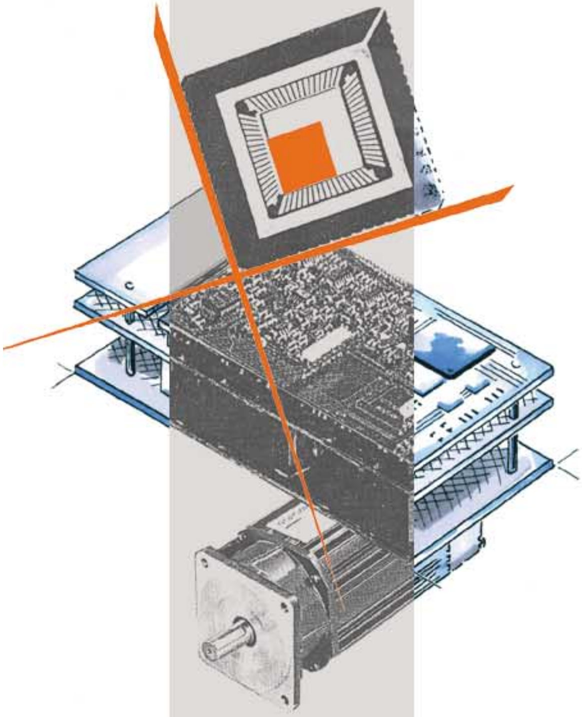
Manual Basic Functions ND31 and ND32

Version 4/2004



NOVOTRON

für Dynamik und Bewegung

Please read the following information about the symbols used in the manual:

Danger! Voltages may cause serious or fatal injury!

> **Noncompliance with instructions can endanger the life and sanity of persons!**

Caution ! ☛ **Caution! Make sure to handle the device correctly!**

> **Noncompliance with instructions can lead to the destruction or can cause malfunction of the device or the entire equipment!**

☞ **Link or recommendation**
Link to other sections of the text or recommendation for practical usage

1 2 — **Sequencing of an instruction**

Menu ***Limit values*** — **Designation of a menu or submenu**

Command *Channel1* — **Designation of a command or function**

[ ], [enter] — **Designation of a key or key combination**

Graphical representation of registers

SwVersion	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0xFF3D	7	6	5	4	3	2	1	0

SwVersion Designation
0xFF3D Address
R/W Read/Write
R ReadOnly

1 General Information

1.1 About this manual

The entire documentation of NOVODRIVE comprises 7 parts:

1 Manual **Basic Device ND31 and ND32**
Standard

2 Manual **Bus Functions ND31 and ND32**
On demand

3 Manual **Basic Functions ND31 and ND32**
On demand *)

4 Manual **Additional Functions ND31 and ND32**
On demand

5 Reserved

6 Manual **Start-up ND31 and ND32**
Standard

7 **Instructions for installation/exchange of ND31 and ND32**
Standard (leaflet)

The symbols used in the manuals are listed and explained on the inside front cover.

*) This manual

1.2 After-sales service

NOVOTRON GmbH
Mauserstrasse 31
71640 Ludwigsburg
Germany

Phone: +49 - (0)71 41 - 29 69 - 0
Fax: +49 - (0)71 41 - 29 69 - 22

1.3 Designations

ND 31 - 32 07 X S - 0 0 0 0 0 0 - 00

00 - Customization

0 - Standard
1 - Table interpolation

0 - 8 k memory
1 - 128 k memory

0 - Resolver
1 - ERN 1387 SinCos/EXE
2 - ECN/EQN on demand
3 - Hall sensor (open collector) + encoder
6 - Resolver + SinCos
7 - Hall sensor (TTL) + encoder

0 - NOVOBUS RS232
1 - CAN Bus NOVO
2 - InterbusS on demand
3 - Profibus on demand
4 - Sercos on demand
5 - NOVOBUS LWL
6 - CAN Bus Pro
7 - CAN Open on demand
8 - NOVOBUS RS 422

0 - Standard
1 - CAN coding
2 - PLC + CAN coding
3 - PLC 8051
4 - galv. isolation CAN Bus
5 - Reserved
6 - galv. isolation CAN Bus with GND

0 - no positioning control
1 - NOVOMERIK positioning control

T - trapezoidal EMF
S - sinusoidal EMF

V - 19“ module
X - prepared for compact case

02 - 2 A Rated current
04 - 4 A Rated current
05 - 5 A Rated current
07 - 7 A Rated current
10 - 10 A Rated current
12 - 12 A Rated current
20 - 20 A Rated current

32 - 320 V DC link voltage
56 - 560 V DC link voltage

31 - Version
32 - Version

ND - NOVODRIVE

2 Safety Instructions

NOVODRIVE contains voltages that can be fatal !

Wiring

Before switching on NOVODRIVE, carefully check the wiring. Make sure all plugs are properly connected and the device is properly grounded.

Protection

Make sure no voltage-carrying parts may be accidentally touched and NOVODRIVE safety components are in place and properly connected.

Emergency power-off

Provide an emergency power-off by which the motor can be stopped at any time.

Discharge time and contact voltage

After being switched off the electrolytic capacitors require at least five (5) minutes to discharge. That means: After being switched off the device still contains dangerous voltage for up to five minutes. During this time, do not touch the device or disconnect any plug.

In case the motor is still turning after the supply voltage has been switched off, hazardous contact voltage may be present in the device until its standstill. Discharge of the capacitors then begins after the standstill.

Inrush current limitation

Frequent switching of the supply voltage should be avoided, since thereby the inrush current limiter of NOVODRIVE may be overcharged, which may lead to the destruction of the inrush current limiting resistor. Wait one minute between switching on and switching off again.

Switching on/off sequence

When switching on, first apply the 24 VDC supply voltage for the NOVODRIVE control section before connecting with the power supply. When switching off, proceed vice versa.

☞ Please read:
„Basic Device ND31 and ND32“ manual,
Chapter 2, „Safety Instructions“

3 Basic Functions

This manual informs you about the possible configuration and settings of NOVODRIVE and explains the parameters you can read in and read out.

Do not modify a parameter if you are not absolutely sure about the parameter's meaning and the impact of its modification!

There are three ways to change the settings of NOVODRIVE:

- by means of the start-up software,
- by means of the RS 232 interface, NOVOBUS protocol,
- by means of the CAN-Bus, NOVOTRON protocol.

Start-up software

The start-up software allows you to comfortably adjust many parameters. If these options should not be sufficient for your application, you find the necessary information in this chapter. All parameters can also be read in and read out over NOVOBUS. The NOVOBUS protocol provides you with the necessary write and read commands.

NOVOBUS

Control of NOVODRIVE by means of NOVOBUS

Apart from the special command Reset, the control of NOVODRIVE is done by reading registers in and out directly in the drive. This allows you to do the configuration of the drive, to determine its status and to do the setpoint setting.

3.1 Short Overview

3.2 NOVODRIVE Memories

NOVODRIVE has two different memories:

- Integrated 512 Byte RAM.
  This memory accommodates the parameter set and the internal variables.

Address range (byte address) from	to	Assignment
0xFD80	0xFDFF	internal (no access !)
0xFE00	0xFE7F	variable
0xFE80	0xFE9F	internal
0xFEA0	0xFF3F	parameter set
0xFF40	0xFF7F	variable

- External battery backed-up RAM with word addressing.
  The storage capacity can be 8 kByte or 128 kByte.

Capacity	Address range (word address)		Assignment
	from	to	
8K / 128 k	0x0000	0x1FFF	used for internal system parameters (no access !)
128 k	0x2000	0x39FF	reserved for additional programs
128 k	0x4000	0x79FF	reserved for add. programs and motion cycles of table interpolation
128 k	0x8000	0xFFFF	reserved for motion cycles of table interpolation

Overwriting system parameters in the address range from 0x0000 to 0x1FFF of the external memory causes Error 101. In this case, the device needs to be sent in for repair.

Each write procedure to the memory over NOVOBUS has immediate consequences on the control. Of each parameter modification a replication is made in the external memory, i.e. the modification does not get lost if the 24 V supply voltage is switched off and switched on again.

3.3 Registers Overview

Address	Bit	Name	Description	see Chapt.
0x01E0	16	„@Tabelle“	Start for table interpolation	AF 12
0x01E2	16	„@PSrelativ“	Start for relative positioning	AF 8
0x01E4	16	„@Referenz“	Start for home	AF 5
0x01E6	16	„@PSabsolut“	Start for absolute positioning	AF 9
0x01E8	16	„@Feininterpolator“	Start for fine interpolation	BusF 6.4
0x01EA	16	„@dummy“	Dummy address	BF 3.6.7
0x01EC	16	„@AnInput2“	Start for torque setting	BF 3.8.4
0x01EE	16	„@Istwertsetzen“	Start for actual value setting	AF 6
0x01F0	16	„@PSonline“	Start for online positioning control	AF 10
0x01F2	16	„@Nullsuche“	Start for zero point search	AF 7
0x01F4	16	„@GPIOmanager“	Start for IO-Manager	BF 3.8.3
0x01FA	16	„@Kurve“	Start of cam disk function	AF 13
0xC02C	16	„BetriebStd“	Operating hours counter	BF 3.6.8.2
0xC02E	16	„SperreStd“	Operating hours counter	BF 3.6.8.2
0xC030	8	„BetriebMin“	Operating hours counter	BF 3.6.8.2
0xC031	8	„SperreMin“	Operating hours counter	BF 3.6.8.2
0xF000	8	„_iaist“	Actual value of motor current Phase A	BF 3.6.3.2
0xF001	1	„_ibist“	Actual value of motor current Phase B	BF 3.6.3.2
0xFDF6	16	„KurveOut“	Output cam disk function	AF 13
0xFE00	16	„errorcode“	Error code	BF 3.4.2
0xFE02	16	„scope_cnt“	Oscilloscope timing	BF 3.7
0xFE04	16	„scope_signal1“	Address of Signal 1	BF 3.7
0xFE06	16	„scope_signal2“	Address of Signal 2	BF 3.7
0xFE08	16	„scope_trigger“	Address of trigger signal	BF 3.7
0xFE0A	16	„GPO“	Output register digital outputs	BF 3.8.3
0xFE0C	16	„GPIN“	Input register digital inputs	BF 3.8.3
0xFE10	16	„Warning“	Error warning	BF 3.4.2
0xFE12	16	„nsollF“	Filtered speed setpoint	BF 3.6.5
0xFE1A	16	„FIsoll“	Fine interpolator output	BusF 6.4
0xFE1C	16	„VzSchlInc“	Tracking error signed (increments)	BF 3.6.3.6
0xFE1E	16	„VzSchlUmdr“	Tracking error signed (revolutions)	BF 3.6.3.6
0xFE20	16	„Phi1“	Commutation position	BF 3.9.3.9
0xFE2A	16	„ps-umdrehung“	Distance for relative positioning (revolutions)	AF 8
0xFE2C	16	„ps-impuls“	Distance for relative positioning (increments)	AF 8
0xFE3A	16	„?Tabelle“	Table initial address	AF 12
0xFE3C	16	„CANinput1“	Setpoint CAN Bus	BusF 6.3
0xFE3E	16	„CANinput2“	Setpoint CAN Bus	BusF 6.3
0xFE40	16	„CANinput3“	Setpoint CAN Bus	BusF 6.3
0xFE42	16	„NBinput/CANinput4“	Setpoint CAN Bus / NOVOBUS	BusF 6.3
0xFE46	16	„nsoll2“	Speed setpoint of position controller	BF 3.6.5
0xFE48	16	„STROD“	Encoder counter value	BF 3.8.6
0xFE4C	16	„InternSoll“	Speed setpoint of e.g. positioning control	AF 5 - 12
0xFE4E	16	„psa_positionH“	Absolute target position (revolutions)	AF 9
0xFE50	16	„psa_positionL“	Absolute target position (increments)	AF 9
0xFE52	8	„psa_status“	Status of absolute positioning	AF 9
0xFE54	16	„LageSoll“	Position setpoint (increments)	BF 3.6.3.5
0xFE56	16	„UmdrSoll“	Position setpoint (revolutions)	BF 3.6.3.5
0xFE5C	16	„AnInput1“	Value of Analog Input 1	BF 3.8.4
0xFE5E	16	„mSoll“	Torque setpoint	BF 3.6.3.2
0xFE60	16	„Sollwert“	Setpoint address	BF 3.6.5
0xFE62	16	„nSoll“	Speed setpoint of ramp generator	BF 3.6.5

Address	Bit	Name	Description	see Chapt.
0xFE68	16	„nist“	Actual speed value	BF 3.6.3.3
0xFE6A	16	„Pa0“	Fine interpolator, position pre-estimation	BusF 6.4
0xFE6C	16	„Pa1“	Fine interpolator, position pre-estimation	BusF 6.4
0xFE6E	8	„RautojuCSR“	Control register for auto adjustment	BF 3.6.9
0xFE6F	8	„Pa2“	Fine interpolator, position pre-estimation	BusF 6.4
0xFE72	16	„MaxSchleppInc“	Memory for maximum tracking error	BF 3.6.3.6
0xFE74	16	„Lage-nsoll“	Speed setpoint of position differentiator	BF 3.6.5
0xFE7A	8	„BalCnt“	Counter for braking circuit	-
0xFE7C	16	„LageIst“	Actual position value (increments)	BF 3.6.3.5
0xFE7D	8	„scope_delay“	Trigger delay oscilloscope	BF 3.7
0xFE7E	16	„UmdrIst“	Actual position value (revolutions)	BF 3.6.3.5
0xFE84	24	„Seriennummer“	Serial number in BCD format	BF 3.6.8.1
0xFE8B	8	„HwVersion“	Hardware configuration	BF 3.6.1.2
0xFEA0	16	„Steuerbits“	Bit field for activating various functions	BF 3.6.1.6
0xFEA2	16	„?SPS“	Pointer for selecting additional functions	BF 3.6.7
0xFEA8	16	„?Rampe+“	Pointer on acceleration ramp	BF 3.6.3.4
0xFEAA	16	„?Rampe-“	Pointer on braking ramp	BF 3.6.3.4
0xFEAC	16	„?CANout“	CAN parameter	BusF 6.3.5
0xFEAE	16	„?512us“	Pointer for selecting additional functions	BF 3.6.7
0xFEB0	16	„?FILage“	Pointer on position setpoint of fine interpol. (incr.)	BF 6.4
0xFEB2	16	„?CANControl“	CAN parameter	BusF 3.6.2.4
0xFEB4	8	„BremseT1“	Time between enable and release of brake	BF 3.6.10
0xFEB5	8	„BremseT2“	Time between activation of brake and disable	BF 3.6.10
0xFEB6	16	„?FForward“	Pointer on speed pre-control	-
0xFEB8	16	„?nSoll“	Pointer on input value of speed controller	BF 3.6.5
0xFEBA	16	„?mMax“	Pointer on torque limit value	BF 3.6.3.2
0xFEBC	16	„?mSoll“	Pointer on torque setpoint	BF 3.6.5
0xFEBE	16	„?Sollwert“	Pointer for speed setpoint	BF 3.6.5
0xFEC0	16	„KurveFakt“	Scaling of cam disk	AF 13
0xFEC2	16	„?GPO“	Pointer on GPO	BF 3.8.3
0xFEC4	16	„?SchRampe“	Pointer on quick-stop ramp	BF 3.6.3.4
0xFEC7	8	„GPOMaske“	Mask for Register GPO	BF 3.8.3
0xFEC8	16	„?512usA“	Pointer for selecting additional functions	BF 3.6.7
0xFECA	16	„?512usB“	Pointer for selecting additional functions	BF 3.6.7
0xFECC	16	„?FIUmdr“	Pointer on position setpoint of fine interpol. (revol.)	BusF 6.4
0xFECE	16	„?102us“	Pointer for selecting additional functions	BF 3.6.7
0xFED0	16	„NPIOffs“	Zero point offset using sine encoder	AF 7
0xFED2	16	„CANIDLSB“	CAN identifier LSB bits	BusF 6.1.1
0xFED6	16	„RefV3“	Zero-point search velocity	AF 7
0xFED8	16	„Window“	Tolerance for in-position message	AF 8 - 10
0xFEDA	16	„RefLage“	Home (position)	AF 6, 7
0xFEDC	16	„RefUmdr“	Home (revolutions)	AF 6, 7
0xFEDE	16	„RefV1“	Home speed 1	AF 5
0xFEE0	16	„RefV2“	Home speed 2	AF 5
0xFEE2	16	„RODinM“	Number of pulses of encoder input	BF 3.8.6
0xFEE4	16	„nMax“	Limit value for speed limitation	BF 3.6.3.3
0xFEE6	16	„ps-v0“	Positioning speed	AF 8 - 10
0xFEE8	8	„CANinitBTR0“	CAN Bus bit rate register	BusF 5.2
0xFEE9	8	„CANinitBTR1“	CAN Bus bit rate register	BusF 5.2
0xFEEA	16	„TCycle“	Cycle time fine interpolator	BusF 6.4
0xFEEE	8	„FIVerrundung“	Fine interpolator, smoothing	BusF 6.4
0xFEEF	8	„TCI2“	Time constant i^2t monitoring	BF 3.6.3.2
0xFEF0	16	„PhiPO“	Commutation position offset	BF 3.9

Adress	Bit	Name	Description	see Chapt.
0xFEF2	16	„AnIn1Offset“	Offset Analog Input 1	BF 3.8.4
0xFEF4	16	„Impulszahl“	Number of pulses of encoder emulation	BF 3.8.7
0xFEF6	8	„nSollFilter“	Setpoint filter	BF 3.6.6.2
0xFEF7	8	„Pole“	Number of poles in motor	BF 3.9
0xFEF8	8	„iKp“	Current controller P-gain	BF 3.6.6.1
0xFEF9	8	„iKi“	Current controller I-gain	BF 3.6.6.1
0xFEFA	16	„Rampe+“	Acceleration ramp	BF 3.6.3.4
0xFEFC	16	„Rampe-“	Braking ramp	BF 3.6.3.4
0xFEFE	16	„SchRampe“	Quick-stop ramp	BF 3.6.3.4
0xFF00	8	„InitIOM1“	Control register for encoder input / output	BF 3.8.6
0xFF01	8	„MaxTempMot“	Max. motor temperature	BF 3.6.4.2
0xFF02	8	„ResolvKomp“	Resolver compensation	BF 3.6.6.1
0xFF03	8	„emk0“	EMF of motor	BF 3.6.6.1
0xFF04	8	„NB_Init“	Baudrate NOVOBUS	BF 3.6.7.1.
0xFF05	8	„Feedback“	Feedback system	BF 3.6.1.3
0xFF06	8	„FICSR“	Fine interpolator Control/Status Register	BF 6.4
0xFF07	8	„bank“	Memory bank selection	BF 3.6.7
0xFF08	8	„CANSlaveBit8“	CAN configuration of CAN process-data read telegram	BusF 6.3
0xFF09	8	„CANSlaveB0“	CAN configuration of CAN process-data read telegram	BusF 6.3
0xFF0A	8	„CANSlaveB1“	CAN configuration of CAN process-data read telegram	BusF 6.3
0xFF0B	8	„CANSlaveB2“	CAN configuration of CAN process-data read telegram	BusF 6.3
0xFF0C	8	„CANSlaveB3“	CAN configuration of CAN process-data read telegram	BusF 6.3
0xFF0D	8	„CANSlaveB4“	CAN configuration of CAN process-data read telegram	BusF 6.3
0xFF0E	8	„CANSlaveB5“	CAN configuration of CAN process-data read telegram	BusF 6.3
0xFF0F	8	„CANSlaveB6“	CAN configuration of CAN process-data read telegram	BusF 6.3
0xFF10	8	„CANSlaveB7“	CAN configuration of CAN process-data read telegram	BusF 6.3
0xFF11	8	„CANCFG“	General CAN configuration	BusF 6.3
0xFF12	16	„?AnOut2“	Pointer on output value for Analog Output 2	BF 3.8.5
0xFF14	8	„AnOut2Fakt“	Scaling factor for Analog Output 2	BF 3.8.5
0xFF15	8	„AnOut2Offs“	Offset for Analog Output 2	BF 3.8.5
0xFF16	8	„CANTimeout“	CAN parameter	BusF 6.3
0xFF17	8	„FKSteuerung“	Offset compensation using sine encoder	BF 3.9.3.6
0xFF18	16	„KommSpurOff“	Offset commutation track using sine encoder	BF 3.9.3.7
0xFF1A	16	„Polabstand“	Pole distance of linear motor in µm	-
0xFF1C	16	„Impulslaenge“	Pulse length of linear motor in nm	-
0xFF1E	16	„KommLaenge“	Internal parameter	BF 3.9.3.7
0xFF20	8	„AnOutConfig“	Analog output configuration byte	BF 3.8.5
0xFF21	8	„MotConfig“	Motor configuration byte	BF 3.6.1.5
0xFF22	8	„AnIn2Fakt“	Scaling factor Analog Input 2	BF 3.8.4
0xFF23	8	„AnIn2Offset“	Offset Analog Input 2	BF 3.8.4
0xFF24	16	„?AnOut1“	Pointer on address for Analog Output 1	BF 3.8.5
0xFF26	8	„AnOutOffs“	Offset Analog Output 1	BF 3.8.5
0xFF27	8	„AnOutFakt“	Scaling factor Analog Output 1	BF 3.8.5
0xFF28	16	„?LageIst“	Pointer on source for actual position value	BF 3.6.5
0xFF2A	8	„Betriebsart“	Mode of operation	BF 3.6.1.4.
0xFF2B	8	„RefUmdrH“	Extension of Register RefUmdr	AF 6, 7
0xFF2C	16	„Schleppfehler“	Maximally tolerable tracking error	BF 3.6.3.6
0xFF2E	8	„i0max“	R.m.s. current limitation	BF 3.6.3.2
0xFF2F	8	„LKd“	Position controller pre-control	BF 3.6.6.4
0xFF30	8	„LKp“	Position controller P-gain	BF 3.6.6.4
0xFF31	8	„nKd“	Speed controller pre-control	BF 3.6.6.3
0xFF32	8	„nKp“	Speed controller P-gain	BF 3.6.6.3
0xFF33	8	„nKi“	Speed controller I-gain	BF 3.6.6.3

NOVOTRON

Address	Bit	Name	Description	see Chapt.
0xFF34	8	„MaxBalCnt0“	Threshold for braking circuit monitoring	-
0xFF35	8	„mMax“	Peak current	BF 3.6.3.2
0xFF36	16	„mMax16“	Peak current	BF 3.6.3.2
0xFF38	8	„CANservice“	CAN identifier	BusF 6.1.1
0xFF39	8	„CANmaster“	CAN identifier	BusF 6.3.1
0xFF3A	8	„CANslave“	CAN identifier	BusF 6.3.1
0xFF3B	8	„CANtime“	CAN identifier	BusF 6.3.1
0xFF3C	8	„Freigabe0“	Start state	BF 3.6.2.2
0xFF3D	8	„SwVersion“	Software configuration	BF 3.6.1.1
0xFF3E	8	„AnIn1Fakt“	Scaling factor Analog Input 1	BF 3.8.4
0xFF3F	8	„nFilter“	Tachometer filter	BF 3.6.6.2
0xFF43	8	„UmdrIstH“	Extension of Register UmdrIst	BF 3.6.3.5
0xFF52	8	„NBcontrol“	Control register for NOVOBUS	BF 3.6.2.3.
0xFF54	8	„MotTemp“	Motor temperature (resistance value)	BF 3.6.4.2
0xFF66	16	„AnIn_msoll“	Value of 8-Bit analog input	BF 3.8.4
0xFF68	16	„SchleppInc“	Actual tracking error unsigned (increments)	BF 3.6.3.6
0xFF6A	16	„SchleppUmdr“	Actual tracking error unsigned (revolutions)	BF 3.6.3.6
0xFF73	8	„ps-status“	Status register for positioning control	AF 8
0xFF79	8	„RefStatus“	Status register for home	AF 5
0xFF7A	8	„scope_status“	Status register for oscilloscope	BF 3.7
0xFF7C	8	„scope_timer0“	Time base of oscilloscope	BF 3.7
0xFF7F	8	„scope_level“	Trigger threshold of oscilloscope	BF 3.7
0xFF88	16	„_Phi“	Commutation track	BF 3.9
0xFFE0	8	„ADDRA“	Internal variable	-
0xFFE2	8	„ADDRB“	Sine encoder channel A	BF 3.9.3.5
0xFFE4	8	„ADDRC“	Sine encoder channel B	BF 3.9.3.5
0xFFE6	8	„ADDRD“	Internal variable	-
0xFFF2	8	„Status“	Status register for NOVODRIVE	BF 3.6.2.1
0xFFF3	8	„CANcontrol“	Control register for CAN Bus	BF 3.6.2.4
0xFFF4	8	„SPScontrol“	Internal control register	BF 3.4.2
0xFFF8	8	„Temp0“	Heat-sink temperature of inverter	BF 3.6.4.1
0xFFF9	8	„Temp1“	Heat-sink temperature of inverter	BF 3.6.4.1
0xFFFA	8	„Temp2“	Heat-sink temperature of inverter	BF 3.6.4.1
0xFFFB	8	„Temp3“	Heat-sink temperature of inverter	BF 3.6.4.1

Description:

BF	Manual „Basic Functions ND31 / ND32“
BusF	Manual „Bus Functions ND31 / ND32“
AF	Manual „Additional Functions ND31 / ND32“

Example:

AF, 12 Manual „Additional Functions ND31 / ND32“, Chapter 12

3.4 Behavior of NOVODRIVE

3.4.1 Switching sequence

When switching on NOVODRIVE, always maintain the following switching sequence:

1 Switch on 24 V supply voltage.
2 Provide 230 V or 400 V mains voltage, as the case may be.
3 Wait until you get the ready-to-operate message.
4 Enable NOVODRIVE.
5 Start NOVODRIVE.

When switching off NOVODRIVE, proceed vice versa:

1 Stop NOVODRIVE.
2 Wait until the motor has stopped.
3 Switch off mains voltage.
4 Switch off 24 V supply voltage.

3.4.2 Device states

Disable State (Register **Status** bit 0 = 1)	--> enable --> <-- disable <--	**Enable State** (Register **Status** bit 0 = 0)
• Inverter is not active. • Motor can be moved, as long as no mechanical brake is active . • Tracking error of position control is reduced to 0. • Controlled stopping is not possible.		• Inverter is active. • Motor stands or moves in a controlled way.

☞ **Do not enable NOVODRIVE, if it is not ready to operate.**

State determination:
Enabled = (Register **NBcontrol** bit 0 = 0)
AND (Register **CANcontrol** bit 0 = 0)
AND (Register **SPScontrol** bit 0 = 0)
AND (hardware enable GPIN5 = 24 V)
AND no „internal blocking“ through errors

To enable the drive, the above equation must read TRUE.

By means of **Freigabe0**, it can be specified which of the registers get the enable value after reset.

The control of the drive over the NOVOBUS protocol is normally done by means of **NBcontrol**. **CANcontrol** can only be written to over the CAN Bus. **SPScontrol** is reserved for internal use and cannot be written to.

Example:**Freigabe0** = binary 00000001

after reset:

NBcontrol	=	binary 1000 0001 (disable and stop)
CANcontrol	=	binary 0000 0000 (enable and start)
SPScontrol	=	binary 0000 0000 (enable and start)

Hardware enable is mandatory. The drive can then be enabled and disabled over bit 0 and started or stopped over bit 7of **NBcontrol**.

Stop State (Register **Status** bit 7 = 1)	--> start --> <-- stop <--	**Start State** (Register **Status** bit 7 = 0)
• Setpoint for speed is always 0. • Quick-stop ramp for speed is active, controlled braking takes place until standstill.		• Ramp values for acceleration and braking are active. • Various additional functions work only in case of enabled and started drive.

State determination:

Start = (Register **NBcontrol** bit 7 = 0)
AND (Register **CANcontrol** bit 7 = 0)
AND (Register **SPScontrol** bit 7 = 0)
AND (hardware start GPIN3 = 24 V)
AND enable active
AND no „internal stop“ through limit switch

To start the drive, the above equation must read TRUE. By means of **Freigabe0**, it can be specified which of the registers get the start value after reset.

Example: see enable example above; the start signal is always in bit 7.

Error state

- If an error occurs, the drive changes into the „Error“ status. The Register **Status** takes the binary value x01x xxx1. The drive is internally disabled and stopped.
- The error code lies in the Register **Errorcode**. It appears also in the 7-segment display. The error is cleared by writing the value 0xAF00 into **Errorcode**.
- If the error is cleared, the drive changes into the „internally disabled“ state. Register **Status** takes the binary value x001 xxx1.
- Either before or after the clearing of the error, the drive has to be disabled from outside in order to clear the state „internally disabled“.

Register	**Errorcode**
Address	0xFE00
Size	16 bit
Access	R/W
Function	Error code
Value range	see Chapter 3.5 write 0xAF00 for clearing

Errors that have no negative effect on the drive are ignored. Their error code is stored in the Register **Warning**.

Register	**Warning**
Address	0xFE10
Size	16 bit
Access	R/W
Function	Error code for warnings
Value range	see Chapter 3.5

3.4.3 Ready-to-operate

Ready-to-operate = Reset finished
AND no error
AND no undervoltage (mains voltage on)

Ready-to-operate is signaled over the ready-to-operate contact at connector X3.

3.4.4 Limit switch

The limit switches can trigger the following actions, depending on the configuration:

- limit switch inputs can be deactivated,
- limit switches can cause an error and disable the drive,
- limit switches can disable the drive internally; register **Status** then takes the binary value 0000 1xx0. The drive can leave the limit switch position again by changing its direction. No further error treatment is necessary.

The behavior can be specified over **SwVersion** (see Chapter 3.6.1.1).

The evaluation of the limit switches with regard to polarity and direction is determined by **NB_Init** (see Chapter 3.6.1.7).

3.5 Error Messages

Error codes are stored in registers as 3-digit BCD numbers.

Error number	Error	Description
001 to 100	Internal errors	Send device for repair
101	external memory	The part containing the system parameters in the battery backed-cleared up external memory has been cleared; send in device for repair
102 to 103	Internal errors	Error during initialization of variables; probably defective memory; send in device for repair
104 to 107	not assigned	
108	Betriebsart	Impermissible mode of operation
109	Internal error	Send device for repair
110	TempMotHw	Error during measuring of motor temperature
111	TempKKHw	Error during measuring of transistor temperature
112	StromA	Error during measuring of current A
113	StromB	Error during measuring of current B
114 to 139	not assigned	
140	Internal error	Send device for repair
141	Internal error	Send device for repair
142 to 210	not assigned	
211 to 223	Internal errors	Send device for repair
224 to 303	not assigned	
304	Overvoltage	DC link voltage too high; braking resistor not connected; mains voltage too high
305	Undervoltage	See 976
306	not assigned	
307	Inverter	Short-circuit of motor lead or defective inverter
308	Overcurrent	Overcurrent; current controller oscillates
309	Resolver	Defective resolver cable or resolver cable wired wrong
310	EndSchaltP	Positive limit switch has responded
311	EndSchaltN	Negative limit switch has responded
312 to 313	not assigned	
314	EndSchalter	Both limit switches have responded at the same time
315	i2t	R.m.s. current limitation has responded
316 to 399	not assigned	
400	OverTempTr	Temperature monitoring of inverter transistors has responded
401	OverTempMot	Motor temperature sensor has responded
402 to 499	not assigned	
501 to 503	NOVOBUS error	Error during communication over NOVOBUS
504	NOVOBUS	Invalid synchronous byte; frequently indicates invalid length of previous synchronous byte telegram
505	NOVOBUS Commandbyte	Invalid command byte in telegram
506	NOVOBUS Parameteradresse	Invalid address
507	NOVOBUS Checksumme	Invalid check sum
508 to 510	other NOVOBUS errors	Reserved
511 to 514	not assigned	
515	CAN error	Invalid command byte

Error number	Error	Description
516	CAN error	Invalid parameter address
517	CAN error	Invalid check sum
518	CAN error	Timeout through missing clock pulse telegram
519 to 523	CAN error	Error during communication over CAN bus
524 to 530	CAN Open error	Error codes of software extension CAN Open; see relevant documentation
531 to 576	CAN	Transmission error on CAN bus
577 to 600	not assigned	
601	PSOverflow	Error in positioning control
602	PSRampe	Ramp pointers do not point at the same registers
603	PSRampe	Ramp pointers do not point at Rampe+
604	PSRampe	Invalid ramp value
605 to 615	PSOverflow	Error in positioning control; distance too long, ramp too flat, speed too low
616	Ps?512us	Pointer ?512 µs points at wrong start address
617	PS?sollwert	Pointer ?sollwert points at wrong address
618	PSimax	Internal error
619		Internal error
620	DisPS	No positioning control in place
621 to 650	not assigned	
651 to 660	ENDAT	Error of software extension ENDAT; see relevant documentation
661 to 699	not assigned	
700	Schleppfehler	Monitoring of tracking error has responded
701 to 704	not assigned	
705	nmax	Speed is higher than nmax (with monitoring active) or 1,5 x nmax (with monitoring inactive); or: Failure of position measuring system (see also 309)
706	SINCOS	Error of signals of a sine encoder
707 to 799	not assigned	
800	ParamPole	Invalid number of motor poles
801	Encoder error	Edges on channels A and B closer than 100 ns
802	Pointer error	A program pointer in the parameter set points at an invalid address
803	Error 512us cycle	Overflow 512 µs cycle, computing load too high
804 to 878	Internal errors	Send in device for repair
879	Ablaufsteuerung	Autokomm function not successfully completed; reset drive to continue
880	Ablaufsteuerung AK_nichtbereit	Autokomm function cannot be started because of an error
881	Ablaufsteuerung noAutokomm	Command execution before execution of Autokomm function
882	Ablaufsteuerung Pos_Sperre	Start of a positioning procedure with disabled controller
883	Ablaufsteuerung FR_Feedback	Step/direction setting possible only with resolver as feedback system
884	CAN Profile Wegüberschreitung	Switch or marker not found within predefined path during home or zero-point search
885	CAN Profile Falsche Funktions-nummer	Invalid number of function indicated
886	not assigned	
887	H8 Version	H8 Version too old for software extension
888 to 899	not assigned	

Error number	Error	Description
900 bis 919	Internal errors	Send device for repair
920	Overcurrent	See 308
921	Overvoltage	See 304
922	Undervoltage	See 976
923 to 930	Internal errors	Send device for repair
931 to 933	Internal errors	Hardware error in connection with encoder outputs; send device for repair
934 to 936	Internal errors	Hardware error in connection with encoder inputs; send device for repair
937	Internal errors	Send device for repair
938 to 939	not assigned	
940 to 946	Internal errors	Send device for repair
947	PER Cos	Resolver error cosine; frequently indicates wrong wiring at X2
948	PER Sin	Resolver error sine; frequently indicates wrong wiring at X2
949	ExtRAM	Error in external memory; send in device for repair
950 to 956	Internal errors	Send device for repair; defect frequently caused by grounding of motor temperature sensor at connector X1 and X6, respectively
957	CAN rdwr	Error in CAN communication
958	CAN Master0	Error in CAN communication; clock pulse telegram without previous process-data write telegram
959	CAN MasterLost	Error in CAN communication; overflow of process-data write telegrams
960 to 969	Internal errors	Send device for repair
970	BallastImax	Peak-current monitoring of braking circuit; braking resistor either not connected or defective
971	BallastIeff	R.m.s. current monitoring of braking circuit; braking circuit overloaded
972	BallastTrTemp	Transistor monitoring of braking circuit; braking circuit overloaded
973	Resolver1	Unstable position measuring; defective resolver wiring or defective device
974	Resolver2	Resolver-cable monitoring; see Error 309
975	Restart	Enable after reset
976	Undervoltage	DC link voltage too low; control mains supply
977	LageSoll	Setpoint change too large in position setting mode
978	Motor cable	Motor-cable monitoring;motor is broken or bad setting of current controller
979	Overvoltage	see 304
980 to 991	Internal errors	Error of register in Potential-ASIC; send in device for repair
992	CAN BusOff	Bus-off status of CAN bus; defective CAN driver or wrong wiring (L and H exchanged)
993 to 996	Internal errors	Send device for repair
997	Up/Download	Error during quick upload/download over NOVOBUS
998 to 000	not assigned	

3.6 Basic Functions

3.6.1 Configuration

3.6.1.1 Register SwVersion

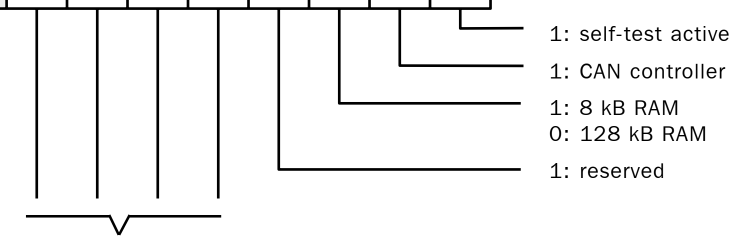
SwVersion	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFF3D	7	6	5	4	3	2	1	0	Bit

Assignments

bit 0 0: limit-switch monitoring off
1: limit-switch monitoring on
bit 1 1: limit switch generates error
0: limit switch trips stop
bit 2 1: automatic reactivation of CAN bus after Bus-off
bit 3 1: motor cable monitoring can be switched on/off
bit 4 1: tripping of i²t monitoring generates error message 315
bit 5 1: activates synchronization of controller cycle (see manual „Additional Functions“)
bit 6 0: reserved
bit 7 1: an internal test mode is activated, so that NOVODRIVE reverses the motor at the speed set; this operation mode is particularly suited to optimize the controller parameters

3.6.1.2 Register HwVersion

HwVersion	R	R	R	R	R	R	R	R	
0xFE8B	7	6	5	4	3	2	1	0	Bit



0	ND31-3202
1	ND31-3204
2	ND31-3207
3	ND31-3212
4	ND32-5605
5	ND32-5610
6	ND32-5620

Assignments

bit 0 1: self-test active when switched on
bit 1 1: CAN controller in place
bit 2 1: 8 kByte external RAM
0: 128 kByte external RAM
bit 3 1: reserved
bit 4...7 device type according to table above

3.6.1.3 Register Feedback

Feedback	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFF05	7	6	5	4	3	2	1	0	Bit

Feedback:

Bit 2	Bit 1	Bit 0	
0	0	0	resolver
0	0	1	digital encoder
0	1	0	reserved
0	1	1	Hall and digital encoder
1	1	1	sine encoder

- Bit 3: 0: reserved
- Bit 4: 0: reserved
- Bit 5: 0: reserved
- Bit 6: 1: negative direction
- Bit 7: 0: speed range 0 ... 6000 r/min
  1: speed range 0 ... 18000 r/min

Extension (starting with H8 Version 2.01 of 07/01/1997)

Feedback	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFF05	7	6	5	4	3	2	1	0	Bit

Feedback:

Bit 2	Bit 1	Bit 0		
0	0	0	resolver	
0	0	1	digital encoder	
0	1	0	reserved	
0	1	1	Hall and digital encoder	
1	0	0	commutation:	resolver
			speed:	resolver
			position:	sine encoder
1	0	1	commutation:	resolver
			speed:	sine encoder
			position:	sine encoder
1	1	0	reserved	
1	1	1	sine encoder	

- Bit 3: 0: reserved
- Bit 4: 0: reserved
- Bit 5: 1: negative direction for second position measuring system
- Bit 6: 1: negative direction
- Bit 7: 0: speed range 0 ... 6000 r/min
  1: speed range 0 ... 18000 r/min

NOVOTRON

Extension (starting with H8 Version 3.0)

Feedback	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFF05	7	6	5	4	3	2	1	0	Bit

Feedback:

0	0	0	resolver		
0	0	1	digital encoder		
0	1	0	reserved		
0	1	1	Hall and digital encoder		
1	0	0	commutation:	resolver
			speed:	resolver
			position:	sine encoder
1	0	1	commutation:	resolver
			speed:	sine encoder
			position:	sine encoder
1	1	0	commutation:	Hall
			speed:	sine encoder
			position:	sine encoder
1	1	1	sine encoder	

0: reserved

1: monitoring of sine encoder active

1: negative direction for second position measuring system

1: negative direction

0: speed range 0 ... 6000 r/min

1: speed range 0 ... 18000 r/min

Assignments

bit 0...5	by these bits the feedback system is selected
bit 4	monitoring of sine encoder generates Error 706, if signal levels become too low
bit 5	reversion of rotation direction of second measuring system
bit 6	reversion of rotation direction of first measuring system
bit 7	selection of speed range: depending on the selected speed range, also the parameters **emk0**, **ResolvKomp**, the ramp values and speed values, and the settings of the controllers for speed and position must be adjusted

3.6.1.4 Register Betriebsart

Betriebsart	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFF2A	7	6	5	4	3	2	1	0	Bit

Additional info ()
Setpoint source (B)
Controlled variable (A)

With register **Betriebsart** the setpoint source and the controlled variable can be specified. For the changes to become effective, a reset has to be executed afterwards. By programming this register, the respective pointers are rewritten after reset. If this is not desired, initialize **Betriebsart** with 0xFF.

In the following table, bits 7, 6 and 5 specify the controlled variable (A), and bits 4, 3 and 2 determine the setpoint source. bits 1 and 0 contain additional information about the feedback system used.

Assignments	**A**	**B**	**C**	**Controlled variable, settings, feedback**
	765	432	10	
Torque control	001	001	00	Torque control, Analog Input 1
	“	010	00	Torque control, Analog Input 2
	“	011	00	Torque control, CAN
	“	100	00	Torque control, digital
Speed control	010	001	00	Speed control, Analog Input 1
	“	010	00	Speed control, Analog Input 2
	“	011	00	Speed control, CAN
	“	100	00	Speed control, digital
Speed control with position controller	011	001	00	Speed control with position controller, Analog Input 1
	“	010	00	Speed control with position controller, Analog Input 2
	“	011	00	Speed control with position controller, CAN
	“	100	00	Speed control with position controller, digital

Position control	100	001	00	Position control, setting via encoder
	“	010	00	Position control, setting via step/direction
	“	011	01	Position control, digital setting with standard feedback (resolver)
	“	“	10	Position control, digital setting with encoder as a second measuring system
	“	100	01	Position control, setting over CAN with standard feedback (resolver)
	“	100	10	Position control, setting over CAN with encoder as a second measuring system
Internal settings	“	101	01	internal setting (e.g. positioning control) with standard feedback (resolver)
	“	101	10	internal setting (e.g. positioning control) with encoder as a second measuring system
Test mode	101	001	00	Test mode (reverse) without position controller
	“	010	00	Test mode (reverse) with position controller
Custom setting	111	111	11	All settings remain in place.

☞ Digital setpoint setting means that the setpoint has to be written into **Sollwert**. This can be done over NOVOBUS or over CAN Bus.

3.6.1.5 Register MotConfig

MotConfig	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFF21	7	6	5	4	3	2	1	0	Bit
	1	1	1	1	1	1	1	0	motor temperature sensor at power connector (X1)
	1	1	1	1	1	1	1	1	motor temperature sensor at resolver connector (X2)

Extension (starting with H8 Version 3.0)

MotConfig	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
00xFF21	7	6	5	4	3	2	1	0	Bit

Bit 0:

1: motor temp. sensor at resolver connector (X2)

0: motor temp. sensor at power connector (X1)

Bits 3, 2, 1: 1 1 1 — 1: reserved

Bit 4: 1: reserved

Bits 6, 5: 1 1 — 1: reserved

Bit 7: cam disk function
0: Input • 16 (12bit Input)
1: Input • 1 (16bit Input)

NOVOTRON

3.6.1.6 Register Steuerbits (starting H8 Version 3.03)

Steuerbits	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFEA0	7	6	5	4	3	2	1	0	Bit

	0: CAN timeout trips stoppage 1: CAN timeout trips disable
1 ≠ 2	monitoring of encoder input active
	0: reserved
4 ≠ 5	Switch-off of Analog Input 1
6 ≠ 7	Error 705 if speed limit is exceeded

Steuerbits	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFEA0	15	14	13	12	11	10	9	8	Bit

	1: Position pre-estimation active (CAN)
	1: reserved
11 ≠ 12	Analog Output 1 12bit resolution
11 = 12	Analog Output 1 14bit resolution
13 ≠ 14	**nKd** operates as real D-gain
1	reserved

3.6.1.7 Register NB_Init

With **NB_Init** the baud rate of NOVOBUS is being set.

NB_Init	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFF04	7	6	5	4	3	2	1	0	Bit
	X	X	X	X	X	X	0	0	38400 baud
	X	X	X	X	X	X	0	1	19200 baud

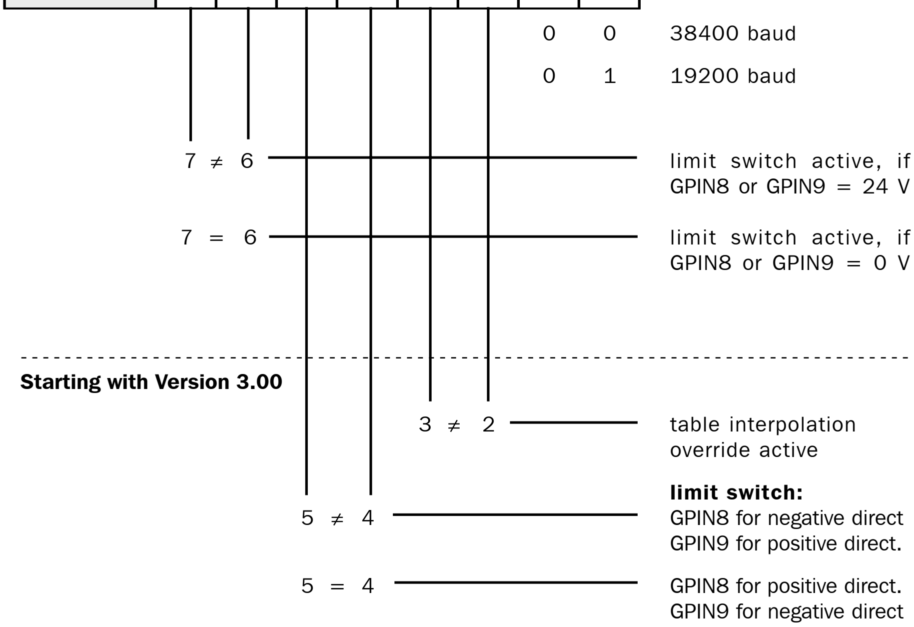
☞ For a change in the baud rate to become effective, a reset has to be executed. Afterwards, the start-up software has to be restarted with the new baud rate as command line parameter.

Example: ND31 19200

☞ In the Windows version of the start-up software, the baud rate must be changed in the NBServer program.

Extension (starting with H8 Version 2.01 07/01/1998)

NB_Init	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFF04	7	6	5	4	3	2	1	0	Bit



If not used, override must be reset again.

NOVOTRON

3.6.2 State values

3.6.2.1 Register Status

ByteStatus	R	R	R	R	R	R	R	R	
0xFFF2	7	6	5	4	3	2	1	0	Bit

Bit	
0	1: disable
1	1: reserved
2	1: current limitation
3	1: limit switch
4	1: internally disabled
5	1: error
6	1: reset
7	1: quick-stop

Assignments

bit 0	1: drive disabled 0: drive enabled
bit 1	0: reserved
bit 2	1: current limitation active
bit 3	1: limit switch has been tripped
bit 4	1: internally locked, waiting for disabling
bit 5	1: error
bit 6	1: drive in reset state
bit 7	1: quick-stop active

3.6.2.2 Register Freigabe0

With **Freigabe0** the start state of NOVODRIVE is programmed, i.e. if NOVODRIVE is in the disabled or active status. This register is evaluated only after a reset.

Freigabe0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFF3C	7	6	5	4	3	2	1	0	Bit

Bit	
0	1: disable with **NBcontrol**
1	1: disable with **CANcontrol**
2	1: disable with **SPScontrol**
3	0: reserved
4	0: reserved
5	0 reserved
6	0: reserved
7	0: reserved

☞ Disable with register **SPScontrol** is used only by software extensions.

3.6.2.3 Register NBcontrol

With **NBcontrol** the drive can be disabled and stopped over NOVOBUS.

NBcontrol	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFF52	7	6	5	4	3	2	1	0	Bit

- Bit 0 – 1: NOVOBUS disable
- Bit 1 – 0: reserved
- Bit 2 – 0: reserved
- Bit 3 – 0: reserved
- Bit 4 – 0: reserved
- Bit 5 – 0 reserved
- Bit 6 – 0: reserved
- Bit 7 – 1: NOVOBUS stop

3.6.2.4 Register CANcontrol

With **CANcontrol** the drive can be disabled and stopped over the CAN bus.

CANcontrol	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFFF3	7	6	5	4	3	2	1	0	Bit

- Bit 0 – 1: CAN disable
- Bit 1 – 0: reserved
- Bit 2 – 0: reserved
- Bit 3 – 0: reserved
- Bit 4 – 0: reserved
- Bit 5 – 0 reserved
- Bit 6 – 1: clear error
- Bit 7 – 1: CAN stop

CANControl is evaluated indirectly over the pointer **?CANControl**. This means that **CANControl** can be transfered to process-data write telegrams.

☞ **see „Bus Functions“ manual, Chapter 6.3.4**

Register	**?CANControl**
Address	0xFEB2
Size	16 bit
Access	R/W
Standard value	0xFFF3

3.6.3 Actual values, setpoints, limit values

3.6.3.1 Current scaling

	Phase current (8Bit) F_{iph}	R.m.s. current (8 Bit) F_{ieff}	R.m.s. current (16 Bit) F_{ieff16}
NOVODRIVE	[A/Ink]	[Aeff/Ink]	[Aeff/Ink]
ND31-3202	0,0589	0,3045	0,000135
ND31-3204	0,206	0,1206	0,000470
ND31-3207	0,206	0,1206	0,000470
ND31-3212	0,353	0,2060	0,000805
ND32-5605	0,150	0,0885	0,000035
ND32-5610	0,300	0,1769	0,000690
ND32-5620	0,600	0,3540	0,001380

3.6.3.2 Current values and limit values

Register	Description	Address	Size	Access	Scaling
Setpoints and actual values of phase currents in motor					
_iaist	Phase A actual value	0xF000	8 bit signed	R	F_{iph}
_iasoll	Phase A setpoint	0xF00C	8 bit signed	R	F_{iph}
_ibist	Phase B actual value	0xF001	8 bit signed	R	F_{iph}
_ibsoll	Phase B setpoint	0xF00D	8 bit signed	R	F_{iph}
Peak value					
Imax	Limit value of NOVODRIVE	0xFE88	8 bit signed	R	F_{ieff}
mmax	Limit value 8 bit, user-defined	0xFF35	8 bit signed	R/W	F_{ieff}
mmax16	Limit value 16 bit, user-defined	0xFF36	16 bit signed	R/W	F_{ieff16}
R.m.s. current					
iOmax	Limit value, user-defined	0xFF2E	7 bit unsigned	R/W	F_{ieff}
i2	(Actual value)^2	0xC01A	16 bit unsigned	R	$F_{ieff} * F_{ieff}$
Torque setpoint					
msoll		0xFE5E	16 bit signed	R	F_{ieff16}

With **mmax** and **mmax16** the peak current can be determined. The values depend on the motor used. The smaller value determines the maximum peak current.

The effective current can be limited with **I0max**. When the r.m.s. current limit is reached, it can be determined with bit 2 in **SwVersion** whether NOVODRIVE generates Error 315 or whether the peak current is to be limited to the r.m.s. current.

Before software version 2.01, the time constant for calculating the effective current is 5 minutes.

Starting from Version 2.01 of 07/01/1998, the time constant can be set between 18 ms and 25,5 s. To activate time constant TCi2, bit 7 in **I0max** must be set. If the bit is not set, the standard value 25,5 s is in effect.

Register	**TCi2**
Address	0xFEEF
Size	8 bit unsigned
Access	R/W
Scaling	$TCi2 = 256 * e^{\frac{-0.1s}{\tau}}$
Value range	18 ms ... 25,5 s
Responding time	$t_a = -\tau * \ln\left(1 - \frac{iOmax^2}{msoll^2}\right)$

Example Choosen t = 10 s, this leads to **TCi2** = 0xFD
I0max = 100 % rated current
If the motor is disabled, the torque setpoint increases up to 200 % of the rated current of the device. After 2,87 s the i^2t-monitoring responds.

3.6.3.3 Speed ranges

Scaling of speed and ramp values

Speed range	Feedback bit 7	Fn	F_{Rampe}
0...6000 r/min	0	0,223517 r/min per increment	6,821 r/min/s per increment
0...18000 r/min	1	0,894068 r/min per increment	27,285 r/min/s per increment

Speed setpoints and actual values

Register	Description	Address	Size	Access	Scaling
nist	Actual speed value	0xFE68	16 bit signed	R	Fn
nsoll	Setpoint after ramp generator	0xFE62	16 bit signed	R	Fn
nsoll2	Setpoint from position controller	0xFE46	16 bit signed	R	Fn
nMax	Limitation of speed setpoint	0xFEE4	16 bit signed	R	Fn

Starting from Version 3.06 the actual speed is monitored. If the limit value is exceeded, Error 705 is generated.

The limit value can be set over **Steuerbits**:

Register	**Steuerbits**	limit value for **nist**
	bit 6 ≠ bit 7	**nMax**
	bit 6 = bit 7	1,5***nMax**

3.6.3.4 Ramp generator

The speed setpoint is passed through a ramp generator. A discrete ramp value can be set for acceleration, for braking and for quick-stop. Ramp value 0 means that the ramp is deactivated.

The ramp values are determined over pointers:

Register	Description	Address	Size	Access	Standard value
?Rampe+	Pointer on ramp value for acceleration	0xFEA8	16 bit unsigned	R/W	0xFEFA (Rampe+)
?Rampe-	Pointer on ramp value for braking	0xFEAA	16 bit unsigned	R/W	0xFEFC (Rampe-)
?SchRampe	Pointer on ramp value for quick-stop	0xFEC4	16 bit unsigned	R/W	0xFEFE (SchRampe)

For the ramp values three registers are available, i.e. various setting combinations are possible. If, for example, all ramp values are to be identical, the three pointers can be set on **Rampe+**. In this case, **Rampe-** and **SchRampe** have no effect.

To switch off a ramp, there is also the option to set the respective pointer on **Null**.

Register	Description	Address	Size	Access	Scaling
Rampe+	Value register	0xFE68	16 bit unsigned	R/W	F_{Rampe}
Rampe-	Value register	0xFEFC	16 bit unsigned	R/W	F_{Rampe}
SchRampe	Value register	0xFEFE	16 bit unsigned	R/W	F_{Rampe}
Null	Zero value	0x0042	16 bit unsigned	-	-

3.6.3.5 Position and revolutions

Register	Size	Access	Resolution			Revolutions	Position (0...360°)
UmdrSoll, Lagesoll	32 bit	R	0,00549 °			16 bit	16 bit
			Address			0xFE56	0xFE54
UmdrIst, LageIst	32 bit	R	0,00549 °			16 bit	16 bit
			Address			0xFE7E	0xFE7C
UmdrIstH	8 bit	R			8 bit		
			Address	0xFF43			

UmdrSoll / **LageSoll** and **UmdrIst** / **LageIst,** respectively, together constitute a signed 32-bit value. In **UmdrIstH** the overflow from the 32-bit position value are being summed.

These registers are not writable. In the position control mode the position setpoint is determined by the pointer parameter **?LageSollExt**.

3.6.3.6 Tracking error

If a position controller is used, tracking error monitoring can be executed. If the limit value for the tracking error is exceeded, Error 700 is generated and the inverter is disabled. Monitoring is active, if **SchleppFehler** has a value < 0x8000 and if the position controller has been activated.

If the position controller is disabled, the tracking error is cleared.

The maximum tracking error value is recorded in **MaxSchleppInc**. To set back the recording, write 0 in **MaxSchleppInc**.

Register	Size	Access	Resolution	Revolutions	Position (0...360°)
VZSchlUmdr, VZSCHlInc Actual tracking error signed	32 bit	R	0,005493°	16 bit	16 bit
			Address	0xFF1E	0xFF1C
SchleppUmdr, SchleppInc Actual tracking error unsigned	32 bit	R	0,005493 °	16 bit	16 bit
			Address	0xFF68	0xFF6A
MaxSchleppInc Recording of max. tracking error value, limited to 1 revolution	16 bit	R/W	0,005493°		16 bit
			Address		0xFF72
SchleppFehler Limit value of tracking error	16 bit	R/W	1,40625 °	8 bit	8 bit
			Address	0xFF2C	

3.6.4 Temperatures

3.6.4.1 Heat-sink temperature

These values are read-only.

The heat-sink temperature is measured at four different positions and is compared with a limit value.
If the heat-sink temperature is higher than the limit value, the inverter generates Error 400 and shuts off.

Heat-sink temperature T0	**Temp0**	Address: 0xFFF8
Heat-sink temperature T1	**Temp1**	Address: 0xFFF9
Heat-sink temperature T2	**Temp2**	Address: 0xFFFA
Heat-sink temperature T3	**Temp3**	Address: 0xFFFB

Size 8 bit, unsigned

Scaling in [°C]

$$t1 = \frac{1}{\dfrac{\ln\left(\dfrac{56{,}32}{\mathrm{Temp0}} - 0{,}1\right)}{3425} + \dfrac{1}{298}} - 272{,}5$$

3.6.4.2 Motor temperature

The actual value of the motor temperature is read-only.
The limit value of the motor temperature may be read and written.

Temperature threshold	**MaxTempMot**	Address: 0xFF01
Motor temperature	**MotTemp**	Address: 0xFF54

Size 8 bit, unsigned

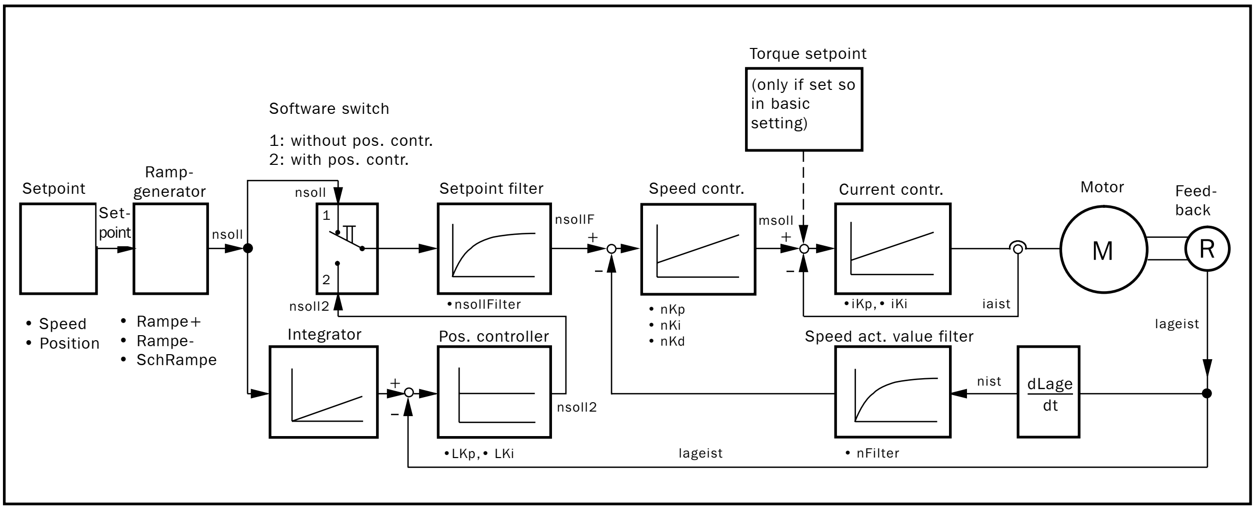
Scaling 1 bit equals 58,75 Ohm

The motor temperature can be calculated from the characteristic of the sensor accommodated in the motor.

If **MaxTempMot** is exceeded, Error 401 is generated.

It is possible to connect break contacts or PTCs, but not NTCs.

3.6.5 Controller structure



- The structure of the controllers and the setpoint setting is partially interconnected by pointers. During start-up the pointers are initialized according to the setting in **Betriebsart** and **Feedback**.
- In case of normal applications, the initialization through **Betriebsart** etc. is sufficient. If changes are to be made during operation (e.g. switching position controllers on and off or changing the setpoint source), this can be done at any time by modification of the pointers. If you do so, be aware that this can be accompanied by setpoint jumps. Therefore the drive must be disabled during such procedures.

⚠ **Do not modify the pointer parameters, if you are not absolutely sure about the impact of your manipulation!**

☞ After a reset the pointer parameters are set back to the standard values set in **Betriebsart** and **Feedback**.

☞ To get an individual setting, select the value „custom-setting“ for **Betriebsart**.

Activation of the position controller is done by switching the internal setpoint pointer **?nsoll** onto the output of the position controller.

⚠ **Although the position controller is not active, a tracking error can occur. If the position controller is activated while the tracking error is unequal zero, it can cause the motor to move suddenly in an uncontrolled way. In order to clear the tracking error, it is necessary to disable the drive before activating.**

Register	**?nsoll**
Address	0xFEB8
Size	16 bit
Access	R/W
Value range	Name Address Notes **nsoll** 0xFE62 Output of ramp generator **nsoll2** 0xFE46 Output of position controller

- **Setpoint**
  Except for the torque control, the setpoint is always a speed-related setpoint. It is set over the pointer parameter **?Sollwert**.

Register	**?Sollwert**
Address	0xFEBE
Size	16 bit
Access	R/W
Function	Pointer on speed setpoint
Value range	Name Address Notes **Sollwert** 0xFE60 digital setpoint **InternSoll** 0xFE4C see additional functions **FISoll** 0xFE1A see fine interpolator **AnInput1** 0xFE5C see analog inputs **AnIn_msoll** 0xFE66 **CANinput1** 0xFE3C see CAN Bus NOVOTRON **CANinput1** 0xFE3E **CANinput1** 0xFE40 **CANinput1** 0xFE42 **Lage-nSoll** 0xFE74 see position differentiator

Register	**Sollwert**
Address	0xFE60
Size	16 bit signed
Access	R/W
Function	Register for digital setpoint setting over NOVOBUS. Can be used for torque, speed and position setting. For the setpoint setting, the register must be written to with NB_WriteWord.
Scaling	see torque, speed and position, respectively

- **Torque setpoint**
  To change into the torque control operating mode, set **nKp** (see speed controller) to zero. Setpoint setting is done over the pointer **?msoll**.

Register	**?msoll**
Adresse	0xFEBC
Size	16 bit
Access	R/W
Function	Pointer on torque setpoint in the torque control mode
Value range	see **?Sollwert**

If the speed control is switched off, the motor can suddenly race off, since there is no more speed or position control active.

- **Position differentiator**
  To determine the setpoint in the position setting mode, a differentiator is used. The differentiator can also be used with a digital encoder to determine the speed.

  In case of a position change which corresponds to a speed of more than 6000 r/min, Error 977 is generated.

Input:

Register	**?LageSollExt**
Address	0xFEA6
Size	16 bit
Access	R/W
Function	Pointer on a 16-bit position value from which the speed is computed
Value range	Name / Address / Notes - / 0x0000 / position differentiator off **_RODin** / 0xC402 / see encoder input **STROD** / 0xFE48 **KurveOut** / 0xFDF6 / see cam disk function see also **?Sollwert** 16-bit position setting

Output:

Register	**Lage-nSoll**
Address	0xFE74
Size	16 bit
Access	R/W
Function	Speed value computed from the position change at input for use see **?Sollwert**

Example Position setting over encoder input:
Setting **Betriebsart** = 100 001 00 (binary) initializes the pointers after the reset as follows:

?LageSollext	= 0xC402	(counter of encoder input)
?Sollwert	= 0xFE74	(**Lage-nSoll**)
?nsoll	= 0xFE46	(**nsoll2**)

3.6.6 Controller parameters

All controller parameters can be modified by the user.

3.6.6.1 Current controller

NOVODRIVE works with a PI current controller. To compensate the counter-EMF of the motor, the parameter **emk0** is offered. Depending on the speed range and device type selected, the scaling of the EMF compensation is:

$$\text{voltage gradient of motor [mV/rpm]} = \text{emk0} * F_{EMK}$$

In case of high speeds, the lagging of the current's actual value behind the setpoint is disadvantageous. This can be compensated with the **ResolvKomp** parameter.

Register	Description	Address	Size	Access	Value range
iKp	P-gain	0xFEF8	8 bit unsigned	R/W	0 … 255
iKi	I-gain	0xFEF9	8 bit unsigned	R/W	0 … 255
emk0	EMF compensation	0xFE03	8 bit unsigned	R/W	0 … 127
ResolvKomp	Current phase pre-control	0xFF02	8 bit unsigned	R/W	0 … 127

Both the EMF compensation and the current phase pre-control depend on the speed range selected:

		Speed range 0 … 6000 rpm	Speed range 0 … 18000 rpm
ND31	F_{EMK}	2,75 mV/rpm	11,0 mV/rpm
ND32	F_{EMK}	4,81 mV/rpm	19,2 mV/rpm
recommended values for the current phase pre-control	ResolvKomp	12	40

3.6.6.2 Filter

Tachometer filter

With the tachometer filter it is possible to filter the actual speed value. The tachometer filter is a first-order filter. The higher the value set, the stronger the filtering.

Tachometer filter **nFilter** Address: 0xFF3F

Size 8 bit, unsigned, range: 0x00 - 0x6F.

Scaling

$$\text{Time constant tacho filter} = \frac{512\ \mu s}{1 - \frac{nFilter}{128}}$$

Setpoint filter

With the setpoint filter it is possible to filter the speed setpoint. The setpoint filter is a first-order filter. The higher the value set, the stronger the filtering.

Setpoint filter **nSollFilter** Address: 0xFEF6

Size 8 bit, unsigned, range: 0x00 - 0x3F.

Scaling

$$\text{Time constant setpoint filter} = \frac{512\ \mu s}{1 - \frac{nSollFilter}{128}}$$

3.6.6.3 Speed controller

NOVODRIVE works with a PI speed controller with pre-control.

P-gain	**nKp**	Address: 0xFF32
I-gain	**nKi**	Address: 0xFF33
pre-control	**nKd**	Address: 0xFF31

Size 8 bit, unsigned, range: 0x00 - 0x7F.

As of H8 Version 3.0

- **nKd** can be used as D-gain.

Activation: **Steuerbits** (0xFEA0) bit 13 ≠ 14

3.6.6.4 Position controller

NOVODRIVE works with a P position controller with pre-control.

P-gain	**LKp**	Address: 0xFF30
pre-control	**LKd**	Address: 0xFF2F

Size 8 bit, unsigned, range: 0x00 - 0x7F.

As of H8 Version 3.0

- **LKd** can be used as D-gain.
Activation: **LKd** bit 7 =1

- **LKp** can operate with a four times stronger gain.
Activation: **LKp** bit 7 = 1

3.6.7 Program pointers

For the use of additional functions several program pointers are offered. The use of the pointers is described in the respective chapters of the „Additional Functions“ manual.

Name	Address	Size	Access	Function
?Init	0xFEA4	16 bit	R/W	One-time activation after reset
?SPS	0xFEA2	16 bit	R/W	For background processes such as Ablaufsteuerung etc.
?512usA	0xFEC8	16 bit	R/W	Calculation of setpoints through table interpolation or fine interpolator
?512us	0xFEAE	16 bit	R/W	General use; activation in any controller cycle
?512usB	0xFECA	16 bit	R/W	

Program pointers that are not used must have the value @Dummy (0x01EA).

If an invalid value is written into a program pointer, Error 802 is generated.

It is possible to load customer specific software extensions into the external memory. Due to the segmentation of the memory a further parameter must be set, if a software extension is used.

Register	**Bank**
Address	0xFF07
Size	8 bit
Access	R/W
Function	selection of a memory bank
Value range	see description of software extension

3.6.8 Drive information

3.6.8.1 Serial number

The serial number can be read out from **Seriennummer**.

Register	**Seriennummer**
Address	0xFE84 ... 0xFE86
Size	24 bit
Access	R
Representation	BCD number, highest-order digit first

3.6.8.2 Operating hours

The operating hours can be read out from the memory. There is one counter for active time (NOVODRIVE enabled) and one counter for passive time (NOVODRIVE disabled).

	Name	Address	Size	Access
active time	**BetriebStd**	0xC02C	16 bit unsigned	R
	BetriebMin	0xC030	8 bit unsigned	R
passive time	**SperreStd**	0xC02E	16 bit unsigned	R
	SperreMin	0xC031	8 bit unsigned	R

3.6.8.3 H8 Version

Register	Software version number
Address	0x0110 ... 0x0117
Size	8 Byte
Access	R
Coding	ASCI

Example:

0x33	0x2E	0x30	0x39	0x00	0x00	0x00	0x00	0x00
3	.	0	9	\0				

Register	Software version date
Address	0x0118
Size	16 bit
Access	R
Coding	Day bit 0..4 Month bit 5..8 Year bit 9..11 - The value 0 means the year 1993.

Example:

0x10FB =	0001000	0111	11011
	2001	7	27

3.6.9 Auto adjustment of commutation angle

Starting with H8 Version of 06/05/1997

For optimal use of NOVODRIVE and motor, NOVODRIVE must know the position of the sensor (e.g. resolver). For this purpose, the **PhiP0** parameter has been conceived. NOVODRIVE can automatically determine the position of the sensor.

During the procedure the motor moves jerkily! Make sure that nobody gets hurt and nothing damaged!

In case of Hall feedback for trapezoidal commutation, the function cannot be used!

Prerequisites

- Motor and sensor are connected to NOVODRIVE.
- NOVODRIVE is connected to 24 V logic supply and 230 V / 400 V mains voltage.
- Inverter is disabled via NOVOBUS and enabled via hardware.
- Motor can move freely (no load connected).
- Rated current of motor is set correctly.
- Current controller is set correctly.
- Number of poles in motor is set correctly,
- Reset.

Automatic resolver setting activation:
Write 0x80 into register **RautojuCSR** (address 0xFE6E).

Automatic resolver setting process:
After the activation, NOVODRIVE enables the inverter and supplies the motor with current so that the motor takes a certain position. NOVODRIVE keeps the motor in this position for 10 seconds. During this time, **RautojuCSR** has the value 0x81. After 10 seconds the procedure is finished, the inverter is disabled, **RautojuCSR** takes the value 0x00.

☞ In the version of 06/05/1997, the function works correctly only if the basic setting is on positive rotation direction.

In case of applications with negative rotation direction, switch into positive direction first, execute the procedure, then set back on negative direction. The error has been eliminated with H8 of 08/14/1997.

Possible errors	Cause and clearing
Error 307 inverter	Set current controller lower
Current controller oscillates	Set current controller lower
Motor does not move as intended or takes a certain (preferred) position	Wrong indication of number of poles in the motor or phase sequence A1, A2, A3 wrong
Error 705	Motor took its position too fast. Reduce tolerable peak current for execution of adjustment or raise limit value for maximum speed

3.6.10 Brake function

☞ From H8 Version 06/05/1997.

For some applications (e.g. vertical axis) it can be reasonable that the stopping brake is activated before the drive is disabled. Also, it is reasonable that the stopping brake is not released until the drive is enabled. NOVODRIVE supports these functions.

The hardware signal for controlling the braking relais is the digital output GPO3 (connector X3, Pin B24).

For programming delay times use registers **BremseT1** and **BremseT2**.

With **BremseT1** you can specify how many 10-ms cycles after enabling the drive are to go by until the brake is released, i.e. until GPO3 switches to 1 (24 V). If you write 0xFF or 0xFE in **BremseT1**, the function is deactivated.

The time delay (in 10-ms cycles) for disable to become effective is programmed in **BremseT2**, GPO3 instantly drops off (0 V).

☞ The function is not in effect until a reset has been made.

Register	**BremseT1**
Address	0xFEB4
Size	8 bit
Access	R/W
Scaling	10 ms / increments 0xFF no brake function 0xFE no brake function

Register	**BremseT2**
Address	0xFEB5
Size	8 bit
Access	R/W
Scaling	10 ms / increments

If the GPIO Manager is used simultaneously, set bit 2 for GPO3 in Register GPOMaske!

3.7 Oscilloscope

3.7.1 Signal selection

NOVODRIVE has a function for recording any register over a definable period of time. The signal selection is done by writing the register address in **scope_signal1** (0xFE04) and **scope_signal2** (0xFE06), respectively. To specify the trigger time, write the trigger source in **scope_trigger** (0xFE08).

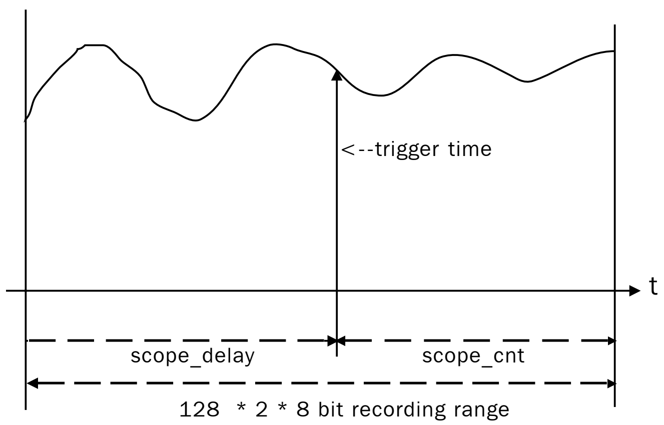
The recording is done with two channels with 8 bit resolution and a length of 128 sampling points. This means that always the 8 most significant bits of a signal are recorded.
If a single signal is to be recorded with 16 bit resolution, it is possible to write the address of the register in **scope_signal1** and the address increased by 1 in **scope_signal2**. After reading out the recording the 16-bit values can be computed as:

$$\text{Value}_{16bit} = (256 * \text{Value Channel 1} + \text{Value Channel 2}) * \text{Scaling}$$

For scaling of the values see description of the single registers. In case of 16-bit registers notice that the scaling value increases by a factor of 256, since only the upper 8 bits are recorded.

Example 1: By setting **scope_signal1** on 0xFE62 (**nsoll**) and **scope_signal2** on 0xFE68 (**nist**) the actual value and the setpoint for speed is recorded. The scaling of the two values is 0,2235*256 = 60,16 rpm/increment.

Example 2: By setting **scope_signal1** on 0xFE68 and **scope_signal2** on 0xFE69 the behavior of the actual speed is recorded with 16 bit exactness. The scaling is 0,2235 rpm/ increment.



3.7.2 Time base

The sampling rate of the storage oscilloscope is written in **scope_timer0** (0xFF7C). This means that always after a certain time T_{AB} a value is being saved.

T_{AB} = 512 μs × s**cope_timer0**

In total, 128 sampling points are saved per channel.

The time base must be set when the recording is disabled.

Example: **scope_timer0** = 1:

Every 512 μs a sampling point saved in the scope buffer. The full scope buffer comprises 128 x 512 μs = 65,536 ms.

scope_timer0 = H' 05:

Every 5 x 512 μs = 2,56 ms a sampling point is saved in the scope buffer. The full scope buffer comprises 128 x 2,56 ms = 327,68 ms.

3.7.3 Trigger threshold

The trigger threshold is written in **scope_level** (0xFF7F) with inverted bit 7.

Example: Trigger threshold **nist** (0xFE68) = 4000 r/min.

4000 / 0.223517 = 17895 = 0x45E7. Of this only the higher byte: 0x45.

Invert bit 7: 0x45 turns into 0xC5

Write 0xC5 in **scope_level**.

3.7.4 Trigger delay

Trigger delay requires programming **scope_delay** (0xFF7D) and **scope_cnt** (0xFE02). **Scope_delay** and **scope_cnt** must be reprogrammed before each recording. The values are computed as follows:

Given is the requested delay time Tvz (see graphic on page 3-51):

scope_cnt = 0x80 + Tvz/T_{AB} (T_{AB} is the sampling time)

for Tvz>=0 **scope_delay** = 0x00

for T_{VZ}<0 **scope_delay** = -T_{VZ} / T_{AB}

Example: Sampling time 1,024 ms (scope_timer0 =2), delay time -20ms.

This means: 19 (20 ms / 1,024 ms = 19,53) sampling points before and 107 sampling points after the trigger time are to be recorded.

scope_delay	= 19	= 0x13
scope_cnt	= 0x80 - 0x13	= 0x6D

3.7.5 Scope status

The register **scope_status** (0xFF7A) controls the recording.

scope_status	R/W	R/W	R/W	R/W	R	R	R	R	
0xFF7A	7	6	5	4	3	2	1	0	Bit

bit 7 Status of oscilloscope
bit 7 = 0 recording active
bit 7 = 1 recording stopped

bit 6 Trigger status
bit 6 = 0 no triggering active
bit 6 = 1 triggering active

bit 5 Trigger ready initialize with 1

bit 4 Trigger edge
bit 4 = 0 negative-edge trigger
bit 4 = 1 positive-edge trigger

bit 3 .. 0 reserved = 0

bit 7 is automatically set on 1 at the end of a recording.

☞ A recording can be aborted by setting bit 7 on 1.

3.7.6 Recording process

- Modifications in the setting may be done only when the recording is not active.
- At the beginning, write the desired values into **scope_delay, scope_cnt, scope _timer0, scope_signal1, scope_signal2, scope_trigger and scope_level**.
- Then, **scope_status** is set to 0x20 in case of a negative-edge trigger and to 0x30 in case of a positive-edge trigger; thereby the recording is started.
- After the delay time has elapsed, the trigger condition is checked.
- Bit 5 takes the value 0, as soon as the trigger signal has fallen below the trigger threshold in case of a positive-edge trigger and as soon as it has exceeded the trigger threshold in case of a negative-edge trigger.
- Bit 5 takes the value 1 again, as soons as the trigger threshold has been crossed again.
- By this the trigger signal is tripped, bit 6 takes the value 1.
- As soon as the triggering has taken place, the time programmed in **scope_cnt** starts.
- After that the recording is stopped automatically and bit 7 in **scope_status** takes the value 1.
- The data recorded are now ready for being read out from the scope memory.

☞ The scope memory is conceived as a 256 byte FIFO memory. The scope memory can be read by register **_FIFODdat**. In order to read out the scope memory completely, 256 times a ReadByte access to **_FIFOdat** must be made.

Register	**_FIFOdat**
Address	0xFF83
Size	8 bit, sign depends on signal
Access	R
Function	FIFO memory of scope data. The sampled values have a size of 8 bit. The values of signal 1 and signal 2 are stored alternately.

3.7.7 Auto Trigger

A recording without triggering can be made by setting bit 7 in **scope_status** on 0 for the time of a recording and then setting it back to 1 again.

3.7.8 Oscilloscope registers overview

Register	Description	Address	Size	Access
scope_Signal1	Address of Signal 1	0xFE04	16 bit unsigned	R/W
scope_Signal2	Address of Signal 2	0xFE06	16 bit unsigned	R/W
scope_trigger	Address of trigger signal	0xFE08	16 bit unsigned	R/W
scope_cnt	see graphic	0xFE02	16 bit unsigned	R/W
scope_delay	see graphic	0xFF7D	8 bit unsigned	R/W
scope_status	see Chapter 3.7.5	0xFF7A	8 bit unsigned	R/W
scope_timer0	Time base	0xFF7C	8 bit unsigned	R/W
sope_level	Trigger threshold	0xFF7F	8 bit unsigned	R/W

3.8 Signal Inputs and Outputs

3.8.1 Digital inputs

The digital inputs of NOVODRIVE can be read in two different ways:

- by reading directly the registers listed in the table,
- by reading over the GPIOManager.

Digital input	Address	bit	X3 Pin	0 V =
GPIN1	0xFFBB	4	A34	„1“
GPIN2	0xFFBF	1	A24	„1“
GPIN3	0xFFBF	0	A27	„1“
GPIN4	0xFFBB	3	A25	„1“
GPIN5	0xC405	7	A26	„1“
GPIN6	0xFFB7	3	A21	„1“
GPIN7	0xFFB7	5	A23	„1“
GPIN8	0xFFBB	0	A32	„1“
GPIN9	0xFFBB	2	A33	„1“
GPIN10	0xFFB7	2	A22	„1“

3.8.2 Digital outputs

The digital outputs of NOVODRIVE can be set in two different ways:

- by writing into the registers listed in the table by means of NOVOBUS or CAN commands NB_OR and NB_AND;
  only the respective bit may be modified, the other bits must remain in their state,
- by writing over the GPIOManager.

Digital output	Address	Bit	X3 Pin	switches after	0 V =
GPO1 (*)	0xFFB7	6	B22	0 V	„1“
GPO2	0xFFB7	4	B21	0 V	„1“
GPO3	0xFFB7	1	B24	24 V	„1“
GPO4	0xFFBB	1	B28	24 V	„1“
GPO5	0xFFC1	2	B27	24 V	„1“
GPO6	0xFFBB	5	B26	24 V	„1“
GPO7	0xFFBB	6	B25	24 V	„1“
GPO8 (*)	0xFF84	0	B23	24 V	„1“
GPO9	0xFFC1	0	B30	24 V	„1“
GPO10	0xFFC1	1	B29	24 V	„1“

☞ (*) In order to be able to use GPO1 and GPO8 as digital outputs, the analog outputs must be switched off.

3.8.3 GPIOManager

Starting with H8 Version of 06/05/1997.

A much more convenient way to read the digital inputs and set the digital outputs is offered by the GPIOManager. After the activation by setting **?512usB** to value **@GPIOManager**, the inputs and outputs are readable and writable by means of the registers **GPIN** and **GPO**. The inputs and outputs are updated every 1 ms.

To prevent a collision with other outputs that have been assigned with certain functions (e.g. brake output), those outputs must be masked by setting the respective bit in the register **GPOMaske**.

Register	**?512usB**
Address	0xFECA
Size	16 bit
Access	R/W
Function	Activation of GPIOManager
Assignment	Name / Address / Notes **@Dummy** 0x01EA GPIOManager off **@GPIOManager** 0x01F4 GPIOManager on

Register	**GPIN**
Address	0xFE0C
Size	16 bit
Access	R
Function	Status of digital inputs
Assignment	bit 0 GPIN1 ... ... bit 9 GPIN10 bit 10 Tast8 ... ... bit 13 Tast5 bit 14...15 not assigned

Register	**GPO**
Address	0xFE0A
Size	16 bit
Access	R/W
Function	Status of digital outputs
Assignment	bit 0 GPIN1 ... ... bit 9 GPIN10 bit 10...15 not assigned

Register	**GPOMaske**
Address	0xFEC7
Size	8 bit
Access	R/W
Function	Masking of digital outputs that may not be set by the GPIOManager, e.g. brake output
Assignment	bit 0 =1 no change of GPIN1 bit 7 = 1 no change of GPIN8

Register	**?GPO**
Address	0xFEC2
Size	16 bit
Access	R
Function	Pointer on output value of digital outputs
Assignment	Name Address Notes **GPO** 0xFE0A standard value

3.8.4 Analog inputs

NOVODRIVE has two analog inputs.

	Analog Input 1		Analog Input 2
Input range	-10 V ... +10 V		-10 V ... +10 V
Resolution	14 bit	12 bit	8 bit
Conversion time	0,8 ms	0,2 ms	8 μs
Connection +	Peripherals X3 Pin A1		Peripherals X3 Pin A3
Connection -	Peripherals X3 Pin A2		Peripherals X3 Pin B16
Notes	Can be disabled with **Steuerbits** bit 4 and 5. The bit width is defined with **Steuerbits** bit 11 and 12.		Measuring is only done, if pointer **?512us** is set to **@AnInput2** (0x01EC).

Register	Name	Address	Size	Name	Address	Size
Measuring value	**AnInput1**	0xFE5C	16 bit	**AnIn_msoll**	0xFF66	16 bit
Offset correction	**AnIn1Offset**	0xFEF2	16 bit	**AnIn2Offset**	0xFF23	8 bit
Scaling (-128...+127)	**AnIn1Fakt**	0xFF3E	8 bit	**AnIn2Fakt**	0xFF22	8 bit

The measured values can be accessed by pointers. The procedure is described in the respective software modules.

Example: Setpoint setting over Analog Input 1

?Sollwert = **AnInput1** (0xFE5C)

☞ Before H8 Version 3.0, the offset correction by register **AnIn1Offset** is only done at reset. As of Version 3.0, the offset correction is updated regularly by register **AnIn1Offset**. All other offset corrections are immediately reflected in the measured values.

In case of Analog Input 1 the input value may overflow, if a high offset value is set and if the input voltage exceeds +10 V or falls below -10 V.

☞ Analog Input 2 can only be used, if the encoder input is not connected, i.e. it may not be used together with encoder or step/direction for position setting, or with a digital encoder or sine encoder as feedback system.

3.8.5 Analog outputs

NOVODRIVE has two analog outputs.

AnOutConfig	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0xFF20	7	6	5	4	3	2	1	0	
	1	1	1	1	1	1	1	0	Analog Output 1 active
	1	1	1	1	1	1	0	1	Analog Output 2 active
	1	1	1	1	1	1	0	0	Analog Output 1 and 2 active
	1	1	1	1	1	0	0	0	16-bit mode

	Analog Output 1	Analog Output 2
Output range	-9 V ... +9 V	-9 V ... +9 V
Max. output current	5 mA	5 mA
Resolution	8 bit	8 bit
Connector	Peripherals X3 Pin A5	Peripherals X3 Pin B3

Register	Name	Address	Size	Name	Address	Size
Pointer on	**?AnOut1** output value	0xFF24	16 bit	**?AnOut2**	0xFF12	16 bit
Offset correction	**AnOutOffs**	0xFF26	8 bit	**AnOut2Offs**	0xFF15	8 bit
Scaling	**AnOutFakt** (-128...+127)	0xFF27	8 bit	**AnOut2Fakt**	0xFF14	8 bit

If the analog outputs are used, GPO8 and GPO1 cannot be used as digital outputs!

☞ A change in **AnOutConfig** requires a reset to become effective. All other parameters need no reset.

Example: Value for actual speed outputted over Analog Output 1, scaling 100%

?AnOut1 = 0xFE68
AnOutFakt = 127
AnOutOffs = 0

16-bit mode
Register **?AnOut1** is set to the output value. The scaling and offset is done by **AnOutFakt** and **AnOutOffs**. The output voltage is outputted at X3 Pin B3.

3.8.6 Encoder input

☞ The configuration of this function is made during the start-up by means of the start-up software. There is no possibility to modify the parameters described here during operation.

NOVODRIVE has an encoder input that can be connected both to digital encoders and sine encoders. Digital encoders can be used to determine either the actual position or the position setpoint. Sine encoders can be interpolated up to 1024 times, if used as position measuring system. The encoder input can also be used for step/direction setting (stepper motor emulation). The functionality is specified with **Betriebsart** and **Feedback** during reset. The maximum input frequency is 2 MHz.

An encoder can be connected either at the sensor connector X2 or at the peripherals connector X3 (for assignment of connectors see Manual „Basic Device“, Chapter 5.2 and 5.3). However, both connections lead to the same input!

If a sine encoder is chosen for feedback, the following pulse numbers are possible:

64, 128, 256, 512, 1024, 2048, 4096

In case of a digital encoder all pulse numbers between 17 and 61680 can be chosen, for which the following condition must apply:

Pulse number * RODinM = 1048576

Register	**RODinM**	
Address	0xFEE2	
Size	16 bit unsigned	
Access	R/W	
Function	Number of pulses of encoder input or of step/direction input	
Scaling	Encoder input	$\text{RODinM} = \dfrac{1048576}{\textit{pulse no.}}$
	Step/direction	$\text{RODinM} = \dfrac{4194304}{\textit{pulse no.}}$

InitIOM1	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0xFF00	7	6	5	4	3	2	1	0

- Bit 0: x — 0= no rounding; 1= rounding of digits after decimal point
- Bit 1: x — 0= no reset; 1= reset of counter at zero pulse
- Bit 2: x — 0= frequency/ direction input; 1= encoder input
- Bit 3: 1 — reserved
- Bit 4: 0 — reserved
- Bit 5: 1 — reserved
- Bit 6: 0 — reserved
- Bit 7: 1 — reserved

☞ The new pulse number requires a reset to become effective.

The counter of the encoder input can be read out directly in the hardware.

Register	**_RODin**
Address	0xC402
Size	16 bit unsigned
Access	R
Scaling	2^{16} / revolution

Starting with H8 Version 3.00 the value can be read out also from the register **STROD**, as long as register **Feedback** does not say sine encoder.

Register	**STROD**
Address	0xFE48
Size	16 bit unsigned
Access	R
Scaling	like **_RODin**

3.8.7 Encoder emulation

☞ The configuration of this feature is made during the start-up by means of the start-up software. There is no possibility to modify the parameters described here during operation.

For operation with resolver feedback, NOVODRIVE offers an encoder emulation. The number of pulses ranges from 1 to 1024 pulses per revolution. The actual value of the resolver is outputted. The zero point is fixed. The maximum output frequency is 156,2 kHz.

Register	**Impulszahl**
Address	0xFEF4
Size	16 bit
Access	R/W
Value range	1 ... 1024 pulses

☞ The new pulse number requires a reset to become effective.

3.9 Feedback Systems

3.9.1 Resolver

The resolver serves to measure position and speed and to commutate the motor. As an option, an additional encoder can be used for position feedback (operation with a second measuring system).
The connection of a resolver is done according to the „Basic Device“ manual, Chapter 5.2. Apart from the motor specifications, the following parameter are needed:

- **Feedback** = xxxx x000 (feedback system = resolver)
- **Pole**
- **PhiPO**

Register	**Pole**
Address	0xFEF7
Size	8 bit
Access	R/W
Function	Number of poles in motor
Value range	2, 4, 6, 8, 10, 12

Register	**PhiPO**
Address	0xFEF0
Size	16 bit unsigned
Access	R/W
Function	Shifting of the resolver position for correct commutation of the motor
Scaling	12 bit / 360°

☞ Both values require a reset to become effective.

First start-up of resolver

The controllers and the limit values for the current must be set quite low. For determination of **PhiPO** use the Autojustage function (see Chapter 3.6.9).

After being switched on, the motor can react in various ways:

1 motor can be stopped and run at low speed,
2 motor jumps to a position or oscillates between two fixed positions,
3 motor runs uncontrolled at high speed.

The motor's behavior determines how to proceed further:

- **1** and **2** sometimes can only be distinguished by the fact that in case of **1** the motor reacts to setpoints whereas in case of **2** it does not.
- In case of **2** exchange two motor cables (e.g. A2 and A3) and repeat Autojustage.
- In case of **3** repeat Autojustage to optimize **PhiPO**.

If the motor principally runs in a controlled way and the limit values have been adjusted to the motor, start with determining the control parameters.

☞ If the motor does not react at all, the controllers and limit values are set too low.

3.9.2 Digital Hall sensors and digital encoder

Speed and position measuring is made by means of a digital encoder. For the commutation of the motor, Hall sensor signals are needed. The connection of encoder and Hall sensors is made according to Chapter 5.2 Position Sensor Connection (X2) in the „Basic Device“ manual. Apart from the motor specifications, the following parameters are needed:

- **Feedback** = xxxx x011 (feedback system = Hall and encoder)
- **Pole** = 2
- **PhiPO**
- **RODinM** (number of pulses of encoder, see Chapter 3.8.6)

Register	**Pole**
Address	0xFEF7
Size	8 bit
Access	R/W
Function	Number of poles in motor
Value range	For Hall sensors always value 2

Register	**PhiPO**
Address	0xFEF0
Size	16 bit unsigned
Access	R/W
Function	Shifting of Hall signals for correct commutation of the motor
Scaling	12 bit / 360°

☞ Both values require a reset to become effective.

First start-up of digital Hall sensors and digital encoder

Since two independent measuring systems are to be put into operation, the parameterization becomes more difficult than for the resolver.

First, the correct function of the encoder should be checked. To do so, check Register **LageIst**. If the motor is slowly turned manually, the register must continuously pass through the range 0 - 0xFFFF or 0 - 360°, respectively. **LageIst** can be displayed under „Monitor“/ “Register“ in the Windows version of the start-up software or over the oscilloscope.
The zero pulse can be displayed under „Motor“/“digitaler Encoder“. When turning the motor, the zero pulse may be active only for a short moment. If it is constantly active, a pole error of the zero pulse must be assumed.

If up to this point the encoder works correctly, the Hall signals can be tested. To do so, check register **_Phi**.

Register	**_Phi**
Address	0xFF88
Size	16 bit
Access	R
Scaling	16 bit / 360°

_Phi indicates the position value from the Hall sensors. In case of digital Hall sensors, **_Phi** moves continuously at 60° intervals in one direction, if the motor is turned manually. Depending on the number of poles in the motor, **_Phi** passes through the range 0 - 0xFFFF or 0 - 360°, respectively, several times within one revolution of the motor shaft.

Additionally, it must be checked now whether **LageIst** and the commutation position **_Phi** have the same counting direction, if the motor is turned manually in one direction. If need be, exchange the Hall signals.
Because of the coarse position resolution of digital Hall sensors for trapezoidal commutation, the adjustment of the commutation angle by means of **PhiPO** cannot be done automatically. The parameter must be determined experimentally. Possible values are 0°, 120° and 240° electrically.

At the beginning, the controllers and the limit values for the current must be set quite low. After being switched on, the motor can react in various ways:

1 motor can be stopped and run at low speed,
2 motor jumps to a position or oscillates between two fixed positions,
3 motor runs uncontrolled at high speed.

The motor's behavior determines how to proceed further:

- **1** and **2** sometimes can only be distinguished by the fact that in case of **1** the motor reacts to setpoints whereas in case of **2** it does not.
- In case of **2** interchange two motor cables (e.g. A2 and A3) .
- In case of **3** modify **PhiPO** until the motor stops.

If the motor runs in a controlled way and the limit values have been adjusted to the motor, start with determining the control parameters.

3.9.3 Sine encoder

This description applies to H8 Version of 02/10/1997 and later.

☞ All modifications of the parameters described here require a reset to become effective.

3.9.3.1 Mode of operation

Sine encoders provide two sinusoidal incremental signals mutually shifted by 90°. In order to roughly measure a position, it is sufficient to count the sine pulses. Theoretically, for the fine measuring the position can have an arbitrarily high resolution. For the incremental signals applies:

$A - /A = U*\sin \varphi$
and $B - /B = U*\cos \varphi$

From this it is possible to compute the position within a pulse:

$$\varphi = \arctan \frac{A - \overline{A}}{B - \overline{B}}$$

NOVODRIVE here performs a 1024-fold interpolation. The complete position can be obtained by combining the pulse value with the computed position within the pulse.

3.9.3.2 Parameterization

The connection of a sine encoder is done according to Chapter 5.2 Position Sensor Connection (X2) of the „Basic Device“ manual. For parameterization of the sine encoder evaluation the following parameters are needed:

- **Feedback** (see Chapter 3.6.1.3)
- **RODInM**
- **Pole**
- **PhiPO**
- **KommSpurOffset**
- **KommLaenge**

3.9.3.3 Possible operation modes with sine encoder

a) **Feedback** = xxxx x111
Sine encoder feedback for commutation, speed measuring and position measuring; the commutation position is initialized from the commutation track.

b) **Feedback** = xxxx x110
Digital or analog Hall signals for commutation;
sine encoder feedback for speed measuring and position measuring.

c) **Feedback** = xxxx x100
Resolver feedback for commutation and speed measuring;
sine encoder feedback for position measuring.

d) **Feedback** = xxxx x101
Resolver feedback for commutation;
sine encoder feedback for speed measuring and position measuring.

3.9.3.4 Number of pulses

Register	**RODinM**	
Address	0xFEE2	
Size	16 bit	
Access	R/W	
Value range	Number of pulses	RODinM
	64	0x4000
	128	0x2000
	256	0x1000
	512	0x0800
	1024	0x0400
	2048	0x0200
	4096	0x0100

For NOVODRIVE the number of pulses selected means one logical motor shaft revolution. A logical revolution is always resolved with 16 bit. A logical motor shaft revolution needs not necessarily comply with a mechanical motor shaft revolution.

Examples Encoder ERN1387 (Heidenhain) with 2048 pulses per revolution.

Example 1: **RODinM** = 0x0200.

During one encoder shaft revolution with 2048 increments, the 16-bit position value passes its entire value range one time. One logical revolution complies with one mechanical motor shaft revolution. 11 bit of the 16-bit position value come from the pulse counter, 5 bit come from the interpolation.

The distance resolution is 1/65536 U.

Example 2: **RODinM** = 0x4000.

Already after 64 increments one logical revolution has taken place. During one mechanical revolution, 2048 / 64 = 32 logical revolutions take place. 6 bit of the 16-bit position value come from the pulse counter, 10 bit come from the interpolation.

The distance resolution is 1/2097152 U.

Position scaling

$$\textit{number of positions per revolutions} = \frac{\left(\text{encoder pulses} / \text{revolution}\right) * 2^{16}}{\text{programmed number of pulses}}$$

Speed scaling (only in case of speed measuring by means of sine encoder)

One increment means:	[r/min]	
Speed range 0...6000 r/min	14648,4375 /	(number of positions per revolution)
Speed range 0...18000 r/min	58593,75 /	(number of positions per revolution)

Example ERN1387: (2048 pulses per revolution)

RODinM	Programmed number of pulses	Speed range	Position scaling	Speed scaling
0x0200	2048	0...6000 r/min	65536 positions per revolution	0,223517417 r/min per inc.
0x0200	2048	0...18000 r/min	65536 positions per revolution	0,894069668 r/min per inc.
0x4000	64	0...6000 r/min	2097152 positions per revolution	0,0069491931 r/min per inc.
0x4000	64	0...18000 r/min	2097152 positions per revolution	0,027939677 r/min per inc.

3.9.3.5 Check of incremental signals

The start-up software allows to view incremental signals by means of the oscilloscope. There are some differences between the DOS version and the Windows version:

- Select the signals **Sinusencoder Kanal A** and **Sinusencoder Kanal B**. In the DOS version the addresses below must be entered. The signals are the scanning values of the pulses. The time base should be set to 5 - 10 ms. If the motor is now slowly turned, the signals are supposed to change. However, the signals may never reach the lower or upper limit. In the DOS version the upper and the lower half of the oscilloscope is displayed the other way round, that is why the signals may never reach the zero line. If they do reach the zero line, the signal amplitude of the sensor is higher than 1Vss, i.e. an error in the sensor must be assumed.

- Select the signals **LageIst** (without zoom) and **LageIst** (plus zoom function). The addresses for the DOS version are 0xFE7C and 0xFE7D. The signal FE7D is the lower-order byte of **LageIst**. As time base select 5 -10 ms again. If the motor is now slowly turned, both signals should change continuously. On 0xFE7D and **LageIst** plus zoom function, respectively, a certain noise may be present.

If both steps lead to a positive result, it can be assumed that the measuring system is connected correctly.

Register	**Sinusencoder Kanal A**
Address	0xFFE2
Size	8 bit
Access	R

Register	**Sinusencoder Kanal B**
Address	0xFFE4
Size	8 bit
Access	R

3.9.3.6 Automatic correction of offset error with sine encoder and analog linear measuring systems

Disable the controller and set **FKSteuerung** to 0. Move the motor back and forth. Then set **FKSteuerung** on 0xFF again. By this the measuring system's errors are captured and compensated automatically.

By this the subdivisional error, e.g. in case of RG2 von Renishaw, can be improved from +- 1,6 μm to +-0,2 μm.

Register	**FKSteuerung**
Address	0xFF17
Size	8 bit
Access	R/W
Assignment	0x00 = offset correction active 0xFF = offset correction finished

3.9.3.7 Parameter for commutation

In total, four parameters are necessary for the commutation.

Register	**Pole**
Address	0xFEF7
Size	8 bit
Access	R/W
Function	Number of poles in motor
Value range	2, 4, 6, 8, 10, 12 for linear motors see Chapter 3.9.5.3

Register	**PhiP0**
Adress	0xFEF0
Size	16 bit unsigned
Access	R/W
Function	Shifting of commutation position relative to zero pulse of the incremental measuring system
Scaling	12 bit = 360°

Register	KommSpurOffset
Address	0xFF18
Size	16 bit
Access	R/W
Function	Shifting of commutation position relative to zero point of the commutation track
Scaling	12 bit = 360°

Register	KommLaenge
Address	0xFF1E
Size	16 bit unsigned
Access	R/W
Function	For computing the commutation position from the incremental position measuring
Scaling	$\textbf{KommLaenge} = 2^{16} * \frac{\text{mechanical revolution}}{\text{logical revolution}}$

Example: ERN1387

RODinM	Programmed number of pulses	Logical revolutions per 1 mechanical revolution	KommLaenge
0x0200	2048	1	32768 = 0x8000
0x4000	64	32	1024 = 0x400

3.9.3.8 Commutation with sine encoder
(only **Feedback** = xxxx x111)

☞ This paragraph applies only to the setting **Feedback** = xxxx x111. For all other settings see operation with resolver and Hall sensors.

A correct commutation of the motor currents requires two measures:

1 initialization of the commutation position after power-on,
2 update of commutation position from the incremental signals.

There are several possibilities to do the initialization:

- use of a commutation track (e.g. in case of ERN1387),
- use of analog Hall sensors as commutation track (in case of linear motors),
- execution of automatic adjustment every time after the supply voltage has been switched on,
- use of the Ablaufsteuerung software and execution of the Auto-komm function after power-on;
- use of the XCAN software and execution of the Autokomm function after power-on.

In case of **Feedback** = xxxx x111, the update of the commutation position is done by means of the incremental signals of the sine encoder, using the parameter **KommLaenge**.

3.9.3.9 Commutation track

Check of commutation track

The commutation track and the incremental track must count in the same direction. To check this, turn the motor manually and check registers **Phi1** and **_Phi**. Both registers must have the same counting direction. If this is not the case, exchange the commutation tracks in the wiring.

Register	**Phi1**
Address	0xFE20
Size	16 bit
Access	R

Register	**_Phi**
Address	0xFF88
Size	16 bit
Access	R

Initialization of commutation position

After power-on, the commutation position is initialized from a commutation track. During this time the motor may not move. Since the commutation track may work inexactly, thereby preventing an optimal commutation, a zero point search function is offered which also improves the commutation position (see manual „Additional Functions“, Chapter 7.1). This function allows to search the zero pulse/point of the measuring system. At this point, both the absolute position and the commutation position is set on a defined value.

Since the commutation track and the zero marker of the linear measuring system are not always placed in a way that they generate the correct phase position of the motor currents, a phase position offset (**PhiPO**) and a commutation track offset (**KommSpurOff**) function is offered.

First start-up
(determination of PhiPO and KommSpurOff)

1 Set **KommSpurOff** to zero.
2 Execute Autojustage (see Chapter 3.6.9).
3 Turn the motor manually to the zero pulse of the measuring system. The zero marker is displayed by the start-up software ND30Cfg on the parameter page for digital encoders. The zero marker can be viewed in the services menu of the DOS version ND31.COM under „I/O-Ports“/“PerChip“/“Inputs bit 4“.
4 Read off **Phi1** and enter this value in **PhiPO**.
5 Reset.
6 Read off **Phi1** again (at zero pulse).
7 **KommSpurOff** = **PhiPO** - **Phi1**. Enter the computed value in **KommSpurOff**.
8 Check: Reset again, then read off **Phi1**. The same value (approximately) as in **4** must be displayed.

If the other parameters, such as current controller, speed controller, number of poles in motor etc. are set correctly, the motor should be ready to run now. If it does not run and jumps to a fixed position, interchange motor connections A2 and A3 and repeat the procedure.

3.9.3.10 Initialization of commutation by means of Auto adjustement or Autokomm function

If no commutation track or Hall sensors are used, the Auto adjustment or Autokomm function must be executed after each power-on. By this the parameter **PhiPO** can be determined. A description of the two functions can be found

- in Chapter 3.6.9 Autojustage,
- in the manual „Additional Functions“, Chapter 16, CAN-Profile, Autokomm,
- in the manual „Additional Functions“, Chapter 15, Ablauf steuerung, Autokomm.

If the Autokomm function is used, bit 2 of register Freigabe0 must always be set to 1. By this the drive remains internally blocked until the function is executed. Moreover, various functions thereby recognize that there are no commutation track or Hall signals.

3.9.4 Use of two measuring systems

3.9.4.1 Connection

☞ Compare with Chapter 5.2 of „Basic Device“ manual.

Resolver and digital encoder / sine encoder at X2

If a resolver or encoder is connected, the pin assignment changes as follows:

PIN	Assignment
1	0 V and temperature sensor
2	Temperature sensor
3	A+
4	A-
5	Rotor R2
6	R+
7	+5 V Up
8	Rotor R1
9	Stator S2
10	Stator S4
11	B+
12	B-
13	R-
14	Stator S3
15	Stator S1

Resolver at X2, digital encoder / sine encoder at X3

Connection of resolver at X2:
see Chapter 5.2 of „Basic Device“ manual.

Connection of encoder at X3:

Assignment	Pins		
	A1	B1	0 V
5 V	A14	B14	
/N encoder input	A16	B15	
N encoder input	A15	B16	
B encoder input	A17	B17	
A encoder input	A18	B18	
/B encoder input	A19	B19	
/A encoder input	A20	B20	

3.9.4.2 Configuration

Commutation and speed control with resolver, position control with digital encoder
Settings:

Feedback	xxxx x000
Polzahl	number of poles in motor
?LageIst	select **LageIst2** (0xFE26) or mode for second measuring system

Commutation and speed control with resolver, position control with sine encoder
Settings:

Feedback	xxxx x100
Polzahl	number of poles in motor
?LageIst	select **LageIst2** (0xFE26) or mode for second measuring system

If the rotation direction of the two measuring systems is not identical, the motor starts spinning in an uncontrolled way. The rotation direction can be adjusted over **Feedback**.

Commutation with resolver, speed control and position control with sine encoder
Settings:

Feedback	xxxx x101
Polzahl	number of poles in motor
?LageIst	**LageIst** (0xFE7C)

Commutation with Hall sensors, speed control and position control with sine encoder
Settings:

Feedback	xxxx x110
Polzahl	2
?LageIst	**LageIst** (0xFE7C)

3.9.5 Synchronous linear motors

In this chapter, only the particular characteristics of linear motors are described. It is recommended that you read the previous chapter first.

3.9.5.1 Mode of operation

Operating linear motors requires the following functions:
- position measuring,
- commutation.

For position measuring, linear measuring systems are available. There are two kinds of linear measuring systems:
- digital encoders,
- sine encoders.

In the following, the analog measuring systems are described in detail. Digital measuring systems can be supported with the feedback settings
- xxxx x011 (encoder + Hall) or
- xxxx x001 (encoder)

Basically, analog linear measuring systems do not differ from sine encoders. The period length of a sine pulse corresponds with a certain distance that depends on the respective measuring system (e.g. Renishaw RG2: 20 μm).

3.9.5.2 Position measuring

The resolution of distance and speed as well as the maximum speed can be manipulated over parameters.

☞ The formulas given here are valid for the speed range of 0...6000 r/min. If the 0...18000 r/min range is selected, (see **Feedback** bit 7), the speed resolution gets four times coarser, yet the maximum speed gets four times higher.

Number of pulses:
Position is internally represented by NOVODRIVE with 32 bits. The 16 upper bits are called „revolutions“, the 16 lower bits are called „increments“. The number of pulses indicates how many sine periods of the linear measuring system are required to increase the „revolutions“ by 1 and to pass the 16-bit position one time. This distance is also called „logical revolution“. By this, the distance resolution, the speed resolution and the maximum speed are specified.

NOVOTRON

Distance resolution:
This variable indicates the distance the motor moves, if **LageIst** changes by 1 increment.

$$\textit{distance res.} = \frac{\textit{(period length measuring system)* numb. pulses}}{65536}$$

Speed resolution:
This variable indicates the lowest settable velocity.

speed resolution [*mm/min*] = 14,64843731 * *distance resolution* [*µm*]

Maximum speed:

maximum speed [*m/min*] = 28,672 * *speed resolution*

Example:

Renishaw RG2:	period length 20 µm
speed range:	0...6000 r/min

Pulses	Distance resol.	Speed resolution	Maximum speed
4096	1,25 µm	18,31054664 mm/min	525 m/min
2048	0,625 µm	9,155273319 mm/min	262,5 m/min
1024	0,3125 µm	4,577636659 mm/min	131,25 m/min
512	0,15625 µm	2,28881833 mm/min	65,625 m/min
256	78,125 nm	1,144409165 mm/min	32,8125 m/min
128	39,0625 nm	0,572204582 mm/min	16,4063 m/min
64	19,53125 nm	0,286102291 mm/min	8,20312 m/min

3.9.5.3 Commutation

For the commutation, NOVODRIVE must be able to determine an initial commutation position after power-on. To do so, NOVODRIVE has two inputs for analog Hall sensors.

If no Hall sensors are used, make sure by means of the external control that the Autokomm-Funktion is started after each power-on before enabling (see Chapter 3.9.3.10.). Otherwise the motor is likely to race off in an uncontrolled way after enabling.

Analog Hall sensors

Hall effect sensors must be mounted at a gap of a ¼ magnet period above the magnets so that they generate sinusoidal output signals. The Hall sensors then provide a sine and a cosine signal per magnet period. The commutation position is computed according to the following formula:

$$\textit{commutation position} = \arctan\frac{C - \overline{C}}{D - \overline{D}} + \textit{commutation track offset}$$

where C and D are the outputs of the Hall sensors.
The connection is described in Chapter 5.2 Position Sensor Connection (X2) of the „Basic Device“ manual.

Commutation settings

Unlike for rotating motors, for linear motors additional parameters must be provided:

- pole distance: distance between north pole and north pole in mm (e.g. 61,4 mm)
- pulse length: period length of measuring system in μm (e.g. Renishaw RG2: 20 μm)

With these two values and the number of poles in the motor, the start-up software computes the internal parameter **KommLaenge**. The formula for computing **KommLaenge** now looks a little different than in Chapter 3.9.3.7:

$$\textit{KommLaenge} = 2^{15} * \frac{\text{(distance for one logical revolution [mm])}}{\frac{\text{poles}}{2} * \text{(north-north distance of magnets [mm])}}$$

☞ Normally, the number of poles in linear motors is 2. If thereby no valid value of **KommLaenge** can be computed and if no Hall sensors are used, the number of poles can be raised to 4 or 8.

For the motor to be able to work correctly in connection with Hall sensors, the following two parameters must be determined:

- **PhiPO**
- **KommSpuroff**

If no Hall sensors are used, both values can be set to 0.

3.9.5.4 First start-up

The start-up is done according to the same guidelines that are used for rotating motors. The controller parameters are determined exactly in the same way. In contrast to a rotating motor, one main difficulty of a linear motor is that its movement is limited by the end of the track on which it runs. Another difficulty lies in the fact that there are three systems that all have a certain direction: the motor, the linear measuring system and the commutation track. Make sure all three systems have the same direction!

The checking of the linear measuring system and the optional commutation tracks and Hall sensors, respectively, is done as described in Chapters 3.9.3.5 and 3.9.3.9.

For enabling NOVODRIVE for the first time, we recommend the following:

- At the beginning, set **Kommspuroffset** to 0.
- Set peak current limitation on a low value.
- Deactivate the position controller.

After being switched on, the motor may race off! Provide suitable stoppers at the ends!

After being switched on, the motor can react in various ways:

1 motor can be stopped and run at low speed,
2 motor jumps to a position or oscillates between two fixed positions,
3 motor runs uncontrolled at high speed.

The motor’s behavior determines how to proceed further:

- **1** and **2** sometimes can only be distinguished by the fact that in case of **1** the motor reacts to setpoints whereas in case of **2** it does not.
- In case of **2** exchange two motor cables (e.g. A2 and A3).
- In case of **3** change **PhiPO** until the motor stops in a controlled way. Then optimize **PhiPO** by setting the address 0xFE6E on 0x80 while the controller is disabled. The motor takes a certain position, and **PhiPO** is determined automatically (see also resolver auto adjustment).

By this the commutation track has been adjusted. If the motor has a zero marker, **KommSpurOff** must be adjusted as in Chapter 3.9.3.9.

After that the controller parameters can be determined.

3.9.5.5 Example: Linear motor without commutation track, operation with analog setpoint and pulse feedback

For this application, the Ablaufsteuerung is used. After the drive has been switched on, start the Autokomm function by means of an external control, then switch to setpoint setting over analog input.

The pulse multiplication of NOVODRIVE cannot be outputted over the encoder emulation. Despite of this it can be advantageous to use a sensor with sine increments to provide the speed controller with a high-resolution actual speed value. For the pulse feedback, the original signals of the measuring system are used. The signals connected to X2 can be also accessed directly at X3:

Meaning	X2	X3
Incremental signal A	3	A18
Incremental signal /A	4	A20
Incremental signal B	11	A17
Incremental signal /B	12	A19
Zero pulse N	6	A15
Zero pulse /N	13	A16

This requires, of course, that the control is able to process the 1Vss signals of the measuring system. Normally, this is possible, but clarify this point with the control manufacturer.

The settings of NOVODRIVE for the measuring system are made in the „General Settings“ menu:

Feedback:	sine encoder
Number of pulses:	depending on the application (see 3.9.5.2)
Motor:	linear motor

Commutation settings:

Pole distance:	north-north distance of the magnets in mm
Pulse length:	length of a complete pulse of the measuring system in μm (e.g. Renishaw RG2H 20 μm)

Motor test operation

MS-DOS based applications require the programs ND31.COM and ND31ABL.EXE. Applications based on Windows 2000 or Windows XP require the start-up software ND30Cfg, which contains all necessary features.

Step	DOS Software ND31.COM	Windows Software ND30Cfg
During the first testing period, the Ablaufsteuerung should be deactivated.	From the menu „Service“ [#], select register **?SPS** by means of the „WriteRAM“ [W] function and enter 0x01EA.	Under „Software“ / „Ablaufsteuerung“ / „Installation“, deselect „Activate“.
Select test mode (motor reversion). It is assumed here that basic settings, such as limit values, limit switches, feedback etc. have already been made.	From the menu „General Settings“ / „Command Service“, set „Speed Regulator“ and „Test Operation“. From the menu „General Settings“ / „Start Condition“, select „also Novobus enable“.	Under „General“ / „Operation Modes“, select „Speed Control“ and „Digital“. Click Reset in the tool bar. Under „Operational Test“ / „Speed Command“, click „Reversing“. Disable and stop the drive over the tool bar.
Supply mains voltage.		
Auto adjustment of the commutation position (see also Chapter 14.12 Resolver Auto adjustment).	From the menu „Service“ / „WriteRAM“, select register **RautojuCSR** (0xFE6E) and enter 0x80. If the hardware enabling (24 V) is at X3, the motor is supplied with direct current and takes a certain position. This process takes 10 seconds. After that the controller is disabled.	Under „Motor“ / „Basic Settings“ activate „Auto Adjustment“.
Enabling of inverter. The motor may not move. If it does move and takes a certain position, exchange two motor phases (Caution: switch off 230 V and wait for discharge of DC link). After that repeat the whole procedure.	Enabling of inverter by means of „Go“ command in the menu „Command“ or „Service“.	Enabling of inverter by means of „Enable“ and „Start“ in the tool bar.
If the motor has not moved:	In the menu „Command“, enter lower setpoints at the beginning, later on enter higher setpoints.	Under „Operational Test“ / „Speed Command“, set lower setpoints at the beginning, later on enter higher setpoints.

The motor should follow the setpoints. If it does not, these might be possible reasons:

- no hardware enabling,
- no software enabling,
- start (GPIN 3) not connected to 24 V,
- wrong motor lead assignment
  -> exchange two phases and start at the beginning again,
- current limitation set too low,
- wrong commutation settings
  -> correct settings and start at the beginning again,
- current controller or speed controller are set wrong.

Error 307 (Short cuircuit):

- current controller set too high
  -> decrease current controller settings, clear error and start at the beginning again.

Error 978 (Motor cable error):

- current controller set too low,
- no mains supply,
- motor lead interrupted,
- motor lead connected wrong,
- interruption in motor winding.

If controllers oscillate:

- „Singing“ noises indicate current controller error
  -> view setpoint of current (**Iasoll**) and actual value of current (**Iaist**) with oscilloscope of start-up software with a 5 ms time base,
- if oscillations occcur at **Iaist** that are not caused by **Iasoll**, the current controller oscillates
  -> set current controller lower,
- if the oscillations also occur at **Iasoll**, the speed controller oscillates
  -> set speed controller lower.

☞ If you have managed to proceed in the process up to this point, you have made sure that motor and sensor wiring are correct and the motor can be controlled. You may now try to optimize the control (see „Start-up“ manual).

Offset setting of analog input

In the menu Grundeinstellungen / Drehzahlregler, select Analog Input 1. The control must supply 0 V. In the menu „Sollwert“, activate offset correction.

On the page IO /Analogeingang 1, activate Analog Input. The control must supply 0 V. Start offset correction.

Connection with control

You may now put the controller into operation together with the control. To do so, you need the NOVODRIVE Ablaufsteuerung.

Requirement: NOVODRIVE is equipped with 128 kByte memory capacity.

The NOVODRIVE Ablaufsteuerung allows the PLC to control NOVODRIVE by means of digital signals. use the IPE.NA3 Ablauf program.

☞ **For a description of the Ablaufsteuerung see „Additional Functions“ manual, Chapter 15.**

Description of the IPE.NA3 program

Set	Function	Activation	Complete message
0	Autokomm (determination of commutat. position)	GPIN1 = 0 V GPIN2 = 24 V	GPO7 = 24 V
1	Analog speed setting Gain 32 %	GPIN1 = 0 V GPIN2 = 0 V	GPO4 = 24 V
2	Error clearing	GPIN1 = 24 V	GPO4 = 24 V

After power-on, first select Autokomm.

Abortion of a function is signaled with GPO10 = 24 V.

A function is selected by

1 connecting the necessary signals to GPIN

and

2 switching GPIN3 (Start) from 0 V to 24 V.

GPIN3 (Start) must be supplied during the whole time the function is active (Autokomm or Error clearing) or the function is desired to stay active (analog speed setting).

Installation of the Abaufsteuerung and loading of IPE.NA3

Step	DOS software ND31ABL.EXE	Windows software ND30Cfg
Installing the Ablaufsteuerung.	Start ND31ABL. Enter COM interface at which NOVODRIVE is connected to the PC. Select „Install“ by means of the arrow key. SPSPOS.HEX is installed on NOVODRIVE.	Under „Software“ / „Ablauf-steuerung“ / „Installation“, install and activate the Interpreter.
Loading the Ablauf program.	Select „Laden“ by means of the arrow key. Indicate IPE.NA3. Select „Download“ by means of arrow key. IPE.NA3 is downloaded to NOVODRIVE.	Under „Software“ / „Ablauf-steuerung“ / „Installation“, select and load the IPE.NA3 program.

GPIN	Port	Bit	X3 Pin	24 V =
1	6	4	A34	0
2	8	1	A24	0
3 (Start)	8	0	A27	0
5 (Enable)	Perchip 3105 Input	7	A26	0

If you use the Windows software ND30Cfg, you may view the status of the inputs and outputs on the „IO“/“Digitale IO“ page.

Optimization by means of the control

In the menu „Sollwert“ you may determine the amplification of the analog input, provided the analog speed setting function is active. The amplification value that has been determined to be optimal must be entered in the IPE.NA3 Ablaufprogramm, and IPE.NA3 must be loaded again on NOVODRIVE.

In the menu „Drehzahlregler“ the parameters of the PI speed controller can be optimized.

☞ **Some tips:**

- In register nFilter, a change from 0 to 1 has a big effect. This is due to the fact that without filter the P-gain of the speed controller is computed every 102,4 μs, but with filtering only every 512 μs.

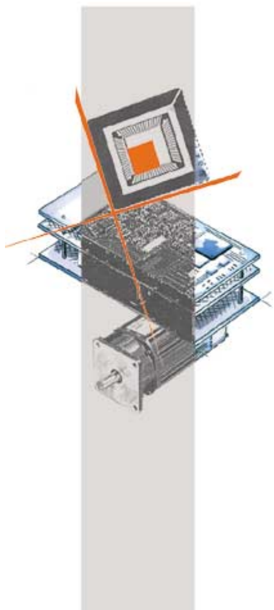
- Effects: If an actual speed value is computed every 102,4 μs, this value is not as highly resolved as with 512 μs. The controller will be more noisy, yet it also gets substantially more dynamic. The P-gain can then be set higher.

- In case of analog setpoints of position controllers that must cope with a poor position resolution (setpoint becomes staircase), it may be reasonable to use the setpoint filter in the NOVODRIVE. For this purpose, the parameter **nSollFilter** is offered. In the DOS software, this parameter can be modified only in the menu „Service“ by means of the „WriteRAM“ function. 0x00 means no filter, 0x3F means maximum. You may set all values that lie in between.

- Control the currents.
  (with oscilloscope function Channel 1 and 2: **Iaist** and **Iasoll**, time base 10 ms).
  The currents must comply with each other as much as possible, without causing **Iaist** to start to oscillate. Set current controller accordingly.

- Control setpoint and actual value of speed.
  (with oscilloscope function Channel 1 **nsoll**, Channel 2 **nist**, time base 100 ms or similar).
  Optimization of compliance of the two signals by means of speed controller:
  - pre-control
  - P-gain
  - I-gain

- If only poor compliance of the actual values and the setpoints can be accomplished (mainly in case of acceleration processes), control **msoll** by means of the oscilloscope function. A horizontal, flat line of **msoll** means that the controller works close to the current limit. The requested acceleration is too high.



NOVOTRON

für Dynamik und Bewegung

N O V O T R O N

Industrie - Automation GmbH
Mauserstrasse 31
D - 71640 Ludwigsburg
Telefon 07141/2969 - 0
Telefax 07141/2969 - 22
e-mail: info@novotron-online.com
http: //www.novotron-online.com