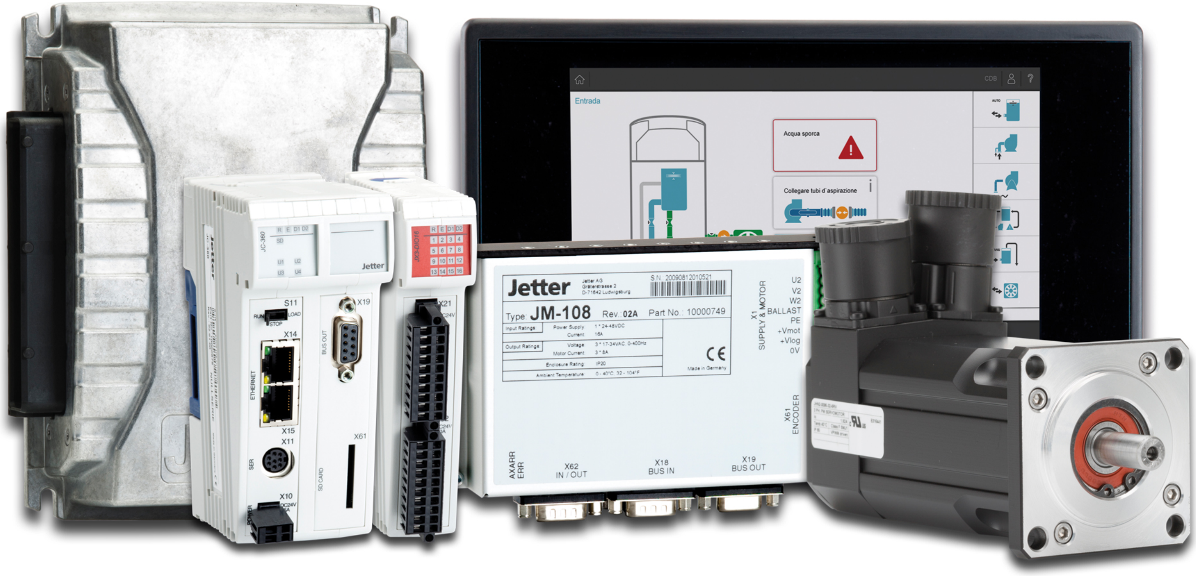


Application-Oriented Manual

Jetter Ethernet System Bus

60881085

We automate your success.

Item # 60881085
Revision 1.02
April 2017 / Printed in Germany

This document has been compiled by Jetter AG with due diligence, and based on the known state of the art.

In the case of modifications, further developments or enhancements to products shipped in the past, a revised document will be supplied only if required by law, or deemed appropriate by Jetter AG. Jetter AG shall not be liable for errors in form or content, or for missing updates, as well as for damages or disadvantages resulting from such failure.

Table of Contents

1 Jetter Ethernet system bus

Introduction

The Jetter Ethernet system bus is based on TCP, UDP/IP and can therefore be used along with other TCP, UDP/IP protocols.

It has been designed for data exchange between the following devices via standard Ethernet.

- Programming device
- Controllers
- Bus node
- Communication modules

Data interchange

The Jetter Ethernet system bus makes a difference between the cyclic and acyclic data interchange between communication participants. Both kinds of data interchange can be executed simultaneously within a network.

Data exchange	Properties
Cyclic	■ **Architecture:** Publish/subscribe ■ **Nodes:** Controllers, bus nodes and communication modules ■ **Access:** Automatically by OS ■ **Access time:** Fast, deterministic ■ **Data:** Registers, inputs/outputs ■ **Configuration:** Hardware Manager in JetSym ■ **Reach:** Subnet
Acyclic	■ **Architecture:** Client/server ■ **Client:** PC and controllers ■ **Server:** PC, controllers, bus nodes and communication modules ■ **Data:** E.g. registers, inputs/outputs, STX variables, application program ■ **Access:** PC or application program ■ **Access time:** Depending on the reaction time of the server ■ **Configuration:** Only when using network registers ■ **Reach:** International

Minimum requirements

The device is operated in a system consisting of various components by Jetter AG. In order to ensure proper interaction of these components, the operating system used and the programming tool JetSym must have at least the release numbers listed below.

Component	As of version
JC-310-JM	V. 1.22.0.00
JC-340	V. 1.22.0.00
JC-350	V. 1.22.0.00
JC-360	V. 1.22.0.00
JC-360MC	V. 1.22.0.00
JC-365	V. 1.26.0.00
JC-365MC	V. 1.26.0.00
JC-440(MC)	V 1.02.0.00
JC-940MC	V. 1.06.0.20
JC-945MC	V. 1.01.0.00
JX3-BN-ETH	V. 1.18.0.02
JX3-COM-EIPA	V. 1.01.0.00
JX3-COM-PND	V. 1.03.0.06
JM-200-ETH	V. 1.22.0.00
JetSym	V. 5.1.2

Contents

The Global Node Number

Definition - Global Node Number

The Global Node Number (GNN) is an ID number to identify Jetter controllers (e.g. controllers, bus nodes) within an Ethernet network.

- The GNN within a network has to be unambiguous for each Jetter device.
- The JetSym Hardware Manager automatically assigns the GNN during configuration.
- The value range of the GNN within a project is 000 ... 199.
- The controller has always got GNN 000.

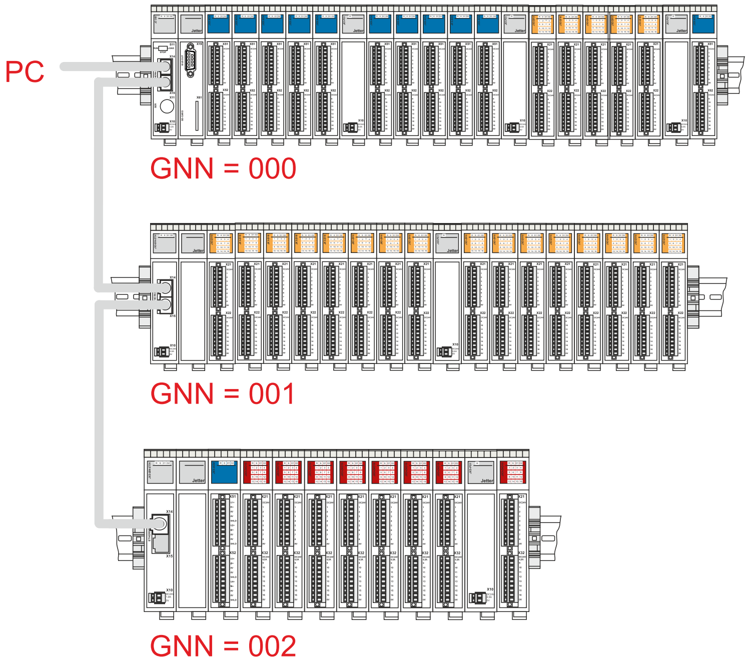
Using the GNN

The Global Node Number is used in the following applications:

- Register number for network registers
- Identification of publications and subscriptions at cyclic data interchange
- Identification of nodes at automatic network configuration (NetConsistency)

Networking example

The following illustration shows networking of a possible JX3 system with a JC-3xx and two JX3-BN-ETH.



1.1 Acyclic data interchange

Introduction

This chapter covers acyclic data interchange on the Jetter Ethernet system bus.

Properties

Acyclic data interchange on the Jetter Ethernet system bus can be characterized as follows:

Property	Description
Architecture	Client/server - Data interchange is initiated by the client. - The server gives a response to the request made by the client. - Usage of unicast frames - Network access is made once.
Client	- JetSym: Programming and debugging of application programs - JetViewSoft: Setting up a visualization application - Controllers: Data interchange out of the application program (NetCopy..., NetBit..., network register)
Server	- PC: E.g. for database applications - Controllers, bus nodes and communication modules: E.g. variable or debugging server
Data	- PC: Registers, inputs/outputs, STX variables, application program - Controllers, bus nodes and communication modules: Registers, STX variables
Access time	- It depends on the data transfer time and on the server's processing time
Configuration	- Network registers: Easy configuration in the application program - Else, both client and server are completely configured by the operating system.
Reach	- Using TCP/IP and UDP/IP frames allow for data interchange exceeding the limits of one's own subnet.

Client

Below, programming the client in the controllers is described. In doing so, the following topics are dealt with:

- Transferring variable/register sets (command group NetCopy())
- Setting and clearing register bits (command group NetBit())
- Transmitting individual register values (network registers)

Examples of the application

- Event-triggered data interchange
- Parameterization
- Configuration

Used protocol

The client of the controller uses the JetIP protocol based on UDP/IP for data transfer.

Server

The server functions do not require any programming or configuration by the user.

Protocols

Acyclic data interchange on the Jetter system bus can be established by the following protocols:

- XCOM protocol by Jetter AG
- JetIP protocol by Jetter AG
- UDP/IP
- TCP/IP
- IPv4

Contents

Command group NetCopy()

Introduction

The NetCopy command is a versatile tool for data interchange between Jetter products via Ethernet.

The NetCopy command lets you copy the following data:

- Register values
- Values of register blocks
- Variable values
- Values of variable blocks

Advantages of NetCopy

Advantages of NetCopy commands as compared with the use of network registers:

- Within the command, you can directly specify any valid IP address.
- Within the command, you can directly specify any valid IP port.
- The entire register address range of a remote node can be directly addressed.
- By means of one command, a large register set or, in case NetCopyList is applied, a large number of registers can be copied.
- The result of the copying process can be evaluated directly.

Access via NetCopy

NetCopy functions with the following nodes:

- Controllers
- Bus node
- Communication modules
- PC

To access other nodes, use the NetCopy command as follows:

If ...	... then ...
... you wish to copy data from the controller to another node,	... use the following commands: ■ NetCopyRegToReg ■ NetCopyVarToReg ■ NetCopyList
... you wish to copy data from another node to the controller,	... use the following commands: ■ NetCopyRegFromReg ■ NetCopyVarFromReg ■ NetCopyList

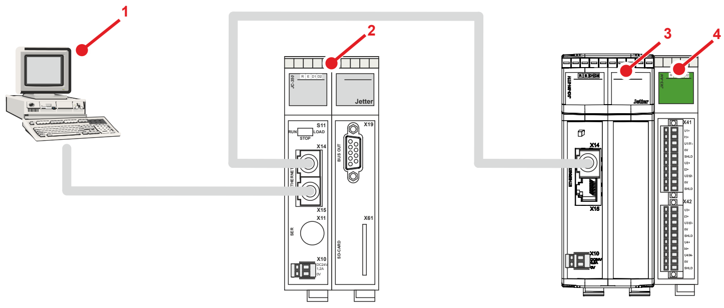
Parameters of the NetCopy commands

For detailed information on the parameters, refer to the JetSym help.

NetCopy - Example featuring a bus node

As you can see in the following illustration, a controller JC-3xx is connected to a PC. The bus node JX3-BN-ETH of IP address 192.168.10.2 is connected to a peripheral module JX3-AI4.

This example describes how to access the module registers of the peripheral module JX3-AI4 in acyclic mode.



Number	Part	Description
1	PC	PC with JetSym
2	JC-3xx	Controller
3	JX3-BN-ETH	Bus node
4	JX3-AI4	Peripheral module with analog inputs

Task

When an event occurs, user scaling of analog input 1 is to be changed.

Solution

The NetCopy command causes values from application program variables to which the user scaling parameters have been stored to be copied to the corresponding registers of the JX3-AI4.

The register number of the peripheral module is seen from the perspective of the JX3-BN-ETH:

1	0	0	x	x	z	z	z	z

with

- xx = 02: First module on the JX3-BN-ETH
- zzzz = 1124 through 1127: Parameter registers of the JX3-AI4 user scaling

```
// Copy values from the local array to the JX3-AI4
nResult := NetCopyVarToReg(IP#192.168.10.2, anParam,
                           100021124, 16, 3, 1);
```

Command group NetBit()

Introduction

The NetBit command is an all-purpose tool to set or clear register bits of Jetter products. The Jetter products are interconnected via an Ethernet network.

Advantages of NetBit

NetBit commands let you both set and clear bits in one go.

Simulating NetBit commands by means of NetCopy commands:

- A NetCopy command lets you copy the register value from the remote node to the local controller.
- A NetCopy command lets you change the state of the bits on the local controller as desired.
- Another NetCopy command lets you copy the register value to the remote node again.

For this, several commands are required. Thus, a register value may be changed during this action by an application program running on the remote controller. The second NetCopy command will then overwrite this value again. There is an undefined data condition, which is prevented by the NetBit functions.

Further advantages of NetBit commands as compared with the use of network registers:

- Within the command, you can directly specify any valid IP address.
- Within the command, you can directly specify any valid IP port.
- The entire register address range of a remote node can be directly addressed.
- The result of executing this command can be evaluated directly.

Access via NetBit

NetBit functions with the following nodes:

- Controllers
- Bus node
- Communication modules

To access other nodes, use the command NetBit as follows:

If ...	... then ...
you wish to set register bits for another node,	use the command ▪ NetBitSetReg
you wish to clear register bits of another node,	use the command ▪ NetBitClearReg

Parameters of the NetBit commands

For detailed information on the parameters, refer to the JetSym help.

Network registers

Introduction

The network registers let you access in transparent mode registers of remote nodes.

Advantages

Advantages of network registers as compared with NetCopy commands:

- Network registers are used just like any other registers in the application program.
- If programs or parts of programs are used for local and distributed applications, a program description is not needed.

Restrictions

The following restrictions apply to network registers as compared with NetCopy commands:

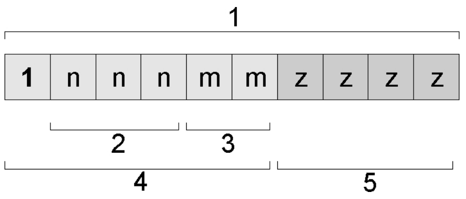
- IP address and IP port of the remote node must be set separately.
- Only part of the register address range of the remote nodes can be accessed directly.
- The outcome of the network access (diagnostics) cannot be logged directly.

Properties

If you access network registers of cyclic data interchange, the controller does not carry out acyclic network register access. The controller accesses the locally stored cyclic data.

Addressing scheme

The addressing scheme for network registers is as follows:



No.	Element	Description
1	Register number	Supports direct access
2	First part of register prefix: Bus node ID, GNN	nnn = 001 ... 199: ID of the network node, referred to as Global Node Number
3	Second part of register prefix: Number of the function module	mm = 02 ... 17: Number of the JX3 module of a remote node mm = 98: Indirect addressing of the register of a remote node mm = 99: Addressing the variable destination window of a remote node
4	Part 1 + 2: Register prefix	1nnnmm: The prefix is preceded by a leading ONE.
5	Module register number	zzzz = 0000 ... 9999

IP address and IP port

Before using a network register, the IP addresses and IP ports of the remote network nodes must be written to two tables in the local register array.

If ...	... then ...
... you carry out network configuration in the JetSym Hardware Manager,	... these tables are generated automatically, see file **ModConfig.da** below.
... you do not carry out network configuration in the Hardware Manager,	... you must generate the tables in your application program.

Content indexing of the tables is done via GNN of the node in the first part of the register prefix (2).

Register	Value range	Properties
235000 + GNN	235000 ... 235199	Register table for **IP addresses**
235400 + GNN	235400 ... 235599	Register table for **IP ports**

Note on the contents of the table:

- GNN = Global Node Number in the range 000 ... 199

File *ModConfig.da*

When you download the configuration files, the Hardware Manager transfers the file **ModConfig.da** to the controller.

The OS of the controller loads this file when the controller is energized or when the corresponding command is automatically issued by the Hardware Manager after download.

The file **ModConfig.da** lists registers with their corresponding values. The OS enters the corresponding values into these registers.

This file also holds the IP addresses (register 235000 + GNN) and port numbers (register 235400 + GNN) of the nodes on the network.

It is no longer required to enter values into registers via application program.

Registers located on JX3 modules

Introduction

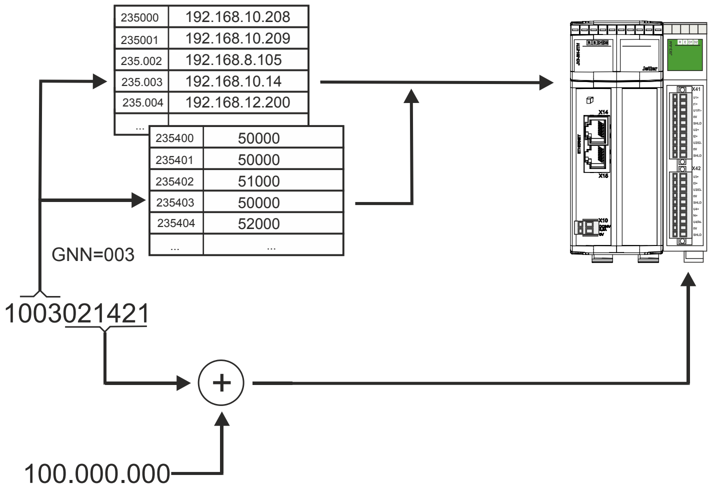
The controller handles access via network registers to module registers of JX3 modules of a remote node (second part of the register prefix mm = 02 ... 17) in a specific way:

If ...	... then ...
... the network register has been configured for cyclic data interchange,	... the controller accesses the locally stored register value.
... the network register has not been configured for cyclic data interchange,	... the controller executes acyclic network access.

Acyclic network access

For acyclic register access to a remote JX3 module, the controller uses the first part of the register prefix as an index to the tables containing the IP addresses and IP ports. The values read out of these tables are used by the controller to address the bus node in the network. For further information on registers 235000 ff, please refer to **Network registers - IP address and IP port** (see page 13).

The controller adds the register offset for the JX3 system bus of a remote node (100,000,000) to the second part of the register prefix and the module register number (no. 3 and 5 in the **addressing scheme** (see page 13)). The controller uses the resulting number to address the register.



235000	192.168.10.208
235001	192.168.10.209
235.002	192.168.8.105
235.003	192.168.10.14
235.004	192.168.12.200
...	

235400	50000
235401	50000
235402	51000
235403	50000
235404	52000
...	...

Action

If you want to access the JX3 module register of a remote network node using register addresses as of 1 billion, proceed as follows:

Step	Action
1	Enter the **IP address** of the remote network node into register **235.000 + GNN**. Value range of the GNN: 1 ... 199
2	Enter the **port number** into register **235400 + GNN**. Value range of GNN: 1 ... 199
⇨	Now you can access the value via register **1nnnmmzzzz.** Value range of GNN = nnn: 001 ... 199 Value range mm: 02 ... 17 Value range zzzz: 0000 ... 9999

This lets you directly access all JX3 module registers of the remote network node.

Example

Via network, a controller and a bus node JX3-BN-ETH are connected.
A JX3-AI4 module is connected to the bus node.

Configuration of the bus node	Value
GNN	3
IP address	192.168.10.14
IP port	50000

Task:

The trailing indicator of the analog channel 4 peak value is to be read.

Solution:

You create a JetSym STX program by taking the following steps:

- Register 235003 is loaded with the IP address of the bus node.
- Register 235403 is loaded with the IP port of the bus node.
- The value of network register 1003021421 is assigned to a local variable.

Indirect addressing of remote modules

Introduction

Indirect addressing of network registers lets you access registers of a remote network node. First enter the number of the remote node register into a table of register numbers in the local controller. Content indexing of this table is carried out via the three low-order figures of the network register number.

Registers - Overview

Overview of the registers allowing indirect addressing of remote nodes:

Register	Value range	Properties
236000 + zzz	236000 ... 236199	Register table for the **register numbers**
1nnn980zzz	1nnn980000 ... 1nnn980199	Register array for the **Content**

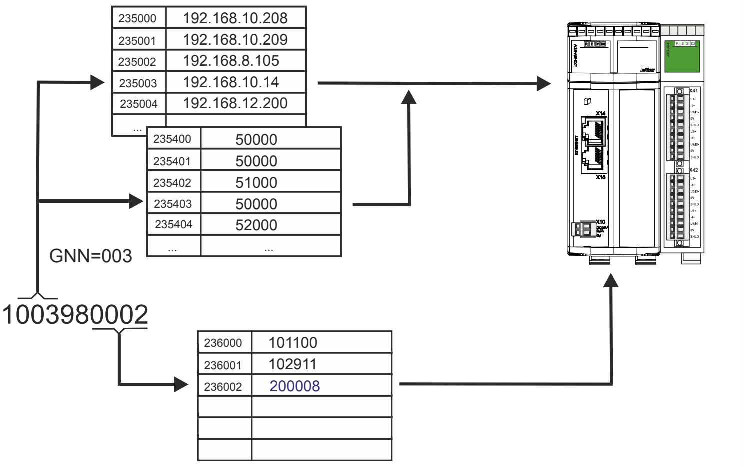
Note on the contents of the table:

- nnn = GNN in the range 000 ... 199
- zzz in the range 000 ... 199

Indirect network register access

For indirect access to a remote node via network register, the controller uses the first part of the register prefix as an index to the tables containing the IP addresses and IP ports. The values read out of these tables are used by the controller to address the bus node in the network. For further information on registers 235000 ff, please refer to **Network registers - IP address and IP port** (see page 13).

The module register number (no. 5 in the **addressing scheme** (see page 13)) is used by the controller as an index to a table of register numbers. The register number read out of this table is used by the controller to address the register in the bus node.



235000	192.168.10.208
235001	192.168.10.209
235002	192.168.8.105
235003	192.168.10.14
235004	192.168.12.200
...	

235400	50000
235401	50000
235402	51000
235403	50000
235404	52000
...	...

236000	101100
236001	102911
236002	200008

Action

If you want to access the register of a remote network node using register addresses as of 1 billion, proceed as follows:

Step	Action
1	Enter the **IP address** of the remote network node into register **235000 + GNN**. Value range of the GNN: 0 ... 199
2	Enter the **port number** into register **235400 + GNN**. Value range of the GNN: 0 ... 199
3	Enter the required **register number** of the remote network node into register **236000 + zzz**.
⇨	Now you can access the value via register **1nnn980zzz.** Value range of the GNN: nnn = 000 ... 199 Value range zzz: 000 ... 199

This configuration lets you indirectly access - via 200 controller registers - all module registers of the remote network node.

Example

Prerequisite:

Via network, a controller and a bus node JX3-BN-ETH are connected.

Configuration of the bus node	Value
GNN	3
IP address	192.168.10.14
IP port	50000

Task:

The global error register of the JX3-BN-ETH is to be read every second.

Solution:

- Register 235003 is loaded with the IP address of the bus node.
- Register 235403 is loaded with the IP port of the bus node.
- Register 236028 is loaded with the error register number 200008.

Addressing with variable destination window

Introduction

Indirect addressing also allows for a variable destination window. You shift the register array of 10,000 registers of the remote network nodes by an offset by entering a value into R 272702 of the remote network nodes.

Registers - Overview

Overview of the registers allowing indirect addressing with variable destination window:

Register	Value range	Properties
1nnn99zzzz	1nnn990000 ... 1nnn999999	**Register content** of a remote network node; The register is in the variable destination window which consists of 10,000 registers.
272702 (of the remote node)	0 ... 2,147,483,647	Variable destination window: The destination window is a register array of a remote network node. This destination window is shifted by this **offset**.

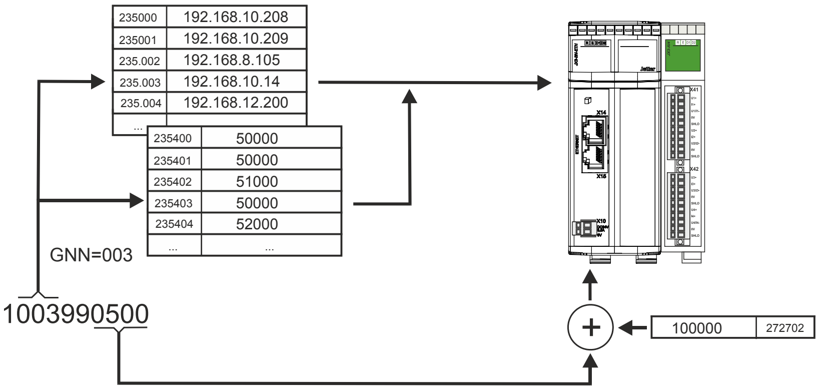
Note on the contents of the table:

- nnn = GNN in the range 000 ... 199
- zzzz in the range 0 ... 9,999

Network register access with variable destination window

For access via network register with variable destination window to a remote node, the controller uses the first part of the register prefix as an index to the tables containing the IP addresses and IP ports. The values read out of these tables are used by the controller to address the bus node in the network. For further information on registers 235000 ff, please refer to **Network registers - IP address and IP port** (see page 13).

The module register number (no. 5 in the **addressing scheme** (see page 13)) is used by the controller to address the register in the bus node. A register number is transmitted to the remote network node by the controller. The remote network node adds the content of register 272702 to this register number and uses the result as register number.



235000	192.168.10.208
235001	192.168.10.209
235.002	192.168.8.105
235.003	192.168.10.14
235.004	192.168.12.200
...	

235400	50000
235401	50000
235402	51000
235403	50000
235404	52000
...	...

Steps to take for addressing with destination window

To use register addresses starting from 1 billion with variable destination window (offset), proceed as follows:

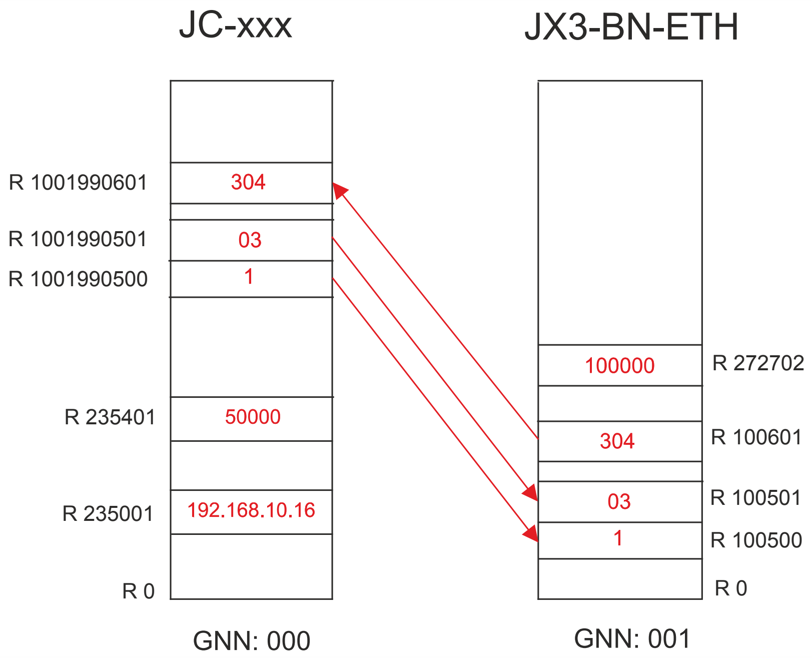
Step	Action
1	Enter the **IP address** of the remote network node into register **235000 + GNN**. Value range of the GNN: 0 ... 199
2	Enter the **port number** into register **235400 + GNN**. Value range of the GNN: 0 ... 199
3	Set the base address of the **destination window:** Enter a value into R 272702 of the remote network node.
⇨	Now, registers **1nnn990000 ... 1nnn999999** let you access the value.

Example

A JetControl is to read a register value from a JX3-BN-ETH. Control system and bus node are interconnected via the Jetter Ethernet system bus.

There are JX3 modules connected to the JX3-BN-ETH, such as a JX3-AO4 of module number 03.

By entering value 100000 into R 272702 of the JX3-BN-ETH, you get read access to the EDS of the connected JX3 modules. In this example, the module code of the JX3-AO4 is to be read. For further information on how to read an EDS, please refer to EDS registers.



Reading is carried out in three steps:

Step	Action
1	Enter value 1 for a JX3 module into R 1001990500.
2	Enter module number 03 into R 1001990501.
3	Read module code 304 for JX3-AO4 from R 1001990601.

Registers for acyclic data interchange

Introduction

In acyclic data interchange, data transmission from a controller to remote network nodes is carried out via JetIP protocol. The client in the controller is supplied with registers for configuration and error diagnostics.

Registers/flags - Overview

Register	Description
232708	Timeout in milliseconds
232709	Response time in milliseconds
232710	Amount of network errors
232711	Error code of last access
232717	Maximum number of retries
232718	Present number of retries

Flags	Description
2075	Network error

R 232708

Timeout

To R 232708, write the timeout (in milliseconds) for acyclic access via network.

Module register properties	
Values	1 ... 65,535 [ms]
Value after reset	250 [ms]

R 232709

Response time

R 232709 displays the total response time of latest acyclic access via network in milliseconds. The total response time includes the time for data transmission and the processing times in the controller and in the remote network node.

Module register properties	
Values	0 ... 65,535 [ms]
Type of access	Read

R 232710

Amount of network errors

R 232710 shows the total number of network errors.

Module register properties	
Values	-2,147,483,648 ... 2,147,483,647 (overflowing)

R 232711

Error code

R 232711 shows the error code of the latest network access.

Module register properties		
Values	0	No errors
	1	Timeout
	3	Error message from remote node
	5	Invalid network address
	6	Invalid amount of registers
	7	Invalid interface number

R 232717

Maximum number of retries

R 232717 lets you set the maximum possible number of network access retries. If a network access could not be made without errors, the controller will repeat the access at the most as often as it has been set in this register. If the network access could still not be made without errors, the controller will cancel the access and create an error message.

Module register properties	
Values	0 ... 255

R 232718

Present number of retries

R 232710 shows the total number of network access retries.

Module register properties	
Values	-2,147,483,648 ... 2,147,483,647 (overflowing)

M 2075

Network error

If a network error occurs, the operating system sets flag 2075. In order to detect further errors this way, you must manually reset the flag.

Flag properties		
Values	0	No network errors since last reset
	1	A network error has occurred

1.2 Cyclic data interchange

Introduction

This chapter covers cyclic data interchange via Jetter Ethernet system bus.

Properties

Properties of cyclic data interchange via Jetter Ethernet system bus:

Property	Description
Architecture	Publish/subscribe ▪ The publishers send the data. ▪ The subscribers receive the data. ▪ Usage of multicast frames
Publisher	▪ Each publisher sends one or several publications. ▪ Data of a publication are consistently transferred in a frame. ▪ The cycle time can be set for each publication.
Subscriber	▪ The subscriber receives one or several publications and assigns them to the corresponding subscriptions. ▪ The subscriber validates the received data.
Data	▪ Registers ▪ Inputs ▪ Outputs
Access time	▪ Very short, as the network nodes access the locally stored interchanged data.
Configuration	▪ In the JetSym Hardware Manager
Reach	▪ Restricted to own subnet

Examples of the application

- Cyclic, deterministic interchange of process data
- Cyclic, deterministic interchange of status information details

The JetSym Hardware Manager generates the configurations for cyclic data interchange using the status information details and the process data of the connected peripheral modules.

Restrictions

For cyclic data interchange, do not use any configuration registers or special registers. Access to these registers can take longer or trigger further action, which may lead to unwanted results.

Multicast in other networks

Please note that the Jetter Ethernet system bus operates with multicasts (multipoint connections). If you couple the Jetter Ethernet system bus with your local network, you have to filter out unwanted multicasts by a router.

As an alternative, the function **JetSync blockage** (see page 86) can be used, too.

Technical specifications

Technical specifications of cyclic data interchange via Jetter Ethernet system bus:

- Usage of multicast frames
- Reserved multicast groups: 255
- Multicast groups available to the user: 0 ... 254
- IP addresses for multicasts: 239.192.0.0 + multicast group
- MAC address for multicasts: 01:00:5E:40:00:00 + multicast group
- Maximum size of user data in a publication/subscription: 256 byte

Contents

Publish/subscribe

Introduction

Publish/subscribe is used as communication architecture for cyclic data interchange in the Jetter Ethernet system bus. The JetSym Hardware Manager generates the configurations for cyclic data interchange and transfers them to the controller. Based on this configuration, the configuration automatically carries out cyclic data interchange.

Basic data interchange

Basic data interchange via publish/subscribe is executed by the publishers and subscribers in the operating system of the Jetter devices at the Jetter Ethernet system bus.

Publisher

- The publishers publish data of the network node, on which they are being processed.
- A data record is published by the publisher. Therefore it is called **publication**.
- A publisher can manage several publications.

Subscriber

- The subscribers which are interested in these data receive the publications and transfer the contents to the data of the network node on which they are processed.
- A data record is received by the subscriber. Therefore it is called **subscription**.
- A subscriber can manage several subscriptions.
- To receive a publication, there must exist a corresponding subscription.
- One publication can be received by subscriptions on various network nodes simultaneously, as the publications are published via multicast frames.

JetSym

When a combination of a controller and one or several network nodes is configured in JetSym, the Hardware Manager generates the configuration files for the publishers and for the subscribers. The Hardware Manager generates one-to-one relationships between the publications and the subscriptions.

Features of publish/subscribe

If, in Hardware Manager, you add network nodes with the modules connected to them, Hardware Manager will automatically generate the module status and the process data belonging to these modules as publish/subscribe variables. For further information on the process data, please turn to the user manual of the respective JX3 module.

Features of publish/subscribe

Parameter	Value	Description
Number of network nodes	000 ... 199	200 network nodes max.: They are entered into the Hardware Manager by their name and as GNN
Maximum amount of process variables per publication/subscription	64	64 process variables max: This corresponds to 256 bytes of process data
Cycle time	1 ... 2,147,483,647 ms	Default: 2 ms

Network nodes are the controller, the communication modules and the bus nodes.

For details on characteristic features of publish/subscribe, please turn to chapter **Hardware Manager** (see page 37).

Configuring and executing publish/subscribe

Publish/subscribe is configured in the JetSym Hardware Manager. Publish/subscribe is executed by the operating system of the respective network node:

- Publishers and subscribers are configured by means of configuration files in the file system of the network nodes.
- The configuration file for the publisher is **/SysConfig/JetSync/Publisher.pub**.
- The configuration file for the subscriber is **/SysConfig/JetSync/Subscriber.sub**.
- Automatic restart of the publishers and subscribers takes place in a controller at each restart of the application program.
- In the other network nodes, automatic restart of the publishers and subscribers takes place during the booting phase.
- For applying publishers and subscribers in a controller, an application program must be executed with at least one task running.

For transferring the configuration, the Hardware-Manager takes the following steps:

Step	Action
1	Stop all publishers and subscribers.
2	Transfer the configuration files to all network nodes.
3	Restart all publishers and subscribers.

Related topics

- **Hardware Manager** (see page 37)

Publish/subscribe - Registers

Introduction

If you transmit cyclic data by publish/subscribe, there are several module registers available for administration, configuration and error detection. You have got read and partial write access to these module registers.

Register overview

Module registers	Description
210004, 200008, 200009	General error registers
250000 ... 250004	Registers for administration of all subscriptions
250x10 ... 250x11	Registers for administration of one subscription
250x20 ... 250x30	Registers for configuring one subscription
254001 ... 254003	Registers for error detection
255000 ... 255004	Registers for administration of all publications
255x10 ... 255x11	Registers for administration of one publication
255x20 ... 255x30	Registers for configuring one publication
Flag 2080	Enable for publishing an error
Flag 2081	Error collection of the subscriber

x = 0 ... 9

Availability

Administration and configuration registers are available as follows:

- For subscriptions and publications, 10 arrays for administration and configuration registers are available.
- The register arrays differ by the hundred's place of the respective register number.
- The placeholder x indicates the number of the register array. Value range of x: 0 ... 9
- External clients use register array x = 1, such as JetSym with visualization application and PCOMX protocol.
- STX functions use register array x = 0.
- In order to gain faster access to individual publish/subscribe administration registers, several register arrays are at your disposal: There are individual publish/subscribe IDs to be called in each register array.

Registers for administration of all subscriptions

There are several registers available which go with all subscriptions.

Register	Name	Description
250000	Status	Status register
250001	Command	Command register
250002	ID in case of error	Displays the ID of the subscription, in which an error has occurred.
250003	Amount	Total amount of subscriptions
250004	CRC	16-bit CRC (**C**yclic **R**edundancy **C**ode) of the subscriber configuration file

Subscriber status

Status registers of all subscriptions

From MR 250000, you can read the collective status of all subscriptions. In case of an error, you first read out the ID of the subscription, in which an error has occurred.

Meaning of the individual bits

Bit 0 **Error in CRC computing of the configuration file**

0 = No error has occurred.

1 = For CRC computing, the configuration file does not exist. For this reason, CRC computing has not taken place.

Bit 1 **Error in connection with a subscription**

1 = An error has occurred in a subscription.
At the moment, this is only a timeout error.

Bit 7 **Subscription is functioning.**

0 = If a subscription fails, bit 7 is reset.

1 = The subscriptions are functioning.

Module register properties	
Type of access	Read

Subscriber command

Command registers of all subscriptions

Via MR 250001, you transmit commands to all subscriptions.

Commands	
102	**Reboot all subscribers**
105	**Stop all subscribers**
110	**Acknowledge error**

Selecting a subscription

The following registers let you select a subscription as follows:

- The index is for selecting subscriptions.
  - If the subscription exists, R 250x11 shows its ID.
  - If the subscription does not exist, R 250x11 shows value **-1**.
- In this case, enter the ID of the subscription into R 250x11.
  - If the subscription exists, the content of R 250x11 is kept.
  - If the subscription does not exist, R 250x11 shows value **-1**.

Register	Name	Description
250x10	Index	Index of the subscriptions: 0: Selects the first subscription 1: Selects the next subscription 2: etc.
250x11	ID	The ID of the subscription is entered

Configuring a subscription

The following registers show the configuration of a subscription, which you have selected via R 250x10 and R 250x11.

Register	Name	Description
250x20	Status	Bit 0: Publication received Bit 1: Timeout
250x21	Mode	0: Cyclic 1: Upon request
250x22	Number of variables	As configured
250x23	Group address	As configured
250x24	Hash	Internal usage
250x25	Sequence number	Internal usage
250x26	Data size	Internal usage
250x27	Timeout in ms	Bus cycle * 3

Register	Name	Description
250x28	Number of received publications	-
250x29	Amount of timeouts	-
250x30	Amount of missing sequence numbers	The subscriber of a publication computes the difference between present and last received sequence number. If the value of the difference is greater than one, certain publications have not been received.

Registers for error detection

If a subscription has not received any process data from the assigned publication before timeout, the subscription will generate an error. Further, the operating system writes the address of the bus node into registers 254001 to 254003, with which communication has been terminated.

This helps you to search for the error exactly in this bus node using NetCopy commands.

Register	Name	Description
254001	GNN	The Global Node Number that was extracted from the ID of the missing publication
254002	IP address	
254003	Port number	

Important Note:

These are the prerequisites for all three registers to display these values correctly:

- JetSym was used for engineering the system.
- Only the IDs assigned by Hardware Manager have been used.
- This configuration has also been uploaded to the controller.

Registers for administration of all publications

There are several registers available which go with all publications.

Register	Name	Description
255000	Status	Status register
255001	Command	Command register
255002	ID in case of error	Displays the ID of the publication, in which an error has occurred.
255003	Amount	Amount of all publications
255004	CRC	16-bit CRC (Cyclic Redundancy Code) of the publication configuration file

Publisher status

Status registers of all publications

From MR 255000, you can read the collective status of all publications. In case of an error, you first read out the ID of the publication, in which an error has occurred.

Meaning of the individual bits		
Bit 0	**Error in CRC computing of the configuration file**	
	0 =	No error has occurred.
	1 =	For CRC computing, the configuration file does not exist. For this reason, CRC computing has not taken place.
Bit 1	**Error in connection with a publication**	
	1 =	An error has occurred in a publication.
Bit 7	**Publication is functioning**	
	0 =	If a publication fails, bit 7 is reset.
	1 =	The publications are functioning.

Module register properties	
Type of access	Read

Publisher command

Command registers of all publications

Via MR 255001, you transmit commands to all publications.

Commands	
102	**Reboot all publishers**
105	**Stop all publishers**
110	**Acknowledge error**

Selecting a publication

The following registers let you select a publication:

- The index is for selecting publications.
    - If the publication exists, R 255x11 shows its ID.
    - If the publication does not exist, R 255x11 shows value **-1**.
- In this case, enter the ID of the publication into R 255x11.
    - If the publication exists, the content of R 255x11 is kept.
    - If the publication does not exist, R 255x11 shows value **-1**.

Register	Name	Description
255x10	Index	Index of the publications: 0: Selects the first publication 1: Selects the next publication 2: etc.
255x11	ID	The ID of the publication is entered

Configuring a publication

The following registers show the configuration of a publication, which you have selected via R 255x10 and R 255x11.

Register	Name	Description
255x20	Status	Bit 0: Publication transmitted
255x21	Mode	0: Cyclic 1: Upon request
255x22	Number of variables	As configured
255x23	Group address	As configured
255x24	Hash	Internal usage
255x25	Sequence number	Internal usage
255x26	Data size	Internal usage
255x27	Timeout in ms	Bus cycle
255x28	Number of publications sent	-
255x29	Number of retries	-
255x30	Number of transmit errors	-

Network registers, network inputs and outputs

Introduction

The network registers, network inputs and outputs let you access in transparent mode, at cyclic data interchange, registers, inputs and outputs of remote nodes. The controller accesses the local image of the cyclic data.

Prerequisites

These are the prerequisites for using the registers, inputs and outputs at cyclic data interchange:

- Via publish/subscribe, the data are interchanged in cyclic mode.

Properties

Network registers, network inputs and outputs are not used in cyclic data interchange:

- If network registers of non-cyclic data interchange are accessed, the controller generates acyclic network register access.
- If network inputs and outputs of non-cyclic data interchange are accessed, the controller does not generate acyclic network register access. There are no data being transmitted via network.

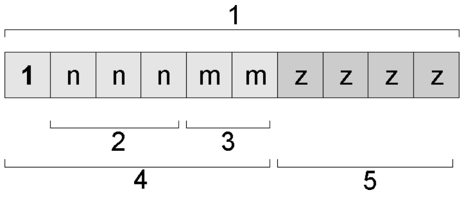
Advantages of network registers, network inputs and outputs

Advantages of network registers, network inputs and outputs in cyclical data interchange as compared with acyclic data interchange:

- The operating system cyclically interchanges data of the registers, inputs and outputs with other network nodes.
- This results in network load optimization.
- This is a very quick access, as, at the instance of use, only the local images of the data have to be accessed.

Register addressing scheme

The addressing scheme for network registers is as follows:



No.	Element	Description
1	Register number	Supports direct access
2	First part of register prefix: Bus node ID, GNN	nnn = 001 ... 199: ID of the network node, referred to as Global Node Number
3	Second part of register prefix: Number of the function module	mm = 02 ... 17: Number of the JX3 module of a remote node mm = 91: Registers of the combined digital inputs and outputs of a remote node
4	Part 1 + 2: Register prefix	1nnnmm: The prefix is preceded by a leading ONE.

No.	Element	Description
5	Module register number	zzzz = 0000 ... 9999

Network registers for accessing JX3 modules

Characteristic feature of the register number for access to remote JX3 modules: The value of the second part of the register prefix is the number of the module at the JX3 system bus (02 ... 17).

In cyclic data interchange, access to the process data of the remote JX3 modules is made via network registers.

For further information on configuration of data interchange and generated variables for access to JX3 modules, please turn to chapter **Hardware Manager** (see page 37).

Register overview - Inputs and outputs

The register number, in which the digital inputs and outputs of the remote nodes have been combined, is characterized by the value being 91 in the second part of the register prefix.

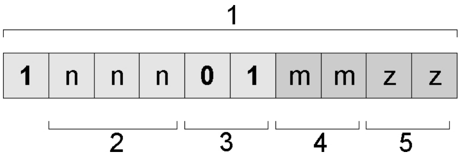
Overview

Registers	Description
1nnn914000 ... 1nnn914030	32 combined inputs
1nnn914060 ... 1nnn914092	16 combined inputs
1nnn914120 ... 1nnn914153	8 combined inputs
1nnn914200 ... 1nnn914230	32 combined outputs
1nnn914260 ... 1nnn914292	16 combined outputs
1nnn914320 ... 1nnn914353	8 combined outputs

Where nnn = GNN: 000 ... 199

Addressing scheme - Inputs and outputs

The addressing scheme for the digital network inputs and outputs at cyclic data interchange is as follows:



No.	Element	Description
1	I/O number	Supports direct access
2	Bus node ID, GNN	nnn = 001 ... 199: ID of the network node, referred to as Global Node Number.
3	Designation:**01**: I/O 01 as a fixed number	01: 01 indicates that a JX3 module is to be addressed.
4	Module number	mm = 02 ... 17: Number of the JX3 module of a remote node
5	Module-specific I/O number	zz = 01 ... 16: Specifies which input/output on the module is to be addressed

Example

Via network, a controller and a bus node JX3-BN-ETH are connected. A JX3-DO16 is connected to the bus node. The JX3-DO16 has got I/O module number 3.

Task:

The outputs of the JX3-DO16 are to be activated or deactivated as follows:

Step	Description
1	All outputs with odd numbers are active for half a second, while all outputs with even numbers are deactivated.
2	All outputs with even numbers are active for half a second, while all outputs with odd numbers are deactivated.
3	There is a moving light from output 1 to output 16; each corresponding output is activated for 200 ms.
4	Proceed with step 1.

Solution:

Configure the network group in the JetSym Hardware Manager and write an application program. Download both to the network nodes.

Related topics

- **Hardware Manager** (see page 37)

1.3 Hardware Manager

Introduction

The Hardware Manager lets you easily configure the peripheral devices. If possible, always use the Hardware Manager that is part of JetSym. Making configurations by hand is complicated and prone to errors

Detailed information

For detailed information on hardware configuration using Hardware Manager, refer to the JetSym help.

Contents

Hardware Manager

Hardware Manager

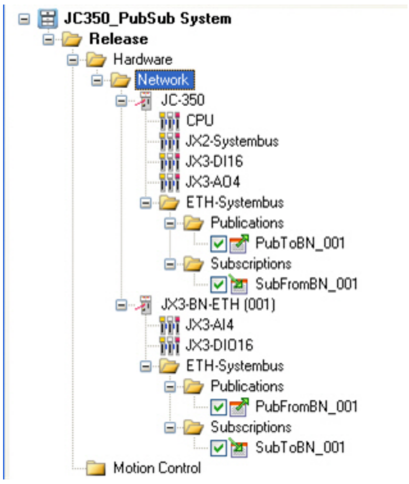
The Hardware Manager manages all connected hardware components.

The Hardware Manager assists you in the following aspects:

- Engineering and configuring control systems and bus nodes
- Engineering modules and axes at the JX2 system bus and configuring axes at the JX2 system bus
- Engineering JX3 modules at a JX3-BN-ETH, JC-3xx and a JC-4xx
- Engineering and configuring Ethernet axes
- Engineering an axis group (path group and technology group)
- Configuring a path group
- Configuring technology group

Launching the Hardware Manager

For launching the Hardware Manager, klick, in JetSym, the tab **Hardware**. As an alternative, launch the Hardware Manager via keys **[Alt] + [5]**.



Related topics

- **Ethernet system bus** (see page 5)

1.4 Error handling on the Jetter Ethernet system bus

Introduction

This chapter covers error handling on the Jetter Ethernet system bus.

Contents

Acyclic data interchange - Error handling

Introduction

The programmer uses the following information for error handling:

- Return values of the commands
- JetIP networking registers and flags

NetCopy() and NetBit()

For error handling, use the return values of the respective command. You will find them in the JetSym online help.

Jetter AG recommends not to execute error handling via the registers and flags of the JetIP network.

Network registers

Error logging for network registers and flags of the JetIP networking:

Registers/flags	Description
Flag 2075	Errors at acyclic data interchange
Register 232710	Amount of errors at acyclic data interchange
Register 232711	Error code of the latest acyclic data interchange

Error message during CRC computing

Detecting the error

Both publisher and subscriber carry out a CRC of their configuration files. The calculated value can be read from registers 255004 and 250004. If there is no configuration file, they report an error.

Root cause of the error

This error may be caused by the following root cause:

- CRC computing failed, because there is no configuration file.

Response of the device to this error

The operating system of the device responds to the error by taking the following steps:

Step	Description
1	The operating system sets bit 0 in the status register of the publisher (R 255000) or of the subscriber (R 250000).

Fixing the root cause

Deploying a configuration file

Acknowledging the error

After deploying a configuration file, restart both publisher and subscriber.

Error message on part of a subscription

Detecting the error

If a subscriber has not received any process data from the assigned publisher before timeout, the subscriber will generate an error. The subscriber for the subscription of which the error has been generated, can run either on a controller or on a network node. The remote network node is a JX3-BN-ETH, for example.

Root cause of the error

The error may be caused as follows:

- Communication with the network client providing the process data is terminated.

Response of the device to this error

The operating system of the device responds to the error by taking the following steps:

Step	Description	
1	Sets bit 1 in R 250000.	
2	Writes the subscription ID to R 250002.	
3	Sets flag 2081.	
4	Writes value 11103 and the ID to the error buffers. The error buffer can be accessed via registers 380000 ff. (error history).	
5	Writes the GNN of the network node communication with which has been terminated to R 254001.	
6	Writes the IP address of the network node communication with which has been terminated to R 254002.	
7	Writes the port number of the network node communication with which has been terminated to R 254003.	
8	**If ...**	**... then ...**
	... flag 2080 is set,	... bit 3 is set in R 210004 and R 200008. The red status LED of the controller is lit.

Fixing the root cause

By means of NetCopy commands, you can precisely locate the error and fix the root cause. This works, because GNN, IP address and port number of the other network node are known.

Acknowledging the error

To acknowledge the error, write command 110 to register 250001.

Controller evaluates errors reported by a remote network node

Access to the status registers

The controller has got read access to the contents of the following status registers of all network nodes on the Jetter Ethernet system bus.

The contents are accessed via registers 39nnn0 through 39nnn5. (GNN: nnn = 001 ... 199).

Registers	JX3-BN-ETH, JX3-COM-EIPA	Controller
Error register	200008	39nnn0
Enhanced error register 1	200009	39nnn1
Enhanced error register 2	200010	39nnn2
JetSync status	240010	39nnn3
Subscriber status	250000	39nnn4
Subscription ID	250002	39nnn5

The operating system writes the ID of the subscription for which last an error has been reported to register 250002.

Locating faults

If the value of register 39nnn0 is unequal zero, an error has occurred. A network node has reported this error to the controller via its status registers.

In consequence, the operating system of the controller reacts by taking the following steps:

<table>
<tr><th>Step</th><th colspan="3">Description</th></tr>
<tr><td>1</td><td colspan="3">The operating system sets bit 10 in R 200009.</td></tr>
<tr><td rowspan="2">2</td><th>If ...</th><th>... or ...</th><th>... then ...</th></tr>
<tr><td>... Bit x = 1 of R 200009,</td><td>Bit x = 1 of R 200010,</td><td>... the operating system sets bit 7 of R 200008.</td></tr>
<tr><td>3</td><td colspan="3">The operating system enters the GNN of the network node having last reported an error to the controller into R 394001.</td></tr>
<tr><td>4</td><td colspan="3">The operating system enters the IP address of the network node having last reported an error to the controller into R 394002.</td></tr>
<tr><td>5</td><td colspan="3">The operating system enters the port number of the network node having last reported an error to the controller into R 394003.</td></tr>
</table>

Fixing the root cause

By means of NetCopy commands, you can precisely locate the error and fix the root cause. This works, because GNN, IP address and port number of the other network node are known.

Make sure the contents of registers 39nnn0 through 39nnn5 are read by the application program. Further registers having got a value unequal zero indicate that further network nodes have reported an error. Make sure you also clear these errors.

1.5 NetConsistency function

Target

The goal of NetConsistency is automated comparison of actual system properties with the set system properties of network nodes. If the actual system properties are not in accordance with the set system properties, the respective issues are automatically replaced within the system by the set system properties.

Application

The user can take the following actions by applying NetConsistency:

- Exchanging a defective system component, a network node by simply adjusting it to the new system component within an engineered plant.

  The JetControl, which is the NetConsistency master, automatically configures the new system component by all kinds of information given in the former system component.
- Easily updating an already existing plant:

  Download of the new system properties to the NetConsistency master JetControl, is required. JetControl automatically recognizes the difference between the former and the actual system configuration. It assigns the new system properties to the respective places.

System properties

Possible system properties are:

- Network parameters (IP address, port number, subnet mask, default gateway)
- Parameter data
- Configuration data

Configuration data

The JetSym Hardware Manager generates the configuration and parameter data. The Hardware Manager transfers the data to JetControl through the feature **Compare program/Download.**

The NetConsistency master

The NetConsistency feature supplies a NetConsistency master defined in the system. Only a JetControl can be a NetConsistency master.

Prerequisites

There are the following prerequisites for using NetConsistency:

- JetSym as of V 5.1.0
- At least one NetConsistency master:

Product	As of version
JC-940MC	V. 1.05.0.08 V. 1.06.6.01 (testing os)
JC-945MC	V. 1.01.0.00
JC-440(MC)	V 1.02.0.00
JC-340, JC-350	V. 1.23.0.04

- NetConsistency slaves: Min. 1, max. 64

Produkt	As of version
JC-310-JM	V 1.22.0.00
JM-200-ETH	V 1.22.0.00
Ethernet axis JM-xxx (JM-2xx-OEM)	V 2.07.0.37
Ethernet axis MC-JM-xxx (JM-2xx-OEM)	V 2.07.0.37
JX3-BN-ETH	V 1.18.0.02
JX3-COM-EIPA	V 1.01.0.00
JX3-COM-PND	V 1.03.0.06

Contents

Topic **Page**

NetConsistency function

Restrictions

- NetConsistency is only available for the Jetter Ethernet system bus.
- The network nodes have to be connected to the same subnet.
- Only if JetIPScan is active, NetConsistency will be executed. JetIPScan is active, if bit 2 of R 202962 is set.
- JetControl executes NetConsistency only once at booting the JetControl, which is the master of NetConsistency.

Function

The NetConsistency feature in its actual version comprises the following system properties:

- Network parameters
  - IP address
  - Subnet mask
  - Default gateway
- Parameter data
- Configuration data

Network parameters

For this, NetConsistency uses JetIPScan. One of the JetIPScan features is to assign network parameters to bus nodes via GNN.

The controller assigns the network parameters to those bus nodes which you have configured in Hardware Manager.

The controller assigns the IP address to those bus nodes which you have configured in Hardware Manager.

As subnet mask, the controller assigns its own subnet mask to the bus node.

As default gateway, the controller assigns its own IP address or its own default gateway to the bus node:

Product	Assigned default gateway
JC-940MC and JC-945MC, if only ETH1 has been configured	Default gateway of the controller
JC-940MC and JC-945MC, if ETH2 and/or ETH3 have been configured	IP address of ETH1 of the controller
JC-340, JC-350 and JC-440	Default gateway of the controller

Parameter and configuration data

For this, NetConsistency uses FTP. Via FTP, parameter and configuration files are transferred to the bus nodes.

The controller stores the parameter and configuration files of all bus nodes in a backup directory. For each bus node, the backup directory holds a folder called *NetNode* plus the GNN attached to the name.

Example: File system of the controller: */SysConfig/Backup/NetNode002*

JetSym transfers all parameter and configuration files to the backup directory of the controller by comparison and download.

For uploading the parameter and configuration files, the addressed bus nodes are rebooted after file transfer. Bus nodes **without** parameter and/or configuration files are not rebooted.

The following products having got parameter and configuration files are rebooted:

Product	As of version
JX3-BN-ETH	V. 1.18.0.02
JX3-COM-EIPA	V. 1.01.0.00
JX3-COM-PND	V. 1.03.0.06

The following products **not** having got parameter and configuration files are **not** rebooted:

Product	As of version
JC-310-JM	V. 1.22.0.00
JM-200-ETH	V. 1.22.0.00
Ethernet axis JM-xxx (JM-2xx-OEM)	V 2.07.0.37
Ethernet axis MC-JM-xxx (JM-2xx-OEM)	V 2.07.0.37

System launch of the bus nodes without non-volatile storage of the IP address

At system launch, the bus nodes use the GNN set via their own DIP switch sliders 1 to 8. This applies, until the network parameters configured in Hardware Manager via JetControl - which is the NetConsisteny master - are assigned to the bus node.

Non-volatile storage via NetConsistency of the network parameters assigned last is not implemented.

We recommend the following: When configuring the bus nodes in Hardware Manager, use the GNN as least significant byte of the IP address.

There is **no** non-volatile storage for the IP addresses of the following products:

Product	As of version
JC-310-JM	V. 1.22.0.00
JM-200-ETH	V. 1.22.0.00
Ethernet axis JM-xxx (JM-2xx-OEM)	V 2.07.0.37
Ethernet axis MC-JM-xxx (JM-2xx-OEM)	V 2.07.0.37
JX3-COM-EIPA	V. 1.01.0.00
JX3-COM-PND	V. 1.03.0.06

System launch of the bus nodes with non-volatile storage of the IP address

The network parameters assigned by NetConsistency are saved to the non-volatile store in the **config.ini** file of the bus nodes, if the DIP switch sliders 9 through 12 of the JX3-BN-ETH are in the position listed below.

DIP switch	Position
9	ON
10	OFF
11	OFF
12	OFF

The GNN of the bus nodes are configured via DIP switch sliders 1 through 8. The coding is binary, which means that, for example, switch 3 in position ON means GNN = 4.

At system launch, the bus nodes apply the network parameters which are stored in **/System/config.ini**. Immediately after this, the network parameters configured in Hardware Manager via JetControl - which is the NetConsistency master - are assigned to the bus nodes. If NetConsistency has already assigned the network parameters configured in Hardware Manager to the bus nodes, these bus nodes already use these for system launch.

The bus nodes store the assigned network parameters in the file **/System/config.ini** in the file system. In this case, the already existing file **/System/config.ini** is overwritten.

The DIP switches of the bus nodes set the GNN. This is for identifying the bus nodes within the system, so the network parameters configured in Hardware Manager can be assigned.

There is non-volatile storage for the IP addresses of the following products:

Product	As of version
JX3-BN-ETH	V. 1.18.0.02
JX3-COM-EIPA	V 1.05.0.02 (Beta-OS)
JX3-COM-PND	V 1.05.0.02 (Beta-OS)

Assigning the network parameters dependent on the GNN and transfer of the parameter and configuration data

Introduction

Via JetIPScan, NetConsistency sets the network parameters automatically for the following devices and automatically transfers the parameter and configuration data.

Via JetIPScan, NetConsistency sets the network parameters automatically for the following devices:

- Ethernet axes JM-xxx (JM-2xx-OEM, JM-200-ETH, JC-310-JM)
- Ethernet axes MC-xxx (JM-2xx-OEM, JM-200-ETH, JC-310-JM)
- JX3-BN-ETH
- JX3-COM-EIPA
- JX3-COM-PND

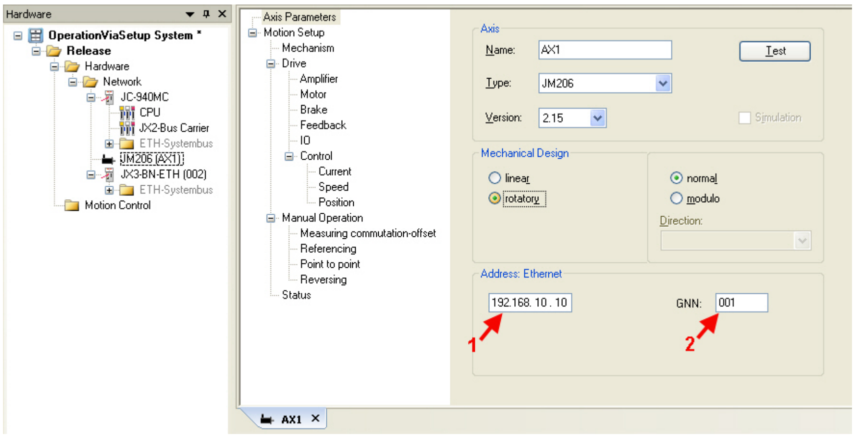
NetConsistency automatically transfers via FTP the parameter and configuration data to the following devices:

- JX3-BN-ETH
- JX3-COM-EIPA
- JX3-COM-PND

Automatically means that when exchanging a network node, you **only** have to take over the GNN (Global Node Number) which has got the same function as the settings of the DIP switch belonging to the former network node.

Any further settings are transmitted to the network node by the JetControl. Via JetIPScan, NetConsistency assigns the network parameters as set in Hardware Manager for the respective network nodes and transfers via FTP the parameter and configuration data as set in Hardware Manager for the respective network nodes.

Assigning the IP address and the GNN to the JM-200 with option -ETH



Axis

Name: AX1

Type: JM206

Version: 2.15

Mechanical Design

○ linear
● rotatory

● normal
○ modulo

Address: Ethernet

192.168. 10 . 10

GNN: 001

Step	Action
1	Set the GNN at the DIP switch (DIP switch sliders 1 through 8) of the MC-JM-xxx or JM-xxx.
2	Start JetSym.
3	Select the device MC-JM-xxx or JM-xxx in Hardware Manager.
4	Select the tab **Axis Parameters.**
5	As an address for **Ethernet Networks (1),** enter the IP address. **A special hint:** Use the GNN as least significant byte of the IP address.
6	As **GNN (2),** enter the Global Node Number of the device. The number has to match the settings of the DIP switch at the device.

Result: IP address and GNN have been assigned to the device.

Setting the DIP switch at the MC-JM-xxx or JM-xxx

The MC-JM-xxx or JM-xxx uses the settings of the DIP switch sliders 1 through 8 as GNN. The coding is binary.

Illustrations

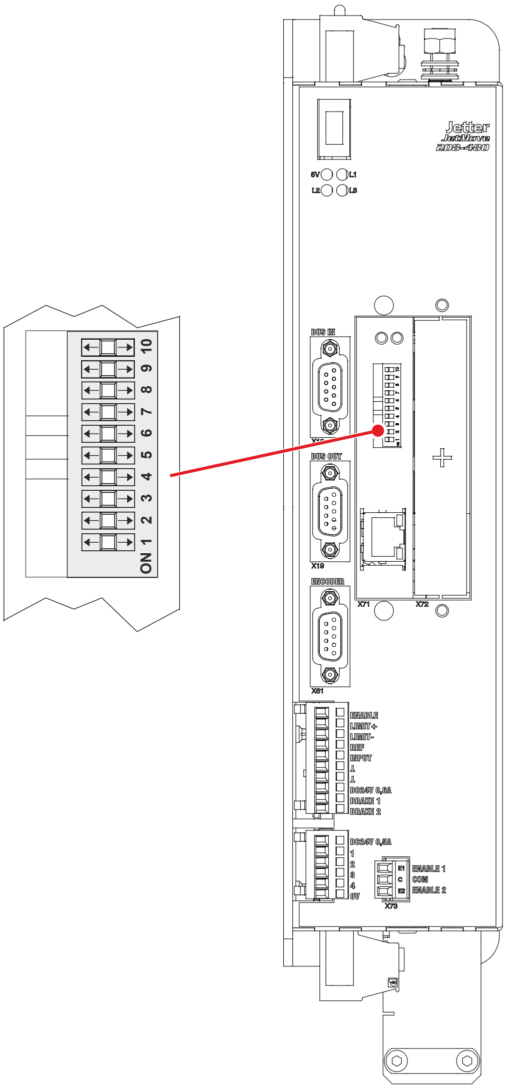
GNN = 4: Switch 3 is set to ON. All other DIP switch sliders are set to OFF.

GNN = 5: DIP switch sliders 1 and 3 are set to ON. All other DIP switch sliders are set to OFF.

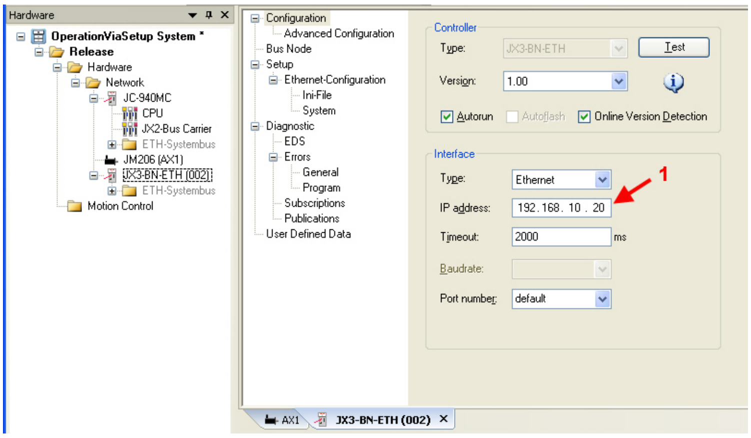
GNN = 8: Switch 4 is set to ON. All other DIP switch sliders are set to OFF.

Position of the DIP switch sliders at the MC-JM-xxx or JM-xxx

If at the digital servo amplifier an Ethernet port is integrated, there is a 10-pin DIP switch available. The illustration below shows the position of the DIP switch sliders.



Assigning the IP address and the GNN to the JX3-BN-ETH and JX3-COM-EIPA/-PND



Type: JX3-BN-ETH

Version: 1.00

☑ Autorun ☐ Autoflash ☑ Online Version Detection

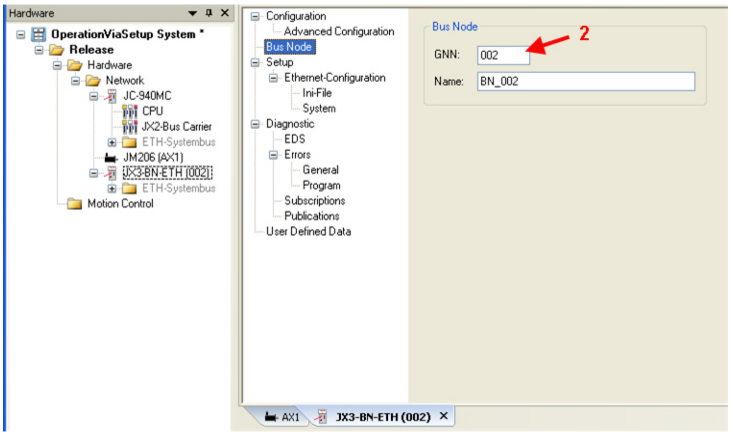
Type: Ethernet

IP address: 192 . 168 . 10 . 20

Timeout: 2000 ms

Baudrate:

Port number: default



GNN: 002

Name: BN_002

Step	Action
1	Set the GNN at the DIP switch (DIP switch sliders 1 through 8) of the JX3-BN-ETH or JX3-COM-EIPA/-PND.
2	Set the GNN operating mode at the DIP switch (DIP switch sliders 9 through 12) of the JX3-BN-ETH or JX3-COM-EIPA/-PND.
3	Start JetSym.
4	Select the device JX3-BN-ETH or JX3-COM-EIPA/-PND in Hardware Manager.
5	Select the tab **Configuration.**
6	As **IP Address (1),** enter the IP address.
7	Select the tab **Bus Node.**
8	As **GNN (2),** enter the Global Node Number of the device. The number has to match the settings of the DIP switch at the device.

Result: IP address and GNN have been assigned to the device.

Setting the DIP switch sliders at the JX3-BN-ETH and JX3-COM-EIPA/-PND

The settings of DIP switch sliders 9 through 12 activate remanent storage of the assigned network parameters in the **config.ini file**.

Set DIP switch slider 9 to ON and DIP switch sliders 10 through 12 to OFF.

The settings of DIP switch sliders 1 through 8 are for configuring the IP address. The coding is binary.

Illustrations

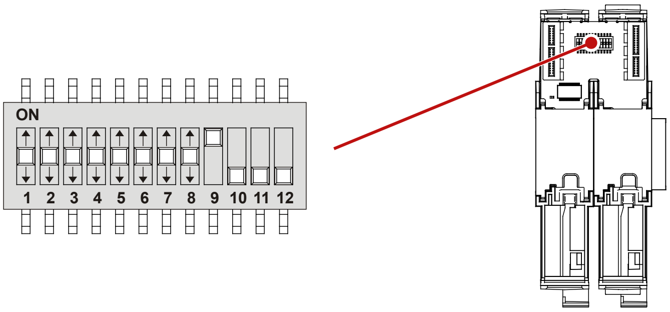
GNN = 4: Switch 3 is set to ON. All other DIP switch sliders are set to OFF.

GNN = 5: DIP switch sliders 1 and 3 are set to ON. All other DIP switch sliders are set to OFF.

GNN = 8: Switch 4 is set to ON. All other DIP switch sliders are set to OFF.

Position of the DIP switch sliders at the JX3-BN-ETH and JX3-COM-EIPA/-PND

The illustration below shows the position of the DIP switch sliders.



Network topology

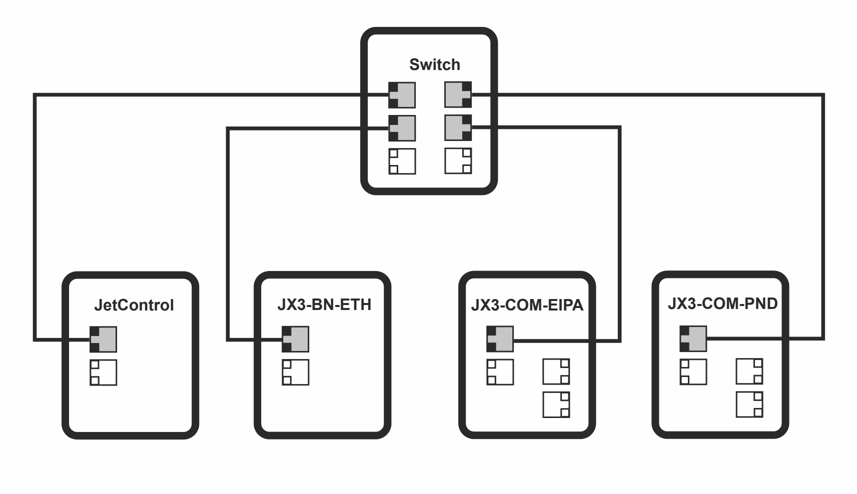
If NetConsistency is applied, the network nodes **must** be arranged in star-shaped topology, see figure *star-shaped topology*.

Background

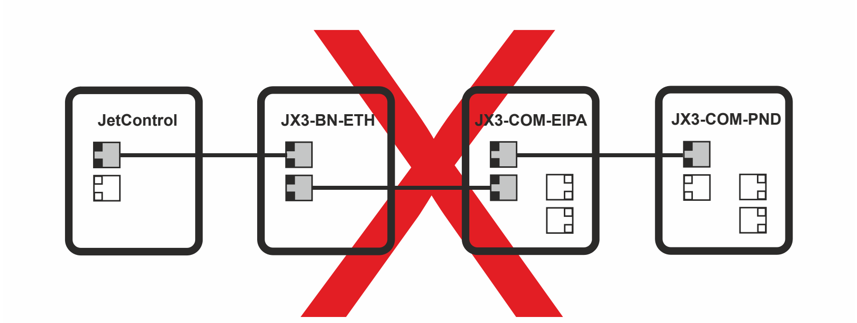
After transferring the parameter and configuration files via FTP, the controller reboots the respective bus nodes.

This means that the Ethernet switch placed on the bus nodes is rebooted and can therefore not forward any more network frames. For line topology - see figure *Line topology*, this means that, for example, JX3-BN-ETH, being the first bus node in the line topology of JetControl is given the command to reboot. Yet, JX3-COM-EIPA being the next bus node in the line cannot receive another reboot command given by the controller, because the Ethernet switch of the first bus node, JX3-BN-ETH, is rebooting at that moment. For this reason, using line topology together with NetConsistency is not permitted.

Star-shaped topology



Line-shaped topology



Compare program/Download

When you have set all parameters in Hardware-Manager, transfer the settings to the system parameters via **Compare program/Download**.

This is done by the following instruction in Hardware Manager:

- Compare program/Download (right mouse button on **release**)

Assigned network parameters

At system launch, the controller assigns the following network parameters to the connected network nodes:

- IP address
- Subnet mask
- Default gateway

IP address

The controller assigns the IP address as set in Hardware Manager.

Subnet mask

The controller assigns its own subnet mask.

Default gateway

The assigned default gateway depends on the controller type:

Product	Assigned default gateway	
JC-340, JC-350, JC-440MC	Default gateway of the controller	
JC-940MC	**If ...**	**... then ...**
	... neither with ETH2 nor with ETH3 network parameters have been configured,	... the controller assigns the default gateway of ETH1.
	... with ETH2 or with ETH3 network parameters have been configured,	... the controller assigns the IP address of ETH1 as the default gateway.
JC-945MC	**If ...**	**... then ...**
	... with ETH3 no network parameters have been configured,	... the controller assigns the default gateway of ETH1.
	... with ETH3 network parameters have been configured,	... the controller assigns the IP address of ETH1 as the default gateway.

Then, the controller transfers the parameter and configuration files to the network nodes and reboots the network node concerned.

Activating and deactivating JetIPScan in JetControl

Introduction

You have to enable JetIPScan by making an entry into the system command register. The settings are remanent.

Enable JetIPScan

To enable JetIPScan, proceed as follows:

Step	Action
1	Switch the device ON.
2	Write value 1112502132 (0x424f6f74) to password register 202960.
3	Enter value 331 into system command register 202961.
⇨	Bit 2 of register 202962 is set and JetIPScan is enabled.

Disable JetIPScan

To disable JetIPScan, proceed as follows:

Step	Action
1	Switch the device ON.
2	Write value 1112502132 (0x424f6f74) to password register 202960.
3	Enter value 330 into system command register 202961.
⇨	Bit 2 of register 202962 is cleared and JetIPScan is disabled.

Program run at system launch

Program run at system launch

The following table shows the program run at system launch:

Step	Description
1	When booting, the network nodes of non-volatile IP address storage take the IP address from the file **config.ini**.
2	When booting the JetControl, each network node is assigned a network configuration (IP address, subnet mask, gateway address) via JetIPScan while the NetConsistency function is executed. Files for network nodes holding parameter and/or configuration files are transferred via FTP. For uploading the new parameter and configuration data, the addressed network nodes are rebooted after file transfer.
3	After the booting the JetControl, and thus, after executing the NetConsistency function, the network nodes can be addressed via the network configurations as set in Hardware Manager. The parameter and configuration data of Hardware Manager are stored to the network nodes.

Program run at NetConsistency

NetConsistency passes the following states of the JetControl boot process:

Step	Description
1	The basic driver is initialized.
2	An instance is initialized.
3	The functions of NetConsistency are executed.

Register description - NetConsistency basic driver

Registers - Overview

Register	Description
470000 ... 470008	Cookie
470009	Version number
470010	Status
470011	Command
470020	Maximum possible amount of instances
470021	Number of instances ready for operation
470030 ... 470035	Restrictions
470040 ... 470157	Locating faults

R 470000 ... R 470008

Cookie

This register shows the beginning of the NetConsistency registers. This way, orientation is simplified.

Module register properties	
Type of access	Read
Value after reset	NetConsistency
Data type	RegString

R 470009

Version of NetConsistency

R 470009 shows the version of NetConsistency.

Module register properties	
Values	IP#0.00.0.00 ... IP#9.99.9.99
Type of access	Read
Value after reset	Version of NetConsistency

R 470010

Status register

R 470010 shows the status of the NetConsistency basic driver.

Meaning of the individual bits

Bit 0	**Error**	
	0 =	No error
	1 =	Error

Bit 2	**Status of initialization**	
	0 =	Basic driver not initialized
	1 =	Basic driver initialized

Module register properties	
Type of access	Read
Value after reset	0x00000004

R 470011

Command register

The value is 0, as there are no commands.

R 470020

Maximum possible number of instances

R 470020 shows the maximum possible number of NetConsistency instances. The actual value is always 1.

Module register properties	
Values	1
Type of access	Read
Value after reset	1

R 470021

Number of instances ready for operation

R 470021 shows the number of NetConsistency instances.

Module register properties	
Values	0 ... 1
Type of access	Read
Value after reset	1

R 470030

Maximum number of error messages for the logger

R 470030 sets the maximum number of error messages which are transferred to the logger by NetConsistency.

Module register properties	
Values	10
Type of access	Read
Value after reset	10

R 470031

Number of error messages transmitted to the logger

R 470031 displays the number of error messages transmitted to the logger by NetConsistency.

Module register properties	
Values	0 ... 10
Type of access	Read

R 470032

Maximum number of warnings for the logger

R 470032 sets the maximum number of warnings forwarded to the logger by NetConsistency.

Module register properties	
Values	10
Type of access	Read
Value after reset	10

R 470033

Number of warnings forwarded to the logger

R 470033 displays the number of warnings transmitted to the logger by NetConsistency.

Module register properties	
Values	0 ... 10
Type of access	Read

R 470034

Maximum possible number of error history entries

R 470034 defines the maximum possible number of error history entries.

Module register properties	
Values	10
Type of access	Read
Value after reset	10

R 470035

Number of entries in the error history

R 470035 displays the number of error messages entered into the error history by NetConsistency.

Module register properties	
Values	0 ... 30
Type of access	Read

R 470040

Error numbers

R 470040 shows the error numbers.

Error name	Error number
NoError	0
GroupFunction	-1
GroupCStandard	-2
GroupJetterFileSystem	-3
GroupJetterLogger	-4
GroupJetterOS	-5
GroupJetterParserXml	-6
GroupJetterPcom	-7
GroupUtility	-8
GroupJetIpScan	-9
Api	-100
Manager	-110
ManagerInit	-111
ManagerDeinit	-112
ManagerMultipleInit	-113
Instance	-120
InstanceInit	-121

Error name	Error number
InstanceDeinit	-122
StateMachine	-140
StateMachineInit	-141
StateMachineDeinit	-142
Error	-150
ErrorInit	-151
ErrorDeinit	-152
Warning	-160
WarningInit	-161
WarningDeinit	-162
Register	-170
RegisterInit	-171
RegisterDeinit	-172
Xml	-180
XmlInit	-181
XmlDeinit	-182
XmlInvalidGnn	-183
XmlInvalidIpAddress	-184
XmlTagNetConsistencyAttrVersion	-185
XmlTagNetNodesAttrCount	-186
XmlTagNetNodeAttrName	-187
XmlTagNetNodeAttrType	-188
XmlTagNetNodeAttrGnn	-189
XmlTagPcomAttrName	-190
XmlTagPcomAttrCommand	-191
XmlTagPcomAttrModuleId	-192
XmlTagPcomAttrTypeId	-193
XmlTagIpAddress	-194
XmlTagJetIPAttrPort	-195
XmlTagJx3SystembusAttrCrcEdsModuleCount	-196
XmlTagFilesAttrCount	-197
XmlTagFilesAttrCrc	-198
XmlTagFileAttrCrc	-199
XmlTagFileAttrPath	-200
XmlTagFileAttrName	-201
JetModuleReadReg	-300
JetModuleWriteReg	-301

Error name	Error number
Utility	-310
JetIPScan	-320
JetIPScanInit	-321
JetIPScanDeinit	-322
Processing	-330
ProcessingInit	-331
ProcessingDeinit	-332

Module register properties	
Values	-2^{16} ... 0
Type of access	Read

R 470041

Time of the error in milliseconds

R 470041 displays the time of the error in milliseconds. When JetControl has been activated for 50 days, an overflow occurs.

Module register properties	
Values	0 ... 2^{32} ms = 0 ... 50 days
Type of access	Read

R 470042

Instance, at which the error occurred

R 470042 displays the instance, at which the error occurred. In fact, only one instance is possible.

Module register properties	
Values	0: First instance
Type of access	Read

R 470043

Number of error parameters

R 470043 shows the number of error parameters.

Module register properties	
Values	0 ... 5
Type of access	Read

R 470044

Error parameter 1

R 470044 shows error parameter 1. The value is only valid, if R 470043 ≥ 1.

Module register properties	
Values	0 ... 2^{32}
Type of access	Read

R 470045

Error parameter 2

R 470045 shows error parameter 2. The value is only valid, if R 470043 ≥ 2.

Module register properties	
Values	0 ... 2^{32}
Type of access	Read

R 470046

Error parameter 3

R 470046 shows error parameter 3. The value is only valid, if R 470043 ≥ 3.

Module register properties	
Values	0 ... 2^{32}
Type of access	Read

R 470047

Error parameter 4

R 470047 shows error parameter 4. The value is only valid, if R 470043 ≥ 4.

Module register properties	
Values	0 ... 2^{32}
Type of access	Read

R 470048

Error parameter 5

R 470048 shows error parameter 5. The value is only valid, if R 470043 = 5.

Module register properties	
Values	0 ... 2^{32}
Type of access	Read

R 470049

Number of characters of the error message

R 470049 shows the number of characters of the error message. The error message has been stored to registers 470050 ... 470157.

Module register properties	
Values	0 ... 300
Type of access	Read

R 470050 ... R 470157

Text of the error message

These registers contain the text of the error message.

Module register properties	
Type of access	Read
Value after reset	""
Data type	RegString

Register description of the NetConsistency instance

Register overview

Register	Description
471010	Status
471011	Command

R 471010

Status register

R 470010 shows the status of the first NetConsistency instance.

Meaning of the individual bits		
Bit 0	**Error**	
	0 =	No error
	1 =	Error
Bit 2	**Status of initialization**	
	0 =	The first instance has not been initialized
	1 =	The first instance has been initialized
Bit 3	**Status of execution**	
	0 =	No execution
	1 =	Execution in process

Module register properties	
Type of access	Read
Value after reset	0x00000004

R 471011

Command register

The value is 0, as there are no commands.

Error handling at NetConsistency

Possibilities of error output

There are the following possibilities of error output:

- Via the logger of NetConsistency and JetIPScan
- Via the enhanced error register R 200009
- Via error number register R 200051 of JetIPScan
- Via error number register R 200061 of NetConsistency

R 200009

Enhanced error register

R 200009 is a bit-coded register.

Meaning of the individual bits		
Bit 12	**Error message by JetIPScan**	
	0 =	No error
	1 =	JetIPScan has reported an error. The error number is contained in R 200051.
Bit 16	**Error message by NetConsistency**	
	0 =	No error
	1 =	NetConsistency has reported an error. The error number is contained in R 200061 and R 470040.

Module register properties	
Type of access	Read

R 200051

Error numbers of JetIPScan

R 200051 shows the error numbers of JetIPScan. The content of this register is identical with JetIPScan MR 13.

Module register properties

Values	0	No error or warning
	5	The user has terminated the function
	1001	The first received response does not match response 2 and 3 (see MR 101x)
	1002	The second received response does not match response 1 and 3 (see MR 102x)
	1003	The third received response does not match response 1 and 2 (see MR 103x)
	-1	All 3 responses are dissimilar (see MR 100x)
	-2	The IP settings of at least one node could not be changed (see MR 140x)
	-3	The JetIPScan function has been invoked, although it is active already
	-10	The length of the set value list is < 1 or > 255, or the pointer to the list is invalid
	-11	A GNN of the set value list is < 1 or > 255, or it is a multiple GNN
	-20 ... -40	Internal error
	-1001 ... -1199	The node has reported the wrong CtrlID or CtrlIDopt (see MR 110x)
	-2001 ... -2199	The node has not called (see MR 120x)
	-3001 ... -3199	Several nodes of the same GNN have called (see MR 130x)
Type of access	Read	

R 200061

Error numbers of NetConsistency

R 200061 shows the error numbers of NetConsistency, see R 470040.

Related topics

- **Register description - NetConsistency basic driver** (see page 58)
- **Register description - JetIPScan** (see page 69)

1.6 JetIPScan - Register description

Introduction

This chapter describes the registers from which the status information of the JetIPScan feature can be read out. You can use these registers for debugging or diagnostics. Further features, such as, for example, checking the network configuration, cannot be triggered this way.

Contents

Topic **Page**

Register numbers

Introduction

Status information is displayed within the registers of a coherent register block. The basic register number of this block is dependent on the controller.

Register numbers

Basic register number	Register numbers
520000	520000 ... 522999

Determining the register number

In this chapter, only the last four figures of a register number are specified. e.g. MR 1499. Add to this module register number the basic register number of the corresponding device to determine the complete register number, for example 521499.

Registers - Overview

Register	Description
MR 0 ... MR 13	Global status
MR 1000 ... MR 1499	Warnings and errors
MR 2000 ... MR 2399	SET and ACTUAL configurations

Global status - Register description

Introduction

The current I/O size can be read from this register.

MR 0

State of the total

In MR 0. the controller signals a summary of status messages in bit-coded mode.

Meaning of the individual bits		
Bit 0	**Function enable** This bit corresponds to bit 2 of the system status register 202962.	
	0 =	JetIPScan client - OFF
	1 =	JetIPScan client - ON
Bit 1	**Collective error message**	
	1 =	Reg 13 contains value 0

Module register properties	
Type of access	Read
Value after reset	Bit 0: Depends on release status. Bit 1: 0

MR 10

State of execution

Corresponds to the feedback value *State*.

Module register properties		
Values	0	The function is not active. Function terminated.
	1	Waiting for response from network nodes
	2	Send an inquiry frame
	3	Check the replies sent by the nodes
	4	Write the configurations of the nodes
Type of access	Read	

MR 11 Number of cycles

Corresponds to the feedback value *Count*.

Module register properties		
Values	0 ... 3	Number of cycles
Type of access	Read	

MR 12 Number of changes

Corresponds to the feedback value *Changed*.

Module register properties		
Values	0 ... 199	Number of changed network nodes
Type of access	Read	

MR 13 Result of the function

Corresponds to the feedback value *Result* and the register content of the global error number 2000051. This register indicates the value of the latest error or warning. Values greater than zero indicate warnings. Values smaller than zero are error messages.

Module register properties		
Values	0	No error or warning
	5	The user has terminated the function
	1001	The first received response does not match response 2 and 3 (see MR 101x)
	1002	The second received response does not match response 1 and 3 (see MR 102x)
	1003	The third received response does not match response 1 and 2 (see MR 103x)
	-1	All 3 responses are dissimilar (see MR 100x)
	-2	The IP settings of at least one node could not be changed (see MR 140x)
	-3	The JetIPScan function has been invoked, although it is active already
	-10	The length of the set value list is < 1 or > 255, or the pointer to the list is invalid
	-11	A GNN of the set value list is < 1 or > 255, or it is a multiple GNN
Values	-20 ... -40	Internal error

Module register properties

	-1001 ... -1199	The node has reported the wrong CtrlID or CtrlIDopt (see MR 110x)
	-2001 ... -2199	The node has not called (see MR 120x)
	-3001 ... -3199	Several nodes of the same GNN have called (see MR 130x)
Type of access	Read	

Warnings and errors - Register description

Introduction

Detailed diagnostics of the warnings and errors which have occurred can be carried out by means of these registers.

If, during checking and setting the IP address of all nodes a warning or an error occurs, the controller sets the corresponding bit in the registers described below. In this case, the bit corresponds to the GNN of the node.

The GNN of the node and the bit number relate as follows:

Bit number = GNN - 1

As a register contains 32 bit, individual groups of 7 subsequent registers each are created (see table).

Register bit	GNN
Register.0	1
Register.31	32
(Register + 1).0	33
(Register + 1).31	64
(Register + 2).0	65
(Register + 2).31	96
(Register + 3).0	97
(Register + 3).31	128
(Register + 4).0	129
(Register + 4).31	160
(Register + 5).0	161
(Register + 5).31	192
(Register + 6).0	193
(Register + 6).6	199

MR 1000 ... 1006

All 3 responses are dissimilar

The controller scans the network configuration three times and compares the three replies. If all three replies are dissimilar, the controller sets the respective bit in these registers.

Meaning of the individual bits	
Bit = 0	No error
Bit = 1	Error

Module register properties	
Bit number	GNN - 1

Type of access	Read

MR 1010 ... 1016

Reply no. 1 is not the same as replies 2 and 3

The controller scans the network configuration three times and compares the three replies. If replies 2 and 3 are the same, yet reply 1 is different, the controller sets the respective bit in these registers.

Meaning of the individual bits	
Bit = 0	No warning
Bit = 1	Warning

Module register properties	
Bit number	GNN - 1
Type of access	Read

MR 1020 ... 1026

Reply no. 2 is not the same as replies 2 and 3

The controller scans the network configuration three times and compares the three replies. If replies 1 and 3 are the same, yet reply 2 is different, the controller sets the respective bit in these registers.

Meaning of the individual bits	
Bit = 0	No warning
Bit = 1	Warning

Module register properties	
Bit number	GNN - 1
Type of access	Read

MR 1030 ... 1036

Reply no. 3 is not the same as replies 2 and 3

The controller scans the network configuration three times and compares the three replies. If replies 1 and 2 are the same, yet reply 3 is different, the controller sets the respective bit in these registers.

Meaning of the individual bits	
Bit = 0	No warning
Bit = 1	Warning

Module register properties	
Bit number	GNN - 1
Type of access	Read

MR 1100 ... 1106

Wrong CtrlID or CtrlIDopt

A node having got the required GNN has called, yet, the CtrlID or CTRLIDopt do not agree with it.

Meaning of the individual bits	
Bit = 0	No error
Bit = 1	Error

Module register properties	
Bit number	GNN - 1
Type of access	Read

MR 1200 ... 1206

The node has not called

The node having got the required GNN has not called.

Meaning of the individual bits	
Bit = 0	No error
Bit = 1	Error

Module register properties	
Bit number	GNN - 1
Type of access	Read

MR 1300 ... 1306

Multiple call

Several nodes using the same GNN have called. Yet, each node must have a unique GNN.

Meaning of the individual bits	
Bit = 0	No error
Bit = 1	Error

Module register properties	
Bit number	GNN - 1
Type of access	Read

MR 1400 ... 1406

The IP settings could not be changed

When the IP settings of a node have been changed, the controller checks whether the node has taken over these changes.

If the node has not taken over these changes, the controller sets the respective bit in these registers.

Meaning of the individual bits	
Bit = 0	No error
Bit = 1	Error

Module register properties	
Bit number	GNN - 1
Type of access	Read

Configuration - Register description

Introduction

These registers can be used to check the SET configuration and the three received ACTUAL configurations When you have entered the GNN in MR 2000, the controller transfers the values to the 4 register arrays.

MR 2000

GNN

Enter the GNN here.

Module register properties	
Values	1 ... 199
Value after reset	1

MR 2010 ... 2015

SET configuration

These registers let you read the default SET configuration.

Register	Command line parameter
2010	NodeID (GNN)
2011	CtrlID
2012	CtrlIDopt
2013	IpAddr
2014	IpMask
2015	Gateway

MR 2110 ... 2123

ACTUAL configuration 1

These registers let you read the first received ACTUAL configuration.

Register	Command line parameter
2110	NodeID (GNN)
2111	CtrlID
2112	CtrlIDopt
2113	IpAddr
2114	IpMask
2115	Gateway
2120	Quantity
2121	MAC address high
2122	MAC address low

Register	Command line parameter
2123	Sent IP address

MR 2210 ... 2223

ACTUAL configuration 2

These registers let you read the second received ACTUAL configuration.

Register	Command line parameter
2210	NodeID (GNN)
2211	CtrlID
2212	CtrlIDopt
2213	IpAddr
2214	IpMask
2215	Gateway
2220	Quantity
2221	MAC address high
2222	MAC address low
2223	Sent IP address

MR 2310 ... 2323

ACTUAL configuration 3

These registers let you read the third received ACTUAL configuration.

Register	Command line parameter
2310	NodeID (GNN)
2311	CtrlID
2312	CtrlIDopt
2313	IpAddr
2314	IpMask
2315	Gateway
2320	Quantity
2321	MAC address high
2322	MAC address low
2323	Sent IP address

1.7 Administrating the connections of the JetIP/TCP and STX debug server

Introduction

This document covers the connection management enhancements of the JetIP/TCP server and of the STX debug server in a JetControl PLC.

If, for example, the Ethernet cable was unplugged or cut, the node was not able to clear the connection. The connection remained active.

The enhanced connection management allows for the server to clear connections according to criteria that can be set by the user.

Number of connections

The number of simultaneously established connections for the TCP server in a JetControl is limited to the following value:

Server	Connections
JetIP/TCP server	4
STX debug server	20

Contents

Automatic termination of connections

Introduction

If the maximum number of simultaneously established connections has been reached, any further connections cannot be established. If further connect requests are made, the user can set the response by the JetIP/TCP server and of the STX Debug server. There are the following possibilities:

- Reject new connection.
- Terminate one existing connection and establish the new one.
- Terminate all existing connections and establish the new one.

Default setting

By default, the server terminates the connection with the longest time of inactivity.

No automatic termination of connections

If the server is not to terminate any of the existing connections, proceed as follows:

Step	Action
1	Enter value 0 into MR 1.

Terminating the connection with the longest time of inactivity

If the server is to terminate the connection that has been inactive the longest time, proceed as follows:

Step	Action
1	Enter value -1 into MR 2.
2	Enter value 1 into MR 1.

Terminating the connection when the set minimum time has expired

If the server is to terminate a connection after a set minimum time of inactivity, proceed as follows:

Step	Action
1	Enter the minimum time [ms] into MR 2.
2	Enter value 1 into MR 1.

If the set minimum value has not been exceeded yet, the server rejects the new connection.

Terminating any connection

If the server is to terminate any of the existing connections, proceed as follows:

Step	Action
1	Enter value 2 into MR -1.
2	Enter value 1 into MR 2.

Terminating all connections which exceed the minimum time of inactivity

If the server is to terminate all existing connections which have exceeded the minimum time of inactivity proceed as follows:

Step	Action
1	Enter the minimum time [ms] into MR 2.
2	Enter value 1 into MR 2.

Register

Register numbers

The register numbers to be used are calculated by adding and the controller-dependent basic register number and the module register number.

Server	Basic register number	Register numbers
JetIP/TCP	230000	230000 ... 230002
STX-Debug	212000	212000 ... 212002

MR 0

Number of connections

The number of currently established connections can be read from module register 0.

Module register properties	
Values	0 ... 4 (JetIP/TCP server) 0 ... 20 (STX debug server)

MR 1

Mode

If the maximum number of connections is active, and if the server is to establish a new connection, module registers 1 and 2 determine the behavior.

Module register properties	
Values	0 ... 2
Value after reset	1

MR 2

Minimum inactivity time

If the maximum number of connections is active, and if the server is to establish a new connection, module registers 1 and 2 determine the behavior.

Module register properties	
Values	-1 ... 2,147,483,647 [ms]
Value after reset	-1

1.8 Executing an ARP request

Use case

Several controllers are interconnected via the Jetter Ethernet system bus. This is the case now. Controller B is exchanged. In this case, the IP address remains the same, but the Ethernet address (MAC address) changes. This way, data interchange between controller A and the new controller B is not possible.

To enable data interchange between the two controllers again, controller A would have to be relaunched.

To prevent a relaunch of controller A, an ARP request must be executed on controller A.

Phases of an ARP request

The controller A inquires from the Jetter Ethernet system bus, which node has got which specific IP address. Controller B reports that it has got this IP address. MAC address and IP address of controller B are aligned with each other. Now, controller A is informed of the MAC address which controller B has got. From now on, data interchange is possible again.

Contents

Executing an ARP request

ARP request

When you enter the IP address of a network node into the corresponding register, the controller triggers an ARP request. This request is used for resolution of an IP address into an Ethernet address (MAC address).

R 104250

Executing an ARP request

Register properties	
Values	Valid IP address

1.9 JetSync blockage

Introduction

In this chapter, the system command registers and the system commands for activating and deactivating the JetSync blockate will be explained in detail.

Contents

Description of system command registers (only JetSync blockage)

Registers - Overview

The following registers are used in this manual:

Registers	Description
R 202960	System password register
R 202961	System command register
R 202962	System status register

R 202960

System password register

Enter system password 1112502132 (0x424F6F74) into this register. Then enter the required command value into the system command register. Now, the controller sets the value of this register to 0.

Register properties	
Value	1112502132 (0x424F6F74)

R 202961

System command register

Enter the system commands into this register. Then the controller executes the command. Then, it sets the value of this register to 0.

Commands	
410	**Disable JetSync blockage**
411	**Enable JetSync blockage for all ports**
412	**Enable JetSync blockage for port X15**

Register properties	
Access	System password register contains the correct password.

R 202962

System status register

The system status register lets you evaluate the system conditions.

Meaning of the individual bits		
Bit 8	**JetSync blockage**	
	0 =	JetSync blockage is not active
	1 =	JetSync blockage is active

Register properties	
Access	Read

Description of the JetSync blockage system commands

System command 410

Disable JetSync blockage

Effect:

- The JetSync blockage is disabled for all ports. Bit 8 in R 202962 is reset.
- The Jetter Ethernet system bus multicast frames are transmitted to all ports (X14, X15 and CPU).

Purpose:

The JetSync blockage enabled by system command 411 or 412 is disabled. Forwarding the Jetter Ethernet system bus multicast frames to all ports again corresponds to the on-state of the controller.

System command 411

Enable JetSync blockage for all ports

Effect:

- The JetSync blockage is enabled for all ports (X14, X15, and CPU). Bit 8 in R 202962 is set.
- Jetter Ethernet system bus multicast frames which are received on a certain port are not forwarded to any of the other ports.
- All other Ethernet frames are forwarded as usual.

Purpose:

This command lets you prevent forwarding Jetter Ethernet system bus multicast frames to the CPU and the other ports. This way, networks are split and thus data traffic - e.g. from the machine network to higher-level networks - is reduced.

Address space

Splitting is carried out on Ethernet level via the multicast address range of the Jetter Ethernet system bus.
0x01 00 5E 40 00 00 ... 0x01 00 5E 40 00 FF

System command 412

Enable JetSync blockage for port X15

Effect:

- The JetSync blockage is enabled for port X15 only. Bit 8 in R 202962 is set.
- Jetter Ethernet system bus multicast frames of the CPU are forwarded to port X14 only.
- Jetter Ethernet system bus multicast frames of port X14 are forwarded to the CPU only.
- Jetter Ethernet system bus multicast frames of port X15 are forwarded to the CPU and to port X14.
- All other Ethernet frames are forwarded as usual.

Purpose:

This command lets you prevent forwarding Jetter Ethernet system bus multicast frames to port X15. This way, networks are split and thus data traffic - e.g. from the machine network to higher-level networks - is reduced.

Address space

Splitting is carried out on Ethernet level via the multicast address range of the Jetter Ethernet system bus.
0x01 00 5E 40 00 00 ... 0x01 00 5E 40 00 FF

Jetter
automation

Jetter AG
Graeterstrasse 2
71642 Ludwigsburg | Germany

Phone +49 7141 2550-0
Fax +49 7141 2550-425
info@jetter.de
www.jetter.de

We automate your success.