

MANUAL

DNP31.0 DeviceNet communication module

Universal Motor Controller UMC100.3



Important notice

Description

The DNP31.0 device allows the connection of the Universal Motor Controller UMC100.3 to a DeviceNet network.

Target group

This manual is intended for the use of specialists in electrical installation and control and automation engineering, who are familiar with the applicable national standards.

Safety requirements

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Using this manual

Symbols

This manual contains symbols to point the reader to important information, potential risks and precautionary information. The following symbols are used:



Sign to indicate a potentially dangerous situation that can cause damage to the connected devices or the environment.



Sign to indicate important information and conditions.



Sign to indicate a potentially dangerous situation that can cause human injuries.

Terms and abbreviations

SMK 3.0	Single Mounting Kit for the fieldbus interfaces
UMC100.3	Universal Motor Controller

Related documents

Technical documentation	Document no.
UMC100.3 Manual	2CDC135032D0204

Table of contents

4	Overview
5	Installation
11	Diagnosis
12	DeviceNet information about the properties of the DNP31.0
23	Technical data
24	Dimensional drawings
25	Ordering data

Overview

This chapter contains a short description of the DeviceNet standard and the DNP31.0 DeviceNet Interface module.

DeviceNet standard

The DeviceNet network has a linear bus topology. Terminating resistors are required on each end of the trunk line. Drop lines as long as 6 meters (20 feet) each are permitted, allowing one or more nodes to be attached. DeviceNet allows branching structures only on drop lines.

The DeviceNet communication interface

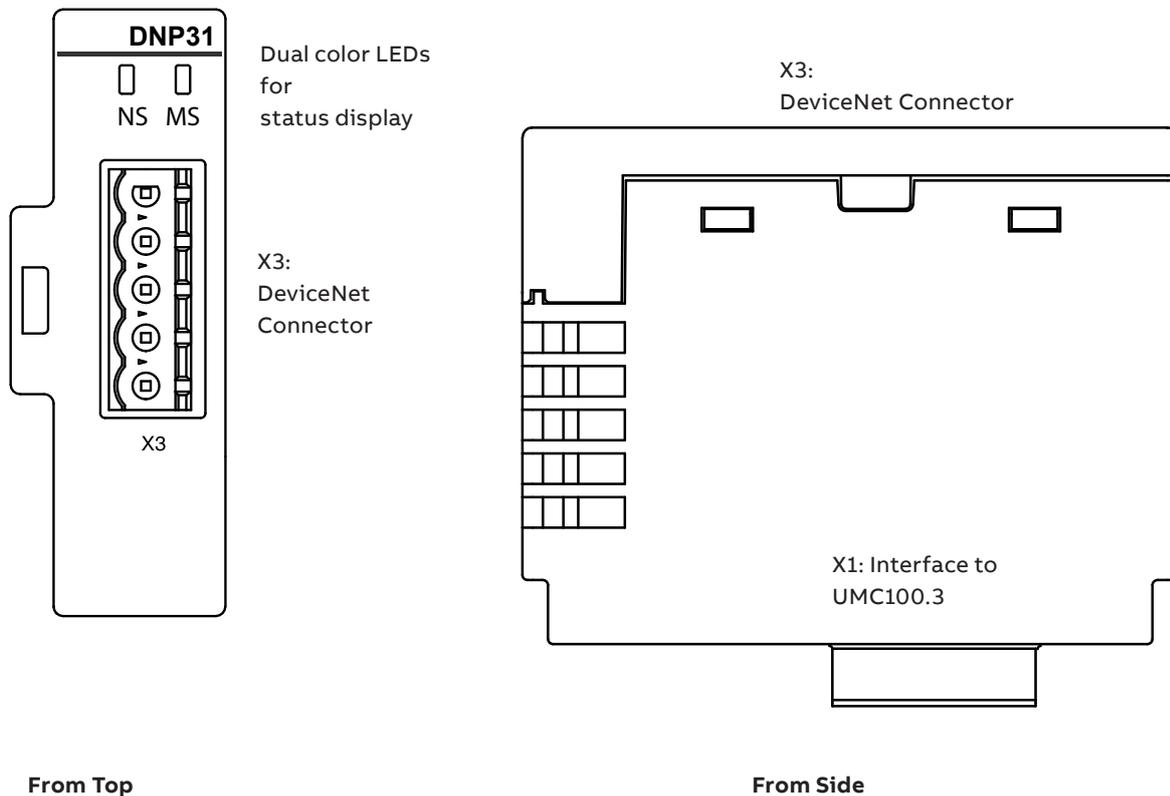
The DNP31.0 communication interface is an optional device for the UMC100.3 which enables the connection of the UMC100.3 to a DeviceNet system. The UMC100.3 is considered as a slave in the DeviceNet network.

- Use UMC100.3 to set bus address and baud rate (or autobaud) of the DeviceNet communication. Once the address is set, it is stored in the DNP31.0, even in case of supply voltage breakdown.

Through the DNP31.0 DeviceNet communication interface it is possible to:

- Give control commands to the drive (Start, Stop etc.)
- Read status information and actual values from the drive
- Read and write UMC100.3 parameter values
- Reset a UMC100.3 trip.

The DNP31.0 acts as a class 2 only slave with predefined master-slave connection set services. These include the explicit messaging, the poll-response service and the change of state/cyclic service. The DeviceNet commands and services supported by the DNP31.0 DeviceNet communication interface are discussed in chapter "communication".



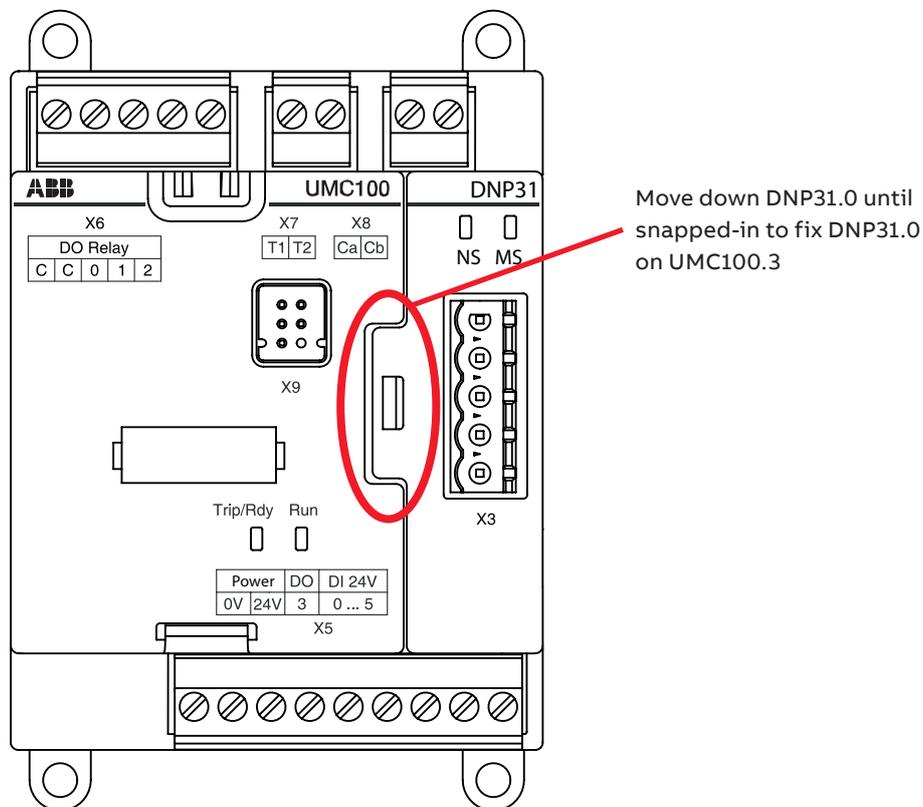
Installation

Mechanical installation

The DeviceNet adapter module DNP31.0 can be mounted either on the UMC100.3 itself or separately from the UMC100.3 using the single mounting kit (SMK3.0).

Mounting the DNP31.0 on the UMC100.3

When the module is installed directly on the UMC100.3 the UMC100.3 with DNP31.0 acts like a device with integrated DeviceNet communication. No additional accessories are needed.



01 Image shows UMC100.3 DC.

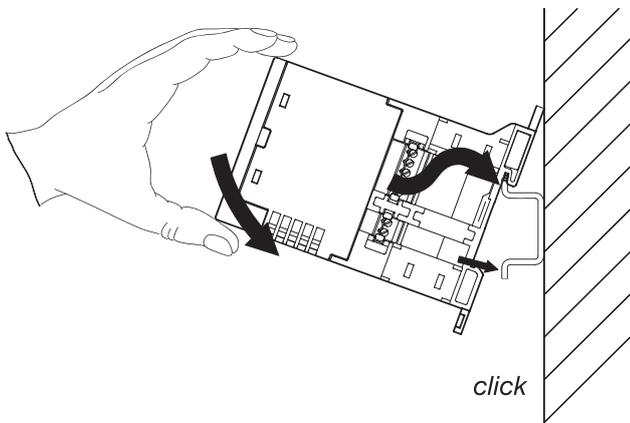
Mounting the DNP31.0 remotely from the UMC100.3

When the communication interface is installed remotely from the UMC100.3 - e.g. in the cable chamber of a motor control center - the DNP31.0 must be supplied separately. This makes it possible to keep the DNP31.0 online even in the case when the drawer is removed. Ready-made cables are available, but it is also possible to use own cables. For more details see section "Using DNP31.0 in a drawout system".

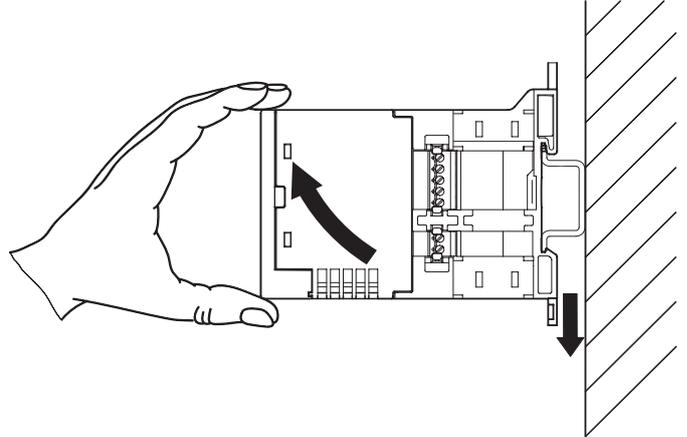
Mounting and dismounting

The single mounting kit (SMK3.0) can be either mounted on a DIN rail hat or fixed with screws on a fitting panel.

Mounting



Dismounting



Mounting the DNP31.0 on the single mounting kit

Step 1:

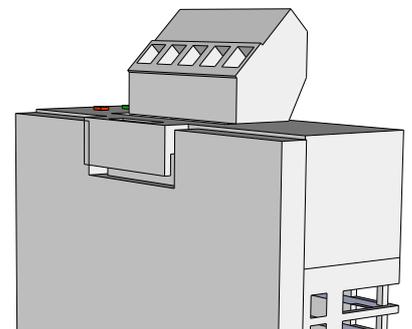
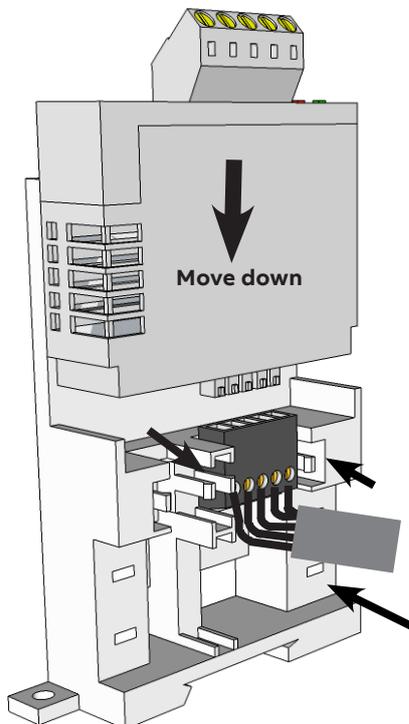
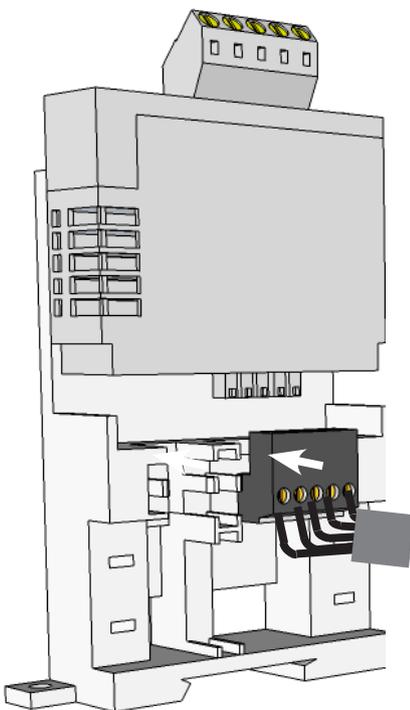
Plug in the communication connectors.

Step 2:

Move down the DNP31.0

Step 3:

Move down the DNP31.0 until snapped in



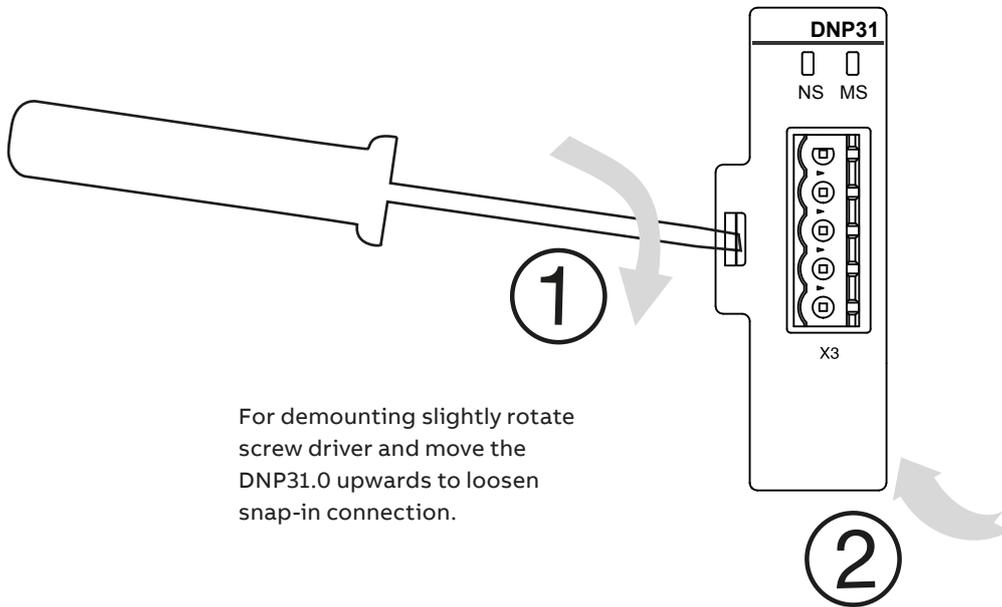
Step 4:

Fix cable with cable ties



Demounting the DNP31.0

Follow the shown procedure for demounting the DNP31.0 from UMC100.3 or SMK.3.



For demounting slightly rotate screw driver and move the DNP31.0 upwards to loosen snap-in connection.

Electrical installation

DeviceNet connection

Connect DeviceNet to X3.

General cabling instructions

- Arrange the bus cables as far away from the motor cables as possible.
- Avoid parallel runs
- Use bushings at cable entries

Bus termination

Bus termination is required to prevent signal reflections from the bus cable ends. The DNP31.0 DeviceNet interface module is not equipped with internal bus termination



The DeviceNet network should be terminated at both ends of the trunk cable with a 121 ohm, ¼ W, 1% Metal film resistor. Connect this resistor between the two signal wires (CAN_H, CAN_L) on the DeviceNet cable.

Setting the communication parameters

The DeviceNet interface module does not provide means to adjust the DeviceNet slave address and baud rate (incl. autobaud). The DeviceNet slave address and baudrate can be set in the UMC100.3 and are copied to the DeviceNet interface module during initial connection between UMC and DNP31.0. When changing communication parameters later on perform a power cycle. The baudrate is set to 'autobaud' by default.

Network and baudrates

The maximum length of trunk cable depends on the data rate and on the type of the cable used.

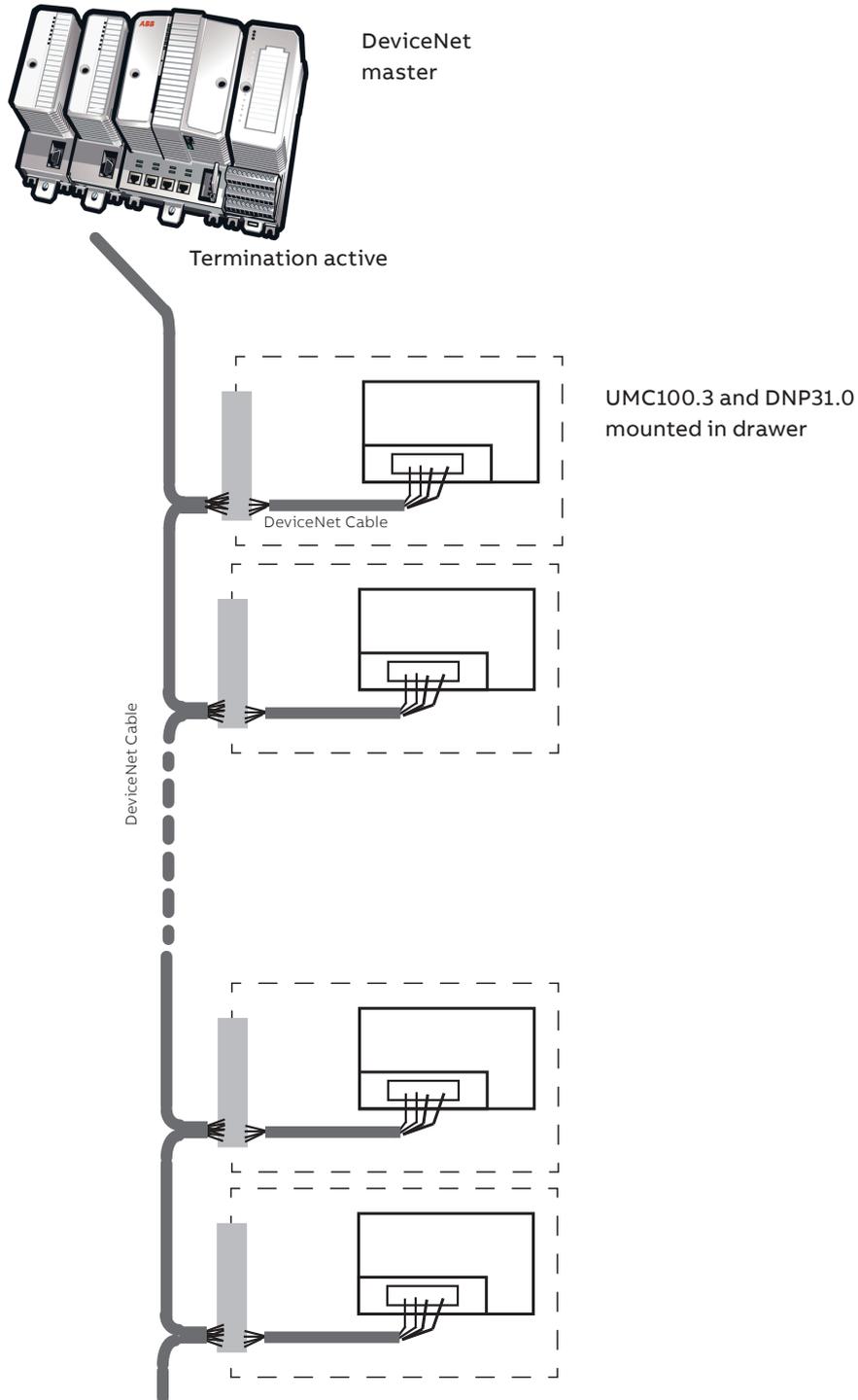
- DeviceNet Cables: YR-29790 (Thick DeviceNet Cable) or YR-29832 (Thin DeviceNet Cable)
- Maximum bus length: 1200 m
- Topology: Multi-drop
- Serial communication type: Asynchronous, half duplex
- Transfer rate: 125, 250 or 500 kbit/s

DeviceNet line with DNP31.0 directly connected to UMC100.3 in a drawout system

The following figure shows a simplified diagram on how to connect the UMC100.3 to a DeviceNet network. Required grounding of the DeviceNet cable is not shown in this figure.



Ground wire and cable shield should be connected to prevent common mode voltage between the network devices from drifting outside the allowable limits.



DeviceNet line with DNP31.0 mounted outside a drawer using single mounting kit (SMK3.0)

The following figure shows a simplified diagram on how to connect the UMC100.3 to a DeviceNet network if the UMC100.3 is inside a drawer and the DeviceNet communication interface is mounted outside the drawer e.g. in the cable compartment.

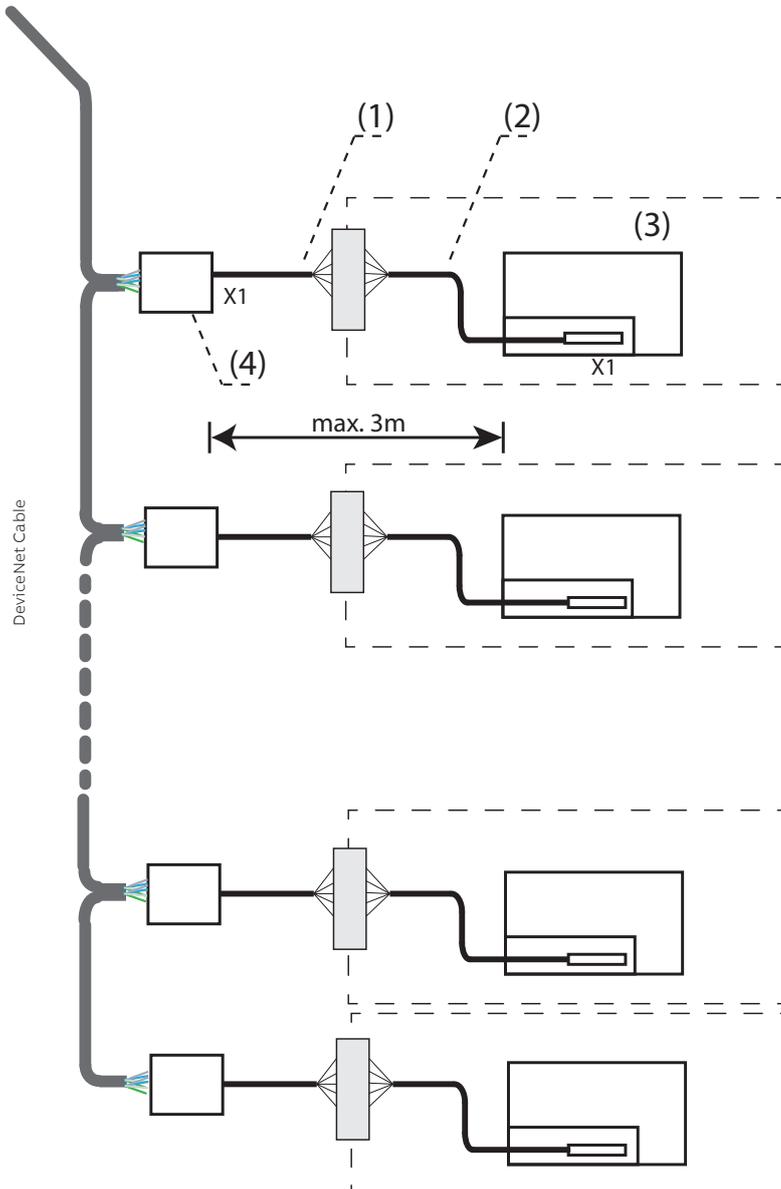
In drawout systems this solution has several benefits:

- DeviceNet address is stored in the DNP31 communication interface in addition to the UMC100. In case of a drawer replacement the new - not addressed - UMC100.3 takes over the bus address automatically.
- Swapping of drawers detected if "Address Check" in UMC100.3 enabled. I.e. it is not possible to accidentally start the wrong motor because of swapped drawer.

Required grounding of the DeviceNet cable is not shown in this figure.



DeviceNet master



(1) CDP24: Ready made connection cable from SMK3.0 X1 to backside of drawer.

(2) CDP18: ready made connection cable from inside drawer to UMC100.3 X1

(3) UMC100.3 mounted inside a drawer

(4) DeviceNet communication adapter with single mounting kit (SMK3.0).

Diagnosis

Diagnosis and behavior in case of an error

The DNP31.0 module provides diagnosis information about the status of the connected device (i.e. UMC100.3), its own status and the status of the DeviceNet connection. Diagnosis information is shown:

- with the locally available LEDs and
- via DeviceNet services.

Diagnosis information is locally displayed using LEDs located on the front side of the DeviceNet interface.

NS LED (red, green)	MS LED (red, green)	Meaning
Off	Off	Power supply missing
Green flashing		Online, but not yet listed in scan list
Red flashing		I/O connection in timeout state
Red		Critical connection error, no communication, e.g. address used twice, bus switched off
Green		Online and listed in scan list
	Green flashing	Waiting for communication to device (UMC100.3)
	Green	Module is working correctly
	Red flashing	Communication timeout on link to device (UMC100.3)
	Red	Unrecoverable module error, module replacement possibly required

Checklist

In case of trouble use the following checklist to track down the problem:

DeviceNet

- Are the termination resistors placed at the end of the line?
- Is the line polarity correct?
- Are the lines by accident swapped?
- Is the maximum line length exceeded?

DeviceNet parameters

- Is the baud rate correct?
- Is the slave address correct?
- Are there two devices with the same address in the system?
- Power supply turned on?

DeviceNet master

- Is the correct EDS file used

DeviceNet information about the properties of the DNP31.0

General information

The DNP31.0 operates as a slave on the DeviceNet network. The unit supports explicit messages and polled connection as well as COS/Cyclic I/O messages of the predefined master/ slave connection set. It does not support the Unconnected Message Manager (UCMM).

The number of inputs and outputs supported by the DNP31.0 depends on the device which is connected (as of writing UMC100.3 only).

Message types

As a group 2 only slave device, the DNP31.0 supports the following message types.

CAN Identifier	Group 2 message type
10xxxxxx111	Duplicate MAC ID check messages
10xxxxxx110	Unconnected explicit request message
10xxxxxx101	Master I/O poll command message
10xxxxxx100	Master explicit request message

xxxxxx = Node address

Class services

As a group 2 only slave device, the DNP31.0 supports the following class services and instance services.

Service code	Service name
14 (0x0E)	Get attribute single
16 (0x10)	Set attribute single
75 (0x4B)	Allocate group 2 Identifier set
76 (0x4C)	Release group 2 Identifier set

Object classes

The DNP31.0 supports the following DeviceNet classes.

Class	Object
001 (0x01)	Identity
002 (0x02)	Message router
003 (0x03)	DeviceNet
004 (0x04)	Assembly
005 (0x05)	Connection
043 (0x2B)	Acknowledge handler
100 (0x64)	ABB discrete input
101 (0x65)	ABB discrete output
102 (0x66)	ABB analog input
103 (0x67)	ABB analog output
105 (0x69)	ABB parameter
112 (0x70)	ABB parameter modular devices
118 (0x76)	ABB any data
128 (0x80)	ABB query

Class code 001 (0x01): Identity object

The identity object is required on all devices and provides identification of and general information about the device. Only one instance (0x01) is supported.

—

Class attributes

None

—

Instance attributes

Attribute	Access	Name	Type	Value
1	Get	Vendor	UINT	760 (0x2F8)
2	Get	Product type	UINT	12 (0xC)
3	Get	Product code	UINT	*1)
4	Get	Revision Major revision Minor revision	STRUCT OF USINT USINT	1 3
5	Get	Device status	UINT	*2)
6	Get	Serial number	UINT	unique
7	Get	Product name Length Name	STRUCT OF USINT STRING[Length]	7 General

*1) The product code is obtained from the device connected to the DNP31.

*2) Device status

byte 0	0x00	not owned
	0x01	owned (allocated)
byte 1	0x00	no fault
	0x04	recoverable fault
	0x08	unrecoverable fault

—

Common services

Service code	Class	Instance	Service name
05 (0x05)	No	Yes	Reset
14 (0x0E)	No	Yes	Get_Attribute_Single

Class code 002 (0x02): Message router object

The message router object provides a messaging connection point through which a client may address a service to any object class or instance residing in the physical device.

—

Class attributes

None

—

Instance attributes

None

—

Common services

None

Class code 004 (0x04): Assembly object

The assembly objects bind attributes of multiple objects to allow data to or from each object to be sent or received over a single connection.

—

Class attributes

None

—

Instance attributes

Assembly object, instance 100, attributes

Attribute	Access	Name	Type	Value
3	Get/Set	Producing data	ARRAY OF [n] BYTE	*1)

*1) The size of the data array depends on the Produced Data size of the device connected to the DNP31.

Assembly object, instance 101, attributes

Attribute	Access	Name	Type	Value
3	Get/Set	Consuming data	ARRAY OF [n] BYTE	*1)

*1) The size of the data array depends on the Consumed Data size of the device connected to the DNP31. For detailed information, please refer to the device manual.

—

Common services

Service code	Class	Instance	Service name
14 (0x0E)	Yes	Yes	Get_Attribute_Single
16 (0x10)	No	Yes	Set_Attribute_Single

Class code 005 (0x05): Connection object

The connection class allocates and manages the internal resources associated with both I/O and explicit messaging connections. The specific instance generated by the connection class is referred to as a connection instance or a connection object.

Three instances of the connection class will be supported. Instance 1 will be the explicit message connection, instance 2 will be the polled I/O connection, and instance 4 will be the COS/Cyclic IO connection.

Class attributes

None

Instance attributes

Connection object, instance 1 = Explicit message connection, attributes

Attribute	Access	Name	Type	Value
1	Get	State	USINT	*1)
2	Get	Instance type	USINT	0 = Explicit Message
3	Get	Transport class trigger	BYTE	0x83 - Server, Transport Class 3
4	Get	Produced connection ID	UINT	10xxxxxx011 *2)
5	Get	Consumed connection ID	UINT	10xxxxxx100 *2)
6	Get	Initial comm. characteristics	BYTE	0x21
7	Get	Produced connection size	UINT	*3)
8	Get	Consumed connection size	UINT	*3)
9	Get/Set	Expected packet rate	UINT	in ms
12 (0x0C)	Get/Set	Watchdog timeout action	USINT	01 = auto delete
13 (0x0D)	Get	Produced conn. path length	UINT	0
14 (0x0E)	Get	Produced connection path	EPATH	Empty
15 (0x0F)	Get	Consumed conn. path length	UINT	0
16 (0x10)	Get	Consumed connection path	EPATH	Empty
17 (0x11)	Get	Production inhibit time	UINT	0

*1) Connection state

0	nonexistent
1	configuring
2	wait for connection ID
3	established
4	timed out
5	deferred

*2) xxxxxx = node address

*3) The size of Produced and Consumed Connection depends on the device connected to the DNP31.0 (as of writing UMC100.3). For detailed information, please refer to the device manual.

Connection object, Instance 2 = Polled I/O message connection, attributes

Attribute	Access	Name	Type	Value
1	Get	State	USINT	*1)
2	Get	Instance type	USINT	1 = I/O Connection
3	Get	Transport class trigger	BYTE	0x82 - Server, Transport Class 2 *4)
4	Get	Produced connection ID	UINT	01111xxxxxx *2)
5	Get	Consumed connection ID	UINT	10xxxxxx101 *2)
6	Get	Initial comm. characteristics	BYTE	0x01
7	Get	Produced connection size	UINT	*3)
8	Get	Consumed connection size	UINT	*3)
9	Get/Set	Expected packet rate	UINT	in ms
12 (0x0C)	Get	Watchdog timeout action	USINT	0 = time out
13 (0x0D)	Get	Produced conn. path length	UINT	6
14 (0x0E)	Get	Produced connection path	EPATH	20 04 24 64 30 03
15 (0x0F)	Get	Consumed conn. path Length	UINT	6
16 (0x10)	Get	Consumed connection path	EPATH	20 04 24 65 30 03
17 (0x11)	Get	Production inhibit time	UINT	0

*1) Connection state

0	nonexistent
1	configuring
2	wait for connection ID
3	established
4	timed out
5	deferred

*2) xxxxxx = node address

*3) The size of Produced and Consumed Connection depends on the device connected to the DNP31.0 (as of writing UMC100.3). For detailed information, please refer to the device manual.

*4) If alloc_choice = polled and ack suppression is enabled then value = 0x80.

Connection object, Instance 4 = COS/Cyclic I/O message connection, attributes

Attribute	Access	Name	Type	Value
1	Get	State	USINT	*1)
2	Get	Instance type	USINT	1 = I/O Connection
3	Get	Transport class trigger	BYTE	*4)
4	Get	Produced connection ID	UINT	01101xxxxxx *2)
5	Get	Consumed connection ID	UINT	10xxxxxx101 *2)
6	Get	Initial comm. characteristics	BYTE	0x01 (acknowledged) 0x0F (unacknowledged)
7	Get	Produced connection size	UINT	*3)
8	Get	Consumed connection size	UINT	*3)
9	Get/Set	Expected packet rate	UINT	in ms
12 (0x0C)	Get	Watchdog timeout action	USINT	0 = time out
13 (0x0D)	Get	Produced conn. path Length	UINT	6
14 (0x0E)	Get	Produced connection path	EPATH	20 04 24 64 30 03
15 (0x0F)	Get	Consumed conn. path Length	UINT	6
16 (0x10)	Get	Consumed connection path	EPATH	20 04 24 65 30 03
17 (0x11)	Get/Set	Production inhibit time	UINT	in ms

*1) Connection state

0	nonexistent
1	configuring
2	wait for connection ID
3	established
4	timed out
5	deferred

*2) xxxxxx = node address

*3) The size of Produced and Consumed Connection depends on the device connected to the DNP31.0 (as of writing UMC100.3). For detailed information, please refer to the device manual.

*4) Transport Class Trigger

0x00	Cyclic, unacknowledged
0x02	Cyclic, acknowledged
0x10	COS, unacknowledged
0x12	COS, acknowledged

Common services

Service code	Class	Instance	Service name
05 (0x05)	No	Yes	Reset
14 (0x0E)	No	Yes	Get_Attribute_Single
16 (0x10)	No	Yes	Set_Attribute_Single

Class code 043 (0x2B): Acknowledge handler object

The acknowledge handler object is used to manage the reception of message acknowledgements. This object communicates with a message producing application object within a device. The acknowledge handler object notifies the producing application of acknowledge reception, acknowledge timeouts and production retry limit.

Class attributes

None

Instance attributes

Attribute	Access	Name	Type	Value
1	Get/Set	Acknowledge timeout	UINT	in ms
2	Get	Retry limit	USINT	1
3	Get	COS producing connection instance	UINT	4

Common services

Service code	Class	Instance	Service name
14 (0x0E)	No	Yes	Get_Attribute_Single
16 (0x10)	No	Yes	Set_Attribute_Single

Class Code 100 (0x64): ABB discrete input object

The ABB discrete input object models the discrete inputs of the device connected to the DNP31. You can use this object in applications as simple as a toggle switch or as complex as a discrete I/O control module. There is a separate instance for each discrete input available on the device.

Class attributes

Attribute	Access	Name	Type	Value
2	Get	Number of Discrete Inputs	USINT	*1)

*1) The number of discrete inputs depends on the device connected to the DNP31.0 (as of writing UMC100.3). For detailed information, please refer to the device manual.

Instance attributes

Attribute	Access	Name	Type	Value
3	Get	Discrete Input Value	BOOL	Input State

Common services

Service code	Class	Instance	Service name
14 (0x0E)	No	Yes	Get_Attribute_Single

Class code 101 (0x65): ABB discrete output object

The ABB discrete output object models the discrete outputs of the device connected to the DNP31. You can use this object in applications as simple as an actuator or as complex as a discrete I/O control module. There is a separate instance for each discrete output available on the device.

Class attributes

Attribute	Access	Name	Type	Value
2	Get	Number of discrete outputs	USINT	*1)

*1) The number of discrete outputs depends on the device connected to the DNP31. For detailed information, please refer to the device manual.

Instance attributes

Attribute	Access	Name	Type	Value
3	Get/Set	Discrete output value	BOOL	Output state

Common services

Service code	Class	Instance	Service name
14 (0x0E)	No	Yes	Get_Attribute_Single
16 (0x10)	No	Yes	Set_Attribute_Single

Class Code 102 (0x66): ABB analog input object

The ABB analog input object models the analog inputs of the device connected to the DNP31.0 (as of writing UMC100.3). There is a separate instance for each analog input available on the device.

Class attributes

Attribute	Access	Name	Type	Value
2	Get	Number of analog inputs	USINT	*1)

*1) The number of analog inputs depends on the device connected to the DNP31.0 (as of writing UMC100.3). For detailed information, please refer to the device manual.

Instance attributes

Attribute	Access	Name	Type	Value
3	Get	Analog input value	INT	Input state

Common services

Service code	Class	Instance	Service name
14 (0x0E)	No	Yes	Get_Attribute_Single

Class Code 103 (0x67): ABB analog output object

The ABB analog output object models the analog outputs of the device connected to the DNP31.0. There is a separate instance for each analog output available on the device.

Class attributes

Attribute	Access	Name	Type	Value
2	Get	Number of analog outputs	USINT	*1)

*1) The number of analog outputs depends on the device connected to the DNP31.0. For detailed information, please refer to the device manual.

Instance attributes

Attribute	Access	Name	Type	Value
3	Get/Set	Analog output value	INT	Output state

Common services

Service code	Class	Instance	Service name
14 (0x0E)	No	Yes	Get_Attribute_Single
16 (0x10)	No	Yes	Set_Attribute_Single

Class code 105 (0x69): ABB parameter object

The ABB parameter object provides the parameter for non-modular devices and for the adapter module (slot 0) of modular devices. The index always corresponds to the parameter number. The attribute is fixed to 3. The use of the ABB parameter object provides a known public interface to a device's configuration data. In addition, this object also provides all the information necessary to define and describe each of a device's individual configuration parameters.

This object allows a device to fully identify a configurable parameter by supplying a full description of the parameter, including minimum and maximum values and a human-readable text string describing the parameter. The complete description of an device's parameters (as of writing UMC100.3) can be found in the device manual. Configuration tools obtain the parameter description automatically from the EDS file.

Class attributes

Attribute	Access	Name	Type	Value
2	Get	Number of parameters (Max. instance)	USINT	*1)

*1) The number of parameters depends on the device connected to the DNP31.0 (as of writing UMC100.3). For detailed information, please refer to the device manual.

Instance attributes

Attribute	Access	Name	Type	Value
3	Get/Set *2)	Parameter value	LINT	Actual value

*2) The Set service is not available for read-only parameters.

Common services

Service code	Class	Instance	Service name
14 (0x0E)	No	Yes	Get_Attribute_Single
16 (0x10)	No	Yes	Set_Attribute_Single

Class code 112 (0x70): ABB parameter modular object

The ABB parameter modular object provides the parameter for **modular** expansion devices which are connected to an adapter module. The index always corresponds to the slot number (beginning with slot 1) and the attribute corresponds to the parameter number of the module plugged into the slot. The use of the ABB parameter modular object provides a known public interface to a device's configuration data. In addition, this object also provides all the information necessary to define and describe each of a device's individual configuration parameters.

This object allows a device to fully identify a configurable parameter by supplying a full description of the parameter, including minimum and maximum values and a human-readable text string describing the parameter. The complete description of a modular device's parameters can be found in the device manual. Configuration tools obtain the parameter description automatically from the EDS file.

Class attributes

Attribute	Access	Name	Type	Value
2	Get	Max. number of expansion modules	USINT	7

Instance attributes

Attribute	Access	Name	Type	Value
3	Get/Set *1)	Parameter value	LINT	Actual value

*1) The Set service is not available for read-only parameters.

Common services

Service code	Class	Instance	Service name
14 (0x0E)	No	Yes	Get_Attribute_Single
16 (0x10)	No	Yes	Set_Attribute_Single

Class code 118 (0x76): ABB any data class

The ABB any object can be used to read parameter data and measured values in a compact format. When required details are available on request.

Class code 128 (0x80): ABB query object

The ABB query object provides an array filled with the external IDs of all modules connected to the adapter module. It is used for modular devices only. By means of this object, configuration tools are able to read the module configuration from the DNP31.0.

Class attributes

None

Instance attributes

Attribute	Access	Name	Type	Value
1	Get	Query external ID list	ARRAY [0..7] OF USINT	External IDs of the expansion modules actually connected *1)

*1) Empty slots are represented by the external ID "00 00".

Common services

Service code	Class	Instance	Service name
14 (0x0E)	No	Yes	Get_Attribute_Single

Technical data

Data at $T_a = 25\text{ °C}$ and rated values, unless otherwise indicated.

General data

Supply voltage	24 V DC (11...24,7 V DC) according to DeviceNet specification
Current consumption	type. 18.5 mA
Supported communication protocols	DeviceNet
Physical Interface	2-wire CAN
Integrated termination resistors	no
Possible bus addresses (set via UMC100)	0 ... 63
Max. baudrate	500 kbaud
Pollution degree	3
Degree of protection	IP20
Mounting	UMC100.3 or SMK3.0
Altitude	4000 m
Duty time	100 %
Weight	0.042 kg
Protective separation	no

Standards / directives

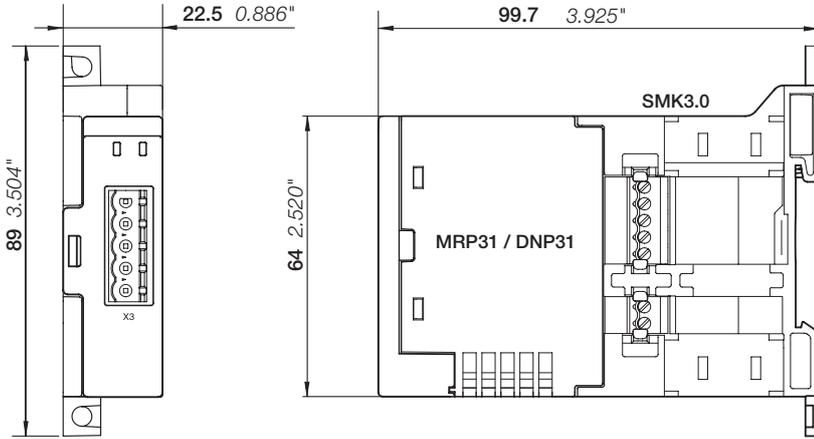
Product standard	DeviceNet specification
RoHS directive	2011/65/EU

Environmental data

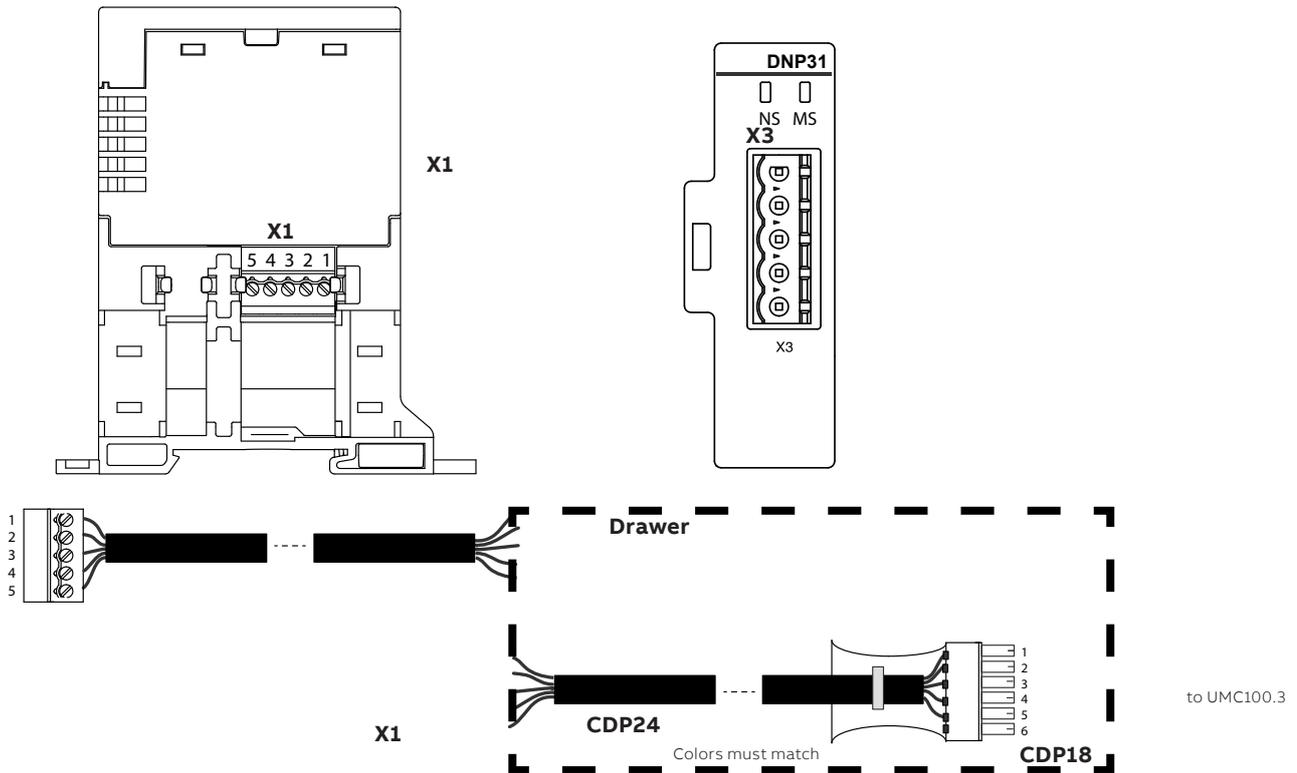
Ambient air temperature	Operation	0 ... +60 °C
	Storage	-25 ... +70 °C
Vibration (sinusoidal) acc. to IEC/EN 60068-2-6 (Fc)		0.7 g / 10 150 Hz
Shock (half-sine) acc. to IEC/EN 60068-2-27 (Ea)		15 g / 11 ms

Dimensional drawings

Main device



Connection cables



Pin	Color	Function
5	Blue	VDD
4	Brown	VCC
3	Black	Communication
2	Grey	Communication
1	White	Diagnosis

Function
V-
CAN_L
Screen
CAN_H
V+

Ordering data

Order code	Type	Description
1SAJ231000R0001	DNP31.0	DeviceNet communication interface
1SAJ929600R0001	SMK3.0 *	Single mounting kit for separate mounting of the communication interface
1SAJ929180R0015	CDP18.150 **	Cable between UMC100.3 and drawer inside, 1.5 m
1SAJ929240R0015	CDP24.150 **	Cable between SMK3.0 and drawer outside, 1.5 m
1SAJ929610R0001	SMK3-X2.10	Terminal block 2-pol. for SMK3.0 supply, 10 pcs.
1SAJ929620R0001	SMK3-X1.10	Terminal block 5-pol. for SMK3.0 communication, 10 pcs.

* is delivered including terminal block for power supply connection

** ready cables with terminal block on one end and one open end

ABB STOTZ-KONTAKT GmbH
Eppelheimer Straße 82
69123 Heidelberg
Germany

**You can find the address of your local sales organization
on the ABB homepage**



abb.com/lowvoltage

Additional information

We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars will take precedence. ABB AG does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.

We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents – in whole or in parts – is forbidden without prior written consent of ABB AG.

© Copyright 2019 ABB. All rights reserved.
Specifications subject to change without notice.