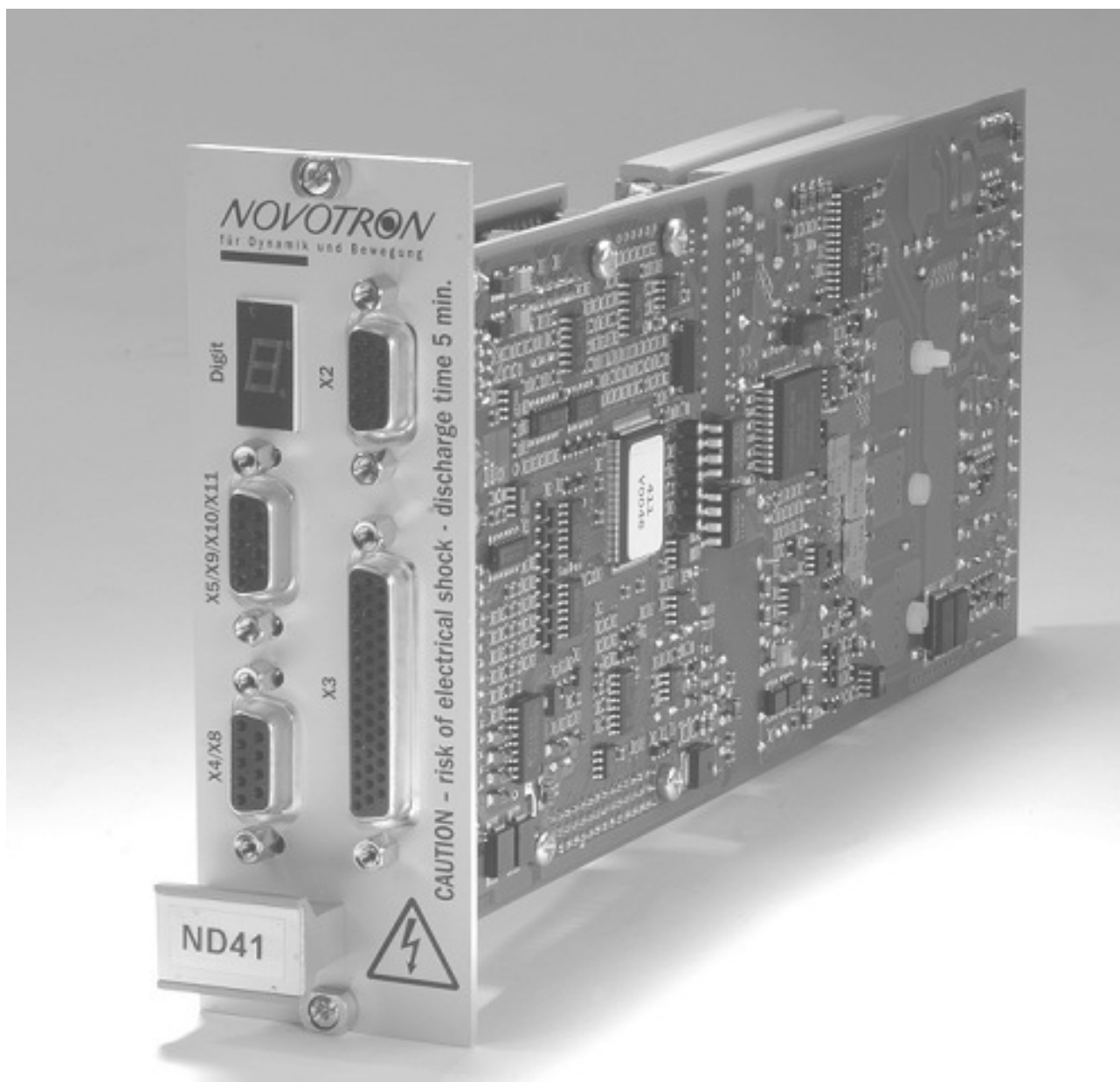


**Manual For Servo Amplifier  
NOVODRIVE ND40**

**Extension Moduls**



Version: 01.01  
Stand: 24/06/2009

## 1 Contents

2 General Information.....	3
2.1 Customer Service.....	3
2.2 List of Abbreviations.....	3
2.3 Symbols.....	3
2.4 Trademarks.....	3
2.5 User Manual Structure.....	3
3 Safety Information.....	4
3.1 Intended Usage.....	4
3.2 General Safety Instructions.....	4
3.3 Connection.....	5
3.4 Operation.....	6
4 Extension Module for Position Measuring Systems.....	7
4.1 EnDat 2.2 Interface.....	7
5 Bus Interface (Option).....	8
5.1 General Information.....	9
5.2 Activation/Deactivation of Control by Bus System.....	13
5.3 PROFIBUS Interface.....	14
5.4 CANopen Interface.....	19
5.5 Device Profiles.....	37
5.6 Description of Process Data Inputs.....	39
5.7 Description of Process Data Outputs.....	42
5.8 Bus Interface Related Error Codes of NOVODRIVE.....	46

## 2 General Information

### 2.1 Customer Service

Address:	NOVOTRON GmbH Mauserstrasse 31 71640 Ludwigsburg Germany
	phone: +49 - 7141 - 2969 - 0
	fax: +49 - 7141 - 2969 - 22
	e-mail: <a href="mailto:info@novotron-online.com">info@novotron-online.com</a> web: <a href="http://www.novotron-online.com">www.novotron-online.com</a>

### 2.2 List of Abbreviations

Abbreviation	Meaning	Abbreviation	Meaning
VAC	AC voltage	VDC	DC voltage
RO	Read Only	RW	Readable and Writeable
WO	Write Only	BCD	binary coded decimal

### 2.3 Symbols



Warning or important information.

Noncompliance may lead to trouble in operation or to property damage.



Hazard that may lead to

- damage of life or health of user or other persons, or
- major property damage.

### 2.4 Trademarks

EnDat is a registered trademark of DR. JOHANNES HEIDENHAIN GmbH.

Windows is a registered trademark of Microsoft Corporation.

COMBICON, MICRO COMBICON and POWER COMBICON are product names of Phoenix Contact.

### 2.5 User Manual Structure

- Volume 1 Basic Device
- Volume 2 Software Reference
- Volume 3 Extension Modules
- Volume 4 setup

### 3 Safety Information



#### **NOVODRIVE works with hazardous voltage!**

Line voltage is present at power inputs, motor connectors, brake choppers, DC link circuits, and the motor temperature sensor input. Connectors carrying hazardous voltage are X1, X6, and X7.

#### 3.1 Intended Usage

Devices of the ND40 series are state-of-the-art servo converters for driving brushless servo motors and linear motors featuring an appropriate position sensor.

NOVODRIVE servo converters are components to be installed in electric machines. A NOVODRIVE servo converter may only be put into operation as such an integrated component.



#### **Safe and troublefree operation**

Safe and troublefree operation is possible only by correct interplay of NOVODRIVE, motor, and position sensor in combination with correct wiring and appropriate parameterization.

Using NOVODRIVE servo converters for any other purpose may lead to property damage or damage to users and other persons.

NOVODRIVE servo converters may be installed and put into operation only if in technically perfect condition, if used in compliance with the intended usage described above, and if users are aware of risks and hazards that may occur when working with the devices.

#### **Machinery Directive**

According to the 2006/42/EG Machinery Directive, machinery manufacturers have to make an appropriate risk assessment for the specific machinery, and they have to take appropriate action in order to prevent damage to persons or property when working with the machinery.

Machinery manufacturers and/or operators are responsible for sticking to applicable guidelines for safe operation and prevention of accidents.

#### **Ambient Conditions**

NOVODRIVE servo converters may not be used in explosion-prone areas or in medical areas or other areas classified hazardous.

#### 3.2 General Safety Instructions

Before you install and put into operation a NOVODRIVE servo converter, read this User Manual fully and carefully. Inappropriate handling of NOVODRIVE servo converters may lead to damage to persons or property. Make sure you take notice of the technical specifications and connecting conditions (check type plate and this User Manual).

### Transportation and Storage

To transport and store NOVODRIVE servo converters, use the original packaging only.

### Repairs and Modifications

Do not dismantle and/or modify NOVODRIVE servo converters.

If a NOVODRIVE servo converter needs repair or modification, this may be done by skilled personnel of NOVOTRON GmbH only.

### Working at the motor or axis

Do not work on running gear until NOVODRIVE has been disconnected from the power supply system and NOVODRIVE's capacitors' are fully discharged.

Provide for sufficient protection, as a defect in NOVODRIVE's power part or in the feedback system may speed up the drive enormously within fractions of a second, leading to uncontrolled movements and very high acceleration of the motor.

## 3.3 Connection

Only skilled personnel may install, put into operation, and maintain NOVODRIVE servo converters.

Skilled personnel are persons familiar with the requirements to assemble, install, put into operation, and work with the product. Such persons need to take notice of and comply with the following norms and guidelines:

- IEC 364 / CENELEC HD 384 or DIN VDE 0100,
- IEC-Report 664 or DIN VDE 0110,
- national directives for accident prevention.

### Grounding

Make sure NOVODRIVE's compact case or 19" rack, as the case may be, is grounded before the line voltage is switched on (see 4.2 Technical Specifications and 5.2 Power Part Connections).

### Wiring

Always check the wiring before you power up NOVODRIVE. Check

- if all connections are correct and all terminal clamps are tight,
- if grounding / shielding is correct,
- if connectors are locked against getting loose.

Do not pull any (i.e. also low-voltage) energized plug connectors, as this may destroy the electronics.

Make sure no voltage carrying parts can be accidentally touched. Cables carrying line voltage must have double or reinforced insulation between wire and surface. Use appropriate sleeves for wire ends.

### Fuse Protection

Make sure NOVODRIVE is equipped with an appropriate and correctly connected fuse protection.

### Emergency Shutdown / Emergency Stop

Provide for an emergency shutdown / emergency stop by which the motor can be brought to a standstill at any time (see section 6 Emergency Shutdown / Emergency Stop).

### 3.4 Operation

#### Discharge Time

NOVODRIVE contains capacitors which keep on carrying hazardous operating voltage for some time after line voltage has been switched off.

Therefore, after disconnecting NOVODRIVE from line voltage wait at least five minutes before you touch any voltage carrying parts (e.g. pins) or loosen connections. As a safety measure, repeatedly measure the DC link voltage and wait until it is below 40 V.

#### Electric Shock Protection

A moving motor can produce hazardous voltage also if line voltage is switched off.

Therefore, the discharge time of the capacitors does not start until the motor has stopped.

#### Power On/Off

Do not switch on and off line voltage of NOVODRIVE frequently in short time, as this may lead to an overload of NOVODRIVE's inrush current limitation, which could destroy the inrush current limiter. Therefore, always wait at least one minute before switching NOVODRIVE on again after you switched it off.

#### Sequence for Switching On/Off

If you want to switch on NOVODRIVE, first switch on the power supply unit providing the 24 VDC supply voltage for the low-voltage part, and then switch on line voltage for the power part. If you want to switch off NOVODRIVE, proceed vice versa.

## 4 Extension Module for Position Measuring Systems

### 4.1 EnDat 2.2 Interface

An extension module allows to use Heidenhain absolute measuring systems with an EnDat 2.2 interface. Information on the position measuring system connected can be requested over variables in the basic device.

### 5 Register

Address	Name	Function
640	EpParam[0]	Measuring system: 0000 xxxx xxxx xxxx No position measuring system 0001 xxxx xxxx xxxx Linear measuring system 0010 xxxx xxxx xxxx Multi-turn encoder 0011 xxxx xxxx xxxx Single-turn encoder 1111 xxxx xxxx xxxx No accepted measuring system identified xxxx 0000 xxxx xxxx Unknown manufacturer/protocol xxxx 0001 xxxx xxxx Heidenhain / EnDat 2.2 xxxx xxxx 0000 xxxx Reserved Unit: xxxx xxxx xxxx 0001 resolution incr. / rev. xxxx xxxx xxxx 0010 resolution in [μm] xxxx xxxx xxxx 0011 resolution in [nm] xxxx xxxx xxxx 0100 resolution in [pm]
641	EpParam[1]	LSB 1...48 Number of bits of position information
642	EpParam[2]	MSB Resolution of measuring system
643	EpParam[3]	LSB (see Parameter 0)
644	EpParam[4]	Internal information
645	EpParam[5]	Internal information
646	EpParam[6]	Type designation of measuring system as ASCII code
647	EpParam[7]	
648	EpParam[8]	
649	EpParam[9]	
650	EpParam[10]	Manufacturer specific Options of measuring system
651	EpParam[11]	Manufacturer specific Cable runtime
652	EpParam[12]	Manufacturer specific Error bits supported
653	EpParam[13]	Manufacturer specific Error bits set following an error
654	EpParam[14]	Manufacturer specific Warn bits supported
655	EpParam[15]	Manufacturer specific Warn bits set following an error

All parameters are read-only parameters.

## 6 Bus Interface (Option)

NOVODRIVE can be equipped with a PROFIBUS-DP or CANopen interface. All interfaces work as slaves and support the exchange of process data and service data. The module's profile is optimized for control of NOVODRIVE by a PLC.

Using the process data interface, all operating modes of NOVODRIVE can be activated and executed, with actual values and setpoints being transmitted in SI units (settings for SI scaling are done over the setup software).

**This section provides information on the following:**

- components to be used for installation,
- activation/deactivation of bus systems in the NOVODRIVE,
- settings that must/can be done for configuration and setup of bus systems,
- NOVODRIVE's state diagram,
- using a PLC for controlling NOVODRIVE.

## 6.1 General Information

### Bus data processing time

The processing time for the data of the bus system depends on

- NOVODRIVE's processing time,
- the data transmission speed of the bus system,
- the bus system capacity and the priority of the data.

NOVODRIVE's processing time for process data with SI scaling is approx 5 ms.

NOVODRIVE's processing time for service data is approx. 20 ms.

### Connection

For information on connection and pin assignment see User Manual Volume 1 'Basic Device', Section 5.3.

### Supply voltage

The bus system needs no external supply voltage.

Vorläufig

### 6.1.1 PROFIBUS

#### GSD file

The GSD file contains three different modules:

- ND30-SPS (for ND30 series only)
- ND40-SPS (for ND40 series; process data for standard applications)
- ND40-SPS + SDO (for ND40 series; process data and service data for more complex applications)

Only one module can be activated at a time.

For setup with a PLC the GSD file 'NOV00A3E.gsd' is needed (as of Version 2.0). The file can be downloaded from [www.novotron-online.com](http://www.novotron-online.com). A function block is not required.

#### Supported baud rates

The baud rate of the PROFIBUS is determined automatically. NOVODRIVE's PROFIBUS interface supports the following baud rates:

Baud rate	9,6 kBit/s	19,2 kBit/s	45,45 kBit/s	93,75 kBit/s	187,5 kBit/s	500 kBit/s	1500 kBit/s	3000 kBit/s	6000 kBit/s	12000 kBits/s
Max. cable length	1200 m	1200 m	1200 m	1200 m	1000 m	400 m	200 m	100 m	100 m	100 m

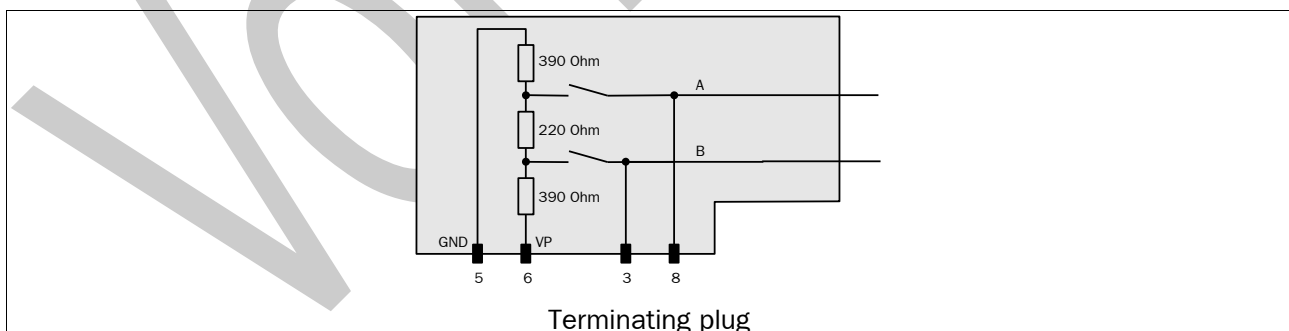
Branch lines up to 1500 kBit/s are possible only at a cable length below 6,6 m.  
In case of transmission speeds exceeding 1500 kBit/s, no branch lines should be connected.

#### Line parameters according to EN50170

Parameter	Value
Line impedance	135...165 Ohm at 3...20 MHz
Operating capacity	< 30 pF/m
loop resistance	< 110 Ohm/km
Wire diameter	> 0,64 mm
Wire cross-section	> 0,34 mm <sup>2</sup>

#### Bus termination

Each bus line must be terminated with appropriate resistors on both ends in order to guarantee a certain open-circuit potential and minimize line reflexions.



## 6.1.2 CANopen

### EDS file

The EDS file contains the objects of the CiA standard plus NOVODRIVE'S objects. For setup with a PLC the EDS file 'NOVOND40.eds' is needed.

The file can be downloaded from [www.novotron-online.com](http://www.novotron-online.com).

### Profiles

NOVODRIVE's CANopen interface supports the CiA Draft Standard Proposal 301 Version 4.1.

### Supported transfer types

- SDO Upload/Download: NOVODRIVE's CANopen interface supports direct access to objects in the expedited mode. Block transfer is not supported.
- PDO: NOVODRIVE's CANopen interface supports four Receive connections and four Transfer connections; user may choose between synchronous transfer and remote transfer.
- Connection monitoring: Select between Heartbeat Producer / Consumer and Nodeguarding by the PLC.

### Supported objects (see EDS file)

- CiA Draft Standard Proposal 301, index range 1000...1FFF .
- Manufacturer specific index range 3064...3400 for access to ND40 registers, and 4000...7FFF for the interpolation buffer.

### Supported baud rates

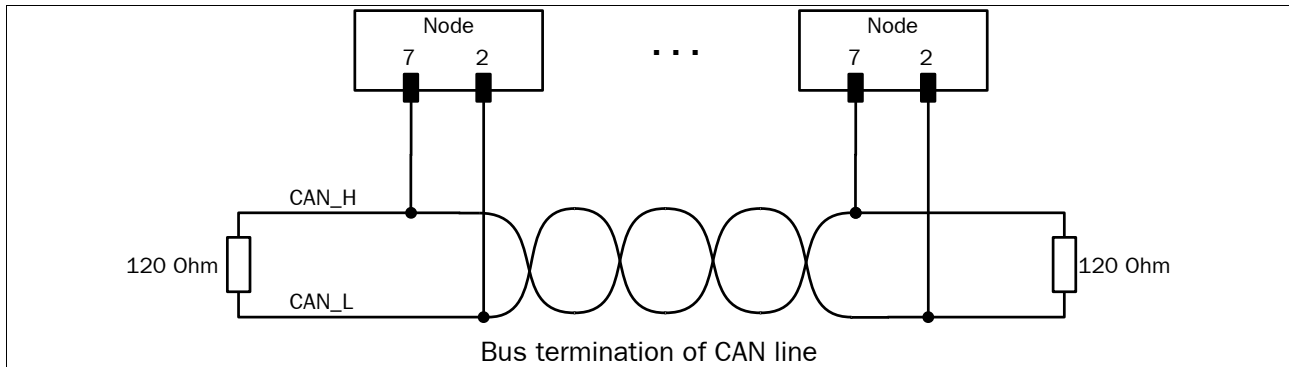
The baud rate of the CANopen is set manually over the setup software (→ Section 6.2). The following table shows what values are accepted as baud rates:

Baud rate	20 kBaud	50 kBaud	125 kBaud	250 kBaud	500 kBaud	800 kBaud	1 MBaud
Value	20000	50000	125000	250000	500000 default	800000	1000000
Max. cable length	5000 m	2500 m	500 m	250 m	100 m	50 m	25 m

### Line parameters according to ISO11898

Parameter	Value
Line impedance	120 Ohm
Operating capacity	< 60 pF/m

## Bus termination



CANopen bus termination must be 120 Ohm. Termination needs to be done also in case of short lines with low baud rates, as it also works as a pull-down resistor for all nodes.

## 6.2 Activation/Deactivation of Control by Bus System

Control by bus system must be activated and deactivated over the setup software. The bus system available is recognized by NOVODRIVE.

Activation and deactivation is done in four steps:

### (1) Deactivation:

- Go to the **'Bus System/Extension Module/PROFIBUS and CANopen'** page and select 'Query Mode only' for ignoring the setpoints of the PLC. NOVODRIVE's actual state continues to be transmitted to the PLC.  
Exception: the error code. Unlike the NOVODRIVE basic device, the PROFIBUS/CANopen interface always reports Error 661 when in the deactivated state.
- If you select 'No use of bus system', the bus interface gets deactivated. Upon restart of the basic device, no bus system will be activated.

### Activation:

- Go to the **'Bus System/Extension Module/PROFIBUS and CANopen'** page and select 'Usage of Drive Profil' for job execution over the state diagram (see 6.5.1), or select 'Usage of Job Control' for selecting jobs over the job control. In both cases, NOVODRIVE will be fully controlled by the PLC over the bus interface.

### (2) Entering the PROFIBUS address and the Node ID, respectively

Go to page **„Bus System/Extension Module/PROFIBUS and CANopen“** to enter the bus address. Note that any new address will not be accepted until the respective value has been saved in the EEPROM and NOVODRIVE has been reset or restarted.

- Valid PROFIBUS addresses: 1...126.
- Valid CANopen addresses (NODE-ID): 1...127.

Any other values generate Error 681.

### (3) Setting the baud rate (for CANopen only)

Go to the **„Bus System/Extension Module/PROFIBUS and CANopen“** page to set the baud rate.

### (4) Saving the settings and resetting of NOVODRIVE

Go to **'EEPROM'** and select 'SAVE'.

Then reset NOVODRIVE (button in the tool bar on the top).

The bus system now is ready to operate.

## 6.3 PROFIBUS Interface

### 6.3.1 General information

All inputs and outputs are predefined regarding their functions. No changes and variations are possible.

The bus interface is separated in two parts:

#### 1) Process data

The following graphic shows the process data interface (PDO):

Logic inputs:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17	Byte 18	Byte 19
ControlFlags	OperationMode (INT)	SpeedSetpoint (INT)*	Position Setpoint (DINT)*				AccelerateLimit (INT)*		DecelerateLimit (INT)*		CurrentPeak Limit1 (INT)*		DataInput16 (WORD)		DataInput32 (DWORD)				

Logic outputs:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17	Byte 18	Byte 19
StatusFlags	ActualOperation-Mode(INT)	ActualSpeed (INT)*	ActualPosition (DINT)*				ErrorCode (BCD-codiert) (WORD)		ScalingOverflow	Limitation	ActualCurrent (INT)*		DataOutput16 (WORD)		DataOutput32 (DWORD)				

Values designated with a \* are scaled in SI units (SI scaling is done over the setup software).

All other values contain bit fields or are numeric values.

For description of process data see Sections 6.6 and 6.7.

#### 2) Service data

The service data interface (SDO) offers free access to the NOVODRIVE registers.

Logic inputs:							Logic outputs:						
Byte 20	Byte 21	Byte 22	Byte 23	Byte 24	Byte 25	Byte 26	Byte 20	Byte 21	Byte 22	Byte 23	Byte 24	Byte 25	Byte 26
<div>SDO Command</div> <div>SDO Client address</div> <div>SDO Client data</div>							<div>SDO Service report</div> <div>SDO Server address</div> <div>SDO Server data</div>						

### 6.3.2 Description of SDO Data Exchange for PROFIBUS

NOVODRIVE's PROFIBUS interface comes with a SDO channel. To activate this channel, select the 'ND40-SPS+SDO' module in the GSD file. The module accommodates the ND40 process data channel and service data channel.

Over the SDO channel it is possible to read *and* write NOVODRIVE registers, provided PROFIBUS is in control of the process (if this is not the case, registers can only be read). Using the SDO channel requires sufficient knowledge about the NOVODRIVE basic device.

#### Function and characteristics of the SDO channel

The SDO channel consists of

- Send: Command, Client Address, Client Data
- Receive: Service Report, Server Address, Server Data

Send		Receive
SDO Command		SDO Service report
SDO Client Address		SDO Server Address
SDO Client Data	[0]	SDO Server Data
	[1]	
	[2]	
	[3]	

### Sequence of an SDO data exchange

Step	For reading a register	For writing a register
1	Set job address to register address.	
2	Set Client Data to '0'.	Set Client Data to the desired value.
3	Set SDO Command according to table below.	
4	Wait until Server Address and Client Address are identical.	
5	Read out Service Report and data. Any value other than '0' in the Service Report indicates an error.	Read out Service Report and data. Any value other than '0' in the Service Report indicates an error.
6	Set Command Sdo to '0'.	

### Job data

Register	Server Data [0]	Server Data [1]	Server Data [2]	Server Data [3]
ReadByte	Data	0x00	0x00	0x00
ReadWord	Data[H]	Data[L]	0x00	0x00
ReadLong	Data[3]	Data[2]	Data[1]	Data[0]
ReadWordX	Data[H]	Data[L]	0x00	0x00
ReadLongX	Data[3]	Data[2]	Data[1]	Data[0]
WriteByte	0x00	0x00	0x00	0x00
WriteWord	0x00	0x00	0x00	0x00
WriteLong	0x00	0x00	0x00	0x00
WriteWordX	0x00	0x00	0x00	0x00
WriteLongX	0x00	0x00	0x00	0x00

Register	SDO Command	Client Data [0]	Client Data [1]	Client Data [2]	Client Data [3]
ReadByte	0x01	0x00	0x00	0x00	0x00
ReadWord	0x02	0x00	0x00	0x00	0x00
ReadLong	0x03	0x00	0x00	0x00	0x00
ReadWordX	0x04	0x00	0x00	0x00	0x00
ReadLongX	0x05	0x00	0x00	0x00	0x00
WriteByte	0x06	Data	0x00	0x00	0x00
WriteWord	0x07	Data[H]	Data[L]	0x00	0x00
WriteLong	0x08	Data[3]	Data[2]	Data[1]	Data[0]
WriteWordX	0x09	Data[H]	Data[L]	0x00	0x00
WriteLongX	0x0A	Data[3]	Data[2]	Data[1]	Data[0]

To verify whether data have been written correctly, the register can be read out again.  
Values are not SI scaled.

### 6.3.3 SDO Command

Value range: BYTE [0...0xFF]

#### Description

Value	Function
0x00	--no job--
0x01	ReadByte
0x02	ReadWord
0x03	ReadLong
0x04	ReadWordX
0x05	ReadLongX
0x06	WriteByte
0x07	WriteWord
0x08	WriteLong
0x09	WriteWordX
0x0A	WriteLongX

### 6.3.4 SDO Client Address

Value range: INT [0...829]

In order to prevent inadvertent deactivation of the PROFIBUS interface, access to certain parameters (e.g. PDOHandler) is restricted.

#### Description

The client address is identical with the respective register address given in User Manual Volume 1 'Basic Device`, Section 9.1.

### 6.3.5 SDO Client Data

Value range: DWORD [0...0xFFFFFFFF] in case of client data as a double word

WORD [0...0xFFFF] in case of client data as a word

BYTE [0...0xFF] in case of client data as a byte

#### Description

Client data must be written into the NOVODRIVE register selected.

### 6.3.6 SDO Service Report

Value range: BYTE [0...0xFF]

#### Description

Value	Function
0x00	Data transfer complete
0x01	Internal error
0x02	Internal error
0x10	Internal error
0x80	PROFIBUS interface may only read registers as it is not in control of the process.
0xEE	Internal error
0xFB	Internal error
0xFC	Wrong SDO Command
0xFD	Invalid register address
0xFE	Invalid SDO Command
0xFF	Internal error

### 6.3.7 SDO Server Address

Value range: INT [0...999]

#### Description

The server address is identical with the respective register address given in User Manual Volume 1 'Basic Device', Section 9.1.

### 6.3.8 SDO Server Data

Value range: DWORD [0...0xFFFFFFFF]

#### Description

Server data read out from the NOVODRIVE register selected.

## 6.4 CANopen Interface

### 6.4.1 General Information

#### CAN Identifier

Object	COB-ID	Address range
NMT SERVICE	000h	000h
SYNC	080h	080h
EMERGENCY	080h + Node-ID	081h ... 0FFh
TPDO1	180h + Node-ID	181h ... 1FFh
RPDO1	200h + Node-ID	201h ... 27Fh
TPDO2	280h + Node-ID	281h ... 2FFh
RPDO2	300h + Node-ID	301h ... 37Fh
TPDO3	380h + Node-ID	381h ... 3FFh
RPDO3	400h + Node-ID	401h ... 47Fh
TPDO4	480h + Node-ID	481h ... 4FFh
RPDO4	500h + Node-ID	501h ... 57Fh
TSD0	580h + Node-ID	581h ... 5FFh
RSD0	600h + Node-ID	601h ... 67Fh
HEARTBEAT / NMT error control	700h + Node-ID	701h ... 77Fh

The Node-ID has to be within the range of 1 ... 127.

### PDO (Process data channel)

There are four process data channels for input data and four process data channels for output data. Each telegram can be 'mapped' with a maximum of six objects per channel.

Mappings can be altered in the preoperational state only. Otherwise Error '0800 0023h' is reported over the CAN bus.

Upon power-on, the telegrams have the following default settings:

RPDO 01...04 (PLC -> NOVODRIVE)				
Byte	RPD01	RPD02	RPD03	RPD04
[0]	ControlFlags	PositionSetpoint*	AccelerateLimit*	DataInput16
[1]	OperationMode		DecelerateLimit*	DataInput32
[2]	SpeedSetpoint*			
[3]				
[4]			PeakCurrentLimit1*	
[5]				
[6]				
[7]				

TPDO 01...04 (NOVODRIVE → SPS)				
Byte	TPD01	TPD02	TPD03	TPD04
[0]	StatusFlags	ActualPosition*	ErrorCode	DataOutput16
[1]	ActualOperationMode		ScalingOverflow	DataOutput32
[2]	ActualSpeed*			
[3]		ActualCurrent*		
[4]				
[5]				
[6]				
[7]				

Values designated with a \* are scaled in SI units (SI scaling is done over the setup software).

All other values contain bit fields or are numeric values.

## Objects available for mapping

RPDO 01...04			
Process data setpoints	Index	Sub-Index	Length
ControlFlags	2000h	0	08h
OperationMode	2001h	0	08h
SpeedSetpoint*	2002h	0	10h
PositionSetpoint*	2003h	0	20h
AccelerateLimit*	2004h	0	10h
DecelerateLimit*	2005h	0	10h
PeakCurrentLimit1*	2006h	0	10h
DataInput16	2007h	0	10h
DataInput32	2008h	0	20h

Values designated with a \* are scaled in SI units (SI scaling is done over the setup software).  
 All other values contain bit fields or are numeric values.

TPDO 01..04			
Process data actual values	Index	Sub-Index	Length
ControlFlags	2800h	0	08h
ActualOperationMode	2801h	0	08h
ActualSpeed*	2802h	0	10h
ActualPosition*	2803h	0	20h
ErrorCode	2804h	0	10h
ScalingOverflow	2805h	0	08h
Limitation	2806h	0	08h
ActualCurrent*	2807h	0	10h
DataOutput16	2808h	0	10h
DataOutput32	2809h	0	20h

Values designated with a \* are scaled in SI units (SI scaling is done over the setup software).  
 All other values contain bit fields or are numeric values.

## Configuration of PDO mapping parameters

Mapping parameters can be set and altered in the following objects:

- 1400h...1403h and 1600h...1603h (TPDO)
- 1800h...1803h and 1A00h...1A03h (RPDO)

More information on the parameters is given in Section 6.4.4.

## 6.4.2 Description of the SDO Data Exchange for CANopen

### Telegram structure

Byte	0	1 + 2	3	4	5	6	7
Function	CommandByte	MainIndex	SubIndex	DataByteLow	...	...	DataByteHigh

### CommandByte

	PLC → NOVODRIVE		NOVODRIVE → PLC	
Function	CommandByte	Data	CommandByte	Data
Write 4 Bytes	23h	[x] + [x] + [x] + [x]	60h	[0] + [0] + [0] + [0]
Write 2 Bytes	2Bh	[x] + [x] + [0] + [0]	60h	[0] + [0] + [0] + [0]
Write 1 Byte	2Fh	[x] + [0] + [0] + [0]	60h	[0] + [0] + [0] + [0]
Read	40h	[0] + [0] + [0] + [0]	43h (4 Bytes) 4Bh (2 Bytes) 4Fh (1 Byte)	[x] + [x] + [x] + [x] [x] + [x] + [0] + [0] [x] + [0] + [0] + [0]
Error Response	80h	[0] + [0] + [0] + [0]	80h	[Emergency error code]

### SDO abort code

NOVODRIVE supports the following abort codes:

Abort code	Description
0601 0000h	Unsupported access to an object
0601 0001h	Attempt to read a write-only object
0601 0002h	Attempt to write a read-only object
0602 0000h	Object does not exist in the object dictionary
0604 0041h	Object cannot be mapped to the PDO
0604 0042h	Number and length of the objects to be mapped would exceed PDO length
0606 0000h	Access failed due to a hardware error
0609 0011h	Sub-index does not exist
0609 0030h	Invalid value for parameter
0800 0000h	General error
0800 0020h	Data cannot be transferred or stored to the application
0800 0023h	Object dictionary dynamic generation failed or no object dictionary is present

### 6.4.3 Object Dictionary

#### CiA Draft Standard Proposal 301

Index	Sub-Index	DataType	Access	Command	Name/Description
1000h	00	Unsigned 32	RO	40h	DeviceType
1001h	00	Unsigned 8	RO	40h	Error register
1005h	00	Unsigned 32	RW	40h / 23h	COB-ID SYNC message
1009h	00	Unsigned 32	RO	40h	Manufacturer hardware version (EWBG)
100Ah	00	Unsigned 32	RO	40h	Manufacturer software version (EWBG)
100Ch	00	Unsigned 16	RW	40h / 2Bh	Guard time
100Dh	00	Unsigned 8	RW	40h / 2Fh	Life time factor
1010h					Store parameters
	00	Unsigned 8	RO	40h	Highest sub-index supported
	01	Unsigned 32	RO	40h	Save all parameters
	02	Unsigned 32	RO	40h	Save communication parameters
	03	Unsigned 32	RO	40h	Save application parameters
	04	Unsigned 32	RW	40h / 23h	Save manufacturer defined parameters
1011h					Restore parameters
	00	Unsigned 8	RO	40h	Highest sub-index supported
	01	Unsigned 32	RO	40h	Restore all default parameters
	02	Unsigned 32	RO	40h	Restore communication default parameters
	03	Unsigned 32	RO	40h	Restore application default parameters
	04	Unsigned 32	RW	40h / 23h	Restore manufacturer defined default parameters
1014h	00	Unsigned 32	RW	Unsigned 32	COB-ID emergency message
1016h					Consumer heartbeat time
	00	Unsigned 8	RO	40h	Highest sub-index supported
	01	Unsigned 32	RW	40h / 23h	Consumer heartbeat time
1017h	00	Unsigned 16	RW	40h / 2Bh	Producer heartbeat time
1018h					Identity object
	00	Unsigned 8	RO	40h	Highest sub-index supported
	01	Unsigned 32	RO	40h	Vendor-ID
	02	Unsigned 32	RO	40h	Product code (n.a.)
	03	Unsigned 32	RO	40h	Revision number (n.a.)
	04	Unsigned 32	RO	40h	Serial number
1200h					SDO server parameter
	00	Unsigned 8	RO	40h	Highest sub-index supported
	01	Unsigned 32	RO	40h	COB-ID client -> server(rx)
	02	Unsigned 32	RO	40h	COB-ID server -> client(tx)
1400h					RPDO communication parameter
1401h	00	Unsigned 8	RO	40h	Highest sub-index supported
1402h	01	Unsigned 32	RW	40h / 23h	COB-ID
1403h	02	Unsigned 8	RW	40h / 2Fh	Transmission type

Index	Sub-Index	DataType	Access	Command	Name/Description
1600h					RPDO mapping parameter
1601h	00	Unsigned 8	RW	40h / 23h	Highest sub-index supported
1602h	01	Unsigned 32	RW	40h / 23h	1 <sup>st</sup> application object
1603h	02	Unsigned 32	RW	40h / 23h	2 <sup>nd</sup> application object
	03	Unsigned 32	RW	40h / 23h	3 <sup>rd</sup> application object
	04	Unsigned 32	RW	40h / 23h	4 <sup>th</sup> application object
	05	Unsigned 32	RW	40h / 23h	5 <sup>th</sup> application object
	06	Unsigned 32	RW	40h / 23h	6 <sup>th</sup> application object
1800h					TPDO communication parameter
1801h	00	Unsigned 8	RO	40h	Highest sub-index supported
1802h	01	Unsigned 32	RW	40h / 23h	COB-ID used by RPDO
1803h	02	Unsigned 8	RW	40h / 2Fh	Transmission type
1A00h					TPDO mapping parameter
1A01h	00	Unsigned 8	RW	40h / 23h	Highest sub-index supported
1A02h	01	Unsigned 32	RW	40h / 23h	1 <sup>st</sup> application object
1A03h	02	Unsigned 32	RW	40h / 23h	2 <sup>nd</sup> application object
	03	Unsigned 32	RW	40h / 23h	3 <sup>rd</sup> application object
	04	Unsigned 32	RW	40h / 23h	4 <sup>th</sup> application object
	05	Unsigned 32	RW	40h / 23h	5 <sup>th</sup> application object
	06	Unsigned 32	RW	40h / 23h	6 <sup>th</sup> application object

**Process data scaled by NOVODRIVE (Manufacturer Object Dictionary)**

Index	Sub-Index	DataType	Access	Command	Name/Description
2000h	00	Unsigned 8	RW	40h / 2Fh	ControlFlags
2001h	00	Signed 8	RW	40h / 2Fh	OperationMode
2002h	00	Signed 16	RW	40h / 2Bh	SpeedSetpoint
2003h	00	Signed 32	RW	40h / 23h	PositionSetpoint
2004h	00	Signed 16	RW	40h / 2Bh	AccelerateLimit
2005h	00	Signed 16	RW	40h / 2Bh	DecelerateLimit
2006h	00	Signed 16	RW	40h / 2Bh	PeakCurrentLimit1
2007h	00	Unsigned 16	RW	40h / 2Bh	DataInput16
2008h	00	Unsigned 32	RW	40h / 2Fh	DataInput32
2800h	00	Unsigned 8	RO	40h	StatusFlags
2801h	00	Signed 8	RO	40h	ActualOperationMode
2802h	00	Signed 16	RO	40h	ActualSpeed
2803h	00	Signed 32	RO	40h	ActualPosition
2804h	00	Unsigned 16	RO	40h	ErrorCode
2805h	00	Unsigned 8	RO	40h	ScalingOverflow
2806h	00	Unsigned 8	RO	40h	Limitation
2807h	00	Signed 16	RO	40h	ActualCurrent
2808h	00	Unsigned 16	RO	40h	DataOutput16
2809h	00	Unsigned 32	RO	40h	DataOutput32

### Registers of NOVODRIVE basic device (Manufacturer Object Dictionary)

To read *and* write NOVODRIVE registers, provided CANopen is in control of the process (if this is not the case, registers can only be read). Using the SDO channel for manipulation of registers requires sufficient knowledge about the NOVODRIVE basic device.

Index	Sub-Index	DataType	Access	Command	Name/Description
3064h ... 33E7h	00	Unsigned8 Unsigned16 Unsigned 32	RO and RW	23h 2Bh 2Fh 40h	Registers see User Manual Volume 1 'Basic Device', Section 9.1, assignment as follows: 3064h      Register 100 ...      ... 33E7h      Register 999 Please note: The register addresses of the basic device are given in decimal values, and the object addresses for CANopen are given in hexadecimal values.
4000h ... 7FFFh	00	Unsigned16	RW	40h / 2Bh	XRAM see User Manual Volume 1 'Basic Device', Section 9.1.9, assignment as follows: 4000h      XRAM Word 0 ...      ... 4FFFh      XRAM Word 4095 5000h      Reserved ...      ... 5FFFh      Reserved 6000h      XRAM Word 8192 ...      ... 63FBh      XRAM Word 9211

## 6.4.4 Description of Objects 1000h...1A03h

### Object 1000h: DeviceType

Sub-Index	Access	Object				
00h	RO	<table><tr><td>Additional Information</td><td>Device profile number</td></tr><tr><td>31</td><td>16 15 0</td></tr></table> <p>Value '0x0000' is returned. (No predefined CiA profile is used.)</p>	Additional Information	Device profile number	31	16 15 0
Additional Information	Device profile number					
31	16 15 0					

### Object 1001h: Error register

Sub-Index	Access	Object										
00h	RO	<div>Error register</div> <div>70</div>										
		<table><tr><th>Bit</th><th>7</th><th>...</th><th>1</th><th>0</th></tr><tr><td>Meaning</td><td>-</td><td>...</td><td>-</td><td>Generic error</td></tr></table>	Bit	7	...	1	0	Meaning	-	...	-	Generic error
		Bit	7	...	1	0						
		Meaning	-	...	-	Generic error						
		Value '01h' is returned.										
		Error messages supported:										
		<table><tr><th>Abort code</th><th>Description</th></tr><tr><td>8130h</td><td>Life guard error or heartbeat error</td></tr><tr><td>8220h</td><td>Max. PDO length exceeded</td></tr></table>	Abort code	Description	8130h	Life guard error or heartbeat error	8220h	Max. PDO length exceeded				
		Abort code	Description									
		8130h	Life guard error or heartbeat error									
		8220h	Max. PDO length exceeded									

### Object 1005h: COB-ID SYNC message

Sub-Index	Access	Object															
00h	RW	<table><tr><td>x</td><td>gen</td><td>frame</td><td>0 0000h</td><td>11-bit CAN-ID</td></tr><tr><td colspan="4">29-bit CAN-ID</td></tr><tr><td>31</td><td>30</td><td>29</td><td>11</td><td>10</td><td>0</td></tr></table>	x	gen	frame	0 0000h	11-bit CAN-ID	29-bit CAN-ID				31	30	29	11	10	0
		x	gen	frame	0 0000h	11-bit CAN-ID											
		29-bit CAN-ID															
		31	30	29	11	10	0										
		<table><tr><th>Bit</th><th>Description</th></tr><tr><td>x</td><td>Can be '0' or '1'</td></tr><tr><td>gen</td><td>0: NOVODRIVE does not generate sync telegrams 1: NOVODRIVE generates sync telegrams (n.a.)</td></tr><tr><td>frame</td><td>0: 11-bit Identifier 1: 29-bit Identifier</td></tr></table>	Bit	Description	x	Can be '0' or '1'	gen	0: NOVODRIVE does not generate sync telegrams 1: NOVODRIVE generates sync telegrams (n.a.)	frame	0: 11-bit Identifier 1: 29-bit Identifier							
Bit	Description																
x	Can be '0' or '1'																
gen	0: NOVODRIVE does not generate sync telegrams 1: NOVODRIVE generates sync telegrams (n.a.)																
frame	0: 11-bit Identifier 1: 29-bit Identifier																
This object determines the CAN Identifier for the sync telegrams.																	
NOVDRIVE interface cannot generate sync telegrams. If 'gen' is set to '1', Error 0609 0030h is returned.																	

### Object 1009h: Manufacturer hardware version

Sub-Index	Access	Object
00h	RO	<div style="text-align: center;"> <div style="border: 1px solid black; display: inline-block; padding: 2px 10px;">Manufacturer hardware version</div> <div style="display: flex; justify-content: space-between; width: 100%;"> <span>31</span><span>0</span> </div> </div> <p>The content of Register 105 'HardwareCode' is returned (ASCII value).</p>

### Object 100Ah: Manufacturer software version

Sub-Index	Access	Object
00h	RO	<div style="text-align: center;"> <div style="border: 1px solid black; display: inline-block; padding: 2px 10px;">Manufacturer software version</div> <div style="display: flex; justify-content: space-between; width: 100%;"> <span>31</span><span>0</span> </div> </div> <p>The content of Register 106 'SoftwareVersion' is returned (ASCII value).</p>

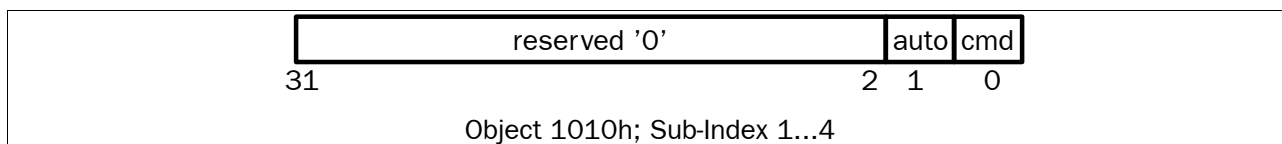
### Object 100Ch: Guard time

Sub-Index	Access	Object
00h	RW	<div style="text-align: center;"> <div style="border: 1px solid black; display: inline-block; padding: 2px 10px;">Guard time</div> <div style="display: flex; justify-content: space-between; width: 100%;"> <span>15</span><span>0</span> </div> </div> <p>This object determines the guard time in [ms] (value range: 0...65535). The value '0' switches off the guard function.</p>

### Object 100Dh: Life time factor

Sub-Index	Access	Object
00h	RW	<div style="text-align: center;"> <div style="border: 1px solid black; display: inline-block; padding: 2px 10px;">Life time factor</div> <div style="display: flex; justify-content: space-between; width: 100%;"> <span>7</span><span>0</span> </div> </div> <p>This object determines the multiplier for the 'Life guarding protocol' (value range: 0...255). The value '0' switches off the guard function.</p>

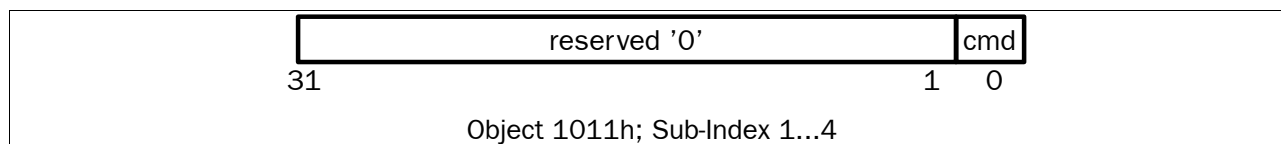
## Object 1010h: Store parameters



Bit	Description
auto	0: NOVODRIVE does not support automatic saving of parameters 1: NOVODRIVE supports automatic saving of parameters (n.a.)
cmd	0: NOVODRIVE does not support saving of parameters upon command 1: NOVODRIVE supports saving of parameters upon command

Sub-Index	Access	Object												
00h	RO	<div style="text-align: center;"><div style="border: 1px solid black; padding: 2px 10px;">SubIndex</div><div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>7</span><span>0</span></div></div> <p>The highest sub-index is returned.</p>												
01h	RO	<div style="text-align: center;"><div style="border: 1px solid black; padding: 2px 10px;">store all default parameters</div><div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>31</span><span>0</span></div></div> <p>Saving of CANopen default parameters is not supported.</p>												
02h	RO	<div style="text-align: center;"><div style="border: 1px solid black; padding: 2px 10px;">store communication default parameters</div><div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>31</span><span>0</span></div></div> <p>Saving of CANopen communication parameters is not supported.</p>												
03h	RO	<div style="text-align: center;"><div style="border: 1px solid black; padding: 2px 10px;">store application default parameters</div><div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>31</span><span>0</span></div></div> <p>Saving of CANopen application parameters is not supported.</p>												
04h	RW	<div style="text-align: center;"><div style="border: 1px solid black; padding: 2px 10px;">store manufacturer defined parameters</div><div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>31</span><span>0</span></div></div> <p>Saving of manufacturer defined parameters is supported upon command. To do so, write 'save' into the register.</p> <table style="margin: 10px auto; border-collapse: collapse;"><tr><td style="border: 1px solid black; padding: 2px 10px;">'e'</td><td style="border: 1px solid black; padding: 2px 10px;">'v'</td><td style="border: 1px solid black; padding: 2px 10px;">'a'</td><td style="border: 1px solid black; padding: 2px 10px;">'s'</td></tr><tr><td style="border: 1px solid black; padding: 2px 10px;">65h</td><td style="border: 1px solid black; padding: 2px 10px;">76h</td><td style="border: 1px solid black; padding: 2px 10px;">61h</td><td style="border: 1px solid black; padding: 2px 10px;">73h</td></tr><tr><td style="text-align: center;">31</td><td style="text-align: center;">24 23</td><td style="text-align: center;">16 15</td><td style="text-align: center;">8 7 0</td></tr></table>	'e'	'v'	'a'	's'	65h	76h	61h	73h	31	24 23	16 15	8 7 0
'e'	'v'	'a'	's'											
65h	76h	61h	73h											
31	24 23	16 15	8 7 0											

### Object 1011h: Restore parameters



Bit	Description
cmd	0: NOVODRIVE does not support the loading of parameters upon command 1: NOVODRIVE supports the loading of parameters upon command

Sub-Index	Access	Object																
00h	RO	<div style="text-align: center;"><div style="border: 1px solid black; padding: 2px 10px;">SubIndex</div><div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>7</span><span>0</span></div></div> <p>The highest sub-index is returned.</p>																
01h	RO	<div style="text-align: center;"><div style="border: 1px solid black; padding: 2px 10px;">restore all default parameters</div><div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>31</span><span>0</span></div></div> <p>Loading of CANopen default parameters is not supported.</p>																
02h	RO	<div style="text-align: center;"><div style="border: 1px solid black; padding: 2px 10px;">restore communication default parameters</div><div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>31</span><span>0</span></div></div> <p>Loading of CANopen communication parameters is not supported.</p>																
03h	RO	<div style="text-align: center;"><div style="border: 1px solid black; padding: 2px 10px;">restore application default parameters</div><div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>31</span><span>0</span></div></div> <p>Loading of CANopen application parameters is not supported.</p>																
04h	RW	<div style="text-align: center;"><div style="border: 1px solid black; padding: 2px 10px;">restore manufacturer defined parameters</div><div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>31</span><span>0</span></div></div> <p>Saving of manufacturer defined parameters is supported upon command. To do so, write 'load' into the register.</p> <table style="margin: 10px auto; border-collapse: collapse;"><tr><td style="text-align: center;">'d'</td><td style="text-align: center;">'a'</td><td style="text-align: center;">'o'</td><td style="text-align: center;">'l'</td></tr><tr><td style="text-align: center;">64h</td><td style="text-align: center;">61h</td><td style="text-align: center;">6Fh</td><td style="text-align: center;">6Ch</td></tr></table> <div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>31</span><span>24</span><span>23</span><span>16</span><span>15</span><span>8</span><span>7</span><span>0</span></div> <p>To rest parameters to default values, write 'fact' into the register. The CANopen interface is then deactivated and motor specific settings are cleared. Both must be reloaded over the setup software.</p> <table style="margin: 10px auto; border-collapse: collapse;"><tr><td style="text-align: center;">'t'</td><td style="text-align: center;">'c'</td><td style="text-align: center;">'a'</td><td style="text-align: center;">'f'</td></tr><tr><td style="text-align: center;">72h</td><td style="text-align: center;">63h</td><td style="text-align: center;">61h</td><td style="text-align: center;">66h</td></tr></table> <div style="display: flex; justify-content: space-between; margin-top: 5px;"><span>31</span><span>24</span><span>23</span><span>16</span><span>15</span><span>8</span><span>7</span><span>0</span></div>	'd'	'a'	'o'	'l'	64h	61h	6Fh	6Ch	't'	'c'	'a'	'f'	72h	63h	61h	66h
'd'	'a'	'o'	'l'															
64h	61h	6Fh	6Ch															
't'	'c'	'a'	'f'															
72h	63h	61h	66h															

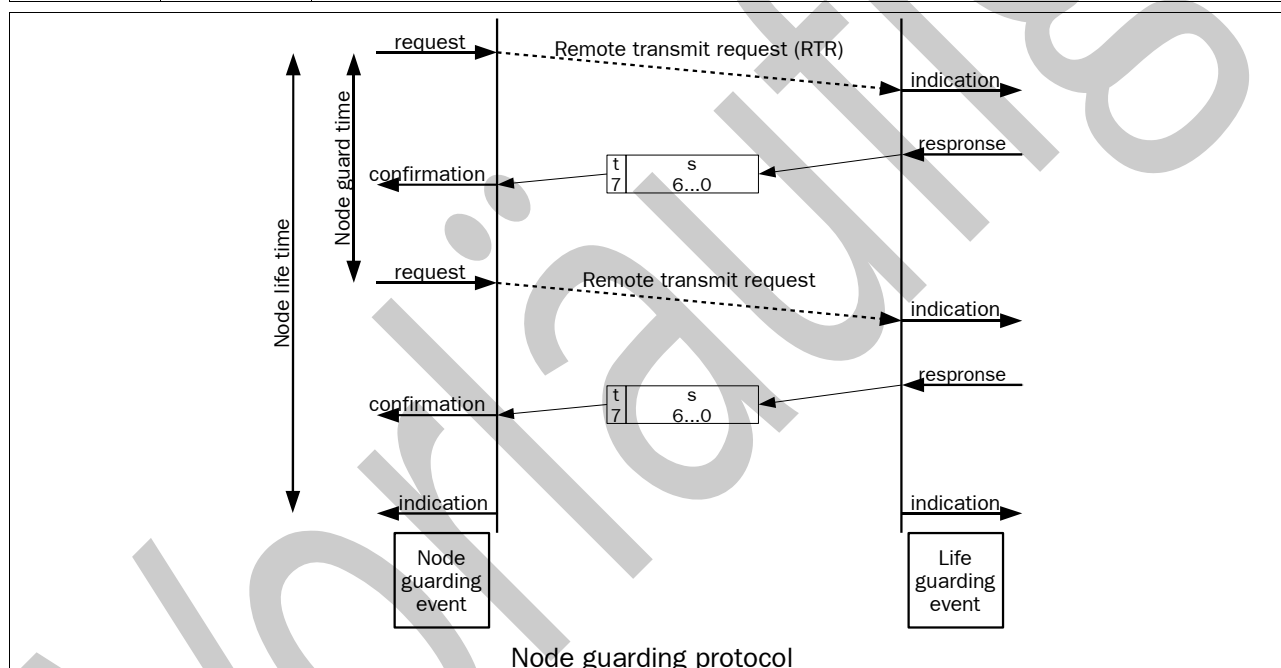
**Object 1014h: COB-ID emergency message**

Sub-Index	Access	Object													
00h	RW	<table border="1"><tr><td rowspan="2">valid</td><td rowspan="2">res.</td><td rowspan="2">frame</td><td>0 0000h</td><td>11-bit CAN-ID</td></tr><tr><td colspan="2">29-bit CAN-ID</td></tr><tr><td>31</td><td>30</td><td>29</td><td>11</td><td>10</td><td>0</td></tr></table>	valid	res.	frame	0 0000h	11-bit CAN-ID	29-bit CAN-ID		31	30	29	11	10	0
		valid				res.	frame	0 0000h	11-bit CAN-ID						
			29-bit CAN-ID												
		31	30	29	11	10	0								
		<table border="1"><tr><th>Bit</th><th>Description</th></tr><tr><td>valid</td><td>0: EMCY exists 1: EMCY does not exist (not supported)</td></tr><tr><td>frame</td><td>0: 11-bit Identifier 1: 29-bit Identifier</td></tr></table>	Bit	Description	valid	0: EMCY exists 1: EMCY does not exist (not supported)	frame	0: 11-bit Identifier 1: 29-bit Identifier							
Bit	Description														
valid	0: EMCY exists 1: EMCY does not exist (not supported)														
frame	0: 11-bit Identifier 1: 29-bit Identifier														
This object determines the CAN address of the emergency channel.															
The emergency channel cannot be switched off.															

Vorläufig

## Object 1016h: Consumer heartbeat time

Sub-Index	Access	Object
00h	RO	<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;">                     SubIndex                 </div> <div style="text-align: center; margin-top: 5px;"> <span style="margin: 0 10px;">7</span> <span style="margin: 0 10px;">0</span> </div> <p>To request the highest sub-index of the address.</p>
01h	RW	<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <div style="display: flex; justify-content: space-between; width: 100%;"> <span>reserviert</span> <span>Node-ID</span> <span>Heartbeat time</span> </div> <div style="display: flex; justify-content: space-between; width: 100%; font-size: small;"> <span>31</span> <span>24</span> <span>23</span> <span>16</span> <span>15</span> <span>0</span> </div> </div> <p><b>Heartbeat time &gt; 0:</b> The consumer heartbeat allows to make NOVODRIVE monitor the PLC. To do so, within a certain time interval (= heartbeat time [ms]) one heartbeat telegram has to arrive before NOVODRIVE goes into the 'Stopped' state and reports an error. The device to be monitored is selected over the 'Node-ID' field.</p> <p><b>Heartbeat time = 0:</b> Monitoring is done by 'Node-Guarding'. The master (PLC) inquires each slave's state in a certain time interval. If the master does not receive a response from the slave within a defined time interval (= guard time * Lifetimefactor [ms]), it recognizes that the respective slave has a time-out. If 'Lifetimefactor' is set to '0', no monitoring takes place.</p>



The NMT master sends an RTR in certain time intervals (node guard time), requesting the slave to send its current state. The response telegram consists of the NMT state 's' and a toggle bit 't'.

### Object 1017h: Producer heartbeat time

Sub-Index	Access	Object
00h	RW	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Heartbeat time &gt; 0</div>  15 0 </div> <p>This object determines the time interval a new heartbeat telegram is sent within. The value '0' stops the sending of telegrams. Values between '1...65535' indicate the time interval in ms.</p>

### Object 1018h: Identity object

Sub-Index	Access	Object
00h	RO	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">SubIndex</div>  7 0 </div> <p>The highest sub-index is returned.</p>
01h	RO	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Vendor-ID</div>  31 0 </div> <p>The content of Register 114 'CiaVendorId' is returned.</p>
02h	RO	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Product code</div>  31 0 </div> <p>The value '0' is returned.</p>
03h	RO	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Revision number</div>  31 0 </div> <p>The value '0' is returned.</p>
04h	RO	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Serial number</div>  31 0 </div> <p>The content of Register 104 'SerialNumber' is returned.</p>

### Object 1200h: SDO server parameter

Sub-Index	Access	Object
00h	RO	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">SubIndex</div>  <div style="display: flex; justify-content: space-around; width: 100px;">70</div> </div> <p>The highest sub-index is returned.</p>
01h	RO	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">COB-ID client -&gt; server (tx)</div>  <div style="display: flex; justify-content: space-between; width: 200px;">310</div> </div> <p>The CAN Identifier for TSD0 is returned.</p>
02h	RO	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">COB-ID server -&gt; client (rx)</div>  <div style="display: flex; justify-content: space-between; width: 200px;">310</div> </div> <p>The CAN Identifier for RSD0 is returned.</p>

### Object 1400h...1403h: RPDO communication parameter

Sub-Index	Access	Object																					
00h	RO	<div><div>SubIndex</div><div>70</div></div> <p>The highest sub-index is returned.</p>																					
01h	RW	<div><div><table><tr><td>valid</td><td>res.</td><td>frame</td><td>0 0000h</td><td>11-bit CAN-ID</td></tr><tr><td colspan="3"></td><td colspan="2">29-bit CAN-ID</td></tr><tr><td>31</td><td>30</td><td>29</td><td>11</td><td>100</td></tr></table></div><table><tr><th>Bit</th><th>Description</th></tr><tr><td>valid</td><td>0: PDO exists 1: PDO does not exist</td></tr><tr><td>frame</td><td>0: 11-bit Identifier 1: 29-bit Identifier</td></tr></table></div>	valid	res.	frame	0 0000h	11-bit CAN-ID				29-bit CAN-ID		31	30	29	11	100	Bit	Description	valid	0: PDO exists 1: PDO does not exist	frame	0: 11-bit Identifier 1: 29-bit Identifier
valid	res.	frame	0 0000h	11-bit CAN-ID																			
			29-bit CAN-ID																				
31	30	29	11	100																			
Bit	Description																						
valid	0: PDO exists 1: PDO does not exist																						
frame	0: 11-bit Identifier 1: 29-bit Identifier																						
02h	RW	<div><div>transmission type</div><div>70</div></div> <p>This object determines the transmission type of the TPDO. The only value supported is 'Feh'. The process data received are being processed immediately.</p>																					

### Object 1600h...1603h: RPDO mapping parameter

Sub-Index	Access	Object
00h	RW	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">number of mapped objects in PDO</div>  7                      0 </div> <p>This object determines the number of objects mapped.</p>
01h	RW	<div style="text-align: center;"> <div style="display: flex; justify-content: space-around; border-bottom: 1px solid black; margin-bottom: 5px;"> <span>MainIndex</span> <span>SubIndex</span> <span>Länge</span> </div> <div style="display: flex; justify-content: space-around;"> <span>31</span> <span>16 15</span> <span>8 7</span> <span>0</span> </div> </div> <p>The first RPDO element mapped. Valid index range: 2000h...2008h.</p>
...	...	...
06h	RW	<div style="text-align: center;"> <div style="display: flex; justify-content: space-around; border-bottom: 1px solid black; margin-bottom: 5px;"> <span>MainIndex</span> <span>SubIndex</span> <span>Länge</span> </div> <div style="display: flex; justify-content: space-around;"> <span>31</span> <span>16 15</span> <span>8 7</span> <span>0</span> </div> </div> <p>The sixth RPDO element mapped. Valid index range: 2000h...2008h.</p>

### Object 1800h...1803h: TPDO communication parameter

Sub-Index	Access	Object								
00h	RO	<div><div>SubIndex</div><div>70</div></div> <p>The highest sub-index is returned.</p>								
01h	RW	<div><div><div><div>valid</div><div>RTR</div><div>frame</div><div>0 0000h</div><div>11-bit CAN-ID</div></div><div>29-bit CAN-ID</div><div>31302911100</div></div><table><thead><tr><th>Bit</th><th>Description</th></tr></thead><tbody><tr><td>valid</td><td>0: PDO exists 1: PDO does not exist</td></tr><tr><td>RTR</td><td>0: Request over RTR 1: Request over RTR not permitted</td></tr><tr><td>frame</td><td>0: 11-bit Identifier 1: 29-bit Identifier</td></tr></tbody></table></div>	Bit	Description	valid	0: PDO exists 1: PDO does not exist	RTR	0: Request over RTR 1: Request over RTR not permitted	frame	0: 11-bit Identifier 1: 29-bit Identifier
Bit	Description									
valid	0: PDO exists 1: PDO does not exist									
RTR	0: Request over RTR 1: Request over RTR not permitted									
frame	0: 11-bit Identifier 1: 29-bit Identifier									
	RW	<div><div>transmission type</div><div>70</div></div> <p>This object determines the transmission type of the TPDO. The value range supported is '01h...F0h'. Data transfer after x sync telegrams. FCh: Data transfer over RTR.</p>								

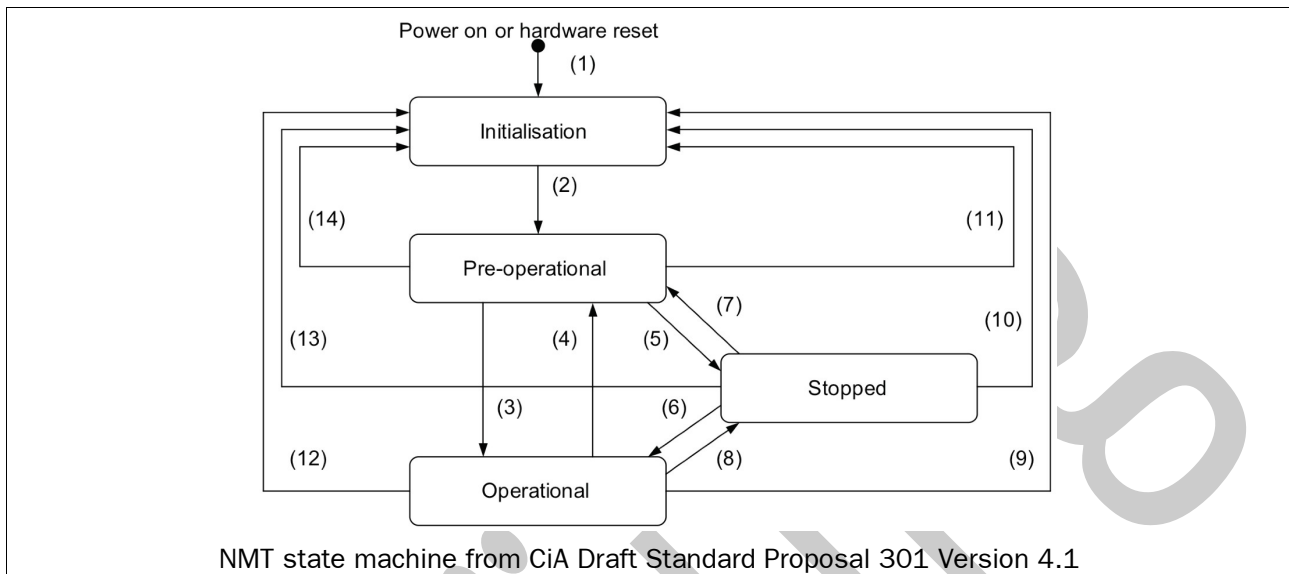
**Object 1A00h...1A03h: TPDO mapping parameter**

Sub-Index	Access	Object
00h	RW	<div> <div>number of mapped objects in PDO</div> <div>7 0</div> </div> <p>This object determines the number of objects mapped.</p>
01h	RW	<div> <div> <div>MainIndex</div> <div>SubIndex</div> <div>Länge</div> </div> <div>31 16 15 8 7 0</div> </div> <p>The first TPDO element mapped. Valid index range: 2800h...2809h.</p>
...	...	...
06h	RW	<div> <div> <div>MainIndex</div> <div>SubIndex</div> <div>Länge</div> </div> <div>31 16 15 8 7 0</div> </div> <p>The sixth TPDO element mapped. Valid index range: 2800h...2809h.</p>

## 6.4.5 Network Management

For altering the NMT state, two bytes need to be transmitted:

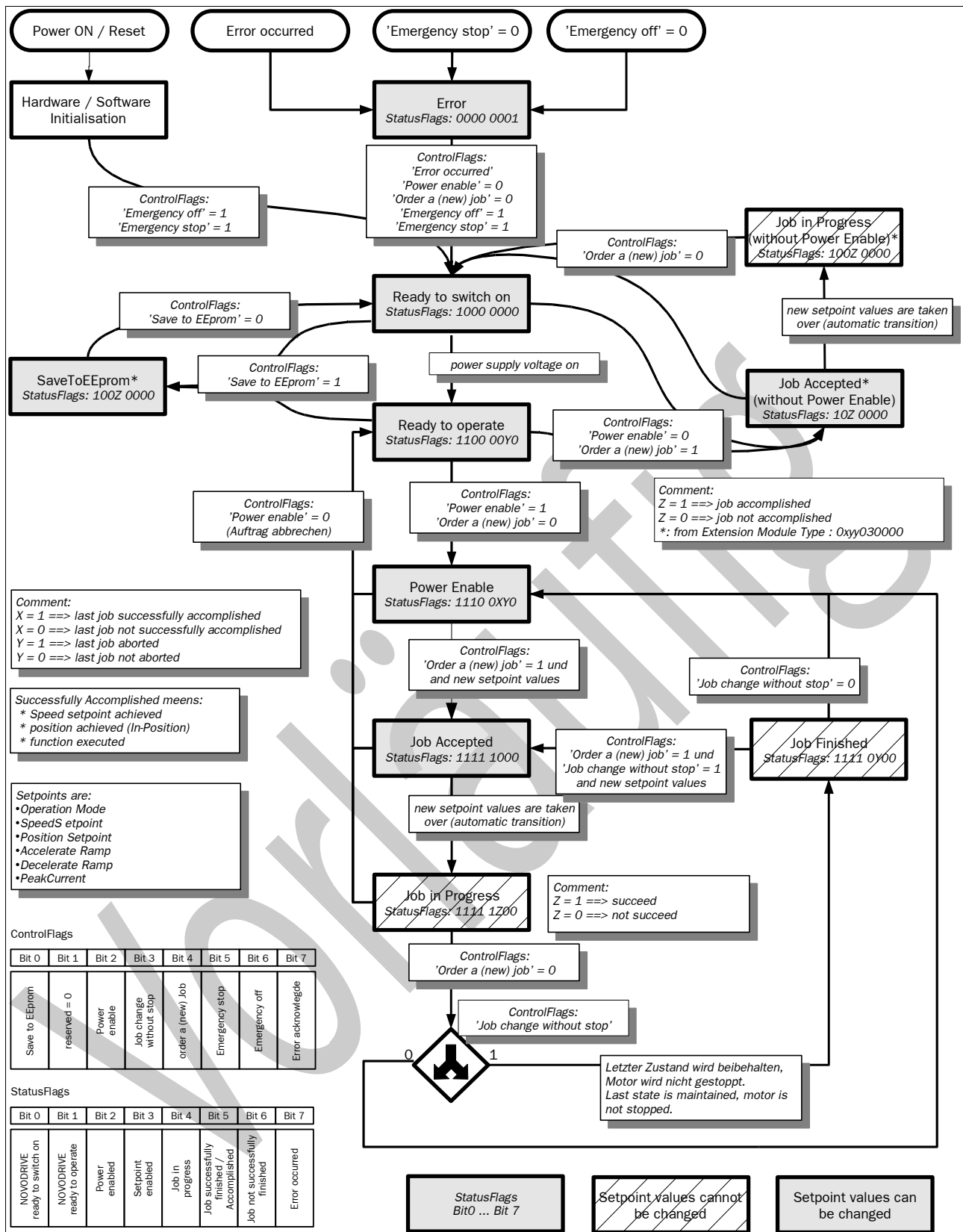
<div> <div>1.Byte</div> <div>2.Byte</div> </div>	
<div> <div>Command</div> <div>Device Address</div> </div>	
Command	Device address
01h, 02h, 80h, 81h, 82h	Device address = 00h: All NMT slaves and nodes connected are addressed. Device address = Node-ID: Only the node is addressed whose Node-ID is identical with the device address.



State transition	Command	Network state after alteration	Effect of state alteration
(1)	-	Initialization	Upon switching on line voltage or reset of NOVODRIVE, initialization is started automatically.
(2)	-	Pre-operational	Upon completion of initialization, the node automatically goes into the pre-operational state and the master decides as to how NOVODRIVE takes part in the communication. Network management, sync, emergency, and SDO channel enabled.
(3), (6)	01 xxh	Operational	Network management, sync, emergency, SDO channel and PDO channel active.
(4), (7)	80 xxh	Pre-operational	Network management, sync, emergency, and SDO channel active.
(5), (8)	02 xxh	Stopped	Only network management telegrams can be received.
(9)	81 xxh	Initialization	Re-initialization of all parameters relevant for communication using the default values of the bus system.
(10)			
(11)			
(12)			
(13)	82 xxh		
(14)			

## 6.5 Device Profiles

### 6.5.1 State diagram upon selection of 'Usage of Drive Profile'



### 6.5.2 Interface upon selection of 'Usage of Job Control'

If you use NOVODRIVE's job control over a bus system, jobs can be saved and activated in NOVODRIVE. Usage of the job control is the same for PROFIBUS and CANopen. For a detailed description of the job control see User Manual Volume 2, 'Software Reference', Section 6.3.

#### Input side

Input functions for job control	Control via bus interface PROFIBUS/CANopen Field 'DigitalInput16'	
Power enable	Bit0	0 = Power disable 1 = Power enable
Setpoint enable	Bit1	0 = Setpoint disable 1 = Setpoint enable 0 → 1 = Start
ID number of job to be started next: 1...127 (bus interface)	Bit8	Job number 1...127
	...	
	Bit14	

#### Output side

Output functions for job control	Control via bus interface PROFIBUS/CANopen Field 'DigitalOutput16'	
Ready to operate (reset done, line voltage connected, no error)	Bit 0 Reserved = 0	
Control output for motor brake	Bit 1 Reserved = 0	
0 = Job is complete. 1 = Job is in progress or waits for error acknowledgement by Setpoint Disable	Bit 8	
0 = Job is in progress 1 = Job is complete and waits for acknowledgement by Setpoint Disable	Bit 9	
0 = Job is in progress 1 = Job has been aborted by Inverter Disable or Setpoint Disable	Bit 10	
0 = Marker output is reset 1 = Marker output is set	Bit 11	

The current state can be read out over 'StatusFlag'. However, NOVODRIVE is controlled over 'DigitalInput16' only.

StatusFlags							
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
NOVODRIVE ready to switch on	NOVODRIVE ready to operate	Power enabled	Setpoint enabled	reserved	reserved	reserved	Error occurred

## 6.6 Description of Process Data Inputs

### 6.6.1 'ControlFlags'

Value range: 8 x BOOL

ControlFlags

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Save to EEprom	reserved = 0	Power enable	Job change without stop	order a (new) Job	Emergency Stop	Emergency Off	Error acknowledge

Bit	Name	Function
0	Save to EEPROM	When the bit is set to '1', current job parameters will be saved in the basic device.
2	Power enable	When the bit is set to '1', power gets enabled.
3	Job change without stop	When the bit is set to '1', input data can be altered and you may change from one operating mode to another without needing to stop the motor. When the bit is set to '0' when „Order (new) job“ = 0, the job gets finished and the motor gets stopped. To abort or finish jobs, the bit must be set to '0'.
4	Order (new) job	By a L → H edge of the bit new input data (and another operating mode, if desired) are accepted. During the time the bit is set to '1', the job is in progress. When the bit is set to '0', the job gets finished.
5	Emergency stop	When the bit is set to '0', the motor gets stopped (controlled deceleration by means of the deceleration ramp).
6	Emergency off	When the bit is set to '0', power gets disabled. The motor decelerates until it comes to a standstill.
7	Error acknowledge	By a L → H edge of the bit an error is cleared when bits 2 and 3 are set to '0'.

### 6.6.2 'OperationMode'

Value range: INT [0...99]

#### Description

All operating modes are described in detail in User Manual Volume 2 'Software Reference'. If an operating mode is selected that does not exist, Error E108 is generated.

Selection of an operating mode is **confirmed** over the 'ActualOperationMode' output.

Changing from one operating mode to another is possible if no job is in progress ('Job change without stop' and 'Order (new) job' = 0) and if jobs are changed without stop („Job change without stop“ = 1 and „Order (new) job“ = 0).

### 6.6.3 „SpeedSetpoint“

Value range: INT [-32768...32767]

#### Description

To set the speed of a job. Input is identical with NOVODRIVE Register 'DigitalSetpoint' as specified in the description of the respective operating mode. Values can be scaled in SI units over the setup software.

If a certain value range is exceeded (both in a positive and in a negative direction), a warning is generated and the setpoint is reduced accordingly.

### 6.6.4 „PositionSetpoint“

Value range: DINT [-2147483648...2147483647]

#### Description

To set the position of a job. Input is identical with NOVODRIVE Register „TargetPosition“ as specified in the description of the respective operating mode. Values can be scaled in SI units over the setup software.

If a certain value range is exceeded (both in a positive and in a negative direction), a warning is generated and the setpoint is reduced accordingly.

### 6.6.5 'AccelerateLimit' / 'DecelerateLimit'

Value range: INT [1...32767]

#### Description

To set limit values for the acceleration / deceleration ramp of a job. Values can be scaled in SI units over the setup software.

If a certain value range is exceeded, a warning is generated and the setpoint is reduced accordingly.

Values that are too low after scaling are automatically replaced by the lowest value permitted (see Section 6.7.6).

### 6.6.6 'CurrentPeakLimit1'

Value range: INT [0...32767]

#### Description

To set limit values for peak current of a job. Values can be scaled in SI units over the setup software. If a certain value range is exceeded, a warning is generated and the setpoint is reduced accordingly. Values that are too high after scaling are automatically replaced by the highest value permitted (see Section 6.7.6).

### 6.6.7 'DataInput16'

Value range: WORD [0...0xFFFF]

#### Description

Input is identical with NOVODRIVE Register „DataInput16“ as specified in the description of the respective operating mode.

When the job control is active, this input takes over the function of the digital inputs.

### 6.6.8 'DataInput32'

Value range: DWORD [0...0xFFFFFFFF]

#### Description

– Function reserved –

## 6.7 Description of Process Data Outputs

### 6.7.1 'StatusFlags'

Value range: 8 x BOOL

StatusFlags							
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
NOVODRIVE ready to switch on	NOVODRIVE ready to operate	Power enabled	Setpoint enabled	Job in progress	Job successfully finished / Accomplished	Job not successfully finished	Error occurred

Bit	Name	Function
0	NOVODRIVE ready to switch on	Bit is set to '1' when NOVODRIVE is ready to use and line voltage can be connected.
1	NOVODRIVE ready to operate	Bit is set to '1' when NOVODRIVE is ready to operate.
2	Power enable	Bit is set to '1' when power is enabled, i.e. NOVODRIVE is in operation.
3	Setpoint enable	Bit is set to '1' when the setpoint is enabled.
4	Job in progress	Bit is set to '1' when a job is in progress. Bit is set to '0' when a job has been finished by the PLC. Before a new job is selected, make sure the bit is set to '0'.
5	Job successfully finished / Accomplished	Bit is set to '1' when a job has been finished successfully (Register 'Status16' Bit 0 = 1). For example: a target position has been reached. Bit 5 = '1' also indicates a successful processing during an job. For example: actual speed complies with speed setpoint.
6	Job not successfully finished	Bit is set to '1' when a job has not been finished successfully.
7	Error occurred	Bit is set to '1' when an error has occurred in NOVODRIVE.

When a job is aborted by the PLC, neither Bit 5 nor Bit 6 is set.

### 6.7.2 'ActualOperationMode'

Value range: INT [0...99]

#### Description

Indicates active operating mode.

### 6.7.3 'ActualSpeed'

Value range: INT [-32768...32767]

#### Description

Indicates actual speed.

Values can be scaled in SI units over the setup software.

### 6.7.4 'ActualPosition'

Value range: DINT [-2147483648...2147483647]

#### Description

Indicates actual position.

Values can be scaled in SI units over the setup software.

### 6.7.5 'ErrorCode'

Value range: WORD [0...0xFFFF]

#### Description

Indicates the error code when an error has occurred in NOVODRIVE.

For a detailed description of all error codes see User Manual Volume 1 'Basic Device', Section 12.5.

### 6.7.6 'ScalingOverflow'

Value range: 8 x BOOL

ScalingOverflow							
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
SpeedSetpoint	ActualSpeed	PositionSetpoint	ActualPosition	AccelerateLimit	DecelerateLimit	PeakCurrentLimit1	ActualCurrent

If one of the bits is set to '1', an overflow has occurred during the scaling of an actual value or a setpoint. No error is generated.

#### Function of bits when set to '1':

Bit	Name	Function
Bit 0	SpeedSetpoint	Speed value does not lie within a certain range.
Bit 1	ActualSpeed	
Bit 2	PositionSetpoint	Position value does not lie within a certain range.
Bit 3	ActualPosition	
Bit 4	AccelerateLimit	Ramp value is '0' after scaling or does not lie within a certain range.
Bit 5	DecelerateLimit	
Bit 6	PeakCurrentLimit1	Current value does not lie within a certain range.
Bit 7	ActualCurrent	

### 6.7.7 'Limitation'

Value range: 8 x BOOL

Limitation							
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
CurrentLimitation	neg. LimitSwitch	pos. LimitSwitch	Reserved	Reserved	Reserved	JobControl activated	referenced

Bit	Name	Function
0	CurrentLimitation	Bit is set to '1' when NOVODRIVE's internal current limitation has responded. The current value is reduced to the highest value possible (→ User Manual Volume 1 'Basic Device', Section 'Register Status16').
1	Neg. LimitSwitch	Bit is set to '1' when the negative limit switch has responded (→ User Manual Volume 1 'Basic Device', Section 'Limit Switches').
2	Pos. LimitSwitch	Bit is set to '1' when the positive limit switch has responded (→ User Manual Volume 1 'Basic Device', Section 'Limit Switches').
6	JobControl activated	Bit is set to '1' when the job control is active.
7	Referenced	Bit is identical with Bit 4 of Register 'Status16' (→ User Manual Volume 1 'Basic Device', Section 'Register Status16').

### 6.7.8 'ActualCurrent'

Value range: INT [0...32767]

#### Description

Indicates the effective current of the motor.

Values can be scaled in SI units over the setup software.

### 6.7.9 'DataOutput16'

Value range: WORD [0...0xFFFF]

#### Description

When the job control is active, this output takes over the function of the digital outputs.

### 6.7.10 'DataOutput32'

Value range: DWORD [0...0xFFFFFFFF]

#### Description

– no function –

## 6.8 Bus Interface Related Error Codes of NOVODRIVE

Error code hex / dec		Display	Error class	Description
0x0636	1590	-	5	Connection to NOVODRIVE has been interrupted.
0x0661	1633	-	1	PROFIBUS interface lost control over the process. Information on actual state still can be displayed.
0x0662	1634	-	5	Process data exchange between PLC and PROFIBUS interface has been interrupted.
0x0663	1635	-	5	PROFIBUS interface lost control over the process (→ User Manual Volume1 'Basic Device', Section 'Register PdoHandler'). No connection of PROFIBUS interface to NOVODRIVE.
0x0680	1664	E 6 8 0	5	Internal error of extension module.
0x0681	1665	E 6 8 1	5	Invalid PROFIBUS address. Selected address is displayed in Register 'ErrorInfo1'.