. Open systems allow the end user to freely choose the best solution for his application between a wide range of products.

PROFIBUS is the leading fieldbus network in Europe, with great acceptance in the rest of the world. Its application areas include manufacturing, process control and building automation.

PROFIBUS is an open fieldbus, standardized in Europe as EN50170, and internationally recognized as IEC61158 and IEC61784. The most important world manufacturers of automation technology offer PROFIBUS interface for its devices.

1.4. NX5001

PROFIBUS-DP Master NX5001 is an advanced master for interface, developed to be used alongside CPUs from Nexto Series. The module supports the protocol for cyclical data exchange, and can be used in any frame from Nexto Series.

Interface allows access of CPUs from Nexto Series in any field device compatible with this protocol, such as module systems of remote I/Os, sensors, transmitters, actuators, etc.

In addition to the use of interface NX5001 in single PROFIBUS networks, two interfaces NX5001 can be used with CPUs from Nexto Series to implement two independent networks, or even four interfaces NX5001 can be used to implement two independent networks (for further information see System Configuration).

Figure 2: NX5001

1.5. Redundancy

There are two types of redundancies that can be configured for NX5001 module:

- Network Redundancy
- Master Redundancy

1.5.1. Network Redundancy

In redundant networks, each slave device has two network connections, forming a double network, connected to two modules NX5001.

Fieldbus network redundancy is a vital feature where high reliability is required. NX5001 network interface is the solution for this type of application, being used in pairs.

In this type of redundancy, the slave device, by having two connections, chooses through which network it will receive and transmit its data. Examples of redundant devices are Altus' PROFIBUS network heads NX5210, PO5063V5 and PO5065.

Each pair of redundant heads controls a bus of Nexto or Ponto Series I/O modules, alternately. One of the redundant heads is communicating through the network (active) and the other one is in stand-by. The stand-by head can take control of the bus if there is a failure in the network of the active head or in its hardware. This change is automatic and transparent to users, keeping the system in operation in case of failure in one of the networks.

At the CPU that controls the network, the NX5001 interfaces manage incoming data from the network, so that only the inputs from the active head are copied to the CPU's variables, while inputs of the stand-by head are ignored. The outputs are sent to the two heads (active and stand-by), but only the active head writes in the output modules.

The CPUs are informed through NX5001 interfaces which head is active on each network node, and if there is any defective device.

In this type of network, hot reconfiguration is allowed, reconfiguring one network while the other remains in operation and vice-versa.

It should be noted that the network can continue operating normally when there are defects in some heads connected to PROFIBUS A network, as well as defects in other heads connected to PROFIBUS B network, as long as both heads from the same group of remote I/O have not failed. In this case, communication with the system of remote I/O is partly distributed between network A and network B.

1.5.2. Master Redundancy

Master redundancy is characterized by the existence of two PROFIBUS-DP NX5001 masters on the same network, where one NX5001 acts as active master and the other as passive master.

PROFIBUS masters in active mode establish connection with the slaves. Its network operation status is OPERATE.

Enabled PROFIBUS masters in passive mode are used to test the PROFIBUS transmission and reception circuits, in order to avoid failures. Passive masters communicate only with active masters. Its operation state is STOP.

Further details regarding the master states can be found in the PROFIBUS Network Utilization Manual - MU299026, or in the Operation Modes chapter.

Regarding its configuration, both masters receive the same bus configuration and the PROFIBUS slaves' configuration.

The active network master has the address configured by the user in the MasterTool IEC XE programmer. The passive master connects to the network through another address. The passive master's address is active master's address subtracted by one. In case the active master's address is zero, the passive master's address will be set to 125.

The passive network master's address is transparent to the user, so there's no specific configuration needed. The NX5001 module calculates and takes this address when it becomes the PROFIBUS network passive master. It's up to the Nexto Series CPU to define if the PROFIBUS-DP NX5001 master acts as passive or active master.

For further details regarding the PROFIBUS-DP NX5001 insertion in the series redundancy context, see Nexto Series CPU Utilization manual (MU214605), chapter Redundancy with NX3030 CPU.

1.6. Documents Related to this Manual

For additional information about Nexto Series, you can consult other documents (manuals and technical characteristics) beyond this one. These documents are available in its last review on www.altus.com.br.

Each product has a document called Technical Characteristics (CE), with the list of features of the product in question. Additionally, the product may have User Manuals (manuals' codes are mentioned at CEs of each product).

It is recommended to consult the following documents as a source of additional information:

Code	Description	Language
CE114000	Nexto Series – Technical Characteristics	English
CT114000	Série Nexto – Características Técnicas	Portuguese
CS114000	Serie Nexto – Características Técnicas	Spanish
MU214600	Nexto Series User Manual	English
MU214000	Manual de Utilização Série Nexto	Portuguese
MU214605	Nexto Series CPUs User Manual	English
MU214100	Manual de Utilização UCPs Série Nexto	Portuguese
MU299026	Manual de Utilização da Rede PROFIBUS	Portuguese
MU209010	Configuração da Remota PROFIBUS – Série Ponto	Portuguese
MU214608	Nexto PROFIBUS-DP Head Utilization Manual	English
MU214108	Manual de Utilização da Cabeça PROFIBUS-DP Nexto	Portuguese
MU209508	Manual de Utilização Cabeça PROFIBUS PO5063V1 e Cabeça Redundante PROFIBUS PO5063V5	Portuguese
MU219511	PO5064 PROFIBUS Head and PO5065 Redundant PROFIBUS Head Utilization Manual	English

Code	Description	Language
MU209511	Manual de Utilização Cabeça PROFIBUS PO5064 e Cabeça Redundante PROFIBUS PO5065	Portuguese
MU209020	Manual de Utilização Rede HART sobre PROFIBUS	Portuguese
MU204631	Manual de Utilização do Repetidor Ótico / FOCUS PROFIBUS	Portuguese
MU299609 **MU299048**	MasterTool IEC XE User Manual Manual de Utilização MasterTool IEC XE	English Portuguese
MP399609 **MP399048**	MasterTool IEC XE Programming Manual Manual de Programação MasterTool IEC XE	English Portuguese

Table 1: Related Documents

1.7. Visual Inspection

Before resuming the installation process, it is advised to carefully visually inspect the equipment, verifying the existence of transport damage. Verify if all parts requested are in perfect shape. In case of damages, inform the transport company or Altus distributor closest to you.

CAUTION

Before taking the modules off the case, it is important to discharge any possible static energy accumulated in the body. For that, touch (with bare hands) on any metallic grounded surface before handling the modules. Such procedure guaranties that the module static energy limits are not exceeded.

It's important to register each received equipment serial number, as well as software revisions, in case they exist. This information is necessary, in case the Altus Technical Support is contacted.

1.8. Technical Support

For Altus Technical Support contact in São Leopoldo, RS, call +55 51 3589-9500. For further information regarding the Altus Technical Support existent on other places, see https://www.altus.com.br/en/ or send an email to altus@altus.com.br.

If the equipment is already installed, you must have the following information at the moment of support requesting:

- The model from the used equipments and the installed system configuration
- The product serial number
- The equipment revision and the executive software version, written on the tag fixed on the product's side
- CPU operation mode information, acquired through MasterTool IEC XE
- The application software content, acquired through MasterTool IEC XE
- Used programmer version

1.9. Warning Messages Used in this Manual

In this manual, the warning messages will be presented in the following formats and meanings:

DANGER

Reports potential hazard that, if not detected, may be harmful to people, materials, environment and production.

CAUTION

Reports configuration, application or installation details that must be taken into consideration to avoid any instance that may cause system failure and consequent impact.

ATTENTION

Identifies configuration, application and installation details aimed at achieving maximum operational performance of the system.

2. Technical Description

2.1. General Features

	NX5001
Space use in Backplane	2 sequential slots
Maximum amount of PROFIBUS Slaves	125
Maximum amount of cyclic input bytes per slave	244
Maximum amount of cyclic output bytes per slave	244
Maximum amount of input bytes	3584
Maximum amount of output bytes	3584
PROFIBUS-DP	Yes
Baud rate	9.6 to 12000 kBits/s, configurable
Redundancy support	Yes
Global Control Command support	Yes
Hot Swap support	Yes
Indication of status and diagnostics	Display, LEDs, web server and CPU internal memory
One Touch Diag (OTD)	Yes
Isolation **Interface PROFIBUS to logic** **Interface PROFIBUS to protection grounding** ⏚ **Logic to protection grounding** ⏚	 1000 Vac / 1 minute 1000 Vac / 1 minute 1250 Vac / 1 minute
Current consumption from bus power source	400 mA
Dissipated power	2 W
IP level	IP 20
Operating temperature	0 to 60 °C
Storage temperature	-25 to 75 °C
Relative humidity	5% to 96%, without condensation
Electronic Circuits' Conformal Coating	Yes
Module dimensions (W x H x D)	36.00 x 114.63 x 117.07 mm
Package dimensions (W x H x D)	42.00 x 122.00 x 147.00 mm
Net weight	200 g
Gross weight (with package)	250 g

Table 2: General Characteristics

Notes:

Baud Rate: Transmission rate can be configured with the following communication speeds: 9.6 kBits/s, 19.2 kBits/s, 45.45 kBits/s, 93.75 kBits/s, 187.5 kBits/s, 500 kBits/s, 1500 kBits/s, 3000 kBits/s, 6000 kBits/s and 12000 kBits/s.

Redundancy Support: It is possible to assemble a PROFIBUS redundant network using two NX5001. This implementation is described in section System Configuration. Needs software version 1.1.0.0 or above/ product revision AE or above.

Global Control Command: This service synchronizes inputs and/or outputs of a given PROFIBUS slave group through Sync, Unsync, Freeze and Unfreeze commands. These commands are available at the NX5001 PROFIBUS-DP master through User Commands. The description of this service is found at Appendix C – Global Control Commands. For further details on the availability of this service and its associated products (NX5001 and MasterTool IEC XE programmer), consult section Compatibility with Other Products. Needs software version 1.2.0.6 or above/ product revision AP or above.

Maximum Number of PROFIBUS Slaves: NX5001 can address up to 31 slaves without needing repeaters or converters. For more than 31 slaves, repeaters and converters must be used.

Logic: Logic is the name given to inner interfaces such as memories, processor and rack interfaces.

Electronic Circuits' Conformal Coating: The electronic circuits' conformal coating protects the product's inner parts against humidity, dust and other harmful elements to electronic circuits.

2.2. Standards and Certifications

Standards and Certifications	
IEC	61131-2: Industrial-process measurement and control - Programmable controllers - Part 2: Equipment requirements and tests
	DNV Type Approval – DNV-CG-0339 (TAA000013D)
	2014/30/EU (EMC) 2014/35/EU (LVD) 2011/65/EU and 2015/863/EU (ROHS)
	S.I. 2016 No. 1091 (EMC) S.I. 2016 No. 1101 (Safety) S.I. 2012 No. 3032 (ROHS)
	UL/cUL Listed – UL 61010-1 UL 61010-2-201 (file E473496)
	TR 004/2011 (LVD) CU TR 020/2011 (EMC)

Table 3: Standards and Certifications

2.3. Performance

The PROFIBUS-DP NX5001 master performance depends on 3 factors:

- Time for Data Transfer between NX5001 and CPU
- Cycle Time of PROFIBUS Network
- Transition time for the PROFIBUS slave I/O bus

The summation of these three factors determines the latency between user application and I/O state change in the PROFIBUS slave.

In other words, it defines the maximum necessary time for a change in an output variable (%Q), made by the user application, alters the PROFIBUS slave, connected to the PROFIBUS-DP NX5001 master, output value, and/or the maximum necessary time for an alteration in the PROFIBUS slave input to be perceived by the user application (%I).

2.3.1. Time for Data Transfer between NX5001 and CPU

This parameter refers to the time required to transfer all variables (%Q and/or %I) between CPU and NX5001 module.

This time is proportional to the total number of input (%I) and output (%Q) variables within the PROFIBUS slave declaration.

Total bytes (Sum of %Ix and/or %Qx)	Transfer Time
20	3.77 ms
7168	9.00 ms

Table 4: Data Transfer Time between CPU and NX5001

2.3.2. Cycle Time of PROFIBUS Network

This parameter refers to the time required for the NX5001 to communicate with all its PROFIBUS slaves.

Besides the total amount of bytes and slaves configured in the PROFIBUS network, there are other parameters configured in the tab General (see Master Parameters section) which influences cycle time:

- **max. T_SDR:** Time the master must wait between sending requests.
- **Baud rate:** The selected communication rate.
- **Slave interval:** Minimum time between two master's accesses to the same slave.
- **Highest station address:** Highest PROFIBUS network device address. Its default value is 125, but it is a good practice to change it to the largest address used in the PROFIBUS network's configuration.

ATTENTION

Further details can be obtained in the PROFIBUS network Utilization Manual - MU299026.

2.3.3. Transition time for the PROFIBUS slave I/O bus

This time represents the necessary time for the PROFIBUS slave I/O channel to alter its state in the bus.

ATTENTION

See the *Transition/Update Time* parameter in the table at the *Features* chapter of the respective I/O module's *Technical Characteristics* document.

2.4. GSD File

Each PROFIBUS-DP device has a file that defines its limits and possibilities for configuration. GSD type files are used to ease interoperability in PROFIBUS network between devices from different manufacturers. These files contain device features to be considered for correct operation on the network, such as number and type of I/O modules, diagnostic messages, possible network parameters, transmission rates and time-out.

Each device to be integrated to a PROFIBUS network shall have a GSD file supplied by its manufacturer.

ATTENTION

GSDs of PROFIBUS heads of Ponto and Nexto Series, and from Nexto's PROFIBUS-DP Master NX5001 are present in the MasterTool IEC XE programmer.
Differently from PROFIBUS masters of AL and Ponto Series, the configuration of PROFIBUS-DP Master NX5001 is performed directly in the MasterTool IEC XE programmer.

2.5. Distance x Baud Rate

In the cable route planning, the table below should be taken under consideration.

Segment length should be as defined in the table.

Baud Rate (kBits/s)	Distance/segment (m)
9.6	1200
19.2	1200
45.45	1200
93.75	1200
187.5	1000
500	400
1500	200
3000	100
6000	100
12000	100

Table 5: Possible Distances for AL-2303 Cable

2.6. Physical Dimensions

Dimensions in mm.

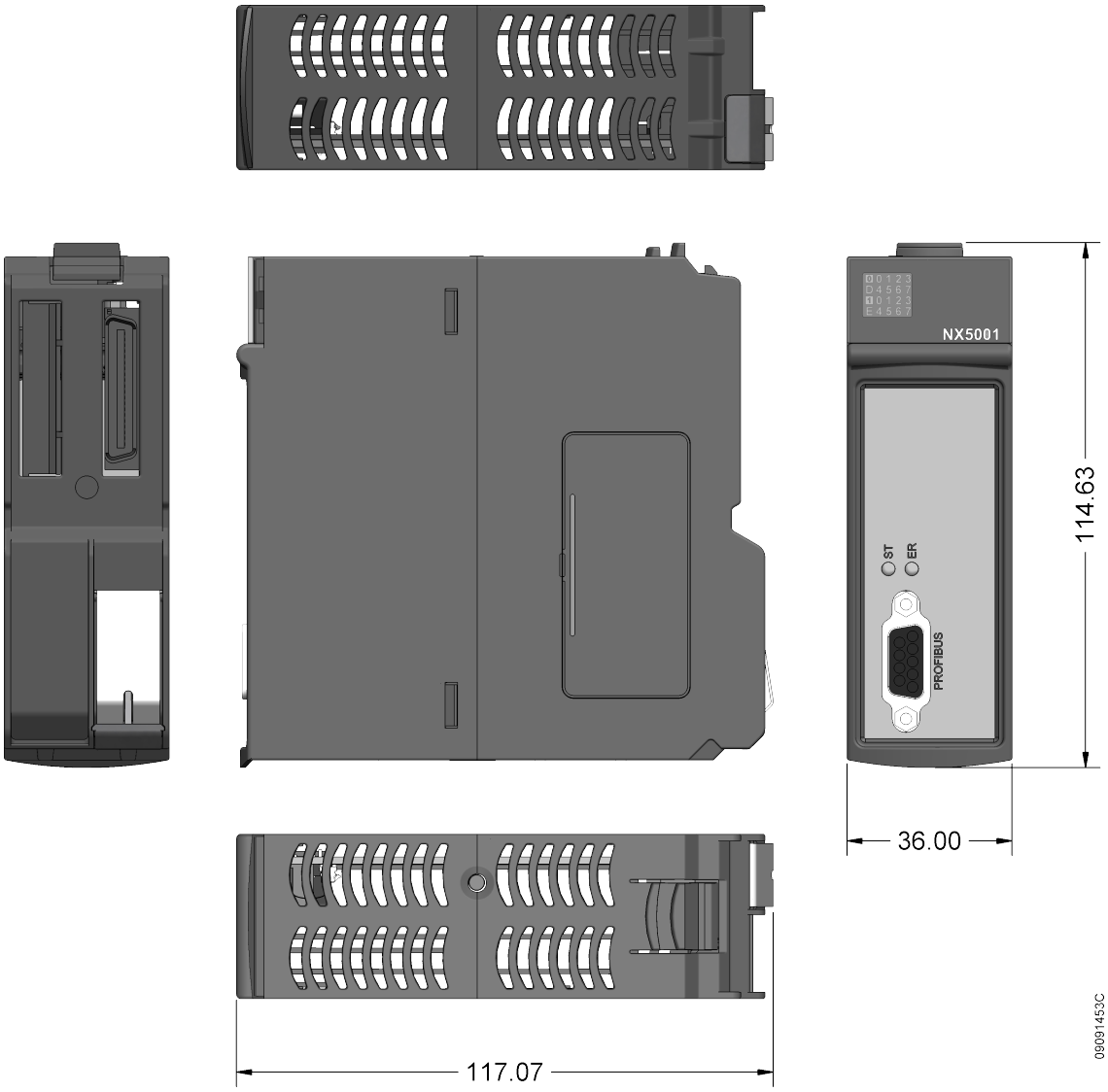

Figure 3: Physical Dimensions from NX5001

2.7. Information for Purchase

2.7.1. Integrated Items

The product package contains the following items:

- One NX5001 module
- Installation guide

2.7.2. Product Code

The following code shall be used for product purchase:

Code	Denomination
NX5001	PROFIBUS-DP Master Module

Table 6: Product Code

2.8. Related Products

The following products must be acquired separately whenever necessary:

Code	Denomination
NX5110	PROFIBUS-DP Head
NX5210	PROFIBUS-DP Redundant Head
PO5063V1	PROFIBUS-DP Fieldbus Head
PO5063V5	Redundant PROFIBUS-DP Fieldbus Head
PO5064	PROFIBUS-DPV1 Head
PO5065	Redundant PROFIBUS-DPV1 Head
AL-2601	PROFIBUS Connector
AL-2602	PROFIBUS Terminator Connector
AL-2605	Terminator with Power Supply Diagnostic
AL-2303	PROFIBUS cable
AL-2431	FOCUS/ PROFIBUS Optical Repeater
AL-2432	FOCUS/ PROFIBUS Optical Repeater with Two Ports
AL-2433	PROFISwitch – Coupler for Redundant Profibus Network

Table 7: Related Products

Notes:

NX5110: PROFIBUS-DP network head enables the connection of Nexto Series' modules to PROFIBUS networks, expanding the CPU's I/O system.

NX5210: PROFIBUS-DP redundant network head enables the connection of Nexto Series' modules to redundant PROFIBUS networks.

PO5063V1: PROFIBUS-DP network head enables the connection of Ponto Series' modules to PROFIBUS networks, expanding the CPU's I/O system.

PO5063V5: PROFIBUS-DP redundant network head enables the connection of Ponto Series' modules to redundant PROFIBUS networks.

PO5064: PROFIBUS-DP network head enables the connection of Ponto Series' modules to PROFIBUS networks, expanding the CPU's I/O system. Additionally, PO5064 module supports DPV1 communication between Ponto Series' modules and any other PROFIBUS network node.

PO5065: PROFIBUS-DP redundant network head enables the connection of Ponto Series' modules to PROFIBUS networks, expanding the CPU's I/O system. Additionally, PO5065 module supports DPV1 communication between Ponto Series' modules and any other PROFIBUS network node.

AL-2601: DB9 connector with standard PROFIBUS pinout. It's suitable for connections between PROFIBUS networks and devices placed on intermediate positions in the network (not in the ends). This connector has a connection either for PROFIBUS network input and output, allowing module exchange without interrupting network activity.

AL-2602: DB9 connector and terminator with standard PROFIBUS pinout. It has inner components for network termination. It's suitable for connections between PROFIBUS networks and devices placed on the network ends.

AL-2605: This device is mounted at the ends of a PROFIBUS network and it eliminates the necessity for AL-2602 connectors. The AL-2605 module was developed for securing the PROFIBUS fieldbus operation even if the modules placed at the network ends are shut down or removed. The product also verifies the fieldbus' power supply, diagnosing failures. It's suitable for any PROFIBUS fieldbus where the reliability and availability are main requirements.

AL-2303: Cable for PROFIBUS network.

AL-2431 and AL-2432: Optical repeaters for PROFIBUS device interconnection through optical fiber. The AL-2432 module has redundancy in optical media, increasing system availability.

AL-2433: The AL-2433 adapter enables the interconnection between non-redundant PROFIBUS-DP slave devices in a redundant PROFIBUS-DP network with NX5001 master.

2.9. Compatibility with Other Products

The table below brings information about compatibility between NX5001 module and MasterTool IEC XE programmer.

NX5001			Compatible Software Version
Version	**Revision**	**Functionality**	**MasterTool IEC XE**
1.2.0.6	AP	Support for Global Control Commands (Sync/Freeze) Support for DPV1 Command	2.01 or above

Table 8: Compatibility of Functionalities with MasterTool IEC XE

The table below indicates the compatibility of the main Altus' products with the NX5001 module.

Product	Software Version	Products Revision
NX5110	1.0.0.12 or above	AD or above
NX5210	1.0.0.12 or above	AD or above
PO5063	1.35 or above	DT or above
PO5063V1	2.07 or above	AV or above
PO5064	1.02 or above	AI or above
PO5063V5	5.07 or above	AV or above
PO5065	1.02 or above	AI or above
PO5063V4	4.35 or above	AV or above
ALT_059A.GSD	1.30 or above	-
ALT_0BAF.GSD	1.30 or above	-
ALT_0BB0.GSD	1.30 or above	-

Table 9: Compatibility with Other Products

3. Configuration

Nexto Series brings to the user the software MasterTool IEC XE, a powerful tool which supplies a complete interface used to program all modules of the series.

ATTENTION

It is not necessary using additional software for parameterization of the PROFIBUS network modules, neither a special cable is required, because all configurations and parameterizations are made, in a simple and easy way, directly in the MasterTool IEC XE programmer and sent to the Master PROFIBUS-DP NX5001 through Nexto CPU.

The assembly of the PROFIBUS network configuration is made in MasterTool IEC XE, so it's necessary that the configuration files (.GSD) of all devices to be connected to the NX5001 master be available and installed.

ATTENTION

The GSD file of NX5001, as well as GSDs from Ponto and Nexto Series remotes, are already installed in the MasterTool IEC XE programmer, ready to be used.

In the example of Appendix A – Usage Example, details are supplied regarding:

- Assembly of PROFIBUS network configuration
- Mapping of Input and Output Points
- Mapping of Diagnostic Area

In Appendix B – GSD Installation, there is an example of GSD installation on MasterTool IEC XE programmer.

The Appendix C – Global Control Commands, has information about programming details, configuration, compatibility issues and operation of support to Global Control Command, whose goal is to synchronize the data channel of the input modules and/or output of a particular group of PROFIBUS slaves.

3.1. System Configuration

This section presents the possible configurations for the PROFIBUS network using the NX5001 interface.

3.1.1. Configuration A: Simple PROFIBUS network

This is the basic configuration. As it can be seen in the figure below, the NX5001 interface is connected to a Nexto Series CPU on the same bus and to the PROFIBUS slaves through a PROFIBUS network.

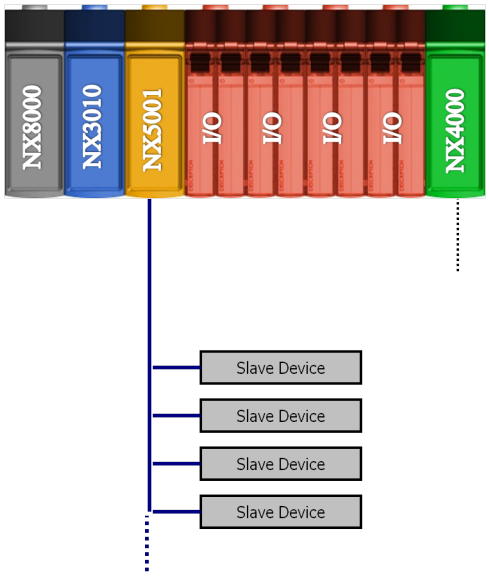

Figure 4: Configuration of a simple PROFIBUS network

3.1.2. Configuration B: Redundant PROFIBUS network

As shown on the figure below, the PROFIBUS network is redundant. In this case, two NX5001 are connected to a Nexto Series CPU in the same bus.

The redundant PROFIBUS network allows normal operation during a failure in one of the redundant networks, offering higher availability, which is required in critical applications.

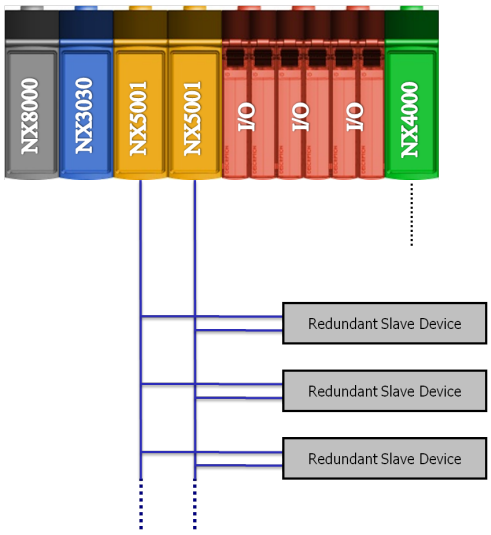

Figure 5: Configuration of a redundant PROFIBUS network

3.1.3. Configuration C: Two independent simple PROFIBUS networks

In this configuration, two NX5001 modules are connected to a Nexto CPU in the same bus. Each NX5001 is connected to an independent PROFIBUS network. In this case there's no redundancy. The architecture is seen in the figure below.

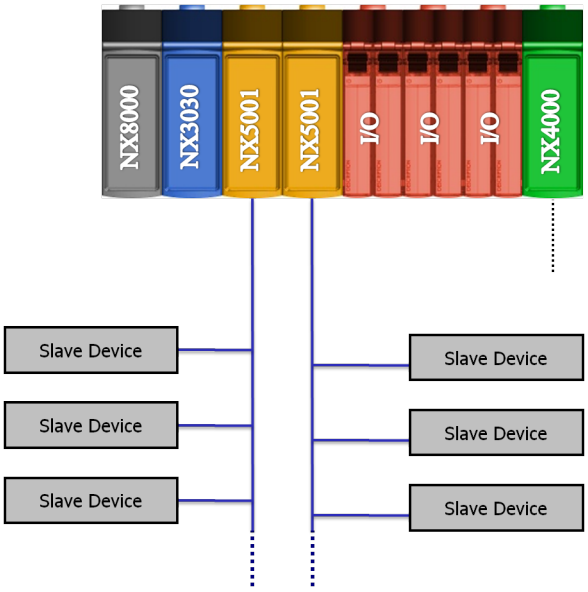

Figure 6: Configuration of two independent simple PROFIBUS networks

3.1.4. Configuration D: Two independent redundant PROFIBUS networks

This is the PROFIBUS master with the most complex configuration supported by the Nexto Series. It's the redundant version of configuration C. As shown on the figure below, there are two independent and redundant PROFIBUS networks.

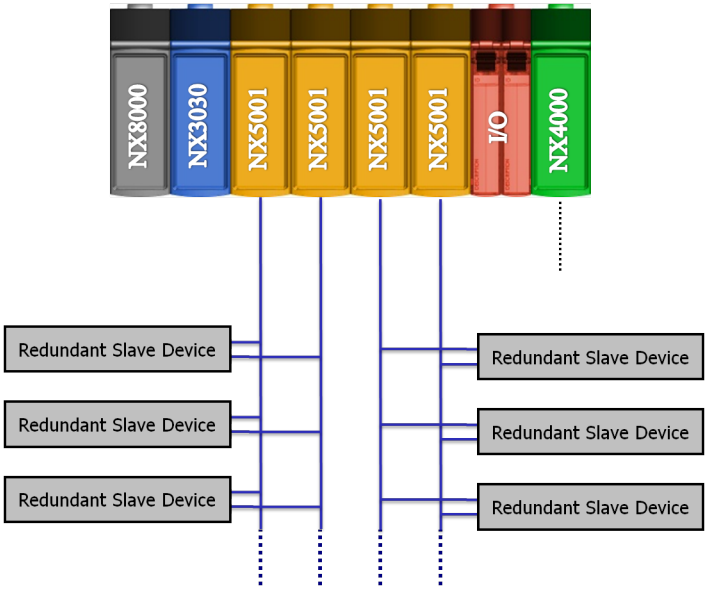

Figure 7: Configuration of two independent redundant PROFIBUS networks

3.1.5. Configuration E: Four independent simple PROFIBUS networks

Configuration E has four NX5001 connected to a Nexto CPU on the same rack. Each NX5001 is connected to an independent PROFIBUS fieldbus. In this case there's no redundancy. The architecture is shown in figure below.

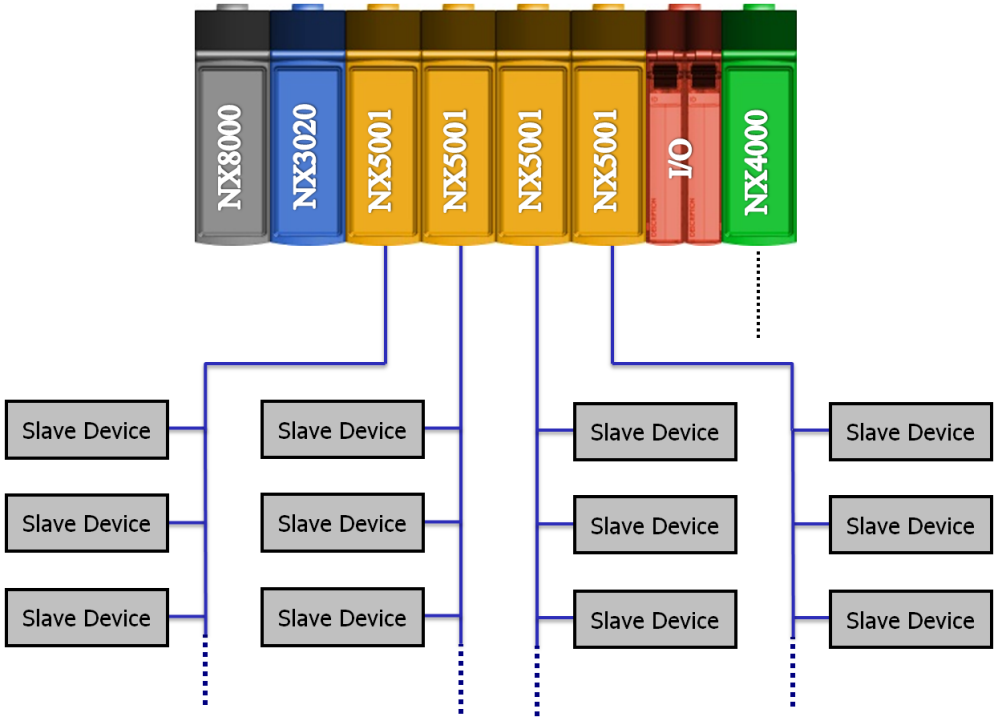

Figure 8: Configuration of four independent simple PROFIBUS networks

4. Diagnostics

One of the features of Nexto Series is the generation of diagnostics of abnormalities, being them failures, errors or operating modes, enabling the operator to identify and easily solve problems that may happen to the system.

Master NX5001 provides a complete diagnostic of the PROFIBUS network, as well as general diagnostics of module operation.

Diagnostics can be subdivided into four major groups:

- General Diagnostics
- Master Diagnostics
- Network Redundancy Diagnostics
- Diagnostics of Slave Devices

Diagnostics can be consulted by the user in the structure defined by the programmer MasterTool IEC XE (details in Diagnostics Mapping).

Memory area Offset (%QB(n)) allocated for the diagnostics of module NX5001 is defined by the user at its configuration screen, tab *Module Parameters* (further details in Master Parameters). This offset defines the first byte of General Diagnostics.

ATTENTION

The data within the diagnostics structure are only valid when the NX5001 PROFIBUS-DP Master is connected to the bus. In case it isn't detected by the CPU, General and Master diagnostics will be zeroed/set to False, while Network and Slave diagnostics will be kept in their last read value.

4.1. General Diagnostics

The Diagnostics set which form the *General Diagnostics* provide information to the user regarding the behavior of PROFIBUS-DP Master NX5001 module.

The table below shows the description of each bit of *General Diagnostics*:

Direct Variable		Diagnostic Message	AT Variable	Description
Variable	**Bit**		**DG_NX5001.tGeneral.***	
%QB(n)	0..7	Reserved		
%QB(n+1)	0	MODULE W/ DIAGNOSTICS	bActiveDiagnostics	TRUE – Module has active diagnostics
		NO DIAG		FALSE – Module has not active diagnostics
	1	MODULE W/ FATAL ERROR	bFatalError	TRUE – Module has a fatal error
		-		FALSE – No fatal error
	2	CONFIG. MISMATCH	bConfigMismatch	TRUE – Parameterization error
		-		FALSE – Parameterization ok
	3	WATCHDOG ERROR	bWatchdogError	TRUE – Watchdog detected
		-		FALSE – Watchdog not detected
	4	OTD SWITCH ERROR	bOTDSwitchError	TRUE – Module's OTD switch failure
		-		FALSE – OTD switch ok
	5	-	bBusCommunicationError	TRUE – Bus communication failure
		-		FALSE – No errors in the bus communication

Direct Variable		Diagnostic Message	AT Variable	Description
Variable	Bit		DG_NX5001.tGeneral.*	
	6..7	Reserved		
%QB(n+2)	0..7	Reserved		
%QB(n+3)	0..7	Reserved		
%QB(n+4)	0	Reserved		
	1	ABSENT CONFIG.	bNX5001NoCfg	TRUE – NX5001 hasn't received configuration from the CPU
		-		FALSE – NX5001 received configuration from the CPU
	2..3	Reserved		
	4	-	bStsEnableInterface	TRUE - NX5001 was enabled by user command
		-		FALSE - NX5001 was disabled by user command
	5	Reserved		
	6	-	bStsMstRedundEnable	TRUE - Indicates there's master redundancy (control variable)
		-		FALSE – Indicates the master redundancy isn't enabled (control variable)
	7	Reserved		
%QB(n+5)	0	PB SLAVE UNCONFIGURED	bSlaveNotConfigured	TRUE – There is at least one slave not parameterized
		-		FALSE – All slaves are parameterized
	1	PB SLAVE NOT PRESENT	bSlaveNotPresent	TRUE – There is at least one absent slave in the network
		-		FALSE – All modules are connected to the field network
	2	PB SLAVE W/ DIAG.	bSlaveWithDiagnostic	TRUE – There is at least one slave with active diagnostics
		-		FALSE – No slaves with active diagnostics
	3..4	Reserved		
	5	COMM. FAILURE	bPbusCommFail	TRUE – PROFIBUS communication failure
		-		FALSE – PROFIBUS fieldbus is operational
	6..7	Reserved		

Table 10: General Diagnostics

Notes:

Direct Representation Variable: "n" is the address defined at *%Q Start Address of Module Diagnostics Area*, at the NX5001 module configuration screen – tab *Module Parameters* in MasterTool IEC XE.

AT Variable: the whole name of an AT variable is composed by *DG_modulename.tGeneral.* plus the variable name defined on the table above. Observe that *modulename* is the name used for the specific module. For reference, the default name given by MasterTool IEC XE is the product code (e.g. NX5001) or the code followed by n, where n is incremented at each

new module. For instance: in order to access the NX5001 module's fatal error diagnostics, the following AT variable must be used: *DG_NX5001.tGeneral.bFatalError*.

bStsMstRedundEnable: When used in a system with Half-Cluster redundancy, after switching from one Half-Cluster to the other, this diagnostic will remain off for a few seconds.

4.2. Master Diagnostics

Master diagnostics inform the network's general status from the master's point of view, with an overview of active stations or PROFIBUS network error. If there is redundancy, each NX5001 module has its own diagnostics that are associated with different variables.

Diagnostics which indicate network conditions are in memory area between %QB(n+6) to %QB(n+7). Between %QB(n+8) to %QB(n+21) is the reserved memory area. Next is the memory area that presents the status of each of the network's slave devices (whether the device is configured, active or with diagnostic), between %QB(n+22) to %QB(n+69).

A summary of the description of each variable can be found in the table below.

Direct Variable	AT Variable DG_NX5001.tMstStatus.*	Description
%QB(n+6)	tGlobalBits	Global errors
%QB(n+7)	byState	Master state
%QB(n+8)	byReserved02	Reserved
%QB(n+9)	byReserved03	Reserved
%QB(n+10)	wReserved04	Reserved
%QB(n+12)	wReserved06	Reserved
%QB(n+14) .. %QB(n+21)	abyReserved[0] .. abyReserved[7]	Reserved
%QB(n+22) .. %QB(n+69)	abySlv_Cfg abySlv_State abySlv_Diag	Network mapping: slave state and diagnostics

Table 11: Master Diagnostics

4.2.1. Global Errors

Bits %QB(n+6) are used to indicate the main errors on bus and PROFIBUS master. The table below shows the description of each one of these bits.

Direct Variable		AT Variable DG_NX5001 .tMstStatus.tGlobalBits.*	Description
Variable	Bit		
%QB(n+6)	0	bCtrlError	Configuration error: configuration in at least one of slave devices is different than the master. This error is caused by incorrect parameterization.
	1		Reserved.
	2	bNonExchangeError	At least one slave device is not communicating, or has notified a serious error.
	3..7		Reserved.

Table 12: Global Errors

4.2.2. Master States

Value displayed in %QB(n+7) represents operation mode of master NX5001. According to the table below, it is possible to distinguish master operation states:

Direct Variable		AT Variable DG_NX5001.tMstStatus.byState	Description
Variable	**Value (Hex)**		
%QB(n+7)	00	OFFLINE	Master off
	40	STOP	Master stopped
	80	CLEAR	Master stopped with protected outputs
	C0	OPERATE	Master operational

Table 13: Master Operation Modes

Details on master's status are described at PROFIBUS Network User Manual - MU299026, or in the section Operation Modes.

4.2.3. Network Maps

This memory area is divided in the following way:

- %QB(n+22) to %QB(n+37): configured slaves area
- %QB(n+38) to %QB(n+53): present slaves area
- %QB(n+54) to %QB(n+69): slaves with diagnostics area

4.2.3.1. Map of Configured Slaves

In this map, it is possible to check which slaves were configured by the master.
Each bit is related to one slave. The format of this area can be observed below.

Direct Variable	AT Variable DG_NX5001. tMstStatus.*	Bit							
		7	6	5	4	3	2	1	0
		Physical Addresses on PROFIBUS Network							
%QB(n+22)	**abySlv_Cfg[0]**	7	6	5	4	3	2	1	0
%QB(n+23)	**abySlv_Cfg[1]**	15	14	13	12	11	10	9	8
%QB(n+24)	**abySlv_Cfg[2]**	23	22	21	20	19	18	17	16
%QB(n+25)	**abySlv_Cfg[3]**	31	30	29	28	27	26	25	24
%QB(n+26)	**abySlv_Cfg[4]**	39	38	37	36	35	34	33	32
%QB(n+27)	**abySlv_Cfg[5]**	47	46	45	44	43	42	41	40
%QB(n+28)	**abySlv_Cfg[6]**	55	54	53	52	51	50	49	48
%QB(n+29)	**abySlv_Cfg[7]**	63	62	61	60	59	58	57	56
%QB(n+30)	**abySlv_Cfg[8]**	71	70	69	68	67	66	65	64
%QB(n+31)	**abySlv_Cfg[9]**	79	78	77	76	75	74	73	72
%QB(n+32)	**abySlv_Cfg[10]**	87	86	85	84	83	82	81	80
%QB(n+33)	**abySlv_Cfg[11]**	95	94	93	92	91	90	89	88
%QB(n+34)	**abySlv_Cfg[12]**	103	102	101	100	99	98	97	96

Direct Variable	AT Variable DG_NX5001. tMstStatus.*	Bit							
		7	6	5	4	3	2	1	0
		Physical Addresses on PROFIBUS Network							
%QB(n+35)	abySlv_Cfg[13]	111	110	109	108	107	106	105	104
%QB(n+36)	abySlv_Cfg[14]	119	118	117	116	115	114	113	112
%QB(n+37)	abySlv_Cfg[15]			125	124	123	122	121	120

Table 14: Map of Slaves Configured by the PROFIBUS-DP Master NX5001

When the bit of the corresponding PROFIBUS slave is set to true, it means that this slave is configured in the *Device Tree* of MasterTool IEC XE programmer. If it remains false, it means that this slave has not been configured.

4.2.3.2. Map of Present Slaves

This map indicates which devices are present in the PROFIBUS network, that is, those that are being accessed by the master. Each bit is related to one slave. The format of this area can be observed in the table below.

Direct Variable	AT Variable DG_NX5001. tMstStatus.*	Bit							
		7	6	5	4	3	2	1	0
		Physical Addresses on PROFIBUS Network							
%QB(n+38)	abySlv_State[0]	7	6	5	4	3	2	1	0
%QB(n+39)	abySlv_State[1]	15	14	13	12	11	10	9	8
%QB(n+40)	abySlv_State[2]	23	22	21	20	19	18	17	16
%QB(n+41)	abySlv_State[3]	31	30	29	28	27	26	25	24
%QB(n+42)	abySlv_State[4]	39	38	37	36	35	34	33	32
%QB(n+43)	abySlv_State[5]	47	46	45	44	43	42	41	40
%QB(n+44)	abySlv_State[6]	55	54	53	52	51	50	49	48
%QB(n+45)	abySlv_State[7]	63	62	61	60	59	58	57	56
%QB(n+46)	abySlv_State[8]	71	70	69	68	67	66	65	64
%QB(n+47)	abySlv_State[9]	79	78	77	76	75	74	73	72
%QB(n+48)	abySlv_State[10]	87	86	85	84	83	82	81	80
%QB(n+49)	abySlv_State[11]	95	94	93	92	91	90	89	88
%QB(n+50)	abySlv_State[12]	103	102	101	100	99	98	97	96
%QB(n+51)	abySlv_State[13]	111	110	109	108	107	106	105	104
%QB(n+52)	abySlv_State[14]	119	118	117	116	115	114	113	112
%QB(n+53)	abySlv_State[15]			125	124	123	122	121	120

Table 15: Map of Present Slaves (Active) on PROFIBUS Network

When the bit of the corresponding PROFIBUS slave is set to true, it means that the master and the slave are exchanging data with each other (communication OK). If it remains false, it means that the master and slave are not exchanging data with each other (The device is not connected to the network or there is some kind of error).

4.2.3.3. Map of Slaves with Diagnostics

In this map you can check which slave has active diagnostics. Each bit is related with one slave. The format of this area can be observed below.

Direct Variable	AT Variable DG_NX5001. tMstStatus.*	Bit							
		7	6	5	4	3	2	1	0
		Physical Addresses on PROFIBUS Network							
%QB(n+54)	abySlv_Diag[0]	7	6	5	4	3	2	1	0
%QB(n+55)	abySlv_Diag[1]	15	14	13	12	11	10	9	8
%QB(n+56)	abySlv_Diag[2]	23	22	21	20	19	18	17	16
%QB(n+57)	abySlv_Diag[3]	31	30	29	28	27	26	25	24
%QB(n+58)	abySlv_Diag[4]	39	38	37	36	35	34	33	32
%QB(n+59)	abySlv_Diag[5]	47	46	45	44	43	42	41	40
%QB(n+60)	abySlv_Diag[6]	55	54	53	52	51	50	49	48
%QB(n+61)	abySlv_Diag[7]	63	62	61	60	59	58	57	56
%QB(n+62)	abySlv_Diag[8]	71	70	69	68	67	66	65	64
%QB(n+63)	abySlv_Diag[9]	79	78	77	76	75	74	73	72
%QB(n+64)	abySlv_Diag[10]	87	86	85	84	83	82	81	80
%QB(n+65)	abySlv_Diag[11]	95	94	93	92	91	90	89	88
%QB(n+66)	abySlv_Diag[12]	103	102	101	100	99	98	97	96
%QB(n+67)	abySlv_Diag[13]	111	110	109	108	107	106	105	104
%QB(n+68)	abySlv_Diag[14]	119	118	117	116	115	114	113	112
%QB(n+69)	abySlv_Diag[15]			125	124	123	122	121	120

Table 16: Map of Slaves with Diagnostics

When the bit of the corresponding PROFIBUS slave is set to true, it means that there are diagnostics to be read and it is found in its respective diagnostic structure. If it remains false, it means that there's no diagnostics.

ATTENTION

In case the slave sends extended diagnostics, the bit corresponding to the slave's address will be true.

4.2.3.4. Map's Logical Meaning

The result of two bits combination (present and with diagnostic) from a device is shown in the following table.

	Present = FALSE	Present = TRUE
With Diagnostic = FALSE	There is no data exchange between master and slave device. Check if slave device is configured and active.	Slave device is present in the network. There is data exchange between master and slave device.
With Diagnostic = TRUE	There is no data exchange between master and slave device. Master has device diagnostics, which can be read in the area of device diagnostics.	Slave device is present in the network. There is data exchange between master and slave device. Master has device diagnostics, which can be read in the area of device diagnostics.

Table 17: Result of the Combination of Presence and Diagnostics Bits of a Slave

4.3. Network Redundancy Diagnostics

Network redundancy diagnostics are only generated if there is redundant operation. Each NX5001 module updates its own part on the memory area specified for this purpose at MasterTool IEC XE configuration.

The diagnostics are presented as a set of bits, one for each physical address of the network, by specifying which redundant device is currently active in the network at the moment.

It is possible to view slave status also through structure *DG_NX5001.tRedund.tActivNtw.bSlave_x*, where "x" is the slave address in the PROFIBUS network. Its value ranges from 0 to 125.

When the corresponding slave bit is true, it means that it is the pair's active slave.

Direct Variable	AT variable DG_NX5001. tRedund.tActivNtw.*	Bit							
		7	6	5	4	3	2	1	0
		Physical Address on Network							
%QB(n+70)	bSlave_0 .. bSlave_7	7	6	5	4	3	2	1	0
%QB(n+71)	bSlave_8 .. bSlave_15	15	14	13	12	11	10	9	8
%QB(n+72)	bSlave_16 .. bSlave_23	23	22	21	20	19	18	17	16
%QB(n+73)	bSlave_24 .. bSlave_31	31	30	29	28	27	26	25	24
%QB(n+74)	bSlave_32 .. bSlave_39	39	38	37	36	35	34	33	32
%QB(n+75)	bSlave_40 .. bSlave_47	47	46	45	44	43	42	41	40
%QB(n+76)	bSlave_48 .. bSlave_55	55	54	53	52	51	50	49	48
%QB(n+77)	bSlave_56 .. bSlave_63	63	62	61	60	59	58	57	56
%QB(n+78)	bSlave_64 .. bSlave_71	71	70	69	68	67	66	65	64
%QB(n+79)	bSlave_72 .. bSlave_79	79	78	77	76	75	74	73	72
%QB(n+80)	bSlave_80 .. bSlave_87	87	86	85	84	83	82	81	80
%QB(n+81)	bSlave_88 .. bSlave_95	95	94	93	92	91	90	89	88
%QB(n+82)	bSlave_96 .. bSlave_103	103	102	101	100	99	98	97	96
%QB(n+83)	bSlave_104 .. bSlave_111	111	110	109	108	107	106	105	104
%QB(n+84)	bSlave_112 .. bSlave_119	119	118	117	116	115	114	113	112
%QB(n+85)	bSlave_120 .. bSlave_125			125	124	123	122	121	120

Table 18: Network Redundancy Diagnostic

ATTENTION

The diagnostics can indicate the device as active or passive in both networks, simultaneously, during a transition (or switchover). This can happen because information for diagnostic are sent by different networks, and also copied in different times. After the switchover, the diagnostics are stable.

4.4. Diagnostics of Slave Devices

This section brings an overview of the diagnostics format generated by PROFIBUS slaves, and details on the diagnostics generated to PROFIBUS heads of Ponto Series, Nexto Series and Other PROFIBUS Slaves.

4.4.1. PROFIBUS Diagnostics

Diagnostics of slave devices show in detail the error situation in a particular slave.

The diagnostics of the network's devices are automatically copied by the NX5001 master to the defined memory area. This area is populated by the existing diagnostics present on the network as they are generated by the slave devices.

For a redundant network, both NX5001 masters can copy device diagnostics to the same area. The general format of diagnostics generated by a PROFIBUS slave will be detailed below at the sections Standard Diagnostics and Extended Diagnostics, and they comply with the following format:

Byte	Meaning
0	Status 1
1	Status 2
2	Status 3
3	Status 4
4	Status 5
5	Status 6
6 - 243	Extended Diagnostics

Table 19: Format of PROFIBUS Diagnostics Frame

4.4.1.1. Standard Diagnostics

The standard diagnostics, defined by norm, consist of 6 bytes and are independent of manufacturer or slave device. The following table shows the description of the diagnostic bits generated by the modules.

Direct Variable		Description
Variable	Bit	
%QB(n)	0	**Station_non_Existent** TRUE: the slave was not found in the network.
	1	**Station_Not_Ready** TRUE: slave is not ready for communication.
	2	**Cfg_Fault** TRUE: indicates that slave configuration is different from the configuration present in the master.
	3	**Ext_diag** TRUE: indicates that the slave has a message of extended diagnostic to be read by the master.
	4	**Not_Supported** TRUE: indicates that the slave received a non-supported command.
	5	**Invalid_Slave_Response** TRUE: indicates that the slave answer to the master was not recognized.
	6	**Parameter_fault** TRUE: indicates that there was an error sending the parameters to the slave.
	7	**Master_Lock** TRUE: indicates that the slave was parameterized by other master.
%QB(n+1)	0	**Prm_Req** TRUE: turned on by the slave to inform that it must be parameterized and configured.
	1	**Static_Diagnostic** TRUE: turned on by the slave to inform that the diagnostic shall be read by the master.
	2	Always turned on by the slave (TRUE).
	3	**Watchdog_On** TRUE: turned on by the slave when activating its watchdog.
	4	**Freeze_mode** TRUE: turned on by the slave when receiving the command Freeze.
	5	**Sync_Mode** TRUE: turned on by the slave when receiving the command Sync.
	6	Reserved.

Direct Variable		Description
Variable	Bit	
	7	**Deactivated** TRUE: slave was stated as inactive in parameterization.
%QB(n+2)	0	Reserved.
	1	Reserved.
	2	Reserved.
	3	Reserved.
	4	Reserved.
	5	Reserved.
	6	Reserved.
	7	**Ext_Diag_Overflow** TRUE: turned on if information of extended diagnostic from the slave overcame the size defined in GSD (Ext_Diag_Data).
%QB(n+3)		**Master_Add** Address of master which parameterized the slave. In case no master has parameterized the slave, the value stays in 255.
%QB(n+4) and %QB(n+5)		**Ident_Number** Identifier of slave device (device number, as registered on PROFIBUS Committee).

Table 20: Standard Diagnostics for Slave Device

4.4.1.2. Extended Diagnostics

The following bytes describe in detail the situation in the slave. This level of detail varies according to the slave device and/or manufacturer.

If the slave sends the extended diagnostic, the bit *Ext_Diag* is turned on.

Extended diagnostics can be subdivided into three parts:

- Device Oriented Diagnostics
- Module Oriented Diagnostics
- Channel Oriented Diagnostics

4.4.1.2.1. Device Oriented Diagnostics

Device oriented diagnostic is the first part of extended diagnostics. Its function is to identify the situation on the device.

Its format is particular for each device (see device manual) and it has an overhead, as shown in the table below.

Direct Variable		Description
Variable	Bit	
%QB(n+6)	0	Fixed in zero (FALSE).
	1	Fixed in zero (FALSE).
	2..7	Overhead. Block size in bytes, including overhead.
%QB(n+7)		Diagnostic bytes specific of the device.

Table 21: Device Oriented Diagnostic

4.4.1.2.2. Module Oriented Diagnostics

The format of this part of the diagnostic can be seen in standard EN50170.

This diagnostic indicates which sub-modules and channels have diagnostics, its detailing is done in the channel oriented diagnostic.

If a module oriented diagnostic is zeroed (status OK on all channels), the number of channel diagnostics is zero.

4.4.1.2.3. Channel Oriented Diagnostics

Each channel diagnostic occupies three bytes, with the following format:

Direct Variable		Description
Variable	**Bit**	
1st Byte	0..5	Identifier: indicates sub-module number.
	6	FALSE: fixed in zero.
	7	TRUE: fixed in one.
2nd Byte	0..5	Channel number: indicates channel number on sub-module.
	6..7	I/O: indicates the direction: -00 – reserved -01 – input -10 – output -11 – input and output
3rd Byte	0..4	Error Type: see Table 23
	5..7	Channel Type: see Table 24

Table 22: Channel Oriented Diagnostic

Error Type	Description
0	Reserved
1	Short-circuit
2	Undervoltage
3	Overvoltage
4	Overload
5	Over temperature
6	Open loop
7	Higher limit exceeded
8	Lower limit exceeded
9	Error
10 .. 15	Reserved
16 .. 31	Specific from manufacturer

Table 23: Error Type

Channel Type	Description
000	Reserved
001	1 bit
010	2 bits
011	4 bits
100	1 byte
101	1 word
110	2 words
111	Reserved

Table 24: Channel Type

For further details on the diagnostic device, refer to EN50170 standard and the manufacturer's manual of slave device.

For PROFIBUS heads of Ponto and Nexto Series, diagnostics are decoded from PROFIBUS format for bit map and are displayed to the user in %QB variables, and predefined structures, as documented in the manuals and CEs of the modules that constitute your network.

For the other slaves, 244 bytes are allocated, and they follow the pattern of diagnostic frame of a PROFIBUS slave.

4.4.2. Altus Slaves - Nexto Series

For PROFIBUS heads of Nexto Series, diagnostics are decoded and displayed in bit map format, as described in the chapters Diagnostic of PROFIBUS heads' manuals and the CEs of the modules that constitute the slave bus.

MasterTool IEC XE programmer provides pre-defined symbolic structures for diagnostic of PROFIBUS heads from Nexto Series and their modules, as can be seen in Diagnostics Mapping, in Appendix A – Usage Example.

4.4.3. Altus Slaves – Ponto Series

For PROFIBUS heads of Ponto Series, diagnostics are decoded and displayed in bit map format, as described in the chapters Diagnostic of PROFIBUS heads' manuals and the CEs of the modules that constitute the slave bus.

MasterTool IEC XE programmer provides pre-defined symbolic structures for diagnostic of PROFIBUS heads from Ponto Series and their modules, as can be seen in Diagnostics Mapping, in Appendix A – Usage Example.

The table below shows an example of diagnostics decoding for PROFIBUS heads from Ponto Series. To ease the interpretation, information is always omitted from *Module Oriented Diagnostic*.

The first byte indicates the physical address in the network of the device that originated the diagnostic (value between 0 and 125).

The next bytes contain the head diagnostics and other diagnostics of modules that constitute your network.

The head diagnostics are formed by two clusters of 12 bytes each. The 12 first have the A network head information, and the following 12 have the B network information.

ATTENTION

For non-redundant slaves, the diagnostics will be available in the A network variables.

Direct Variable	Description	Network
%QB(n)	Physical address on network of slave device with diagnostics.	-
%QB(n+1)	Number of diagnostic bytes.	A
%QB(n+2)	Programmed parameters.	
%QB(n+3)	Sustaining time without master.	
%QB(n+4)	Time of module startup after hot swap.	
%QB(n+5)	Head current status.	
%QB(n+6)	General diagnostics.	
%QB(n+7)	Reserved.	
%QB(n+8)	System general status.	

Direct Variable	Description	Network
%QB(n+9)	Status of modules 0 to 7.	
%QB(n+10)	Status of modules 8 to 15.	
%QB(n+11)	Status of modules 16 to 19.	
%QB(n+12)	Address key value.	
%QB(n+13)	Number of diagnostic bytes.	
%QB(n+14)	Programmed parameters.	
%QB(n+15)	Sustaining time without master.	
%QB(n+16)	Time of module startup after hot swap.	
%QB(n+17)	Head current status.	
%QB(n+18)	General diagnostics.	B
%QB(n+19)	Reserved.	
%QB(n+20)	System general status.	
%QB(n+21)	Modules 0 to 7 state.	
%QB(n+22)	Modules 8 to 15 state.	
%QB(n+23)	Modules 16 to 19 state.	
%QB(n+24)	Address key value.	
%QB(n+25)	From this byte on, the PROFIBUS network head diagnostics are placed. 10 bytes for diagnostics are allocated for each module. So the total amount of bytes is the number of modules multiplied by 10. Details of the modules' diagnostics can be seen in the Diagnostics chapter in their respective CEs.	-

Table 25: Diagnostics Example of Ponto Series

4.4.4. Other PROFIBUS Slaves

For other PROFIBUS slaves, the MasterTool IEC XE programmer provides diagnostics according to the Table 19.

The first 6 bytes are copies of the standard diagnostics from a PROFIBUS frame, common to any module, from any manufacturer.

Extended diagnostics provided by the PROFIBUS slave are allocated over the next 238 bytes.

There are no pre-defined structures for these slaves, data are allocated only in %QB variables. Up to the version 2.00 of MasterTool IEC XE programmer, for other slaves, always were allocated 244 bytes of diagnostics following the format according to the Table 19. From the version 2.01 of MasterTool IEC XE was included a parameter called "*Allocate Diagnostic Area According to the Device Description*" in configuration of the NX5001 module. This parameter defines if the allocation of the slave diagnostics will be fixed in 244 bytes (parameter value equal to *False*) or if the diagnostics quantity will be read from parameter *max_diag_data_len* present in the slave GSD file (parameter value equal to *True*).

4.4.4.0.1. Decoder

It is possible to decode extended diagnostics to PROFIBUS heads which are not from Ponto or Nexto Series using the library and the data structures available on *NX5001_Diagnostic_Structs.library* provided by the MasterTool IEC XE programmer.

This library is used to decode the diagnostics of the slave from PROFIBUS format to bit map.

Output data format follows the following mapping:

- Byte 0 to 5: Contains information of the 6 mandatory bytes (Standard Diagnostics).
- Byte 6 to 45: Contains device diagnostics.
- Byte 46 to 243: Contains channel diagnostic, as per module. For each channel, 4 bytes (DWORD) are allocated, where each bit represents the value of a code, that is, indicates codes from 1 (bit 0) up to 32 (bit 31).

When the parameter "*Allocate Diagnostic Area According to the Device Description*" is disabled, a data structure with the maximum number of bytes called *T_DIAG_SLV_GENERIC_1* is automatically allocated for the slave declared on MasterTool IEC XE.

When this parameter is enabled, one of the generic structures called *T_DIAG_SLV_GENERIC_1_XXX* is automatically allocated for the slave declared on MasterTool IEC XE. In this case, the XXX value corresponds to the size defined on *max_diag_data_len* on the slave GSD file and only the number of diagnostics bytes defined by this parameter will be allocated.

5. User Commands

When adding a PROFIBUS-DP master NX5001 device to the bus configuration, MasterTool IEC XE programmer maps %QB(n) variables in the tab *Bus I/O Mapping*, used to trigger commands to the master device.

User Commands supported by Master PROFIBUS-DP NX5001 are:

Direct Variable		Channel	Description
Variable	**Bit**		
%QB(n)	0	Enable Interface	TRUE: Enables PROFIBUS communication.
			FALSE: Disables PROFIBUS communication.
	1	Reserved	Reserved for internal use.
	2 .. 3	Reserved	Reserved.
	4	Unfreeze	Sends an Unfreeze command when a transition from FALSE to TRUE occurs. Initial Value: FALSE
	5	Freeze	Sends a Freeze command when a transition from FALSE to TRUE occurs. Initial Value: FALSE
	6	Unsync	Sends an Unsync command when a transition from FALSE to TRUE occurs. Initial Value: FALSE
	7	Sync	Sends a Sync command when a transition from FALSE to TRUE occurs. Initial Value: FALSE
%QB(n+1)	0	Group 1	TRUE: Global Control Command is sent to Slaves of Group 1.
			FALSE: Command is not sent to Group 1.
	1	Group 2	TRUE: Global Control Command is sent to Slaves of Group 2.
			FALSE: Command is not sent to Group 2.
	2	Group 3	TRUE: Global Control Command is sent to Slaves of Group 3.
			FALSE: Command is not sent to Group 3.
	3	Group 4	TRUE: Global Control Command is sent to Slaves of Group 4.
			FALSE: Command is not sent to Group 4.
	4	Group 5	TRUE: Global Control Command is sent to Slaves of Group 5.
			FALSE: Command is not sent to Group 5.
	5	Group 6	TRUE: Global Control Command is sent to Slaves of Group 6.
			FALSE: Command is not sent to Group 6.
	6	Group 7	TRUE: Global Control Command is sent to Slaves of Group 7.
			FALSE: Command is not sent to Group 7.
	7	Group 8	TRUE: Global Control Command is sent to Slaves of Group 8.
			FALSE: Command is not sent to Group 8.

Table 26: Description of User Commands

5.1. Project with Redundancy and User Commands

In projects where there's half-cluster redundancy, i.e. where there's PROFIBUS master redundancy, the user commands have the redundancy logic action influence, thus the redundancy logic acts on these bits.

The figure below has a sample code to be applied by the user to avoid that his/her commands interfere with the redundancy logic.

This logic applies to the PROFIBUS communication enable bit, whenever it's necessary a PROFIBUS redundant head switchover, e.g. in a maintenance procedure, where all active slaves must be in a single network.

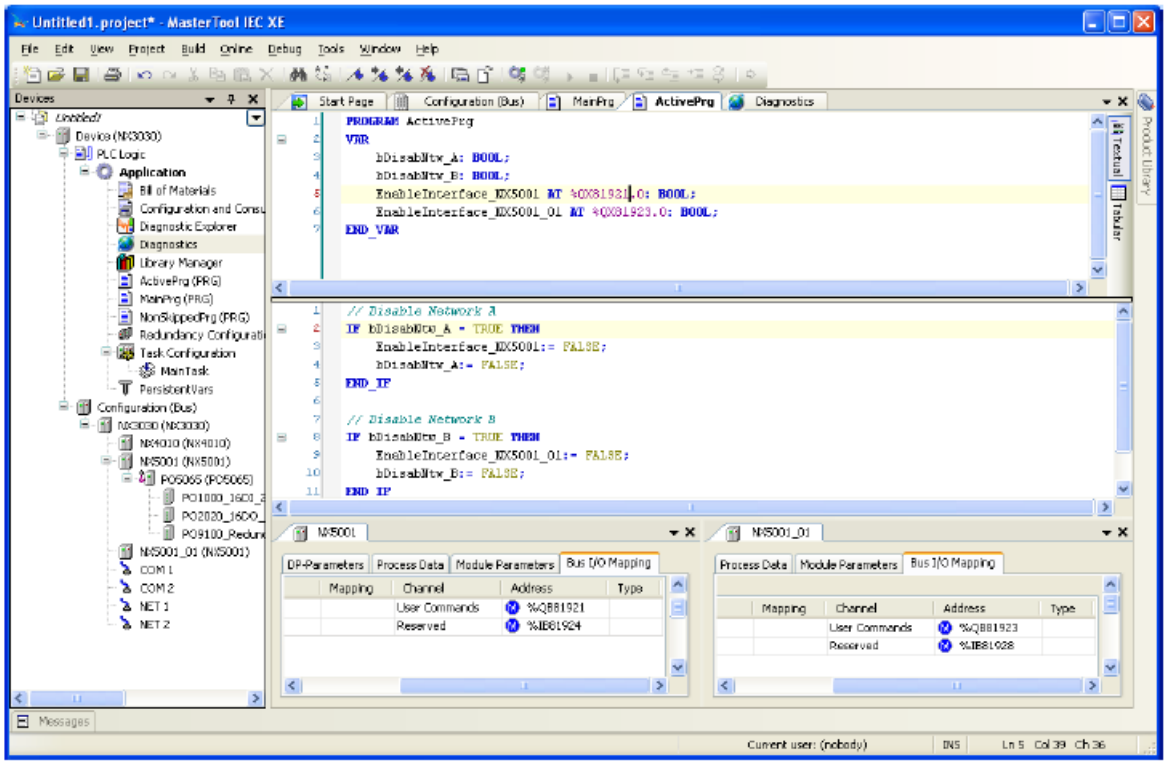

Figure 9: Example Code for Redundant Slaves Switchover Execution

At disabling the communication of one NX5001 PROFIBUS-DP master the communication with its PROFIBUS slaves is ceased. In case there are active slaves in the network, a PROFIBUS heads switchover will happen to another network, as the other NX5001 will continue to communicate.

On the next cycle, the redundancy logic will act and will enable again the NX5001 previously disabled by the example code and its redundant slaves will act as stand-by slaves.

6. DPV1 Commands

The PROFIBUS-DP Master NX5001 supports DPV1 commands (from firmware version 1.2.0.5 and MasterTool IEC XE 2.01). Therefore, it is necessary to include the NX5001 module on the bus and then manually add the library *LibDPV1Command* to the project. It supports read and write commands of the extension PROFIBUS DPV1 in slaves that have implemented this functionality. DPV1 commands enable the exchange of data acyclically with PROFIBUS slaves that support DPV1. Access to these commands is done via function block *DPV1Command*.

ATTENTION

The NX5001 modules do not support DPV1 commands when configured on a project with half cluster redundancy, so the *LibDPV1Command* library should only be used in simple designs, with or without network redundancy.

The input parameters for this function block are described below:

Parameter	Type	Description	Valid Values
uiNX5001Rack	UINT	Rack number of the NX5001 device that will receive the command.	0 .. 31
uiNX5001Slot	UINT	Slot number of the NX5001 device that will receive the command.	0 .. 31
byCommandType	DPV1_COMMAND	DPV1 command to be executed.	1 (READ) / 2 (WRITE)
byRemoteAddress	BYTE	PROFIBUS station address that will receive the command.	0 .. 125
bySlaveSlot	BYTE	PROFIBUS slave slot address that will receive the command.	0 .. 255
byIndex	BYTE	Index address that will be accessed inside the slave.	0 .. 255
byDataLength	BYTE	Size, in bytes, of data to be written/read. If attributed value 0, size will be 240.	1 .. 240
abyDataWrite	ARRAY[0..239] OF BYTE	Data to be written. In case of read, this parameter isn't used.	-
xExecute	BOOL	Trigger to send DPV1 command.	-

Table 27: Function block DPV1Command Input Parameters

The output parameters for this function block are described below:

Parameter	Type	Description
xBusy	BOOL	Returns TRUE during DPV1 command execution.
xDone	BOOL	Returns TRUE if DPV1 command was sent and treated successfully.
xError	BOOL	Returns TRUE if there's an error in the sending or treating of DPV1 command by the PROFIBUS slave.
byGeneralStatus	GENERAL_STATUS	Returns general errors in DPV1 command sending, such as invalid parameters or sending failure. If there are no errors, the return is NO_ERROR.

Parameter	Type	Description
byDPV1ErrorCode1	DPV1_ERROR_CODE	Returns specific errors in DPV1 command treating. The error codes follow the PROFIBUS norm. If there are no errors, the return is DPV1_NO_ERROR.
byDPV1ErrorCode2	BYTE	Returns vendor specific error codes of the device.
byRemoteAddressAnswer	BYTE	Returns the written/read station address.
bySlaveSlotAnswer	BYTE	Returns the written/read slave slot.
byIndexAnswer	BYTE	Returns the written/read index.
byDataLengthAnswer	BYTE	Returns the written/read data size in bytes.
abyDataRead	ARRAY[0..239] OF BYTE	If a reading command, this parameter will contain all slave data read. If a writing command, it will contain the written data.

Table 28: DPV1Command Function Block Output Parameters

To use the function block and send DPV1 commands, the user must inform the NX5001 identification parameters to which the command will be sent (*uiNX5001Rack* and *uiNX5001Slot*), the DPV1 command (READ or WRITE), the regular DPV1 parameters (*byRemoteAddress*, *bySlaveSlot*, *byIndex* and *byDataLength*), and the *abyDataWrite* parameter, used in writing commands.

Once the input parameters are defined, the function block will start working when the *xExecute* parameter turns TRUE. During the function block's operation, this parameter must remain TRUE, and only go to FALSE by the end of the execution, when *xDone* or *xError* turns TRUE. It is important to remind that the user application is the one responsible for monitoring the function block's parameters (*xExecute*, *xBusy*, *xDone* and *xError*). The table below presents a summary of the function block's *DPV1Command* operation, followed by a usage example.

xExecute	xBusy	xDone	xError	Description
FALSE	X	X	X	The function block doesn't execute. If xExecute is set to FALSE after a command, the function block is restarted and the output variables are reset, indicating that a new command may be executed.
TRUE	FALSE	X	X	Triggers a read or write DPV1 command by setting the xExecute input parameter to TRUE.
X	TRUE	X	X	Function block busy. Processing DPV1 read/write command.
TRUE	FALSE	FALSE	TRUE	Function block call failure. Variable byGeneralStatus indicates the type of error (consult Table 30). For errors such as ERROR_DPV1, byDPV1ErrorCode1 indicates the DPV1 service specific error (consult Table 31).
TRUE	FALSE	TRUE	FALSE	Function block call and DPV1 service concluded successfully: the output parameters contain the read/written PROFIBUS slave data.

Table 29: Summary of DPV1Command Function Block Operation

If any parameter or command is invalid, *xExecute* will turn back to FALSE and *byGeneralStatus* indicates which parameter/command is invalid. Also, if there are invalid parameters or there's a communication failure for any reason, the *xError* flag will turn TRUE and *byGeneralStatus* will indicate the nature of the problem, according to the table below:

byGeneralStatus	Error Code	Description
ERROR_NOERROR	0	No errors.
ERROR_SDO_COMM	1	Communication failure between CPU and NX5001 module (SDO frame transmission failure).
ERROR_INVALID_REM_ADD	2	Informed remote address out of range (0 to 125).
ERROR_INVALID_LENGTH	3	Informed data size out of range (1 to 240 bytes).
ERROR_INVALID_COMMAND	4	Informed command invalid (1 to read, 2 to write).
ERROR_INVALID_RACK_NUM	5	Invalid rack number of NX5001 module (0 to 31).
ERROR_INVALID_SLOT_NUM	6	Invalid slot number of NX5001 module (0 to 31).
ERROR_ABSENT_MODULE	7	Absent NX5001 from the bus.
ERROR_MODULE_NOT_DECLARED	8	Non-declared module at the address sent.
ERROR_INVALID_MODULE_DECLARED	9	Module different than NX5001 declared at the address sent.
ERROR_SLAVE_COMM	12	Failure in PROFIBUS communication with slave to which the command was sent.
ERROR_PBUS_COMM	13	Communication failure with all PROFIBUS slaves.
ERROR_DPV1_TIMEOUT	14	Communication time-out between PROFIBUS Master and Slave.
ERROR_PBUS_NOT_OPERATE	15	PROFIBUS Master is not in OPERATE mode.
ERROR_INTERFACE_DISABLED	16	PROFIBUS NX5001 module is disabled by user command.
ERROR_INTERNAL	17	Internal error in PROFIBUS Stack.
ERROR_UNKNOWN	18	Unknown Error.
ERROR_DPV1	128	Error in DPV1 command. For further details, consult variable byDPV1ErrorCode1.

Table 30: byGeneralStatus Possible Values

IF the DPV1 command was correctly sent but a DPV1 service error happens, *xError* flag will turn TRUE, *byGeneralStatus* will indicate *ERROR_DPV1* and *byDPV1ErrorCode1* will indicate the specific error, as determined by the PROFIBUS-DP norm.

byDPV1ErrorCode1	Error Code	Description
READ_ERROR	0xA0	Application. Error Class 0xA
WRITE_ERROR	0xA1	
MODULE_FAILURE	0xA2	
VERSION_CONFLICT	0xA8	
FEATURE_NOT_SUPPORTED	0xA9	
INVALID_INDEX	0xB0	Access. Error Class 0xB
WRITE_LENGTH_ERROR	0xB1	
INVALID_SLOT	0xB2	
TYPE_CONFLICT	0xB3	
INVALID_AREA	0xB4	
STATE_CONFLICT	0xB5	
ACCESS_DENIED	0xB6	
INVALID_RANGE	0xB7	
INVALID_PARAMETER	0xB8	
INVALID_TYPE	0xB9	
READ_CONST_CONFLICT	0xC0	Resource. Error Class 0xC
WRITE_CONST_CONFLICT	0xC1	
RESOURCE_BUSY	0xC2	
RESOURCE_UNAVAILABLE	0xC3	

Table 31: DPV1 Service Error codes and byDPV1ErrorCode1 Possible Values

If the command is sent and treated correctly, the *xDone* flag will turn TRUE and variables *byRemoteAddressAnswer*, *bySlaveSlotAnswer* and *byIndexAnswer* will contain an answer with the command's destiny. Variables *byDataLengthAnswer* and *abyDataIn* will contain the size of the read data and the data if it was a read command, and 0 if it was a write command.

ATTENTION

Only one DPV1 command can be treated at a time. Therefore, if the *DPV1Command* function block is instantiated and called many times in sequence, while the first command sent is still being treated, the other commands will have their *xBusy* parameter set to TRUE, and will remain waiting. After treating the first command, it will start treating the next, and so on as long as their *xExecute* is TRUE. It is possible to treat up to four DPV1 commands simultaneously, as long as they are in four different NX5001 modules.

6.1. Usage Example

Below is presented a usage example of the DPV1Command function block in ST language:

```
PROGRAM UserPrg
VAR
  DPV1Test0 : DPV1Command;
  DPV1Test1 : DPV1Command;
  DPV1Test2 : DPV1Command;
  abyDataWrite : ARRAY[0..239] OF BYTE;
  byStateExec : BYTE;
END_VAR

// Verifies if PROFIBUS Master is operational
IF DG_NX5001.tMstStatus.byState <> OPERATE THEN
  RETURN;
END_IF

abyDataWrite [0] := 16#00;
abyDataWrite [1] := 16#05;
abyDataWrite [2] := 16#82;
abyDataWrite [3] := 16#BE;
abyDataWrite [4] := 16#04;
abyDataWrite [5] := 16#00;
abyDataWrite [6] := 16#8D;
abyDataWrite [7] := 16#91;
abyDataWrite [8] := 16#21;
abyDataWrite [9] := 16#01;
abyDataWrite [10] := 16#00;
abyDataWrite [11] := 16#04;

DPV1Test0(uiNX5001Rack:= 0, uiNX5001Slot := 7, byCommandType := 2,
   byRemoteAddress := 2, bySlaveSlot := 0, byIndex := 2, byDataLength :=
   12, abyDataWrite := abyDataWrite);

DPV1Test1(uiNX5001Rack:= 0, uiNX5001Slot := 7, byCommandType:= 1,
   byRemoteAddress := 2, bySlaveSlot := 0, byIndex := 2, byDataLength :=
   240);

DPV1Test2(uiNX5001Rack:= 0, uiNX5001Slot := 7, byCommandType := 1,
   byRemoteAddress := 2, bySlaveSlot := 0, byIndex := 16, byDataLength :=
   5);

CASE byStateExec OF
  0: // Start the program
    DPV1Test0.xExecute  := FALSE;
    DPV1Test1.xExecute  := FALSE;

    byStateExec := 1; // Enter Write state
```

```
  1: // Write State
    DPV1Test0.xExecute  := TRUE;

    IF DPV1Test0.xDone THEN
      byStateExec := 2; // Enter Read state
    ELSIF DPV1Test0.xError THEN
      byStateExec := 3; // Enter Error state
    END_IF

  2: // Read State
    DPV1Test1.xExecute    := TRUE;

    IF DPV1Test1.xDone THEN
      byStateExec := 0; // Enter Starting state
    ELSIF DPV1Test1.xError THEN
      byStateExec := 3; // Enter Error state
    END_IF

  3: // Error State

END_CASE

IF DPV1Test2.xDone = FALSE THEN
  DPV1Test2.xExecute := TRUE;
ELSE
  DPV1Test2.xExecute := FALSE;
END_IF
```

In the example above, the function block (FB) is instantiated three times. In the POU's body (such as *UserPrg*), some values are attributed to the data array and the FB's instances are called sequentially. In this case, the parameters are passed directly in the FB calls, without middle variables.

After the FB calls comes their execution logic. Instances *DPV1Test0* and *DPV1Test1* execute a writing and a reading of the same index in a slave, so the writing is executed in the first moment and then awaits an answer, either of failure (*xError*) or success (*xDone*). Afterwards, the same address is read and awaits conclusion. In the example, reserved spaces were left for error treatment. *DPV1Test2* instance executes a reading for a different index of the same slave and is executed independently.

7. Installation

This chapter presents procedures for installation of network interface PROFIBUS NX5001.

7.1. Mechanical Assembly

Mechanical assembly of this module is described in Nexto Series User Manual - MU214600.

The module can be installed anywhere in the bus, after the CPU. If a pair of modules is redundant, they must be placed side by side.

Module(s) NX5001 must be declared at the CPU bus through the MasterTool IEC XE, in the desired positions.

7.2. Electrical Installation

DANGER

When performing any installation of an electrical panel, make sure that its power source is TURNED OFF.

The backplane rack installation can be seen on the figure below.

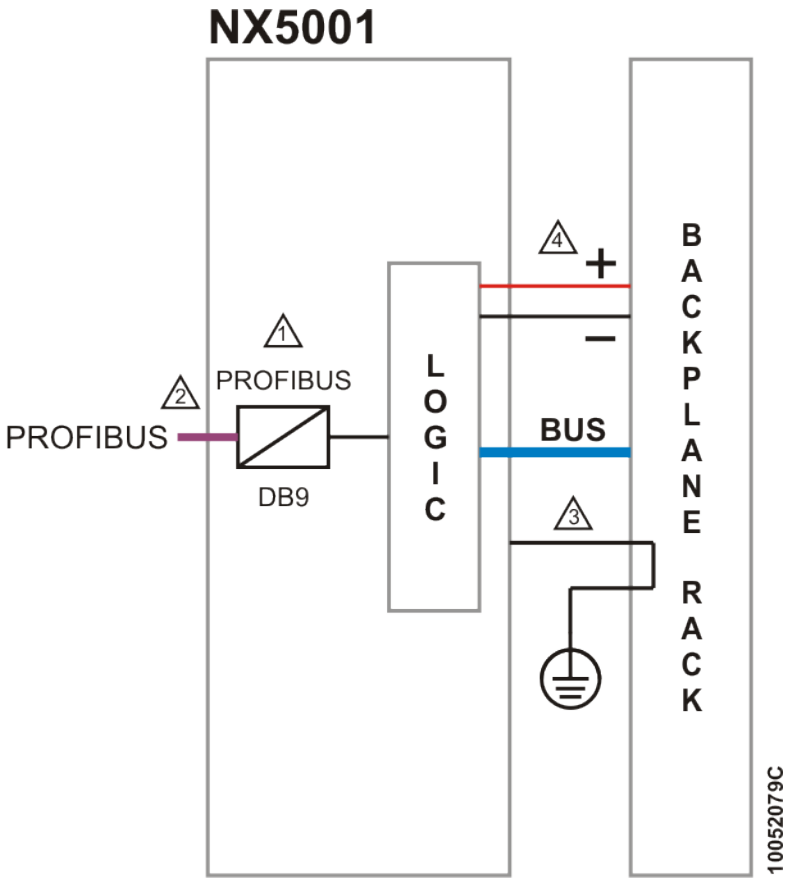

Figure 10: Electrical Diagram from NX5001

Diagram Notes:

- **1** – Standard interface for connection to PROFIBUS networks. Pin 1 of DB9 connector is connected to the protection ground of the Nexto Series rack.
- **2** – Use the supplied AL-2303 cable for PROFIBUS network and one of the following connectors:

- AL-2601 is a connector for PROFIBUS network without internal termination. It can be used to connect any PROFIBUS equipment in a position in which the termination is not required.
- AL-2602 is a connector for PROFIBUS network with internal termination. It must be used in equipment located at the ends of the PROFIBUS network. Altus also offers a second option for networks where reliability and availability are the main requirements. For these cases, module AL-2605 should be used in each end of the network and all PROFIBUS modules should use unterminated connectors (AL-2601). More information about AL-2605 module can be found in the document CE104705. It is mandatory the use of two PROFIBUS network terminations. Each termination should be placed at each end of the network.

- **3** – Module is grounded through the rack of Nexto Series.
- **4** – The NX5001 module is powered by the power supply connected to the same Nexto series rack not requiring external power source.

7.3. Installation of the PROFIBUS Network

The installation of the PROFIBUS network should be performed according to the standard EN50170. Cable and connectors used in installation can be purchased from Altus:

- AL-2601: PROFIBUS Connector
- AL-2602: PROFIBUS Terminator Connector
- AL-2605: Terminator with Power Supply Diagnostic
- AL-2303: PROFIBUS cable

ATTENTION

Addressing errors in slave devices are hard to identify. The PROFIBUS network may not detect errors when two slaves have the same address and some meters of network cable apart. It is recommended to check the addressing keys of each device before network activation.

7.4. Module Hot Swap

NX5001 module can be hot swapped. However, as it is a network interface, some cautions must be taken to make the swap in a safe way for the devices connected to it.

DANGER

Before performing a hot swap, it's important to unload eventual static energy accumulated in the body. For that, touch (with your bare hands) a metallic grounded surface before handling the modules. Such procedure guarantees the static level supported by the module won't be exceeded.

ATTENTION

It's recommended the monitoring of the hot swap diagnostics in the control application developed by the user, in order to guarantee that the value returned by the module is validated before used.
For further information regarding this resource, see Nexto Series User Manual - MU214600.

The procedure for hot swap of NX5001 requires the following cautions:

- Disconnect network cable: disconnecting the cable prevents network communications. CPU will keep the last read values for inputs, and outputs will be in safe mode.
- Unlock the module by pressing the fixation lock.
- Remove the module by pulling firmly.
- Insert the new module in the rack.
- Make sure that the fixation lock is completely locked to the rack. If not, push the module a little harder towards the rack.
- Configure, if necessary, the module with MasterTool IEC XE.
- Reconnect the PROFIBUS cable, reestablishing communication.

8. Operation

This chapter presents general information regarding operation modes, NX5001 PROFIBUS-DP master, PROFIBUS bus and CPU.

8.1. Operation Modes

The table below describes the PROFIBUS network behavior according to the NX5001 PROFIBUS-DP Master operation mode:

NX5001 Operation Modes	Description
OFFLINE	There's no communication in the PROFIBUS bus.
STOP	There's no communication between Master and Slaves. There's only data exchange between the Masters connected to the network.
CLEAR	The Master reads inputs and Slave diagnostics and the outputs' states are defined by the PROFIBUS head.
OPERATE	The Master is on full data exchange stage. On each cyclic data communication, the Master sends the output points data (output writing) to the slaves and receives the input data (input reading). The slave diagnostics messages are also received.

Table 32: NX5001 Operation Modes

ATTENTION

The update of PROFIBUS input data depends, besides of the operation state of NX5001, on the operation state of the slave device. The NX5001 module has diagnostic variables that indicate when the slave device is present and operational (*Data Exchange*). Through these variables, it's possible to check if the inputs are being updated. For a single network, these variables are located at *DG_NX5001.tMstStatus.abySlv_State.bSlave_XX* (where XX is the slave address). For a redundant network, these variables are at *DG_NX5001.tRedund.tActivNtw.bSlave_XX*.

8.2. Output State

The table below relates the CPU's operation state with the module output points' states configured in the PROFIBUS slave bus.

CPU Operation State	Output Points States (PROFIBUS slaves)
RUN	Points are updated according to the application.
STOP	The state of the points is defined by the PROFIBUS head.
BREAKPOINT	The state is frozen and updated according to the application execution.

Table 33: Update of Slaves' Output Points according to CPU State

9. Maintenance

This chapter brings information about general care, common problems found by the user and the procedures to be taken in case of errors.

9.1. Module Diagnostics

One of the features of the Nexto Series is the generation of irregularities diagnostics such as failures, errors or operating modes, enabling the operator to easily identify and solve problems that may happen to the system.

Nexto Series offers five important resources to assist the user during maintenance: Electronic Tag on Display, One Touch Diag, state and diagnostics indicators, HTML page with complete state and diagnostic list, and status mapped in internal memory.

9.1.1. Electronic Tag on Display

This is an important resource that allows the user to check the name (tag) and description that were previously defined in MasterTool IEC XE of any I/O channel or module directly on CPU display. This resource can be extremely useful when the user need to check the function of a particular point or module before connect or disconnect it from the system.

To verify the name (tag) of an I/O of a module, select any I/O point or module information by pressing quickly (less than 1 s) the diagnostics button. After selecting the module or a specific I/O point, the CPU will display the name (tag) and diagnostic information related to the point or module. To access the description of the item or the module previously selected, just press the module's diagnostics button for longer than 1 s.

Further information about this resource can be found in the Nexto Series User Manual - MU214600 and at Nexto Series CPUs User Manual – MU214605.

9.1.2. One Touch Diag

This is another innovative resource of Nexto Series. With this new concept, the user can verify the diagnostics of NX5001 or any other module present in the system, directly from the graphical display on CPU (no diagnostic codes) with just one touch on diagnostic button of corresponding module. "OTD" is a powerful tool that can be used offline (without supervisory application or programmer), making the process of locating and solving problems more effective and faster.

Further information about this resource can be found in the Nexto Series User Manual - MU214600.

Displayed messages are listed on Table 10, under General Diagnostics.

9.1.3. Status and Diagnostics Indicators

All slave modules of Nexto Series have a display with the following symbols: D, E, 0, 1 and numerical characters. The state of the symbols D, E, 0 and 1 are common to all modules from Nexto Series and can be found in the table below.

D	E	Description	Cause	Solution	Priority
Off	Off	Disconnected modules or display failure	-	Verify that the module is completely connected to the rack and if the rack is powered by an external source.	-
On	Off	Normal use	-	-	9 (Lower)
Flashing 1x	Off	Active Diagnostics	There is at least one active diagnosis related to the module	Check which is the active Diagnostics. More information can be found in the Maintenance section of this document.	8
Flashing 2x	Off	CPU in STOP mode	-	Check if the CPU is in RUN mode. More information can be found in the documentation of the CPU.	7
Flashing 3x	Off	Reserved	-	-	6
Flashing 4x	Off	Non fatal fault	Failure in some hardware or software component, which does not have impact on the basic functionality of the product	Check the module diagnostic information. If it is a hardware fault, provide the replacement of this part. If it is a software fault, please contact the Technical Support.	5
Off	Flashing 1x	Parameterization error	The module is not parameterized or not received new parameterization	-	4
Off	Flashing 2x	Master loss	Loss of communication between the module and the CPU	Verify that the module is completely connected to the rack. Check if the CPU is in RUN mode.	3
Off	Flashing 3x	Reserved	-	-	2
Off	Flashing 4x	Fatal hardware error	-	If this failure happens, please, contact Altus support	1 (Higher)

Table 34: Status of Symbols D and E

Segments 0 and 1 are usually switched off, however, when the module is in diagnostics mode (Electronic Tag on Display and One Touch Diag), these two segments start flashing.

Numerical characters are not used by NX5001 module.

9.1.4. Diagnostic LEDs

PROFIBUS-DP Master NX5001 has two LEDs in its front panel to indicate diagnostics related to the PROFIBUS interface:

- **LED ST**: green
- **LED ER:** red

LED ST	LED ER	Meaning	Causes
Off	Off	Module Off. Hardware Failure.	No power supply. Hardware failure.
On	Off	Slave communication established.	All slave communications were reestablished.
On	Flashing	There are present and absent slaves in the PROFIBUS network.	Some PROFIBUS slaves are exchanging I/O data with NX5001, others aren't. PROFIBUS termination problem.
Off	On	No activity in the PROFIBUS network.	Loss of communication with all slaves. PROFIBUS network cable not connected. PROFIBUS network damaged. PROFIBUS termination problem.
Flashing	Off	NX5001 received configuration.	NX5001 has received the CPU configuration, but the communication wasn't enabled by the application.
Flashing 4x	Off	NX5001 is unconfigured.	NX5001 hasn't received the slave and the PROFIBUS bus configuration from the CPU.
On	On	NX5001 initialization.	NX5001 was connected to the Nexto bus, or restarted.

Table 35: LED ST and ER

9.1.5. HTML Page with Complete Status and Diagnostics List

Another way to access diagnostic information is via a HTML page. The Nexto Series CPU has built in HTML page server that provides all status and diagnostics information about the system. These pages can be accessed using a simple browser.

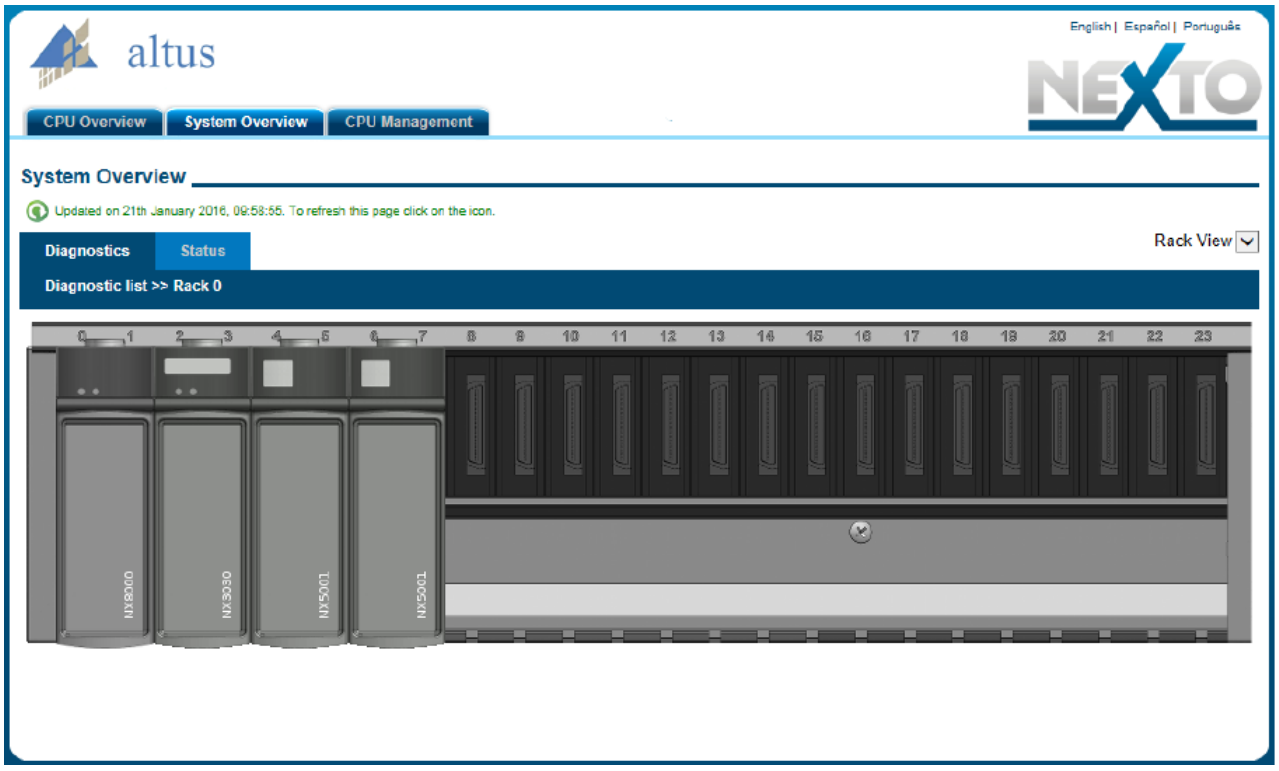

Figure 11: Webpage

Further information about this resource can be found in the Nexto Series User Manual - MU214600.
The list of diagnostic messages displayed by HTML page can be found on table below.
The following messages can be seen on *Status* tab from HTML page:

Message	Meaning
CONFIG - Modules has active diagnostics	Master module PROFIBUS-DP NX5001 has some diagnostics.
CONFIG - There aren't active diagnostics	Master module PROFIBUS-DP NX5001 hasn't active diagnostics.
CONFIG - Module in non-operational state	Master module PROFIBUS-DP NX5001 isn't in normal functioning.
CONFIG - Module in operational state	Master module PROFIBUS-DP NX5001 is in normal operation.
CONFIG - There is an error in module configuration	There're configuration and/or parameterization errors in the PROFIBUS-DP NX5001 Master.
CONFIG - There aren't errors in module configuration	The PROFIBUS-DP NX5001 Master configuration was completed successfully.
INT - Module reset occurred due to watchdog	The PROFIBUS-DP NX5001 Master was restarted by the watchdog logic action.
INT - No reset due to watchdog	The PROFIBUS-DP NX5001 Master wasn't restarted by the watchdog logic action.
INT - One Touch Diag switch is not operational	Failure in the diagnostics button in the PROFIBUS-DP NX5001 Master.
INT - One Touch Diag switch is operational	There isn't any failure in the diagnostics button in the PROFIBUS-DP NX5001 Master.
CONFIG - There is at least one PROFIBUS slave without configuration	There's at least one PROFIBUS slave not configured in the *Device Tree* in the MasterTool IEC XE programmer. It's necessary to consult the NX5001 diagnostics structure to identify which slave isn't configured (for further details, see Master Diagnostics, Map of Configured Slaves).
CONFIG - There aren't PROFIBUS slaves without configuration	All PROFIBUS slaves were configured in the Device Tree in the MasterTool IEC XE programmer.
CONFIG - There is at least one PROFIBUS slave missing	The PROFIBUS-DP NX5001 Master module isn't exchanging data with at least one slave (*DataExchange* command). The slave device may be connected incorrectly to the PROFIBUS network, or has a major failure. It's necessary to see the NX5001 diagnostic structure to identify which slave is absent (details can be found in Master Diagnostics, Map of Present Slaves).
CONFIG - There aren't PROFIBUS slaves missing	The PROFIBUS-DP NX5001 Master module is exchanging data (*DataExchange* command) with all slaves.
CONFIG - There is at least one PROFIBUS slave with active diagnostics	There is at least one PROFIBUS slave with active diagnostics. It's necessary to see the NX5001 diagnostic structure to identify which slave has active diagnostics (details can be found in Master Diagnostics, Map of Slaves with Diagnostics).
CONFIG - There aren't PROFIBUS slaves with active diagnostics	There isn't PROFIBUS slaves with active diagnostics.
CONFIG - There is an error condition in PROFIBUS fieldbus communication	There is failure in the PROFIBUS Network communication. The PROFIBUS bus termination must be checked, together with the cable connection and the slave parameterization.
CONFIG - The PROFIBUS fieldbus communication is operational	There isn't failure in the PROFIBUS Network communication.
CONFIG - The module has not been initialized with parameters	NX5001 hasn't received the bus and/or slave configuration from the CPU.

Message	Meaning
CONFIG - The module has been initialized with parameters	NX5001 has received the bus and/or slave configuration from the CPU.
CONFIG - Module is enabled by user command	The PROFIBUS-DP NX5001 Master module was enabled by application, in other words, the BIT0 from the User Control Variable is in logic level 1.
CONFIG - Module is disabled by user command	The PROFIBUS-DP NX5001 Master module was disabled by application, in other words, the BIT0 from the User Control Variable is in logic level 0.
REDUND - The PROFIBUS master redundancy is enabled	The PROFIBUS-DP NX5001 Master redundancy is enabled, in other words, a project was created with the redundancy configuration option equal to With Redundancy.
REDUND - The PROFIBUS master redundancy is disabled	The PROFIBUS-DP NX5001 Master redundancy is disabled, in other words, a project was created with the redundancy configuration option equal to Without Redundancy.
CONFIG - PROFIBUS master operation mode: OFFLINE	The PROFIBUS-DP NX5001 Master module is in OFFLINE operation mode.
CONFIG - PROFIBUS master operation mode: STOP	The PROFIBUS-DP NX5001 Master module is in STOP operation mode.
CONFIG - PROFIBUS master operation mode: CLEAR	The PROFIBUS-DP NX5001 Master module is in CLEAR operation mode.
CONFIG - PROFIBUS master operation mode: OPERATE	The PROFIBUS-DP NX5001 Master module is in OPERATE operation mode.

Table 36: List of Status Messages of NX5001 displayed on WEB server

9.1.6. Status and Diagnostics Mapped in internal Memory

Information about the module's state can also be obtained through diagnostics readings in MasterTool IEC XE and supervisory systems or HMIs.

In some cases, the diagnostics indication through internal variables can be more specific than through the LEDs or the display, since the latter allows only four types of indication (from one to four blinks). Since in the diagnostics, a large amount of information can be accessed.

For modules such as PROFIBUS fieldbus head of Nexto and Ponto Series, diagnostic information is also sent to the PROFIBUS master equipment.

9.2. Preventive Maintenance

- It must be verified, each year, if the interconnection cables are connected firmly, without dust accumulation, mainly the protection devices.
- In environments subjected to excessive contamination, the equipment must be periodically cleaned from dust, debris, etc.
- The TVS diodes used for transient protection caused by atmospheric discharges must be periodically inspected, as they might be damaged or destroyed in case the absorbed energy is above limit. In many cases, the failure may not be visual. In critical applications, it's recommendable the periodic replacement of the TVS diodes, even if they do not show visual signals of failure.

10. Appendix A – Usage Example

10.1. Architecture

The following example demonstrates how to use the NX5001 module to control a simple PROFIBUS network, consisting of two PROFIBUS slave devices.

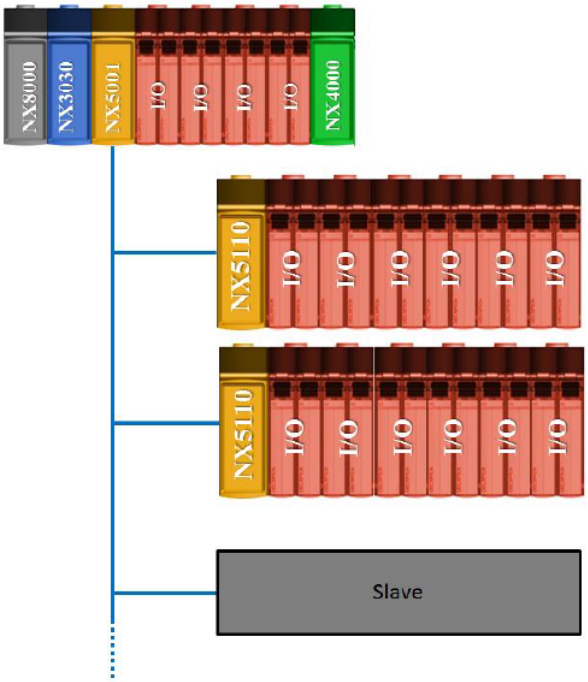

Figure 12: Simple PROFIBUS-DP Architecture

The slave devices are both Nexto Series NX5110 heads, followed by their I/O modules.

10.2. Project Creation

The first step is to create a new project for the NX3010 CPU using the MasterTool IEC XE programmer, as described in the MasterTool IEC XE User Manual - MU299609, chapter Quick Start.

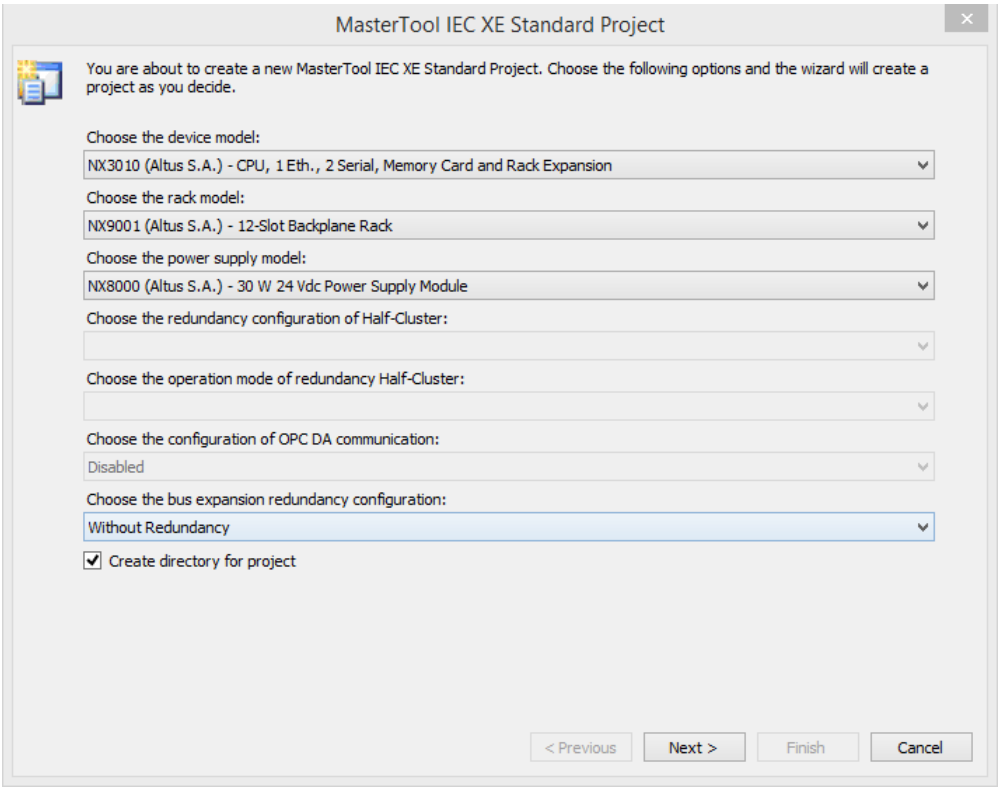

Figure 13: Hardware modules selection

10.3. PROFIBUS Network Configuration

To insert PROFIBUS-DP Master NX5001, you should select the group *Fieldbus Interfaces*.
Drag NX5001 to the desired position on Nexto bus.

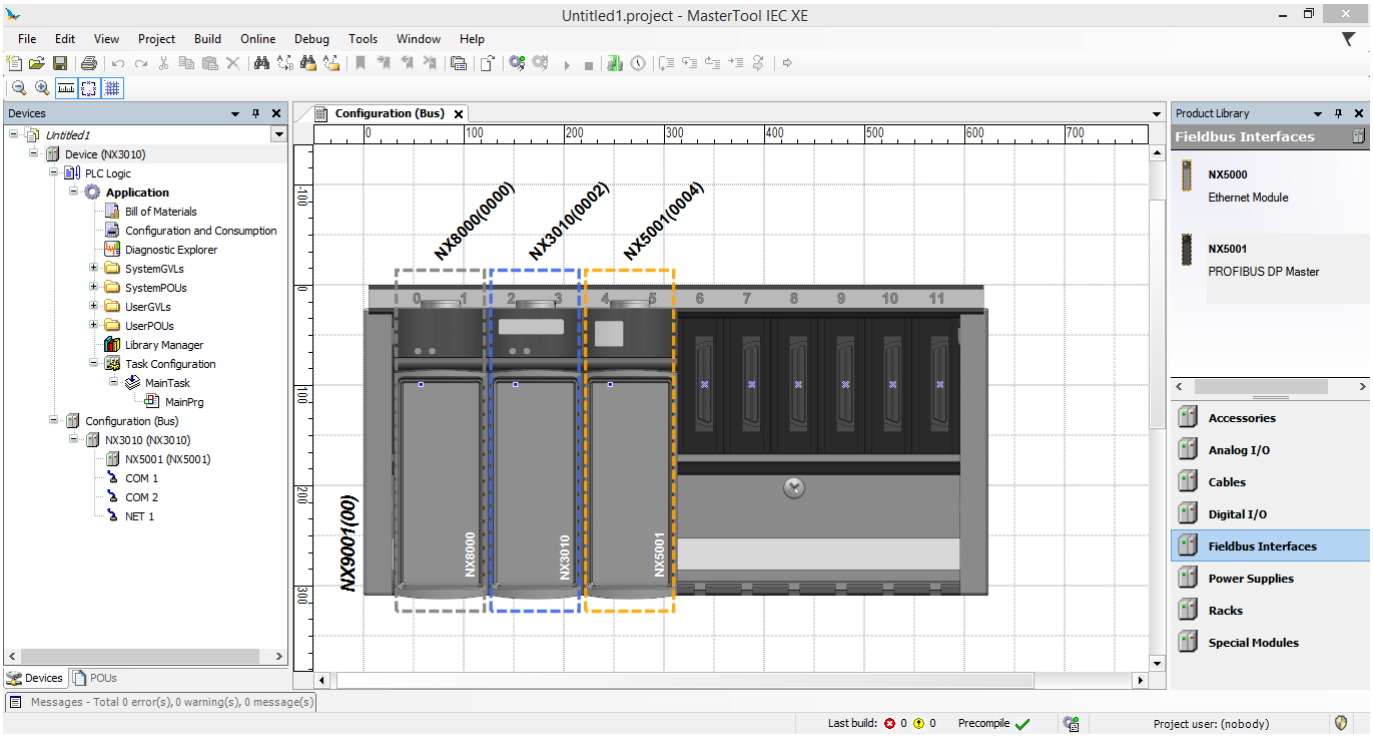

Figure 14: Selection of PROFIBUS-DP Master NX5001

To add slaves to the PROFIBUS network, it's necessary to right-click on NX5001 and select the option *Add Device...*.

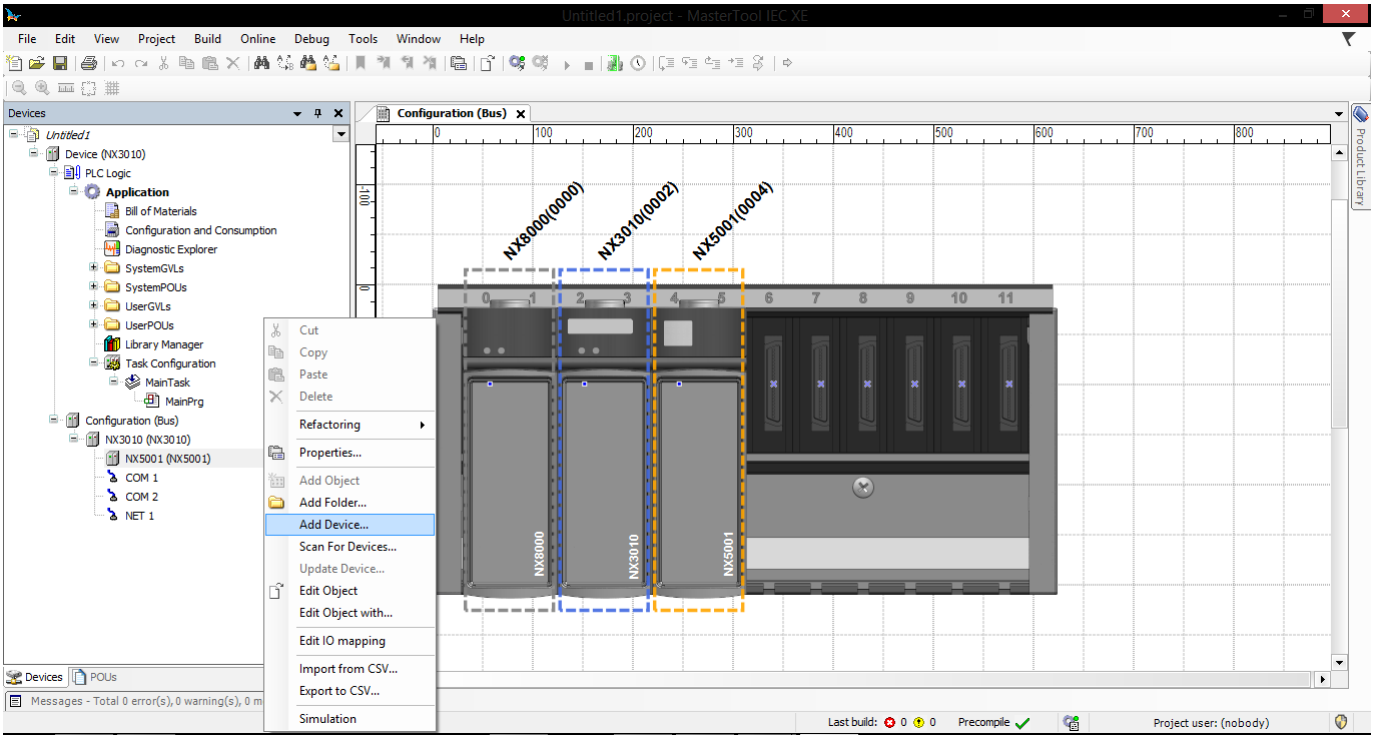

Figure 15: Adding PROFIBUS Slave Devices

Slaves displayed at figure below are present in the MasterTool IEC XE programmer so it is not necessary to install their GSD files.

To assemble the network, select the NX5110 PROFIBUS head.

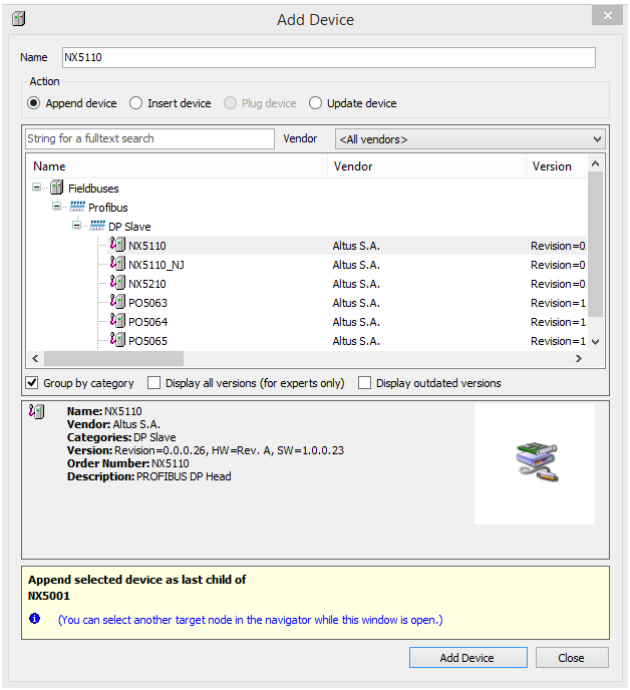

Figure 16: Selection of NX5110 head

NX5110 was included in the project, appearing just below NX5001 on the *Device Tree*.

To add modules that will be in this network node, you have to click right NX5110. Selecting the option *Add Device*, a window will open with the available modules for this slave device.

For this example, four modules were added:

- NX1001 24 Vdc 16 DI
- NX2001 24 Vdc 16 DO Transistor
- NX6000 8 AI Voltage/Current
- NX6100 4 AO Voltage/Current

As can be seen in figure below, the *Device Tree* now has a NX5001 master, followed by the NX5110 slave and four I/O modules.

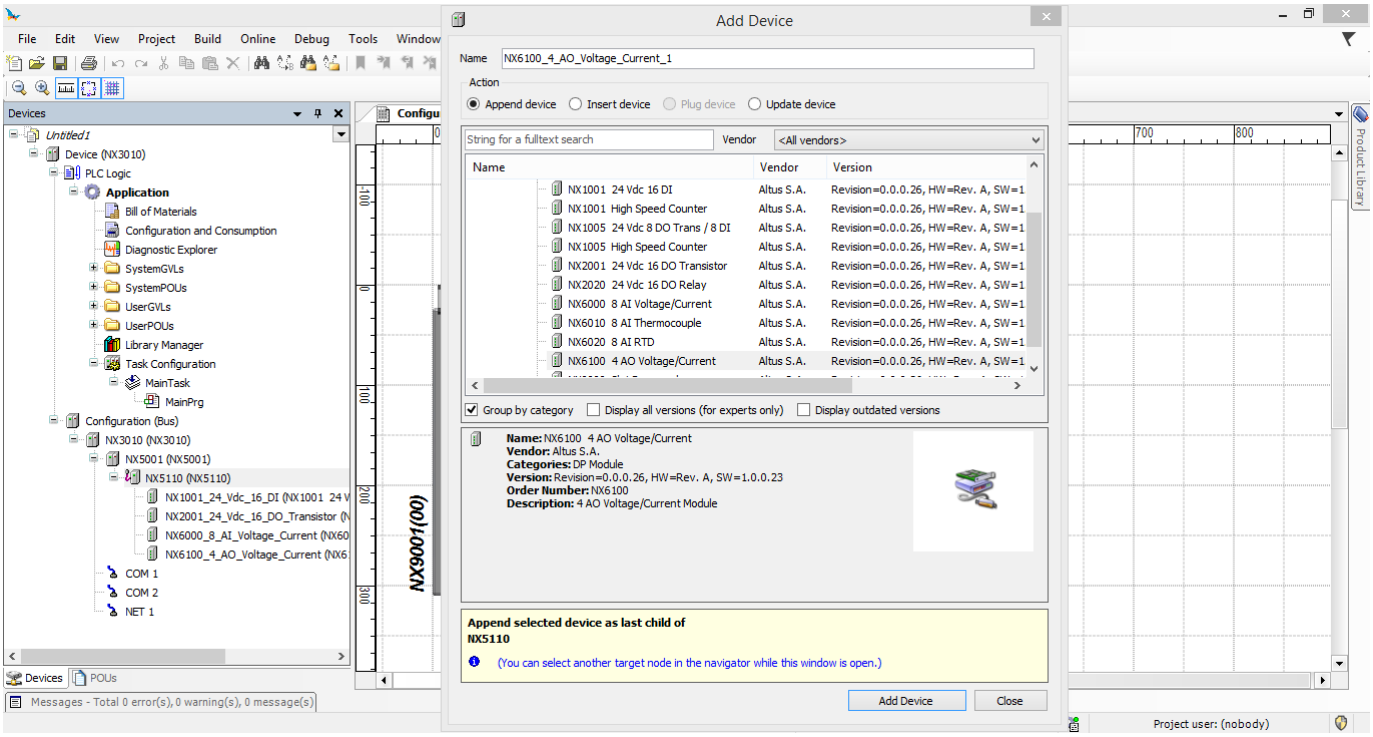

Figure 17: Adding Modules to the PROFIBUS Slave

The exact same procedure must be followed to add the next slave.

10.4. Parameterization of the PROFIBUS Network

10.4.1. Master Parameters

PROFIBUS bus parameters can be edited by clicking on Master PROFIBUS-DP NX5001 and selecting tab *General*. The figure below shows standard screen displayed by MasterTool IEC XE.

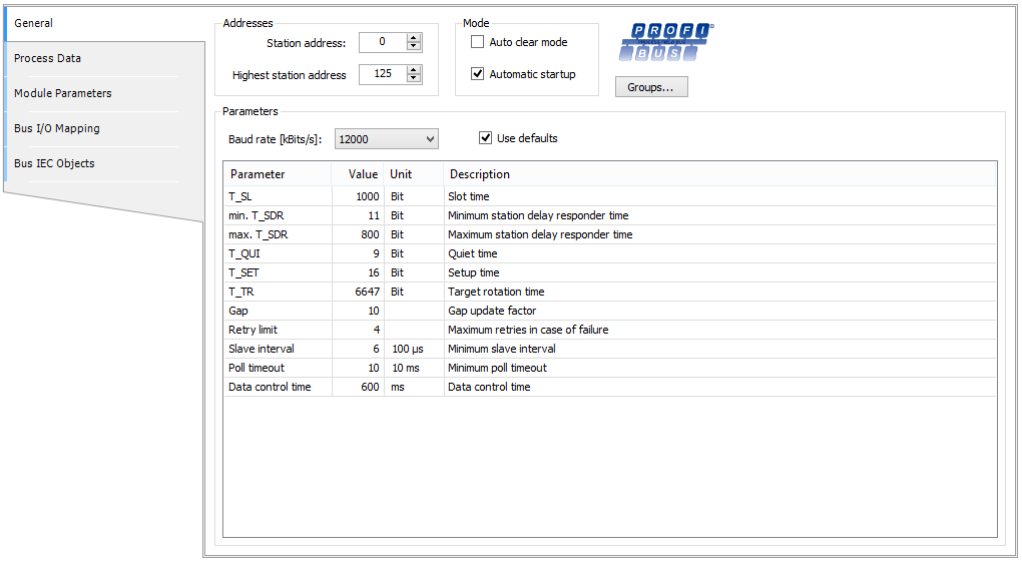

Figure 18: Parameterization of PROFIBUS Bus

Whenever the option *Use defaults* is marked, it is possible to edit only the following parameters:

- Station address
- Auto clear mode
- Automatic startup
- Highest station address
- Baud rate

ATTENTION

Parameters *Auto clear mode* and *Automatic startup* are not supported by the PROFIBUS-DP Master NX5001 in this product review.

Other parameters are changed automatically through alteration of Baud rate. In case it's necessary to change them, the option *Use defaults* must be unmarked.

10.4.1.1. Details of bus and PROFIBUS-DP Master NX5001 parameters

The table below contains relevant information about parameters within tab *General* of PROFIBUS-DP Master NX5001 of MasterTool IEC XE programmer.

Parameters	Description	Default	Possibilities	Note/unit
Station address	Master address on PROFIBUS network.	0	0 .. 125	
Highest station address	Larger device address on PROFIBUS network.	125	0 .. 125	
Auto clear mode	This parameter is not supported in this product review.	Disabled	Enabled / Disabled	
Automatic startup	This parameter is not supported in this product review.	Enabled	Enabled / Disabled	
Baud rate [kBits/s]	Communication rate.	12000	9.6 .. 12000	List of valid values is on Table 38
T_SL	Time that master will wait for an answer. If the time slot has passed and the answer has not been received, the request will be resent until to the repetition limit is reached (Slot Time).	As per selected baud rate	37 .. 16383	Bit
min. T_SDR	Time that the slave must wait before answer a request (Min. Station Delay Responder).	As per selected baud rate	1 .. 65535	Bit
max. T_SDR	Time that master must wait between have sent a request and send a new one (Max. Station Delay Responder).	As per selected baud rate	1 .. 65535	Bit
T_QUI	Number of bit time the master wait in each transmission, before start sending data (Quiet Time).	As per selected baud rate	0 .. 127	Bit

Parameters	Description	Default	Possibilities	Note/unit
T_SET	Setup time.	As per selected baud rate	0 .. 255	Bit
T_TR	Target rotation time.	As per selected baud rate	1 .. 16777215	Bit
Gap	Gap update factor.	10	1 .. 100	
Retry limit	Limit of retries in case of errors.	As per selected baud rate	1 .. 15	
Slave interval	Minimum time between two accesses of one master to the same slave.	6	1 .. 65535	x 100 μs
Poll timeout	Poll time-out is relevant only when there is a master of class 2 on the network, because it defines the maximum time of a communication between two masters.	10	0 .. 65535	x 10 ms
Data control time	Control time interval.	As per selected baud rate	1 .. 65535	ms

Table 37: List of Parameters from PROFIBUS bus and from Master NX5001

10.4.1.1.1. Station Address

This parameter defines the PROFIBUS-DP Master NX5001's network address.

10.4.1.1.2. Highest Station Address (HSA)

This parameter defines the largest bus address up to which the master will look for the next active master to transfer bus right control.

ATTENTION

This parameter must be set to at least the master's address.

10.4.1.1.3. Auto Clear Mode

ATTENTION

Parameter *Auto clear mode* is not supported by PROFIBUS-DP Master NX5001 in this product review.

10.4.1.1.4. Automatic Startup

ATTENTION

Parameter *Automatic startup* is not supported by PROFIBUS-DP Master NX5001 in this product review.

10.4.1.1.5. Baud Rate [kBits/s]

The following Baud Rates are supported by the PROFIBUS-DP Master NX5001:

Baud Rate (kBits/s)
9.6
19.2
45.45
93.75
187.5
500
1500
3000
6000
12000

Table 38: Supported Baud Rates

10.4.1.1.6. T_SET (Setup Time)

Setup time is the minimum period (latency) between receiving a confirmation and sending a new request.

10.4.1.1.7. T_TR (Target Rotation Time)

This parameter is given in *bit times* and it is usually calculated by configuration tools. It's the time to pass the token across the network and returns to its initial master. When there are multiple masters, this includes the total time for each master to complete its I/O cycle, pass the token to the next master and to return to the initial master. Some factors directly influence T_TR: baud rate, the amount of slaves with cyclic data exchange, the total amount of I/Os during data exchange and the amount of masters.

10.4.1.1.8. Gap (Gap Update Factor)

Indicates the number of token rotations between requests to a new master.

10.4.1.1.9. Data Control Time

This parameter defines the control time interval. After this time expires, the master (class 1) sends status reports automatically through the "*Global Control Command*".

The declaration of memory area allocated for PROFIBUS network diagnostics (details in section Diagnostics) is in the tab *Module Parameters*.

MasterTool IEC XE programmer automatically fills this field automatically, but enables the user to edit it by double clicking over the corresponding *Value* column item.

Parameter	Meaning	Minimum	Default	Maximum
%Q Start Address of Module Diagnostics Area	Variable address %QB, whose value indicates the initial byte of the diagnostics area from the PROFIBUS-DP Master NX5001.	0	Inserted by the MasterTool IEC XE programmer according to bus configuration.	The limit depends on the CPU model (see Nexto Series CPUs User Manual)
%Q Start Address of Slaves Diagnostics Area	Variable address %QB, whose value indicates the initial byte of the diagnostics area from the PROFIBUS slaves.	0	Inserted by the MasterTool IEC XE programmer according to bus configuration.	The limit depends on the CPU model (see Nexto Series CPUs User Manual)
Network Redundancy	TRUE: enables the PROFIBUS network redundancy. FALSE: disables the PROFIBUS network redundancy.	FALSE	FALSE	TRUE
Failure Mode	TRUE: enables the switchover in case of PROFIBUS module failure. FALSE: disables the switchover in case of PROFIBUS module failure.	FALSE	TRUE	TRUE
Allocate Diagnostic Area According to the Device Description	TRUE: allocates only the diagnostics area of each slave defined in the GSD file. FALSE: allocates the standard 244 byte diagnostics area for each slave.	FALSE	TRUE	TRUE

Table 39: Details of Parameters Configuration of NX5001 Master

Note:

Failure Mode: This parameter is valid when there is a half-cluster redundancy (for further information see the Nexto Series CPUs User Manual - MU214605).

10.4.2. Slaves Parameters

10.4.2.1. Nexto Series Slaves

After a double click on the NX5110 at the *Device Tree*, the configuration and information tabs are available.

General information about the module, such as name, manufacturer, version and others, are in the *Information* tab.

Parameterization of NX5110 is edited in the parameters available in the main tab, *General*, as shown in the figure below.

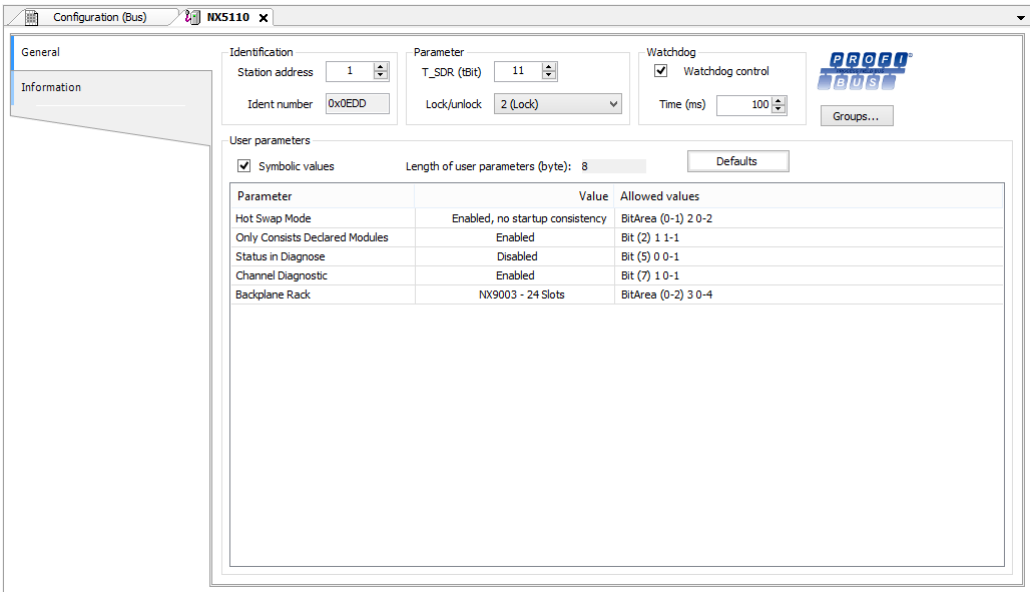

Figure 19: Parameterization of PROFIBUS Slave

Parameters	Description	Default	Possibilities	Note / Unit
Station address	Slave address on PROFIBUS network.	1	0 .. 99	
Ident number	Identifier of slave device (device number, as registered at PROFIBUS Committee).	As per GSD	It is not possible to edit this field	
T_SDR	Time after which a station can answer.	11	0 .. 255	tBit
Lock/Unlock	Slave is blocked or released to other masters.	2	0(T_SDR unlock); 1(Will be unlocked); 2(Lock); 3(Unlock)	
Watchdog control	Checkbox that enables the watchdog.	Enabled	Enabled or Disabled	
Time	Define time base for Watchdog Control.	100	0 .. 65535	ms
Symbolic values	If the symbolic names are specified for parameters on device description file (GS? file), this option can be activated to show symbolic values instead of real values on the column “Value”, at the parameters table.	Enabled	Enabled or Disabled	
Length of user parameters	Total amount of configurable parameters by the user, defined in device description file.	As per GSD	It is not possible to edit this field	Byte

Parameters	Description	Default	Possibilities	Note / Unit
Defaults	Clicking in this option restores the default parameter values.			
Groups	Item for Groups definition. See Appendix C – Global Control Commands.		Enabled or Disabled	

Table 40: List of Parameters of PROFIBUS Slave

Details about parameters of PROFIBUS NX5110 head can be found in its User Manual - MU214608.

Details about the other fields can be found on standard EN50170.

10.4.3. Module Parameters

I/O modules that are present in slave NX5110 bus need to be configured according to the usage specifications. For this, it is necessary to double click on the desired module in the *Device Tree*.

Go onto *General* tab to configure the I/O points.

As can be noted in the figure below, for the module NX6100, this tab is used to configure the type of analog signal of the four output channels. For the module NX6000, this tab should be used to configure the type of analog signal of the eight input channels.

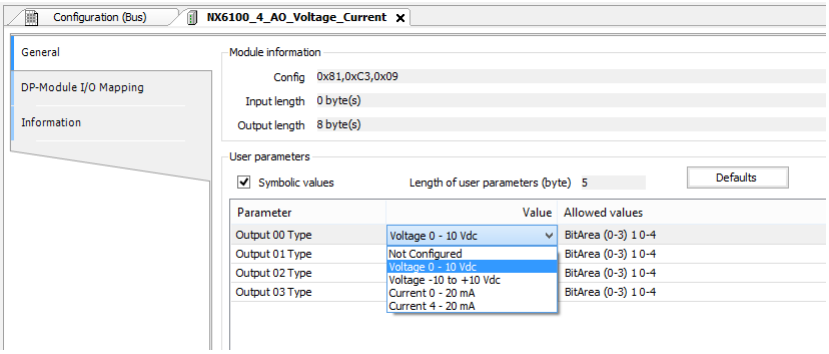

Figure 20: Configuration Tab for Module NX6100

10.5. Mapping

When adding master and slave devices to PROFIBUS in the *Device Tree* at MasterTool IEC XE programmer, memory areas and structures for data allocation regarding devices diagnostics and slave modules input and output points are defined.

The table below shows the relations allowed by MasterTool IEC XE, established between functionalities and variables.

Type of Association	Nexto CPU Variable Type
Digital Input Points	%IBn Variables
Digital Output Points	%QBn Variables
Analog Input Points	%IWn Variables
Analog Output Points	%QWn Variables
Diagnostics	%QXn Variables

Table 41: Variables Allowed in PROFIBUS Relationships

10.5.1. I/O Modules Mapping

PROFIBUS relations can be checked and/or modified within the tab *DP-Module I/O Mapping*. For example, as can be seen in the figure below, for the module NX1001, input variables %IB4 and %IB5 were allocated for this module. That is, the value in each bit represents the status of each input point of NX1001: %IX4.0 represents input 0 and %IX5.7 represents the input 17.

For module NX2001, output variables %QB6 and %QB7 were allocated. Just as for NX1001, each bit of these variables represents the state of an output point of NX2001. Each digital I/O point uses 1 bit, while one analog point uses 2 bytes, for instance: Channel 0 of NX6100 was allocated at %QW10, thus using %QB10 and %QB11.

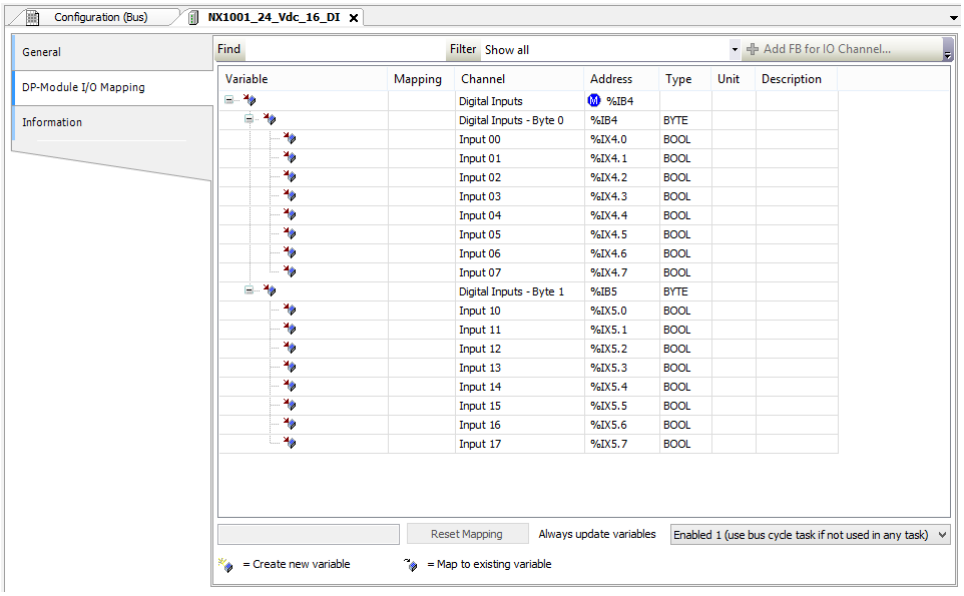

Figure 21: Tab for Configuration of Module NX1001 Mapping

ATTENTION

Select the option *Always update variables* at the right bottom to update the variables at each cycle.

10.5.2. Diagnostics Mapping

When adding a module to Nexto bus, MasterTool IEC XE programmer establishes statements for diagnostics that associates memory areas to the structures.

Regarding PROFIBUS network, as mentioned earlier in this chapter in Master Parameters, the initial memory offset area allocated for diagnostics is parameterized on the tab *Module Parameters*.

From this offset on, the MasterTool IEC XE programmer assembles the definition of associations on the declaration of global variables reserved for diagnostics. To examine them, double click the item *Module_Diagnostics*, above the *Device Tree*.

10.5.3. User Commands Mapping

This mapping is done automatically by MasterTool IEC XE programmer when adding PROFIBUS-DP Master NX5001 to the project.

Defined variables are in the tab *Bus I/O Mapping*, as indicated in the figure below and it can be changed by the user, however this is not considered a good practice.

Details for the user commands can be found in the chapter User Commands.

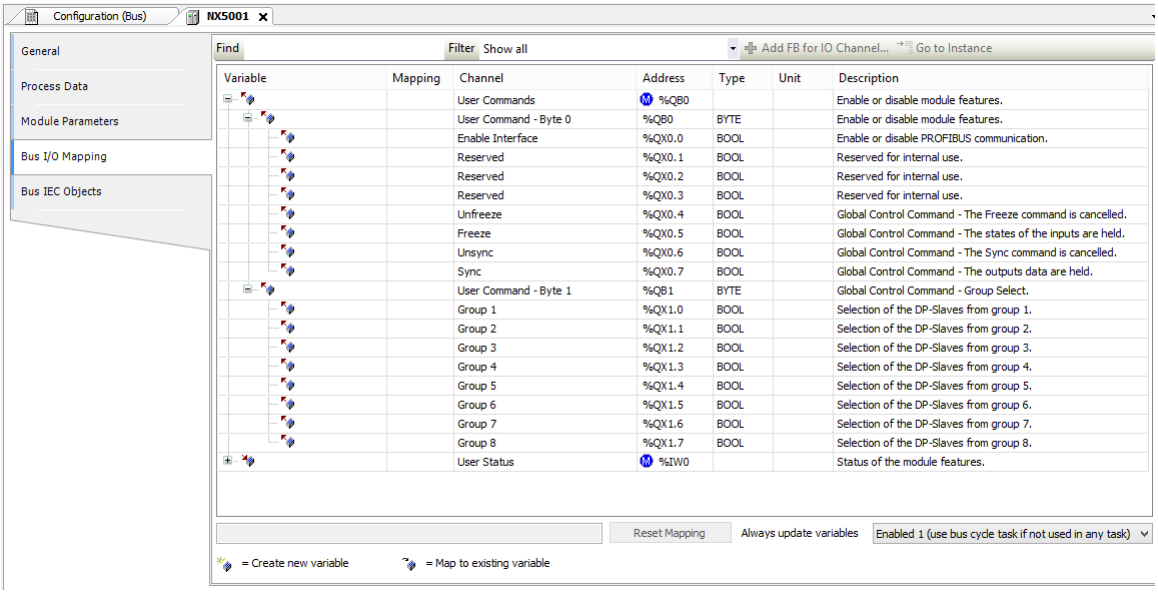

Figure 22: User Commands Mapping

10.5.3.1. Loading PROFIBUS Configuration

After finishing the configuration and parameterization process, it is necessary to send the project to CPU through the Ethernet channel, so it is necessary to change IP address on NET 1 channel in the *Device Tree* to the desired address.

Next, it's necessary to select the IP of the CPU to which the application will be sent.

Access the *Device* option in the *Device Tree*, with a double click. In the *Communication Settings* tab, select the *Gateway* and map the available devices in the network clicking on the *Scan Network* button.

In case there's no *Gateway*, or a new one is to be added, click on the *Add Gateway* button, configuring its IP in the open window.

Next select the CPU IP and click on the *Set Active Path* button, as shown on the figure below:

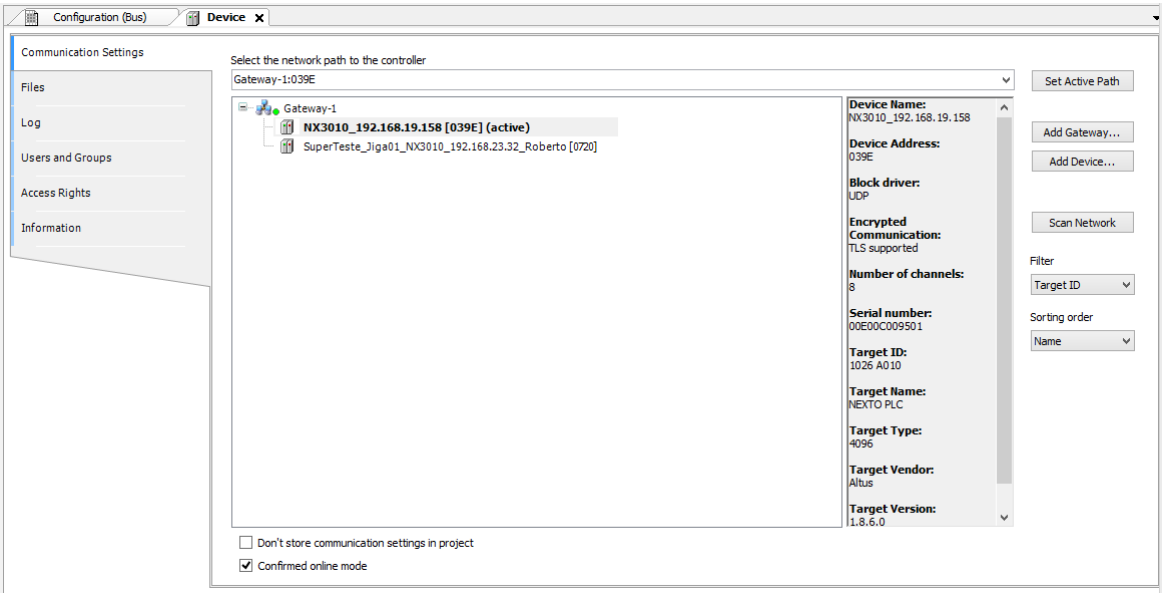

Figure 23: CPU IP Selection

In order to download the project onto the CPU, a *Login* command must be executed. Access the *Online* menu and select *Login*.

At the end of the download, the NX3010 CPU, through Nexto bus, sends the PROFIBUS network configuration to the PROFIBUS-DP NX5001 Master, which configures the PROFIBUS bus and the PROFIBUS slaves.

Next, access the *Debug* menu and click on *Start*, or simply press *F5* in your keyboard to set the application to Run mode.

11. Appendix B – GSD Installation

In case it is necessary to configure a PROFIBUS slave that is not installed in MasterTool IEC XE programmer, the following steps should be performed:

Select the *Tools* menu, then *Device Repository...*

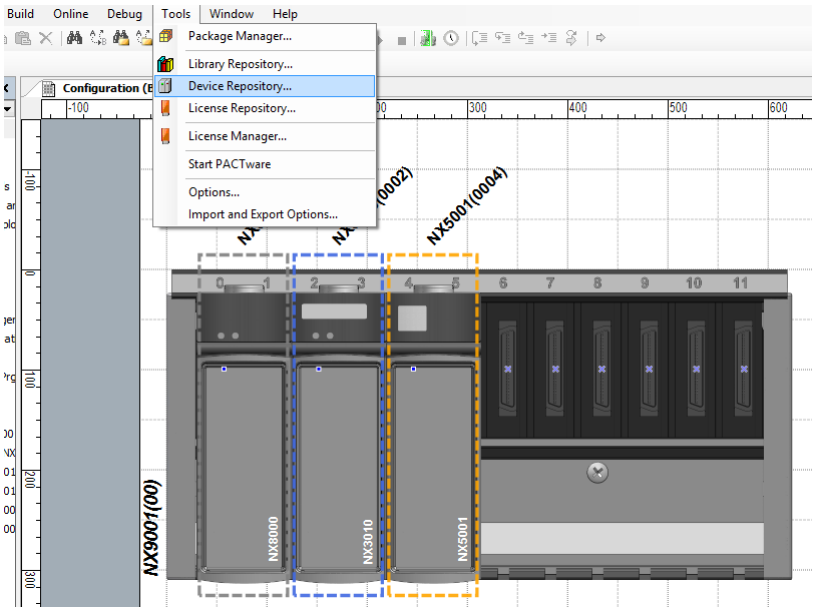

Figure 24: Device Repository Access

Locate the GSD file to be added. Remember to change the type of file to *PROFIBUS DP V5.0 configuration files(*.gs?)*. Select the file and click in *Open*:

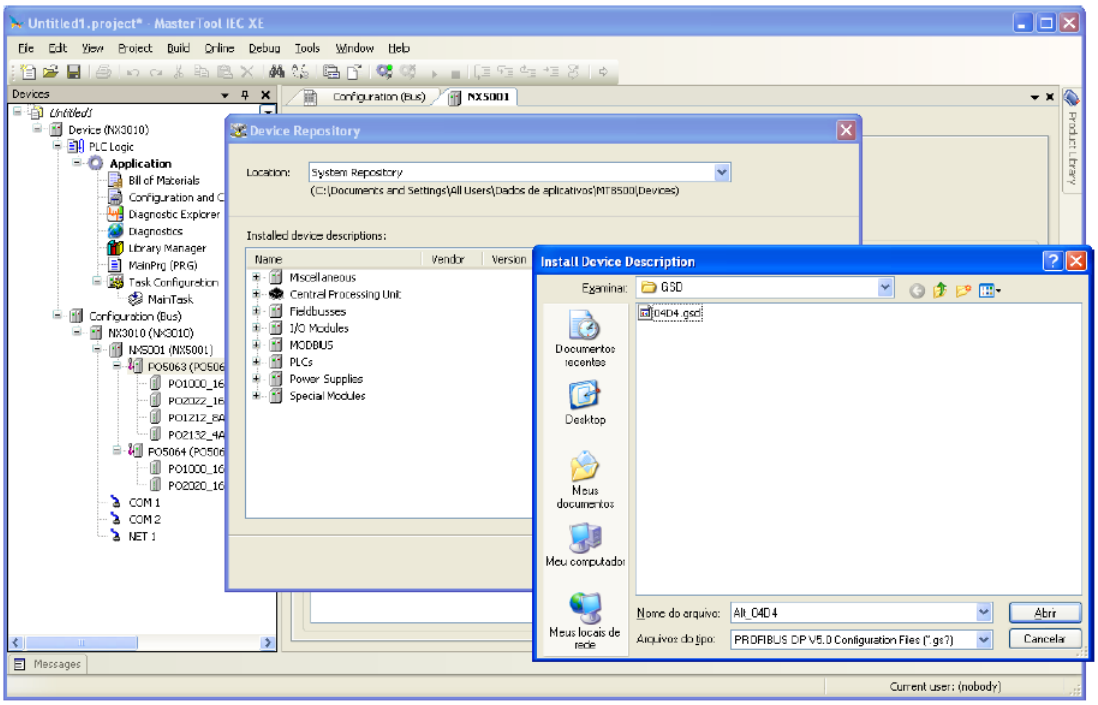

Figure 25: Installing Devices

Once added to the Device Repository, PROFIBUS DP100V slave can be added to the PROFIBUS-DP Master NX5001:

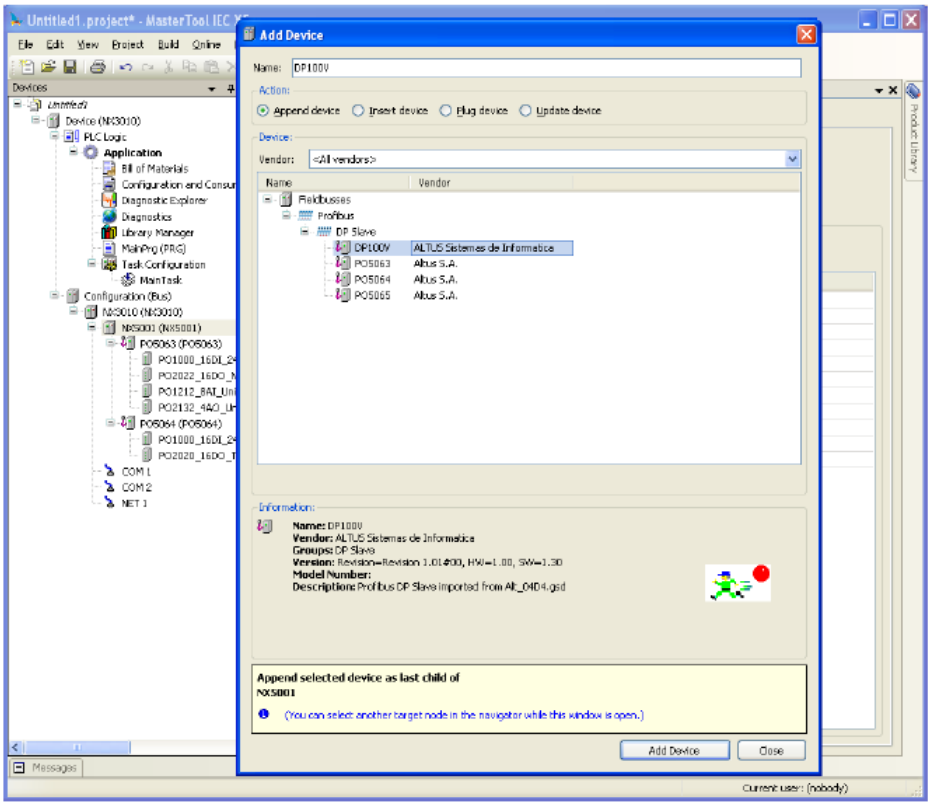

Figure 26: Adding new PROFIBUS slave to NX5001

Now you just have to send the new configuration to NX5001, as indicated in Loading PROFIBUS Configuration.

12. Appendix C – Global Control Commands

(*Global Control Commands*) are used to synchronize the I/O module's channel data of a given PROFIBUS slave group.

NX5001 PROFIBUS-DP master provides *Global_Control* function (SAP 58) commands to send the following control commands:

- **Freeze:** Freezes the slaves' input channels state
- **Unfreeze:** Cancels the Freeze command
- **Sync:** Freezes the slaves' output channels
- **Unsync:** Cancels the Sync command

ATTENTION

These commands will only work if the PROFIBUS slave supports them. Check the *Sync_Mode_supp* and *Freeze_Mode_supp* parameters in the GSD file to make sure.

ATTENTION

The commands are compatible with NX5001 Network Redundancy (Consult Table 39).

ATTENTION

To ensure the proper operation of Sync/Freeze commands in a redundant network, the user must make sure only to issue commands through Network A's NX5001 Bus Master.

The channels in the *Bus I/O Mapping* tab of NX5001 PROFIBUS-DP master are used to trigger the commands (channel *User Commands*) and for status information (channel *User Status*).

The trigger of the commands is a transition from FALSE to TRUE of the bits (%QXn.4 to %QXn.7) and status (%IB(n+1)) different than Busy (1).

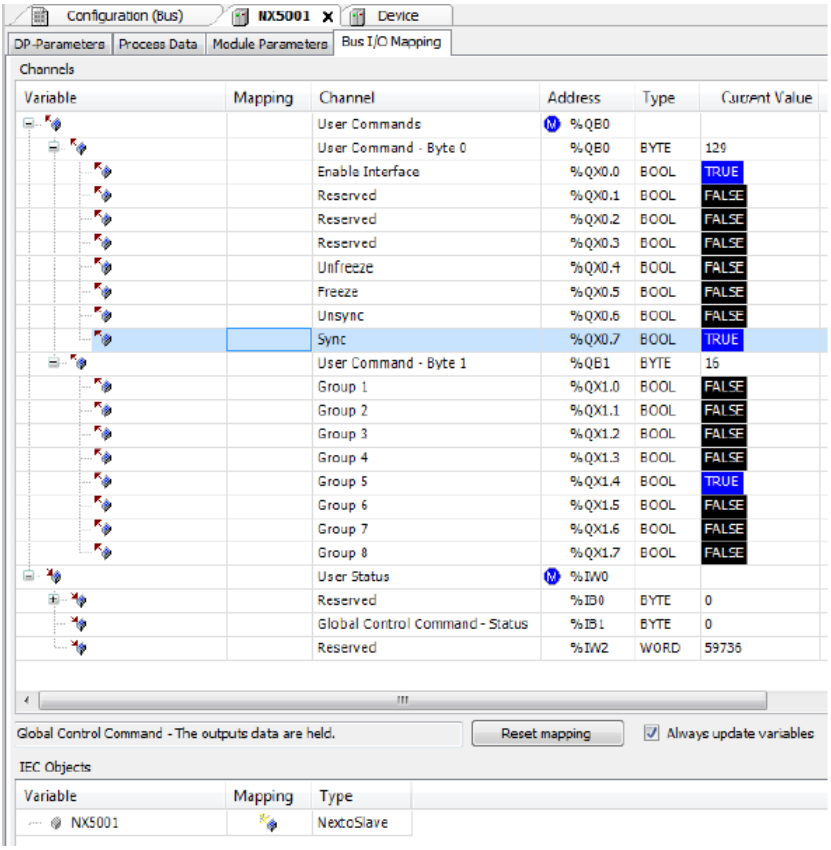

Figure 27: Global Control Command: Parameterization and Status

12.1. Parameters

The input and output parameters are available at the NX5001 PROFIBUS-DP master's *Bus I/O Mapping* tab.

12.1.1. Inputs

The input's parameters are found in User Commands area.

Direct Variable		Channel	Description
Variable	Bit		
%QB(n)	0	Enable Interface	TRUE: Enables PROFIBUS communication.
			FALSE: Disables PROFIBUS communication.
	1	Reserved	Reserved for internal use.
	2 .. 3	Reserved	Reserved.
	4	Unfreeze	Sends an Unfreeze command when a transition from FALSE to TRUE occurs. Initial Value: FALSE
	5	Freeze	Sends a Freeze command when a transition from FALSE to TRUE occurs. Initial Value: FALSE
	6	Unsync	Sends an Unsync command when a transition from FALSE to TRUE occurs. Initial Value: FALSE
	7	Sync	Sends a Sync command when a transition from FALSE to TRUE occurs. Initial Value: FALSE
%QB(n+1)	0	Group 1	TRUE: Global Control Command is sent to Slaves of Group 1.
			FALSE: Command is not sent to Group 1.
	1	Group 2	TRUE: Global Control Command is sent to Slaves of Group 2.
			FALSE: Command is not sent to Group 2.
	2	Group 3	TRUE: Global Control Command is sent to Slaves of Group 3.
			FALSE: Command is not sent to Group 3.
	3	Group 4	TRUE: Global Control Command is sent to Slaves of Group 4.
			FALSE: Command is not sent to Group 4.
	4	Group 5	TRUE: Global Control Command is sent to Slaves of Group 5.
			FALSE: Command is not sent to Group 5.
	5	Group 6	TRUE: Global Control Command is sent to Slaves of Group 6.
			FALSE: Command is not sent to Group 6.
	6	Group 7	TRUE: Global Control Command is sent to Slaves of Group 7.
			FALSE: Command is not sent to Group 7.
	7	Group 8	TRUE: Global Control Command is sent to Slaves of Group 8.
			FALSE: Command is not sent to Group 8.

Table 42: Description of User Commands

ATTENTION

The execution of user commands are active in the rising edge, therefore it is necessary that the command is initially in FALSE logical level, so that when it turns to TRUE, the rising edge is produced.

ATTENTION

If the PROFIBUS-DP network cable is removed, or the interface loses its power, the user command will be lost, since it is detected by rising edge as informed in the box above.

12.1.2. Outputs

The output parameters are found in the User Status channel.

Direct Variable	Description	Possible Values	Meaning
%IB(n)	Reserved		Reserved
%IB(n+1)	Global Control Command - Status	0	Success. Commands sent to the defined groups.
		1	Busy. Service being processed.
		2 .. 255	Error. Commands were not sent to the defined group. Check connectivity and if the error persists, contact Technical Support.
%IW(n+2)	Reserved		Reserved

Table 43: Service Status – Output Parameters

The *Global Control Command Status* indicates if the commands were successfully sent to the PROFIBUS slave Groups.

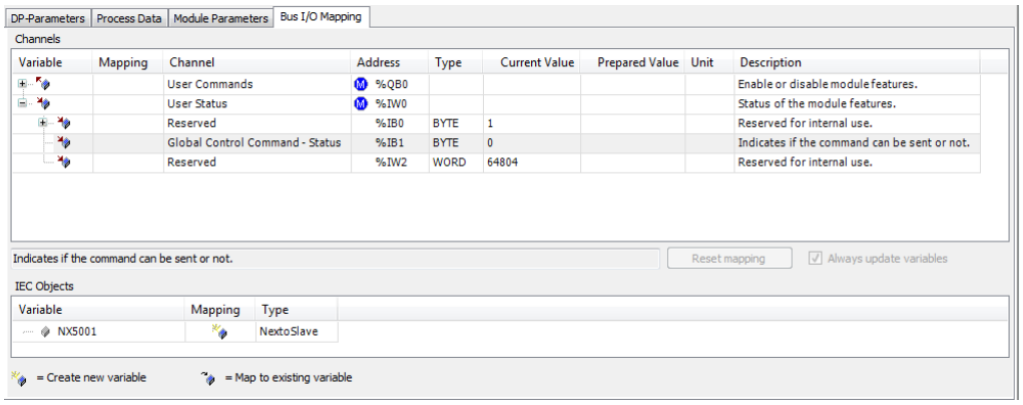

Figure 28: Global Control Command - Status

12.2. Configuration

During the configuration process of the PROFIBUS slave, the user has to define to which Group, or Groups, it should belong. The figure below shows the group configuration of a PROFIBUS slave.

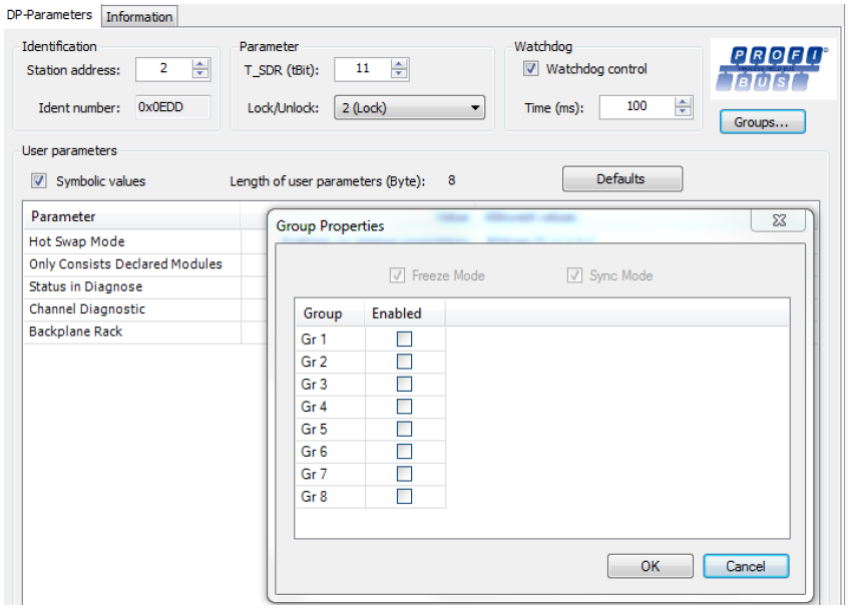

Figure 29: PROFIBUS Slave Group Configuration

Note that *Freeze Mode* and *Sync Mode* are enabled. These are merely a copy of the parameters defined in the GSD file. They indicate that the PROFIBUS slave supports both modes.

12.3. Programming

12.3.1. Groups

The first step of programming is to define which *Groups* will receive commands.

They should be selected in the *Bus I/O Mapping* tab, through the variable %QB(n+1) under User Commands.

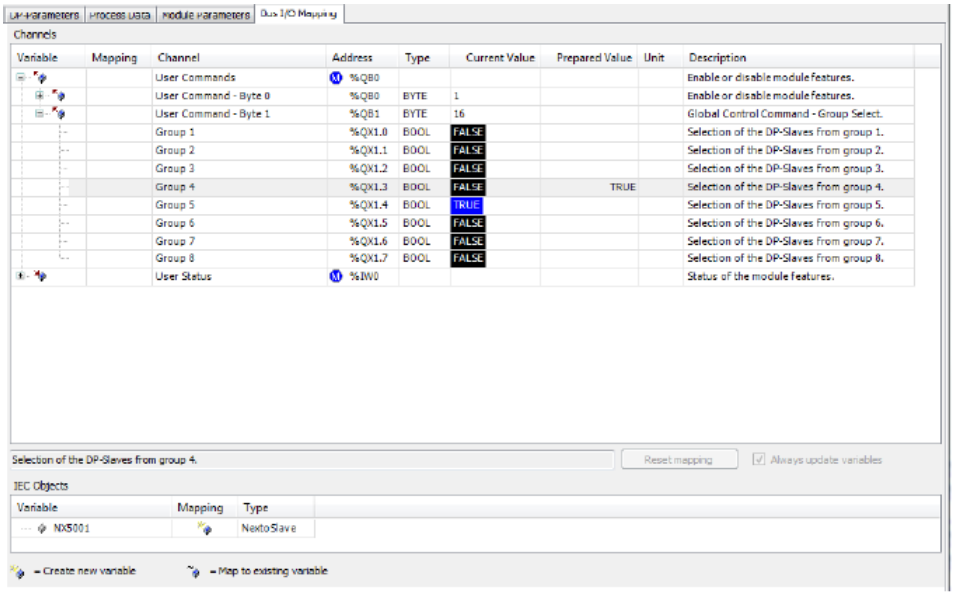

Figure 30: PROFIBUS Slave Group Selection

Details on variable %QB(n+1) are found at Table 42.

12.3.2. Commands

Afterwards, the user must define which commands should be issued by changing the respective value in the bits of variable %QB(n) of the User Commands, described at Table 42.

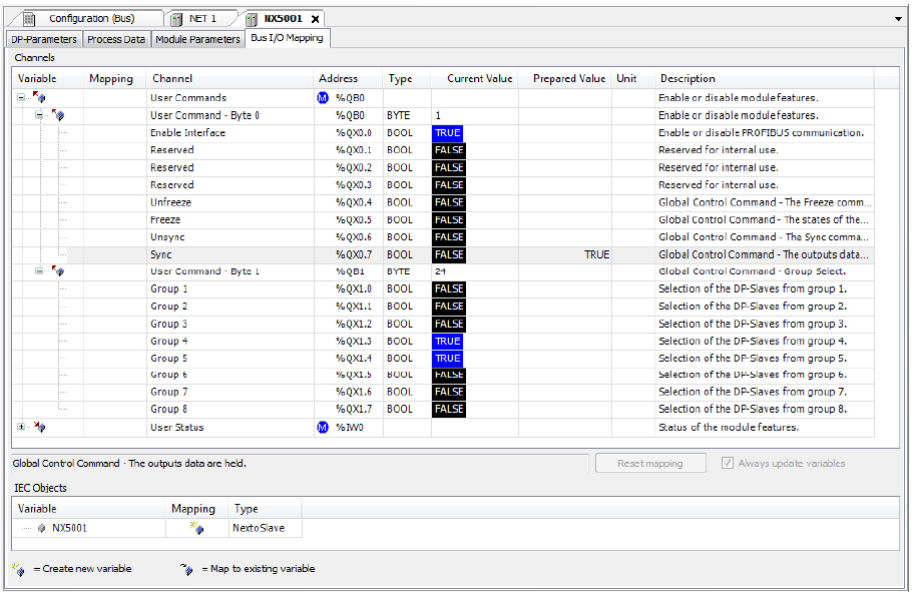

Figure 31: Command Selection

12.4. Operation

Whenever there's a transition from FALSE to TRUE of the *Command* parameters (%QXn.4 to %QXn.7), and the service's *Status* (%IB(n+1)) is different than 1 (*Busy*), NX5001 PROFIBUS-DP master will send the command(s) to the selected Group(s) (%QB(n+1)) through the PROFIBUS *Global_Control* service (SAP 58).

A new command may be issued only when the Status output parameter doesn't indicate Busy.

ATTENTION

Details on the *Global_Control* service, command operation and synchronization process can be consulted at EN50170 norm.