NOVODRIVE ND21

User Manual ND21
Edition: January 1996

We are interested in improving our manuals and to meet our customers requirements. If you have ideas or suggestions to improve this documentation please do not hesitate to contact us.

Novotron GmbH
Mauserstr. 31
71640 Ludwisburg
Germany

Telephone +49-71 41-29 69-0
Fax +49-71 41-29 69-22

INDEX

1. General

1.1. About this Documentation

The documentation for your Novodrive is divided in 3 sections:

- User Manual ND21.

This section is for users who are project managers, designers and software developers. Here you will find the information needed for designing a system using the Novodrive.

- Setup and Parameter Setting of ND21.

This section is for developer and technician who must perform the setup and configuration of the ND21. It will be sent on request by ND21-delivery.

- Instructions for Installation and Replacement.

These instructions are for technicians and electricians who must install or replace the ND21. This flyer will be enclosed by ND21-delivery.

The symbols below will be used in this manual. These symbols will assist you in quickly finding important information.

The commonly used danger symbol, identifies text passages that must, by all means, be read and clearly understood! Not observing these passages could endanger the life and health of yourself and others.

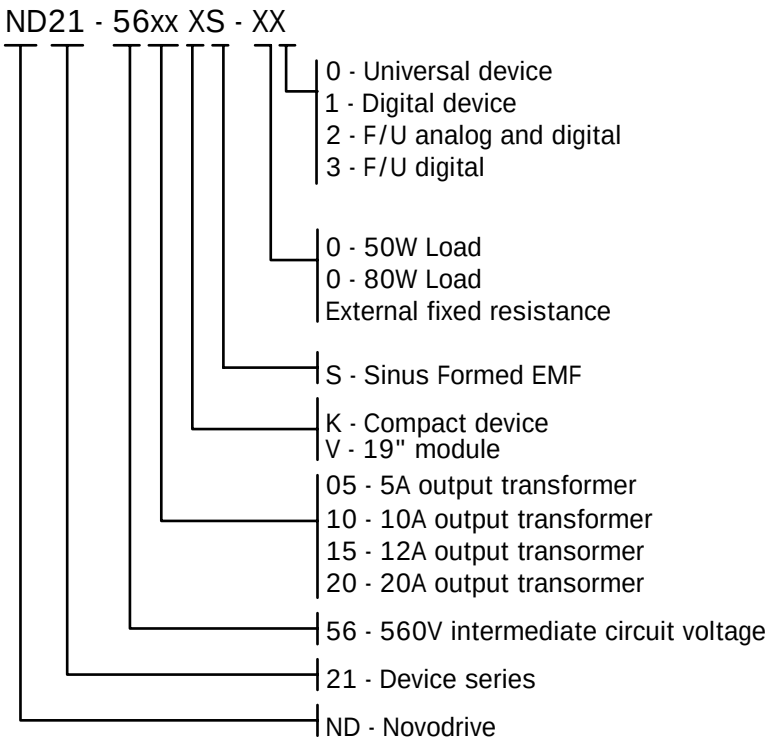
ATTENTION!

Text passages marked with "ATTENTION" must, by all means, be read and clearly understood! Ignoring these passages could lead to the destruction or damaging of the Novodrive or the machine in which it is installed.

1.2. Customer Service

NOVOTRON GmbH
Mauserstraße 31
D - 71640 Ludwigsburg
Telephone: + 49-71 41-29 69-0
Fax +49-71 41-29 69-22

1.3. Designation



Configurations:

ND21-5605KS-XX - Compact device with housing
ND21-5610VS-XX - 19” Model

Output transformers:
05 A Device
10 A device

1.4. Rights

IBM is a registered trademark of the IBM-Corporation.

2. General Safety Remarks

Ther are operating voltages in the ND21 that can be fatal!

Wiring Therefore, check the wiring of the ND21 before switching it on. Ensure that all plugs are properly inserted and grounding has been properly performed.

Saftey Components Ensure that no voltage carrying parts can be accidentally touched and all safety components of the ND21 are present and properly connected.

Emergency Power-Off Provide an “Emergency Power Off” switch so the motor can be switched off at anytime.

Electrolytic Capacitor After power-off the electrolytic capacitor takes approximately one minute to unload! This means: One minute after power-off, fatal voltages are still present in the device. During this time period nothing is to be touched.

Voltage In case the motor continues to turn after the power supply has be switched off, the fatal voltage can be prolonged until a total stop occurs. Only then the unloading of the electrolytic capacitor begins.

On and Off Switching Avoid excessive, repeated and rapid on and off switching of the power supply, this could overload the end switching current limitor of the ND21. This overloading can lead to the destruction of the end switch limit resistance.

2.1. Proper Usage

Novodrive ND21 The frequency converter Novodrive ND21 is a pulse and frequency converter for controlling brushless servo and asynchronous motors. It is a state of the art converter. Using the converter for an application other than that described here, can be damaging to the health of the user or others. Also, the converter the drive or other valuable items can be damaged.

Only use the converter when error free conditions are provided. Always regard safety rules and regulations. Be aware of the manuals and abide the warnings provided.

Compatible Motors Use only brushless servo motors and asynchronous motors with technical data compatible to the converter and its specifications.

Regulations Install the converter only in conformance to the local specifications, standards and regulations.

Operating Environment Do not operate the converter in areas with danger of explosions or in range of medical devices.

Exceptions: The converter is encased in a housing designed, tested and specified for operation in these applications.

2.2. Organizational Measures

Saftey Measures As manufacturer and distributor of a machine in which this converter is used, you are responsible for ensuring that all accident prevention and safety measures have been taken.

Qualified Stuff Ensure that installation and maintenance is only performed by a qualified electrician.

Ensure that the setup is only performed by trained personnel. During installation, the safety warnings in this manual are to be observed and obeyed.

Manuals The designer or developer of a machine in which the converter is installed has read and understood the warnings in the manuals.

Transport and Storage For the transport and storage of the converter the original packing has to be used.

2.3. Safety Units

Emergency Power-Off Machines with moveable parts which are hazardous to people or the machine, must be equipped with a emergency power-off. Install the EPO as **described in chapter 6. Installation**.

2.4. High Voltage

The converter works with hazardous high voltage. Read and abide to the following points.

Ensure that no parts carrying voltage can be accidentally touched.

- Install fuses **as described in chapter 6.2. Fuselage**.
- Emergency Power-off-Installation **as described in chapter 6.4. Emergency-Power-Off Concept**.
- Ensure that proper grounding is provided.
- Make all connections **in accordance with chapter 5. Pinning**.
- Do no disassemble the device. Do not make any changes to the device. Repairs may only be made by the manufacturer.

During setup, abide to all safety regulations and ensure safety features are provided.

2.5. Contact Protection Compact Device

Requirements: Protect against dangerous body current (Draft DIN VDE 160 section 5.5).

The following measurements are necessary to fulfill the requirements above.

- Pulling or inserting the plug of the ND21 is to be non-permissive, when the device is under voltage. Power is only to be applied to the ND21 when all connections screwed to the ND21 are protected against accidental slipping. Working on the plugs of the ND21 is not to be performed when the device is under voltage. All connection are only to be made by a qualified electrician.
- Opening the housing is not allowed!
- Before the power supply is switched on for the first time, ensure all cables have been checked to confirm proper isolation of all wires.
- This is to be repeated at each maintenance interval. Loose clamps are to be retightened. Ensure cables to moveable components, connected to the ND21, are relieved of tension.

  Protect by isolating active components (Draft DIN VDE 0160 section 5.5.1.1.a).

- A minimum of basic isolation protects the active components from the metal housing. The housings must be grounded! A grounding screw for this purpose is provided on the metal housing.

- A wire with a diameter >= 10 mm^2 CU is required for grounding.

Periphery Plug X3

Safe isolation of all signals of the periphery plug X3 from active components is already provided in the ND21 by double isolation **(Draft DIN VDE 160 section 5.6.3.2).** The cable housing for the plug X3 must have a non-conducting surface.

Resolver Plug X2

For safe isolation of all return signals from the active components of the ND21 it is required that all return messages have at least a basic isolation for a measured voltage of 300 Veff. The cable housing for the plug X2 must have a non-conducting surface. Together with the basic isolation for the return signal against active components, a double isolation conforming to **DIN VDE 0160 section 5.6.3.2** is provided.

Power Connection X1

The connection cable and wiring must have a doubled or reinforced isolating between the wires and the surface **(Draft DIN VDE 160 6.5.1.1).**

The connection area is to be protected against accidental contact. Before the power supply is switched on, all cables are to be checked and tested for proper isolation.

Clamps for the respective wire sizes are to be used on the ends of wires.

Bus Connections X4 and X5

The signals of the bus are already separated from active components of the ND21 by double isolation. The housing of the connection cable must have a non-conducting surface.

2.6. 19" Drawer Devices

Requirements: Protection against dangerous body current (Draft DIN VDE 160 section 5.5).

- Pulling or inserting the plug connections of the ND21 is non-permissive when the device is under current. Power is only to be connected to the ND21 when all plugs are protected against slipping by screwing them to the 19" frame or the ND21. Working on the plugs of the ND21 or the 19" frame is not permitted as long as the ND21 is under current. Working on the connections is only to be performed by a qualified electrician.

- The removal of the ND21 from the 19" frame is only to be performed in a power down situation and from a trained electrician. Before the ND21 is first turned on, a check is to be made for proper seating, and all cables are to be checked for proper isolation. This is to be repeated at each maintenance interval of the machine. Loosened screws are to be re-tightened.

- The connection area on the back panel of the 19" frame must be protected against accidental contact. Cables leading to movable components in the machine must be protected against being pulled out of the 19" frame.

Requirements: Protection by isolating active components (Draft DIN VDE 160 section 5.5.1.1a).

- A minimum of basis isolation of the active components against the metal housing is provided. The 19" housing must be grounded. Grounding is performed by using the screw provided on the housing of the 19" drawer.

- A wire with a diameter >= 10 mm^2 CU is required for grounding.

- The 19" frame, the side post of the frame, the cover plate, the ventilator draw and the front panel are separated from the active components of the ND21 by double isolation.

Peripherie plug X3

(See chapter 2.5. Contact Protection Compact Device)

Resolver plug X2

(See chapter 2.5. Contact Protection Compact Device)

Power plug X1

(See chapter 2.5. Contact Protection Compact Device)

Bus Connection X4 und X5

(See chapter 2.5. Contact Protection Compact Device)

3. Function Description

3.1. General

Servo Converter

The ND21 is a servo converter for permanently active synchronized servo motors.

Inputs and Outputs

The ND21 is equipped with all the usual control and signal inputs and outputs such as, enable regulator, ready signal, end switch etc.

Setting Up

Setting up the ND21 is performed with a laptop PC. The memory oscilloscope function built into the ND21 offers the possibility, to display all nominal and actual value courses of the current, torque, speed and rotor position on the laptop monitor without excessive measuring techniques.

This makes adjusting the ND21 very easy, since the effects of parameter change can immediately be evaluated. Parameters are stored in the ND21 and saved on a disc. Parameter lines can be repeatedly copied to other ND21s.

NOVOBUS

Drives can be connected with a controller via Novobus. A ring-shaped wiring is required. Hardware components are not required. The transmission medium is the standarized, serial interface RS232 or RS485.

Positioning Controller

As an addition, Novotron offers a single axis positioning controller for the ND21. Herewith a single cycle control can be programmed in the ND21.

3.2. Power Supply

Circuit Connector

ND21 is equipped with all components required for a positioning axis. This includes a power supply for directly connecting the ND21 to the 400 AC circuit. Protective switches limit the jolt of switching-on current and protect the ND21 from circuit over voltage and transients **(VDE 0160).** A switching circuit is integrated for the internal power supply.

DC Circuit

The braking energy of the motor is absorbed in an intermediate DC circuit. If the capacity of the intermediate circuit condensator is insufficient, the built-in brake chopper becomes active to prevent a too high of an increase in the intermediate circuit voltage.

When inserting several ND21 it is additionally possible to couple the intermediate circuits. Therefore the energy of the brake can be devided in several intermediate capacitors. The

intermediate circuit of ND21 must not be connected parallel to the intermediate circuit of other manufacturers.

Ballast Switching The internal ballast switching is checked electronically. On overload the ballast switching stops and ND21 changes to overload.

Circuit-Breaker A IGBT converter provides the motor with current. Hereby each switch of the converter is individually short circuit proof. All power circuits are isolated from the regulator.

3.3. Automatic Control Part

Position The position, speed and voltage regulation in the ND21 is performed digitally. Even the pulse width modulator of the ND21 works digitally. The voltage regulation and the pulse width modulator are integrated in Novotron ASIC NOVOCHIP, developed especially for the ND21.

Feedback The remainder of the technical regulator functions are performed by a Hitachi H8 micro controller. ND21 needs a resolver for its feedback message system in the motor. The ND21 can also be used for analyzing multi turn resolvers. The resolver or digital conversion is performed by the analogue devices chip 2S82.

The resolution of the rotor position measuring is at low speeds 16 bit, at medium speeds 14 bit, and at high speeds 12 bit. Switching during operation is performed dynamically.

Nominal Values The following choice of possibilities are available for the nominal value input:

- Analogue +/- 10V
- Analogue +/- 24V
- Frequency and direction impulse as well as digital nominal value input via the no cost sensor actor bus (NOVOBUS).
- Internal positioning control.

4. Technical Data

4.1. Nomenclatures

Ratings

Over Voltage Stability	IEC 801-4 Class 2
Radio Interference	EN55011 Class A is upheld by using the Schaffner power filter FN 351-8-29.
Isolation Concept	Two isolation strips: Power switching logic, logic periphery coupling and logic bus coupling. **Note:** The resolver signals are on logical potential and have a basis-isolation against the network. To double or fortify the isolation against the network an isolated cable is required **(see chapter 2. General Safety Remarks)**.
Isolation Concept	2250 VDC
Protection Type	ND21 56XX VS IP00 (Protection Type is determined by 19“frame) ND21 56XX KS IP20 to DIN 40050 and IEC144
Serial interface	RS232: ANSI/EIA232D RS485: EIA485

Signal Processing

Modulation of the output transformer	Digital pulse width modulator
Modulation procedure	Modified sinus-triangle procedure
Current regulator	Digital PI-regulator cycle time 54µs
Resolution of motor current measurement	11 bit
Resolution of rotary field build up	11 bit
Speed regulator	Digital PD-regulator with velocity advanced control, cycle time 432 µs, Resolution 16 bit, Anti-Wind-Up-Switch

Position controller	Digital P-controller with velocity advanced control, cycle time 432 µs, resolution 16 bit.
Position controlling	Resolution 4 Byte, 40 programmable positionsin ND21, Linear approach and braking ramp.
Position relay system	Resolver: Dynamic conformity to the resolution. 16 Bit/rotation at low speeds until approx. 700 rotation. 14 Bit/rotation at medium speeds until 700-2700 rotation. 12 Bit/rotation at speeds higher 2700 rotations
Absolute path measuring system	Absolute resolver supports the system IMAS from the company Baumer Electric.
Motor temperature probe	Selection: Opener, NTC, PTC

Communication

Communication protocol	NOVOBUS
Analogue interface	Voltage (±10 V) Voltage (±20 mA)
Frequence / direction-interface	Stepping motor emulation with a maximum frequenz of 150kHz.
PLC-Connection	**(see chapter 13.3.3. ND21 and Controller and PLC**).

4.2. Electrical Data

4.2.1. Power Supply Connection

Device type ND21-	5605	5610	5615	5620
Connecting rated voltage	400	400	400	400 VAC
Voltage range for connection	150 ... 450 VAC			
Connecting value for rated current	3,5	7,5	10	13,8 kVA
Max. switching current (internally limited)	9	9	9	36 A
Security	3 x 10 A inert	3 x 20 A inert	3 x 20 A inert	3 x 40 A inert
Internal power supply	Built-in switch power supply			
Internal current capacity	20 W			

4.2.2. Intermediate Circuit and End Amplifier

Device type ND21-	5605	5610	5615	5620
DC rail voltage with voltage connection	560 V			
Shut-off threshold with over-voltage	800 V			
Shut-off threshold with under-voltage	200 V			
Intermediate circuit capacity	300	300	300	1200 µF
Power-loss in the output transformer at rated current in	50	100	100	200 W
Remaining voltage-loss with rated current	4 V			
End amplifier tact frequency	9,26	9,26/4,6	4,6	4,6 kHz
Motor tact frequency	18,5	18,5/9,2	9,2	9,2 kHz
Rated current Tu = 25°C	5	10	15	20 Aeff
Peak current Tk = 25°C	10	20	20	40 Aeff

Maximum output current in relation to the cooler temperature:

Temp	25	35	45	55	65	75	85	°C
5605	10	8,7	8,2	7,1	6,1	5,3	4,4	Aeff
5610	20	17,4	16,2	14,1	12,3	10,6	8,8	Aeff
5615	20	20	20	20	18,4	15,9	13.2	Aeff
5620	40	34,8	32,4	28,2	24,6	21,2	17,6	Aeff

As long as the cooler temperature remains at the given value, these currents are valid without time limitation.

By good external venting and an environment temperature of 40°C and 10 A_{eff} motor current, the cooler reaches a temperature (with ND21 5610) of approx. 75°C.

4.2.3. Ballast Switching

Continuous duty loss	Internal (ballast switching) External (option)	50/100 W 5 kW
Pulse power ballast switching		15,5 kW
Switching threshold (Threshold automatik)		725 - 750 V

4.2.4. Ventilation Compact Device

Ventilation	External ventilation (built-in)
Connection	Voltage connection: 220 V_{AC} Current capacity: < 200 mA
Fuselage	315 mA inert

4.2.5. Usable Connectors

Connection for power supply X1 5605,5610,5615, 5620	Phönix Combicon Front-GMSTB 2,5/12-STF PC4/11-ST-7,62
Resolver connection and motor temperature probe X2	High Density D-SUB 15pol (on the ND 21: socket contacts)
Periphery connection X3	High Density D-SUB 44pol (on the ND21: socket contacts)
Bus output connection X4	D-SUB 9pol (on the ND21: pin contacts)
Bus input connection X5	D-SUB 9pol (on the ND21: sochet contacts)

4.2.6. Resolver

Resolver:	Sagem: 21RX360407, 15RX310107
	Litton: JSSBH-15 E-5, JSSBH-21-P4
	Siemen: V23401-H2001-B202
	Tamagawa: TS2018N321 E52, TS2112N21 E11

4.3. Mechanical Data

4.3.1. Measurements

19"Module

5605 u 5610	229 mm x 100 mm x 71 mm
5615 u 5620	229 mm x 100 mm x 142 mm

Compact Device

5605 u. 5610	327 mm x 180 mm x 86 mm
5615 u. 5620	327 mm x 180 mm x 170 mm

4.3.2. Weight

19"Module

5605 u. 5610	1,15 kg
5615	2 kg
5620	2,8 kg

Compact Device

5605 und 5610	3,6 kg
5615	5 kg
5620	6 kg

4.4. Environment Conditions

Storage Temperature	**max. storage temperature**	-25°C bis +70°C
	max. humidity	95 %
Operating Temperature	**Operating temperature**	0°C bis 70°C
	rel. humidity	20 - 75 %
	Altitude over NN	Up to 1000 m over NN power decrease must be expected.

5. Pinning

High voltage! Fatal danger even in switched-off position!

As long as the motor runs, the motor is a generator!

Therefore avoid the uncontrolled drive of the Novodrive in case of interference by building in a brake.

5.1. Power Supply, Motor Connection and Brake Chopper Resistor

Plug X1 16 pole Combicon.

Note: See pinning of Novotron motors as follows. Divergent pinning for ohter motors, **see connection table**.

Pinning

Pinning	Pin	
External brake chopper resistor (Option)	1	-
Main current connection	2	L1
Main current connection	3	L2
Power supply connection	4	L3
-	5	-
DC rail voltage minus	6	- ZKS
DC rail voltage plus	7	+ZKS
Motor connection	8	V
Motor connection	9	U
Motor connection	10	W
Power supply connection	11	PE
Power supply connection	12	PE

Cross-Section of the Power Supply Cable

Type ND21-	**5605**	**5610**	**5615**	**5620**
Main supply 4 x	1,5	2,5	2,5	4mm²
Motor connection 4 x (without brake)	1,5	2,5	2,5	4mm²
Inner cirbuit-bus 2 x	1,5	2,5	2,5	4mm²
Brake chopper resistor 2 x	1,5	2,5	2,5	4mm²

All cables shielded.

Isolation

The used cables and wires have to have a double or reinforced isolation between core and surface **(Draft DIN VDE 160 6.5.1.1)**.

Fuse

NTypeND21-	**5605**	**5610**	**5615**	**5620**
Fuse	3 x 10 A inert	3 x 20 A inert	3 x 20 A inert	3 x 40 A inert

If several drives are to be fused together, for the whole you have to count the sum of each device.

Motor Connection

The motor has to be connected with a shielded cable on connector X1 of ND21. The cable shield has to be based on ND21 and on the motor (the shield is based on both sides). On ND21 the assigned cable clamps have to be used.

The wire cross section can be interpreted to the expected motor current. **See VDE 0113, German issue of EN 6204**.

Voltage loading

Rated cross section	0,75 mm²	1,00 mm²	1,50 mm²	2.50 mm²	4 mm²
Rated voltage	7,5 A_{eff}	10 A_{eff}	13 A_{eff}	18 A_{eff}	25 A_{eff}

The used cables and wires have to have a double isolation between core and surface **(Draft DIN VDE 160 6.5.1.1)**.

For the cable ends cable-end-sleeves with isolation in the corresponed size have to be used.

The connection area has to be fused against accidentally touchs. Before switching on the supply voltage please ensure that the fitting of all cables have been controlled and examined as well as the isolation of all cable ends.

The earthing connection has to be made on the therefore assigned earthing bolt on the housing of the compact device or on the rear wall of the 19“ rack.

Recommended motor cables:

Lütze Silflex NSY
Lapp Ölflex - 400CP

Motor-Connection Table

Manufacturer	Descr.	Pole	u	v	w	dRESOLVER
Novotron	NHD55	4	9	8	10	4800h
	NHD70					
	NHD92					
	NHD115	6	9	8	10	F180h
	NHD142					
	NHD190					
AEG	MS34-62	6	10	8	9	0000h
Siemens	1FT6	6	10	8	9	0000h
	1FT6	8	10	8	9	2000h
Baldor	BSM80A	4	8	9	10	4000h
MOOG	D313	8	8	10	9	0FC0h
Georgii Kobold	KSY464	6	9	8	10	0900h

Note: The dRESOLVER is a ND21 parameter adjusting any desired resolver setting electrically to the ND21.

The variant with the internal brake chopper resistor must not be connected with a external brake chopper resistor!

Brake Chopper Resistor

The value of the external ballast switching has to be 33 ohm. The external ballast switching has to be connected on plug X1, clamp 1 and 7. Cables to the external resistor have to be shielded.

The internal ballast switching is controlled electronically. On overload the ballast switching stops and ND21 switches over to overvoltage. After switching-off the power supply and a waiting period of approximately 5 minutes the ND21 can be started again. Then the brake chopper control is again ready for operation.

The threshold value of the ballast switching is on 720V DC rail voltage. The integrated automatic threshold allows to connect the parallel inner circuits of several ND21. As the ballast switching which is in action can raise its respond threshold up to 30V, it is guaranteed that the ballst energy on every connected ballast switching is divided evenly.

5.2. Resolver Connection (X2)

Danger by uncontrolled running drive!

If the resolver is not connected correctly, the drive could run up uncontrolled.

Therefore please pay attention to the correct connection of the resolver when connecting a motor to the ND21.

Resolver connection ND21

Plug X2

Cable: Cores have to be shielded in pairs.

Note: In the following please find the pinning for Novotron motors. Divergent pinnings for other motor manufacturer **see resolver pinning**.

Connector: 15 pol HD-DSUB - sleeve device-sided, pin contact cable-sided.

Shield: Cable housing D-Sub HD.

Pinning		**Pins**		
		6	-	
Temperature sensor	1	11	-	
		7		
Temperature sensor	2	12	-	
		8	R1	rotor
-	3	13		
		9	S2	stator
Shield R1, R2	4	14	S3	stator
		10	S4	stator
Rotor R2	5	15	S1	stator

Motor Manufacturer

Novotron

AEG MS

Siemens 1FT6

Baldor

MOOG

Georgii Kobold

Resolver Connection

	Novotron Baldor Moog Georgii Kobold	AEG	Siemens
Resolver	**X2**	**X2**	**X2**
Temperature probe	1	1	1
Temperature probe	2	2	2
Rotor R2 / REF	5	5	5
Rotor R1 / REF	8	8	8
Stator S2 / SIN	9	10	15
Stator S4 / SIN	10	9	14
Stator S1 / COS	15	15	9
Stator S3 / COS	14	14	10

Resolver Cable and Resolver Plug

To ensure the save isolation of all return signals from the active somponents of ND21 please provide all return signals with, at least a basis isolation for a rated voltage of $300V_{eff}$.

The calbe housing for plug X2 must have a non-conductoring surface. This creates, together with the basis isolation of the feedback signals, a double isolation according to **draft DIN VDE 0160 paragraph 5.6.3.2** against all active parts of the ND21.

Resolver cables have to be twisted and shielded in pairs.

Recommendation for Resolver Cables

Lütze Superflex (C)Y-PUR-Kombi Order.
No.: 111094 (usable for C-tracks, oilproof)

Lütze Electronic-LIY(C)Y-(C)Y-Kombi Order.
No.: 110652 (oilproof)

5.3. Peripherie Connection (X3)

Cables

Only use cables in conformance to **VDE 0113 paragraph 14**. For the analogue nominal value, the frequency direction standard and the encoder emulation shield cables are required.

Plugs

44 pol HD-DSUB: Sleeve contacts device-sided, pin contacts device-sided.

Pinning	Pins		
	16		Ext. power supply -15V
shunt resistor -	1	31	GPO1
	17		Ext. power supply +15V
shunt resistor +	2	32	Analogue output +
	18		Analogue input +
limit switch N	3	33	Analogue output -
	19		Analogue input -
Incremental-output A	4	34	End switch P
	20		Incremental output B
Incremental-output N	5	35	Ext. power supply null
	21		GPIN1
GPIN2	6	36	Ext. power supply null
	22		GPIN3 / start
GPIN4	7	37	Ext. power supply null
	23		GPO2
GPIN6	8	38	Ext. power supply null
	24		GPIN5 / release
	9	39	Ext. power supply null
	25		DIR1/GPIN8
FREQ/GPIN7	10	40	Ext. power supply 5-24V
	26		-
-	11	41	-
	27		-

-	12		42	-
		28		Read for operation contact 1
Ready for operation contact 2	13		43	-
		29		-
-	14		44	int. null (option)
		30		int. +15 V (option)
internal -15 V (option)	15			

5.3.1. Analogue Input

Resolution: 1 bit + 1 prefix,
automatical offset compensation.

Analogue input +: Plug X3 44pol HD-DSUB Pin 18.

± 10 V analogue input for nominal values or as process signal input resistor R_i = 20 kohm.

Analogue input -: Plug X3 44pol HD-DSUB Pin 19 is connected with the external power supply-zero.

Shunt-resistor +: Plug X3 44pol HD-DSUB Pin 2,
Shunt-resistor -: Plug X3 44pol HD-DSUB Pin 1

Between both connections please find the shunt R = 410 ohm, 0,3 W, 1 %. The resistor changes +/- 24 mA signals to ±10V.

Note: Velocity command through the analogue input **(see manual Setting up and Parameter Setting of ND21, chapter 3.).**

5.3.2. Analogue Output

Analogue output +: Plug X3 44pol HD-DSUB Pin 32.

±10 V analogue output for analogue process periphery, control, loading 5 mA, resolution 8 bit.

Function: A 20 kHz PWM-signal on GPO1 will be issued as a filtered analogue value.

Analogue output -: Plug X3 44pol HD-DSUB Pin 33 connected with the external power supply-zero.

Note: Operation of the analogue output **(see manual Setting Up and Parameter Setting of ND21, chapter 4. Analogue Output).**

The analogue output is not short circuit proof!

Note: For the operation of the analogue in- and output a external power supply is required:

ext. power supply +15V +/-10%, 30mA: X3 44p HD-DSUB Pin7

ext. power supply -15V +/-10%, 30mA: X3 44pol HD-DSUB Pin 16

ext. power supply null: Plug X3 44pol HD-DSUB Pin 33

ext. power supply 5-24V: Plug X3 44pol HD-DSUB Pin 40

(for the external power supply 5-24V even +15V can be used).

5.3.3. Digital Input: Group 1

Signal level of the digital input: GPIN1 - 6, ESCHP, ESCHN

"0": < 2 V

"1": > 4,5 V (max 24 V +10 %)

Input resistor: 4,7 kohm

Input	Function	Active level	Plug X3 Pin
GPIN1	Mode switch	"1"	21
GPIN2	Mode switch	"1"	6
GPIN3	Start	"1"	22
GPIN4	Reference cam	"1"	7
GPIN5	Release	"1"	24
GPIN6	Mode switch	"1"	8
ESCHP	End switch positive	programmable	34
ESCHN	End switch negative	programmable	3

5.3.4. Digital Input: Group 2

"0": < 1,5 V

"1": > 2,5 V (max. 24 V +10 %)

Input resistor: 4,7 kohm (Pull-Down)

FREQ/GPIN7: Plug X3 44pol HD-DSUB Pin 10,
frequency or mode connector,
max. frequency: 500 kHz

DIR1/GPIN8: Plug X3 44pol HD-DSUB Pin 25,
Direction input of the frequency command,
max. frequency: 2,5 kHz

Note: Velocity commend **(see manual Setting Up and Parameter Setting of ND21, chapter 2.).**

Note: Using GPIN8 as mode connector, **see chapter 13. ND21-Positioning Control.**

It can be used either the analogue switch or the frequency-direction-standard.

Note: For the operation of the digital inputs of group 2 an external power supply is required:
ext. power supply 5-24V: Plug X3 44pol HD-DSUB 44pol Pin 40
ext. power supply null: Plug X3 44pol HD-DSUB 44pol Pin 39

5.3.5. Digital Output

(no encoder emulation)

Standard level: 5-24V (depending on the external power supply)

Loading: 100mA

The digital outputs are not short circuit proof!

GPO1: Plug X3 44pol HD-DSUB Pin 31
FET-output, Rdson(FET) = 7,5 Ohm, built-in pull up resistor 6,8K zu ext. power supply 5 - 24V, built-n free wheeling diode.

Function: Cycle control: Order carried out or PWM or free programmable. If GPO1 is used as digital output a analog output it not possible.

GPO2: Plug X3 44pol HD-DSUB Pin 23,
bipolartransistor-output, built in pullup-resistor 6,8kohm for external power supply 5-24 V, built-in free wheeling diode.

Function: "motor stops" "1"
or "reached position" "1"
or "current limit" "1"
or "free programmable"

Note: Programming of the digital Output (**see chapter 14.3. Configuration).**

Note: Encoder Emulation, **see chapter 5.3.6.**

Note: For the operation of the digital ouputs an external power supply is required:
ext. power supply 5-24V:
Plug X3 44pol HD-DSUB 44pol Pin 40
ext. power supply null:
Plug X3 44pol HD-DSUB 44pol Pin 39

Example: The appropriate pull up resistor for GPO1 and GPO2 to coupel a 24V 10mA PLC input with a level of 20V:

$$R = \frac{24V - 20V}{10mA} = 400ohm$$

Power of the ull-up-resistor:

$$R = \frac{24V^2}{400ohm} = 1{,}44W$$

Loading of the 24 V supply by pull up resistor:

$$R = \frac{24V}{400ohm} = 60mA$$

5.3.6. Encoder Emulation (ROD426)

Opto coupling outputs, internal pull up resistors 18 kOhm to the external power supply 5-24V, output voltage limited by Z-diodes to 15V.

Using the whole opto coupling an external pull-up resistor to 5-24V should be fixed.

Size of the external resistor

$$R_{ext} = \frac{U_{ext} \cdot 18K}{(57V - U_{ext})}$$

Incremental output channel A:
Plug X3 44pol HD-DSUB Pin 4

Incremental output channel B:
Plug X3 44pol HD-DSUB Pin 20

Incremental output channel N:
Plug X3 44pol HD-DSUB Pin 5

Note: For the operation of the encoder emulation a external power supply is required:

ext. power supply 5-24V:
Plug X3 44pol HD-DSUB 44pol Pin 40

ext. power supply null:
Plug X3 44pol HD-DSUB 44pol Pin 39

Note: For the operation of the encoder emulation, **see manual Setting Up and Parameter Setting of ND21, chapter 5. Encoder Emulation.**

Output level: 0-15V (limited by Z-diodes)

The encoder signals are not short circuit proof!

5.3.7. Ready for Operation Contact

Ready for operation potential free contact (Closer),
Loading: < 500mA, < 100V.
Ready for operation contact 1: Plug X3 44pol HD-DSUB Pin 28
Ready for operation contact 2: Plug X3 44pol HD-DSUB Pin 13

5.4. Bus Coupling

The ND21 is equiped with a RS232 or RS422/485 interface. RS232, RS422 and RS485 are standardized electronical data interfaces.

NOVOBUS has a loop structure. The controller sends data to drive No. n, that one to drive No. n-1 etc.. Drive No. 0 sends back to the controller.

Data from the drive to the controller are also transmitted from drive to drive until they reach the controller.

To send the data back from drive No. 0 to the controller, a final connector on X4 on drive No. 0 is required which contains the respective bridges for RS232 or RS422/RS485. Lead and return wire are in the same cable.

NOVOBUS must be earthed! Normally it is earthed automatically by the controller. If not, e.g. if light wave conductor components have been fixed between the controller and ND21, alternatively a final connector has to be earthed (Pin 5 on RS232, Pin 3 on RS422/RS485).

The signals of the buses are separate from the active parts of ND21 by the double isolation. For the power supply cable cable housings with a non-conductoring sufac is indispensable.

For the NOVOBUS shielded cables are required.

5.4.1. Serial Interface - Output Plug X4

Cable Standard RS232 or RS485 cable shielded.

Plug DSUB 9 poles, pin contacts on ND21, sleeve contacts on the cable.

Pinning

RS232	RS485	Pins		RS232	RS485
X5.1	/TD	1			
			6	X5.6	TD
X5.2	X5.2	2			
			7	X5.7	X5.7
TD	TGND	3			
			8	X5.8	X5.8
X5.4	X5.4	4			
			9	-	-
SG	-	5			

Note: X5.a means: Internal bridge to plug X5 Pin a (e.g. X5.1 means: Internal bridge to pin X5 Pin 1).

5.4.2. Serial Interface - Input Plug X5

Cable Standard RS232 or RS485 cable shielded.

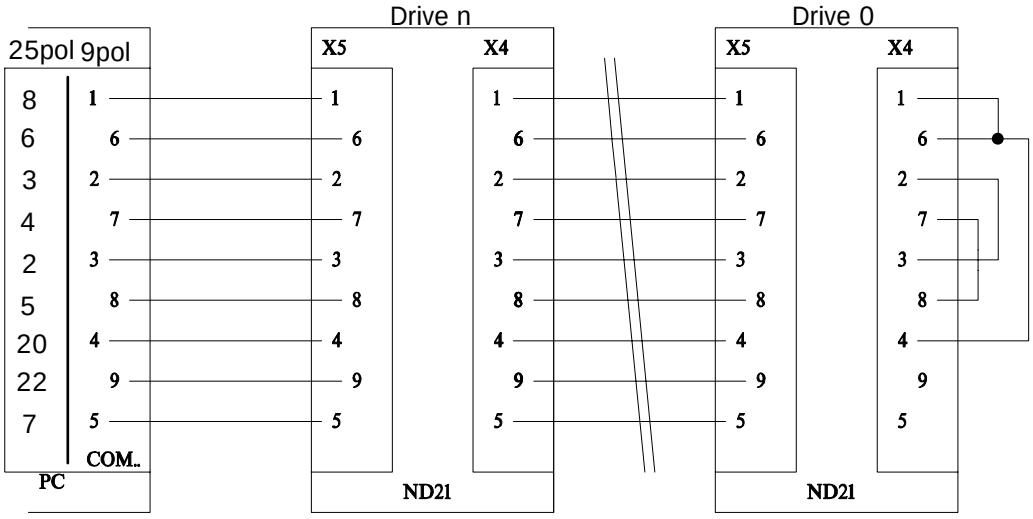
Plug DSUB 9 pole socket contact at ND21, pin contacts at the cable.

Pinning

RS232	RS485	Pins		RS232	RS485
X4.1	/RD	1			
			6	X4.6	RD
X4.2	X4.2	2			
			7	X4.7	X4.7
RD	RGND	3			
			8	X4.8	X4.8
X4.4	X4.4	4			
			9	-	-
SG	-	5			

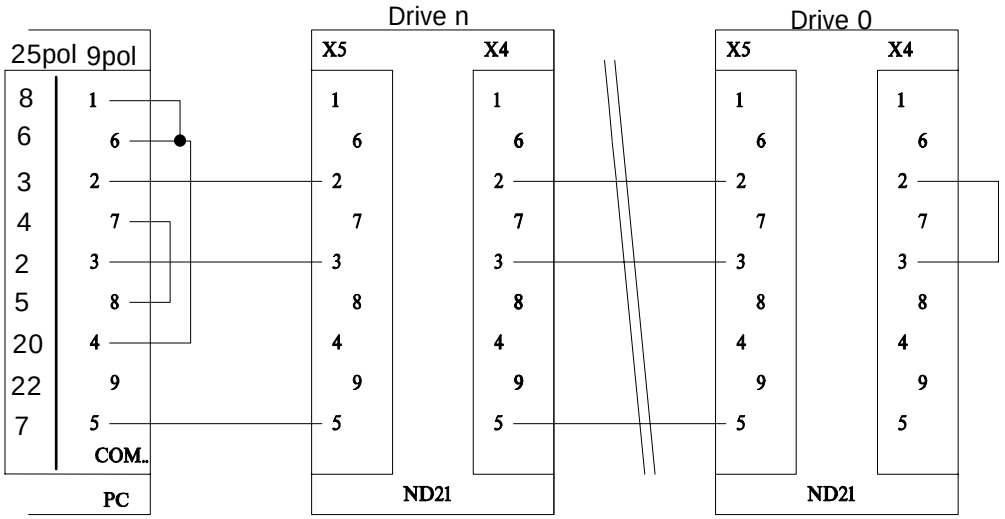
Note: X4.a means: Internal bridge to plug X4 Pin a.

RS232 with connection check



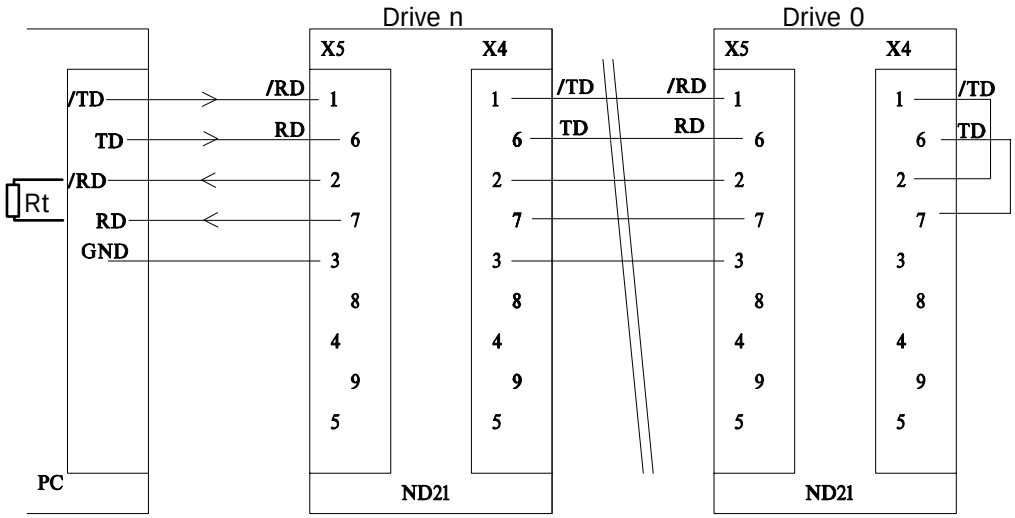
NOVOBUS with RS232 and connection check

RS232 without connection check



NOVOBUS with RS232 without connection check

RS422, RS485



NOVOBUS with RS422, RS485

The receiving line in the control unit must be equipped with a resistance.

RT = 100...120 Ohm (ND21 has an integrated resistior).

5.5. Ventilator Connection

A 2 pole Combicon-Plug can be found at the compact device to supply the installed ventilator:

Power supply: 220 VAC

Power input: < 200mA

Fuse: 315 mA inert

6. Installation

6.1 Mechanical Installation

ATTENTION!

Destruction of the Novodrive!

Operating the Novodrive in a non-suitable environment can be destructible.

Therefore, the installation site must be free of:
Dust, rust, and metal chips, corroding or metal vapors, gases and liquids.

In case of unavoidable condensating moisture, remove the condensation moisture before starting with a suitable heater.

Ventilation

ND21 is only to be installed in an upright position. When installing the compact device the power connection plugs must be at the bottom. When installing the 19" rack the air ventilator drawer must be installed below the ND21. The area round the ND21 ventilation outlets must not be blocked. The ND21 is not to be installed above heat omitting devices.

Environment

The installation can only be performed at a location free of dust, rust, metal chips, corroding or metal vapors, gases or liquids! Condensation moisture is to be avoided. If it is not possible to avoid the condensation if ND21 does not run please ensure that the moisture is completely removed before setting up.

ND21 devices may not be used in areas classified as dangerous, if they are not in approved housings and have not been tested.

6.2. Fuselage

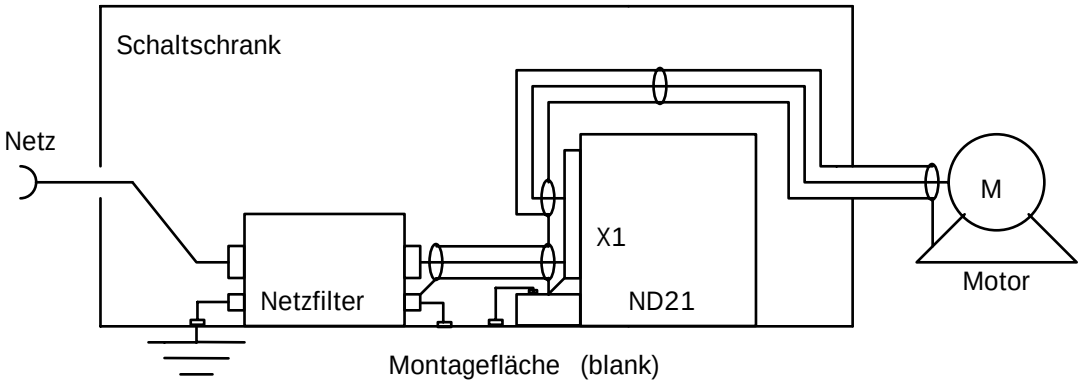
Type ND21-	5605	5610	5615	5620
Fuse	3 x 10 A inert	3 x 20 A inert	3 x 20 A inert	3 x 40 A inert

If several drives are to be fused together, for the whole fuse the sum of each device has to be counted.

Caution „High Voltage"! Fatal danger!

Contacts are only fused if earthing, net and motor connection is executed as described in this chapter.

6.3. Earthing and Shielding



Note: To demonstrate more clearly switch and fuse elements have not been mentioned in the above diagram. The elements have to be inserted that they do not disturb the principle course of the shields and earthing.

To follow the standard **EMV Norm EN55011**, special attention should be paid on shielding and earthing.

Required are shielded motor cables.

Motor Wires The shield is to be fixed with cable clamps on one side of the motor housing and on the other side on the compact device or 19“rack of the ND21.

Shielded power supply wire between power filter and ND21!

Power Filter The shield is to be fixed on the side of the filter with a appropriate earth connection of the power filter and on the side of the ND21 with cable clamps on the compact device or 19“rack of the ND21.

For the compact device we recommend the power filter of Schaffner FN 351-8-29. The power filter must be screwed on the plane assembly board for the effective use of the power filter. Additionally the filter should be earthed with a short connection (2,5mm^2) to the assembly board.

Fuse Wire The fuse wire connection of ND21 (Pin 12 X3) is to be connected with the earthing screw of the compact device or 19“rack. Connection cross section 2,5 mm² for 5605, 5610 and 5615 or 4 mm² for 5620.

Earthing of the compact device and 19“rack.

Earthing ND21 The compact device is to be screwed on the plane assemby plate if possible. Additionally a short earthing connection (10mm^2) should be used between the earth bolt of the compact device or 19“rack towards the assembly plate or earthing busbar. The assembly board has to be earthed properly.

Requirements:

- Put on the shields on both sides.
- Earthing connection short and thick.
- Fixing shields widely.
- Unshielded parts as short as possible.
- Earthing of the switchgear cabinet.
- Wires as short as possible.
- Signal and control wires always separated from the power supply wire.

EMV-compatibility according to EN 55011 is only guaranteed if:

- Convenient power filter is used.
- A connection cable between ND21 and power filter as well as power supply wire between ND21 and the motor is shielded.
- Shielding is conected with the earthing screw of the power filter, the compact device or 19“rack and the motor housing.

6.4. Emergency Power Off Concept ND21

To prevent hazard to individuals or damage to the machine the emergency power off assembly has to be activated. Dangerous parts of the machine or the whole machine can be shwitched off as soon as possible.

Danger of injury by running motor!

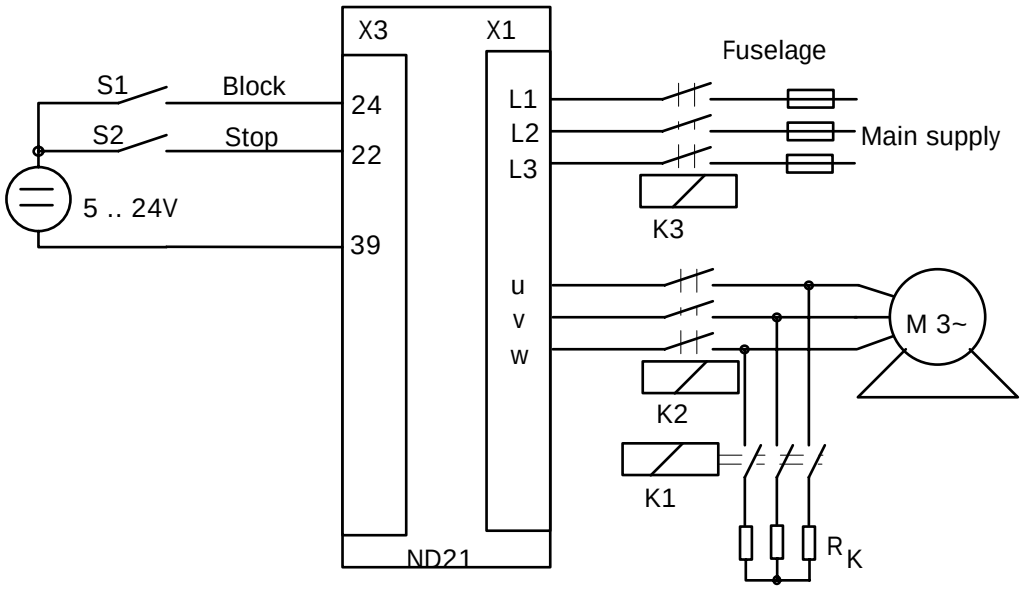
Moving parts can be hazardous to individuals or cause damage to the plant.

Therefore, the plant in which the Novodrive is installed must be equipped with an emergency power off (EPO) assembly. The EPO must stop the plant as quickly as possible.

6.4.1. General

This paragraph is about two different concepts according to the emergency power off:
Short cut braking
Regulated braking

When and which emergency power off concept is to be applied depends on the plant. Depending on the application, the correct emergency power off concept has to be choosen out.



For the emergency power off please abide the following instructions:

- The emergency power-off contactor between Novodrive and motor must be able to separate direct current which is similar to the peak value of the peak current of your Novodrive.

- By short circuit braking the brake resistor must have the correct dimension.

6.4.2. Short Circuit Braking

For short circuit braking of the servo motor the power stage of the ND21 has to be blocked by currentless input 24 on X3. At the same time contactor K1 is to switch on.

Calculation of the brake resistor R_K:

$$R_K = \frac{\text{max. speed (Upm)} \cdot \text{power radient (V/Upm)}}{\sqrt{3} \cdot \text{peak current of the motor}}$$

6.4.3. Emergency Power Off with Controlled Braking

Danger by lag running drive after emergency off.

The drive slowes down by emergency off, therefore please note:

- **<u>Do not</u> block the power stage! Regulated braking is only possible if the power stage is released (neither software nor hardware block).**
- **The control has to be adjusted correctly.**
- **By emergency off the motor stops after eliminating a possible tracking error!**
- **Please note, that a tracking error can occur by mechanical interference.**

ATTENTION! **ND21 can only be stopped when the power stage is released (neither software nor hardware block).**

ATTENTION! **ND21 stops the motor regulated. That requires a correct installation of the control. If the control is installed that a tracking error occures, the tracking error will be eliminated. Only then the motor stops.**

The braking is activated by the pre programmed ramp. Then the power stage can be blocked. Therefore the released input (Pin 24 to X3) is to prepare currentless.

Note: The braking can also be carried out by the analogue speed nominal value input if the nominal value is set on 0.

Additionally it is required to isolate the motor from the main supply. Therefore are two possibilities: Either by separating the main supply and ND21 with contactor K3 or the motor and ND21 with contactor K2. Only one of both contactors is required. Before separating ND21 from the motor, please note that ND21 has to be blocked.

If necessary, the switch off of contactor K2 has to be prolonged by taking adequate measures until braking **referring to chapter 6.4.1.2.** is terminated and the regulator stops. The requirements for a reliable isolation between ND21 and the motor is a complete operation area for the ND21, which guarantees that in case of an emergency power off the contact with current-carrying parts at the ND21 is excluded.

ATTENTION!	**Contactor K2 must be able to isolate direct current which refers to the peak value of the peak current of ND21.**

7. Selecting a Drive Unit

In this chapter you will learn how to find the proper drive unit for a specific usage. In the given calculation example you will see how to select the proper drive for your needs.

7.1. Electrical Selecting

Question: Can a selected motor with a defined torque together with ND21 achieve a certain speed with this torque?

Example: An application requires, a torque of 30Nm for acceleration and should reach a speed of 2200 rpm with this inertia. It is to be tested whether this acceleration can be achieved with ND21 5610 and a motor NHD142E6-180S.

The motor NHD142E6-180S has a continuous idling inertia of 16Nm and can for quick acceleration be 5-fold overloaded. Certainly herewith the requested acceleration is possible. To determine if the specified inertia can be accelerated to the required maximum speed, the following calculating scheme can be used:

Inductive voltage drop on motor:

$U_L = n\ p\ L\ i\ 0{,}0453$

with: NHD142E6-180S

n: Speed [revolutions per minute]	2200
p: Motor polarity	6
i: Effective acceleration current	14,3 A
$i = \frac{M}{(3 \cdot \text{inertia} - \text{constants})}$	$\frac{30\text{Nm}}{(3 \cdot 0{,}7 \frac{\text{Nm}}{\text{A}})}$
L: Motor inductivity phase - phase [H]	0,022 H

When used this result in:

U_L = 188,3 V

Resistive voltage drop on motor:

U_R = R i 0,866

with

R: Effective resistance phase phase - Phase [W] 1,9

herewith:

U_R = 23,5 V

Oposing-EMK of the motors:

U_E = 0,5 $\sqrt{2}$ Vg n / 1000)

with

Vg: Voltage gradient phase - phase [V/1000] 180

herewith U_E = 280 V

Voltage requirement for the motor:

$U = \sqrt{(U_E + U_R)^2 + U_L{}^2}$ = 357,2V

Required inner circuit voltage:

U_{ZK} = $\sqrt{2}$ U= 505 V

At 3 phase 400V power connections the ND21 has an inner circuit voltage of 565V. There is sufficient voltage present to reach the defined speed with the defined torque.

8. Novobus

In this chapter you will learn, how to use the Novobus for controlling your drive unit.

8.1. General

The Novobus protocol and the Novobus driver for IBM PCs are property of Novotron GmbH.

Novobus is an inexpensive solution for networking digital drives and also provides quick communication between controllers i.e. PC or PLC and drive units.

- Exchanging Nominal and actual values (i.e. spin speed nominal and actual value).
- Transmitting new nominal position values for positioning axis.
- Parameter loop for motor regulators (i.e. setting and making on-line changes to regulator structures, regulator parameters and enabling maximum values etc.
- Transmitting control commands (start, stop, regulator stops...).
- Requesting important information (i.e. cooler and motor temperature, limitor switches, ready signals, in-position signals, additional external signals as process information, integrated operating hour counters, status of drive units, error messages).
- Controlling programmable analogue and digital outputs of drive units (i.e. for confirming shields or brakes, transmitting warning signals etc.).

RS232/RS485 The standard transmission hardware for the NOVOBUS are the standard serial interfaces RS232 or RS485 (Standard for all PCs and modern controllers).

No additional hardware extensions i.e. bus controllers, communication cards, protocol chips, intelligent bus plugs are required.

Driver All digital Novotron drive units are standard equipped with the necessary serial interface as well as software drivers for the Novobus protocol.

A Novobus driver as software library for PC applications is provided free of charge (NOVOBUS.LIB).

Loop Structure

The only requirement for communication with the Novobus is a loop connection of the PC with the drive units via RS232 or RS485 cables or via light wave conductors and the respective electro-optic transformers.

Up to 250 axis can be controlled in one loop. The drives are automatically addressed according to their position in the loop. For higher transmission speeds or connecting cycled machine components drives can be distributed to several loops.

8.2. Features

The duration for a byte transfer is 286,46 ms at a speed of 38400 bit/sec.

The required time for completing nominal and acutual value exchange with speed regulated drives is:

Drive per ring	Time
1	0.86 ms
2	2.0 ms
3	2.9 ms
4	3.7 ms
5	4.6 ms
6	5.4 ms

Required time for transferring new nominal value positions for positioning axis is:

Drive per ring	Time
1	4,01 ms
10	40,39 ms
100	401,33 ms
250	1002,79 ms

8.3. Requirements for Transmission

As required, either RS232 or RS485.

Light wave conductor may be used in connection with interface converters.

Transmission speed: 38400 bit/sec (19200, 9600 can be set using setup software).

Transmission is performed with 8 data bits, 1 parity bit (odd parity) and 1 stop bit.

8.4. Bus Structure

NOVOBUS has a loop structure: The drives can be connected to one or more loops.

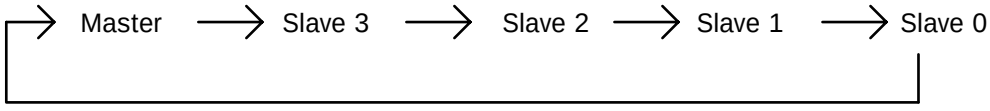
Per loop: 1 Bus-Master (controller)
max. 250 slaves (drives)

In normal operation the slaves can only answer the master transmission. The loop time-out errors, slaves may also independently transmit errors.

8.5. Device Addresses

Up to 250 axis can be controlled by one loop. Drives are automatically addressed according to their position in the loop. Numbering loops of the devices begin with the last one in the loop which has the address “0”. The address of the first device in the loop is N-1, whereas N pertains to the number of slaves.

i.e.: With 4 axis



9. Bus-Definition

The master (controller) transmits telegrams continuously. Most transmissions contain an address (exception: SYNCO and PAUSE). Devices for which a transmission is not intended pass it on, only the device for which the transmission is intended answers the transmission.

The length of the answer is always identical with that of the master transmission.

9.1 Transmission Syntax

Synchronized byte	(obligated)
Address byte	(optional)
Process-Data	(2 byte, optional)
Parameterkanal	(1-7 byte, optional)

All bytes are sent with odd parity. The net transmission length (process data parameter) may consist a maximum of 7 bytes. In a transmission with process data the parameter data may be maximum 5 bytes long.

9.2. Synchronized Byte

The synchronized byte is always the first byte of a transmission. It contains the code for the length of the transmission.

7	6	5	4	3	2	1	0
1	N	S	0	T 2	T 1	T 0	D

N = 1:	Next with short address (or = 0).
S = 1:	Short address.
T2-T0:	Net transmission length (without synchron byte Addressbyte, 0...7).
D = 1:	The transmission contains process data information (2byte data channel)-

Short Address

When S = 1 and N = 0 the same drive that received the last transmission will be addressed.

When S = 1 and N = 0 the next drive will be addressed (Address-Address 0+1).

When S = 0 and N = 0 an address byte will follow.

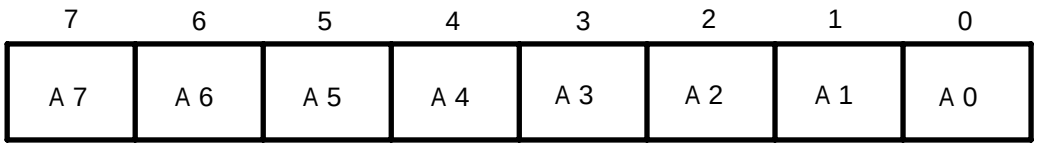
If no process data bus or parameter bus is active a synchronized byte “SYNCO” will be sent to maintain a continuous data flow. “SYNCO” will be forwarded without change by the receiver.

“SYNCO”: H’80

To loosen the data flow a syncrhon byte “PAUSE” can be sent. “PAUSE” being sent as synchron byte will be ignored by the receiver (no respond).

“PAUSE”: H’81

9.3. Address Byte



The master transmission A7-A0 contains:

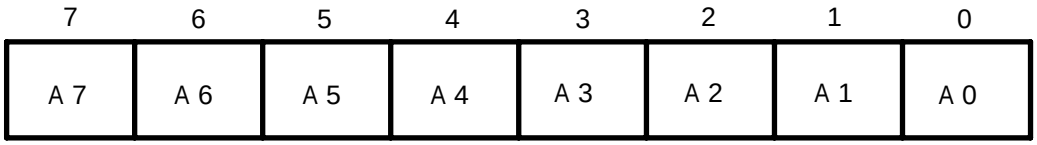
k-N

k = Device address in the loop

N = No. of passive devices (slaves)

All devices increase the address 1 number and forward the transmission with the new address. A drive is addressed when the address after incrementing is “0”. The non-addressed drives pass the entire transmission without checking the contents.

Example: N=5, the controller wants to communicate with the device axis 2: (k-N = 2-5 = -3 H'fd).



Master:	1	1	1	1	1	1	0	1	(H'fd=k-N)
Axis4:	1	1	1	1	1	1	1	0	(H'fe)
Axis3:	1	1	1	1	1	1	1	1	(H'ff)
Axis2:	0	0	0	0	0	0	0	0	(H'00) <- !!!
Axis1:	0	0	0	0	0	0	0	1	(H'01)
Axis0:	0	0	0	0	0	0	1	0	(H'02 = k)

The incremented address is at the adressed axis (axis2) at zero. The master receives the answer for its transmission with the address "k" (in this example, in which axis 2 was addressed: H'02).

9.4. Process Data Bus

With this bus a faster nominal and actual value exchange can be realized. Hereby the lowest bit in the sychron byte is set **(see chapter 9.2. Synchronized Byte).** The data are always 2 bytes long. In the transmission first the byte with the highest value (MSB) and then the one with the lowest value (LSB) is sent.

9.5. Parameter Bus

With the help of the parameter bus the drives can receive parameter, commands can be sent, and information from the drives can be received. A command in the parameter bus consists of a command byte, data bytes (0...4) and a checking byte.

With a Novobus transmission several commands can be sent through the parameter bus and respectively a command in the parameter bus can be divided into several transmissions. The answer to a command is the same length as the command itself (2 to 6 byte). The command byte and the checking byte may not have the value H'00.

9.6. Checking Byte in the Parameter Bus

The master (controller) sends all commands in the parameter bus with a test sum as checking byte (+ check sum). This check sum will be tested by the signaled drive. The command will be carried out when the check sum is correct, otherwise the drive signals an error **(see chapter 9.7. Handling Errors).**

The check sum is the sum of the bytes in a command. In case the sum “0” is transmitted , the check sum will be corrected: Instead of 0, 1 is to be sent. (H’00 is reserved for error messages). The responded drive forms an answer out of its new check sum (-check sum).

The check sum pertains to to the parameter bus and is only made up of the bytes in the parameter bus. Syncronized byte, address byte and process data will not be considered in the check sum.

9.7. Handling Errors

When the drive notices an error in the communication (parity error, framing error, improper synchron byte, incorrect command in parameter bus, improper command parameter or incorrect check sum), it goes into an error mode.

The drive which first detects the error answers with H’00 to all received bytes. The following axis can then detect the error very rapidly when they are addressed with: incorrect synchron, command or checking bytes (may not be H’00), or improper command parameter (in case a H’00 is not accepted here).

A non-addressed drive checks the contents of the transmission. It can can only detect the error when the next syncrhon byte fails. A transmission can have a maximum of 9 bytes long. In case the error is in the address byte, the other axis which have only received H’00 bytes since the error occurred may not notice the error until after the 9 byte.

An error free transmission can only be contained in the address byte (byte 1), in the process data bus (byte 2) and in the data area of the parameter bus (max. 4 bytes). That means a maximum of 7 times consecutively of H’ 00 can be possible. The drives have an error condition that count how many zero bytes have been sent from the preceding drive. In case the drive receives 8 times H’00 without interruption this means that the preceding drive is in an error condition as well.

In this case it checks the sent bytes for a checking sequence. If a checking sequence is detected it returns to normal operating condition (**see chapter 9.8. Check Sequences**).

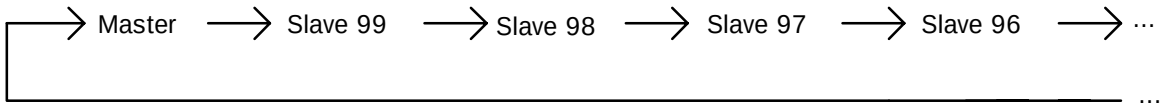
For the next drive to be certain of an error detection, it must receive H'00 at least 9 times. For a drive to detect that the preceding drive is in an error condition it must receive an additional 8 zero byte.

This is why all drives in an error condition send H'00 17 times. If they detect that the preceding drive in the loop is in an error condition they incrementally forward all received bytes.

The master receives (when and error occurs) first a maximum of 25 H'00 and then the address of the drive that detects the error. This provide easy location of errors.

Example: 100 axis

The master needs one byte each out of the RAM of the drive no. 95 to 99. An error bit will be transmitted between the drive no. 98 and 97. The drive no. 97 detects it as a parity error.



The 1. Telegram

1. Byte:	H'88 =	Synchronized byte (follows address byte and 4 bytes in the parameter bus, no process data).
2. Byte:	H'FB =	-5 = 95-100(slave 95 is addressed in a loop with 100 slaves in the ring).
3. Byte:	H'C0 =	Read byte-command for ND21
4. Byte:	H'13 =	AddressL, LSB of address H'FE13
5. Byte:	H'FE =	AddressM, MSB of address H'FE13
6. Byte:	H'D1 =	Checking byte (C0+13+FE = 1D1)

The following transmissions differ only in 2. byte (96-100=-4=H'FC,97-100=-3=H'FD,...)

Master	Slave	Slave	Fehler	Slave	Slave	Slave	...	Slave
	96	98		97	96	95		00
88	88	88		88	88	88	...	88
FB	FC	FD		FE	FF	00		5F
C0	C0	C0		C0	C0	C0		C0
13	13	13		13	13	13		13
FE	FE	FE		FE	FE	>88		88
D1	D1	D1		D1	D1	+A5		A5
88	88	88		88	88	88		88
FC	FD	FE		FF	00	01		60
C0	C0	C0		C0	C0	C0		C0
13	13	13	12	!00	00	00		00
FE	FE	FE		00	!00	00		00
D1	D1	D1		00	00	00		00
88	88	88		00	00	!00		!00
FD	FE	FF		00	00	00		00
C0	C0	C0		00	00	00		00
13	13	13		00	00	00		00
FE	FE	FE		00	00	00		00
D1	D1	D1		00	*00	00		00

Remarks:

>:	Answer byte in a transmission (contents of the memory cell).
+:	New check sum.
!:	The drive detects an error.
***:**	The drive detects that the preceding drive is in error mode.
#:	The 17th sent zero byte after the error detection.

The message H'00 indicates an error to the master, bytes H'61 = 97 means that the error has been noticed from slave 97.

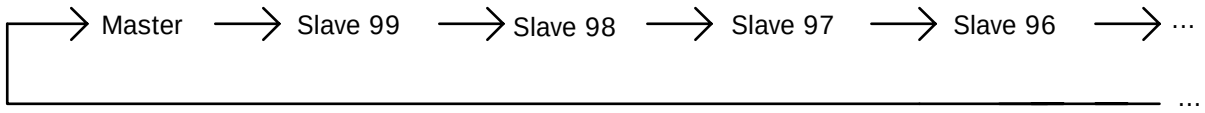
9.8. Checking Sequence

In case the master detects an error in data traffic, it sends a checking sequence to reset the drives. The first 17 bytes of the checking sequence are H'00. Through this the drives that have not yet detected the error are put in an error condition. After the checking sequence the master can repeat the uncompleted commands.

The checking sequence is:

(17 times H'00) H'ff H'44 H'72 H'4c H'41

Example: 100 axis and axis no. 97 - 0 are in error mode. Axis no. 99-98 could not detect any error in the communication. The master has just sent a transmission when it detected the error through an unexpected 0-byte. After detecting the error the master send H'00 17 times. Thereafter it resets the bus with a check sequence.



	Master	Slave 99	Slave 98	Slave 97	Slave 96	Slave 95	...	Slave 00
Started Transmission	88	88	88	00	01	02	...	61
17 Zero Bytes	00	01	02	00	01	02	...	61
	00	00	00	00	01	02		61
	00	00	00	00	01	02		61
	00	00	00	00	01	02		61
	00	!00	!00	00	01	02		61
	00	00	00	00	01	02		61
	00	00	00	00	01	02		61
	00	00	00	00	01	02		61
	00	00	00	*01	02	03		62
	00	00	00	01	02	03		62
	00	00	00	01	02	03		62
	00	*00	*00	01	02	03		62
	00	00	00	01	02	03		62
	00	00	00	01	02	03		62
	00	00	00	01	02	03		62
	00	00	00	01	02	03		62
	00	00	00	01	02	03		62

Check Code

FF	@FF	@FF	@FF	@FF	@FF	...	@FF
44	44	44	44	44	44		44
72	72	72	72	72	72		72
4C	4C	4C	4C	4C	4C		4C
41	41	41	41	41	41		41

Repeating Transmission

88	88	88	88	88	88	...	88
FC	FD	FE	FF	00	01		60
C0	C0	C0	C0	C0	C0		C0
13	13	13	13	13	13		13
FE	FE	FE	FE	>88	88		88
D1	D1	D1	D1	+A5	A5		A5

Remarks:

!: The drive detects an error

***:** The drive detects that the preceeding drive is in error mode

@: The drive detects the first byte of the checking sequence

>: Answer byte in a transmission (contents of the memory cell)

+: New check sum

After the check sequence the bus is again ready for operation.

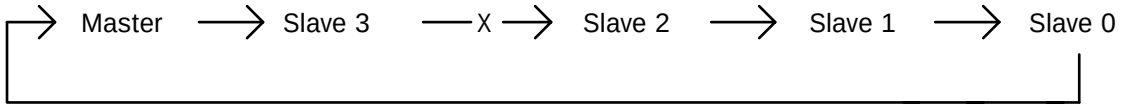
9.9. Time-Out Error

When a byte is not received within at least 10ms, it interpret this as a time-out error. This function can be switched off **(see chapter 14.3.6. The EESelftest Byte).**

When a time-out error occurs, the drive goes into an error condition and waits an further 10 ms to allow other drives to detect and react to the time-out error.

After this waiting period the drive begins to send zero bytes. The other drives detect the error no later the eight receivals of H'00 and then increment the received bytes. The master receives the address of the error source.

Example: Disconnection between axis 3 and axis 2.



Slave2:	...	00	00	00	00	00	00	00	00	00
Slave1:	...	00	00	00	***01**	01	01	01	01	01
Slave0:	...	00	00	***01**	02	02	02	02	02	02

*: The drive detects that the previous drive is in error mode.

The H'02 bytes received inform the master that axis2 has detected an error. The master also knows that either the connection between axis 3 and axis 2 has been broken or axis 3 is not ready for operation.

10. ND21 - Specific Commands

10.1. Read Commands

ReadByte: Reads a byte from the memory of the ND21:

H'2F00-H'2FBF	ROM
H'FD80-H'FF7F	RAM
H'FF80-H'FFFF	On-chip register field

Read Byte

Command	H'C0	AddrL	AddrH	+CheckSum	
Answer	H'C0	AddrL	Data	-Checksum	

ReadWord: Reads a word (2 byte) out of the ND21-memory:

H'2F00-H'2FDF	ROM
H'FD80-H'FF7F	RAM

Read Word

Command	H'C1	AddrL	AddrH	'?'	+CheckSum	
Answer	H'C1	AddrL	DataL	DataH	-Checksum	

('?' = H'3F)

ReadLong: Reads 4 Byte from the memory of ND21:

H'FD80-H'FF7F: RAM

Read Long

Command	H'C7	AddrL	AddrH	'1'	'2'	+ Checksum
Answer	H'C7	AddrL	Data0	Data1	Data3	

Checksum ('1' = H'31, '2' = H'32, '?' = H'3F)

10.2. Write Commands

WriteByte: Writes a byte into the memory of the ND21:

H'FD80-H'FF7F: RAM

Write Byte	**Command**	H'82	Data	AddrL	AddrH	+ CheckSum
	Answer	H'82	Data	AddrL	AddrH	- Checksum

WriteWord: Writes a word (2byte) into the memory of ND21:

H'FD80-H'FF7F: RAM

Write Word	**Command**	H'63	DataL	DataH	AddrL	AddrH	+CheckSum
	Answer	H'63	DataL	DataH	AddrL	AddrH	-Checksum

10.3. Bit Manipulation and Logical Commands

And: Sends a logical AND bit for bis into the memory of ND21.
H'FF00-H'FF7F: RAM (AddrH = H'FF)
RAM := RAM & Data

AND	**Command**	H'A4	Data	AddrL	+Checksum
	Answer	H'A4	Data	AddrL	-Checksum

Or: Sends a logical OR bitwisebit into the memory of ND21.
H'FF00-H'FF7F: RAM (AddrH = H'FF)
RAM := RAM | Data

OR	**Command**	H'A5	Data	AddrL	+Checksum
	Answer	H'A5	Data	AddrL	-Checksum

10.4. Input/Output Commands

WriteIO: Changes the random programmable digital outputs.

Write IO	**Command**	H'c8	Code	+Checksum
	Answer	H'c8	Code	-Checksum

Code	H'00	Clear	GPO1
	H'01	Set	GPO1
	H'02	Clear	GPO2
	H'03	Set	GPO2

10.5. Reset H8

Releases a reset of the ND21 operating program. Every data which has not been safed on the EEPROM will be lost.

Reset	**Command**	H'DD	'!'	+Checksum
	Answer	H'DD	'!'	- Check-sum

('!' = H'21)

11. NOVOBUS PC Driver

11.1. General

A Novobus driver as software library (NOVOBUS.LIB) for PC applications is provided free of charge.

The driver can operate motors in one or two loops. The data transfer is performed via the serial interface of the PC.

The bus driver works in the background via interupt control. The jobs are to be programmed as normal function calls . To avoid unnecessary waiting times, jobs are temperairly stored in a buffer (FIFO).

The driver is equipped with error monitoring and error correction routines. When disturbances in the data transfer occur, the driver attempt to automatically correct them. After several unsuccessfull correction attempts or non-correctable errors, the communication will be stopped and the drive sends an error message.

The bus driver can work in conjuction with a real time system. The different task wanting to work with the bus are synchronized by semaphores.

11.2. Hardware and Software Requirements

NOVOLIB.LIB is a realtime library for IBM compatible MS-DOS computers.

The library was developed for program language Turbo C++ (memory models: small or medium). Object files can also be written in Pascal.

The Novobus functions cannot call functions from the C library. Only the functions H’ 25 (set-interrupt-vector) and H’35 (get-interrupt-vector) can be called.

Only the BIOS-Interrupt INIT0 is used (BIOS- Monitor output for error messages). The Novobus drivers do not use mathematical coprocessors.

The Novobus drivers access the interrupt controllers (PIC 8259) and the serial interface (UART, 8250/16450/16550) directly.

The Novobus drivers are interrupt controlled. At a transmission speed of 38400 bit/s an interrupt occurs every 286 µs. The program may not block the interrupts long, otherwise the computer looses data (UART-overflow).

If this cannot be prevented, the bus must be closed and critical operations restarted, or the module 16550 (UART with FIFO) must be used.

The bus driver can be used with real-time systems (i.e. RTKernel), different task will be synchronized by semaphores. The only functions that the Novobus drivers need are:

- Setting semaphore
- Waiting for semaphore

11.3. Integration of the Library

Novobus.LIB works with the modules “Novobus”, “Nberror” and “NBsemaphore”.

Novobus functions can be found in the module NOVOBUS. The prototypes contain the header file novobus.h. Memory addresses for ND21 can be found in the file nd21.h.

The modules Nberror and NBsemaphore send error messages to the monitor, should they occur. If the usage of another (self-defined) error routine is to be used it must be linked before the Novobus library **(see chapter 11.5. Error Routines**).

The Module NBSEMAPHORE contains two empty function definitions. In case operation with real-time systems is required, the functions NB_Signal() and NB_Wait() must be defined and linked before the Novobus library **(see chapter 11.7. Novobus in Real Time Systems).**

11.4. Opening and Closing the Bus

Novobus drivers can work with one or two loops. A loop can be opened with the function NB_OpenBus1(). The function InitNovobus2() opens the bus with two loops. An opened bus can be closed using the function NB_CloseBus().

With the function NB_Status() inquiries can be made as to whether a bus is active or not.

11.5. Error Routines

When a transmission error occurs the Novobus driver automatically attempts to correct it. The value of the counter value increases with each error correction. The number of automatically corrected errors can be called up with the function NB_BusError(). This function should be performed on a regular basis for monitoring the quality of transmission. Should transmission errors repeatedly occur, the source is to be found and corrected.

If automatic correction is not possible the bus will be stopped and an error message will be sent.

When a non-correctable error occurs the Novobus driver calls the function NB_error(). The library Novobus.lib contains a module NBERROR. If the function NB_error() is not defined in the application program, the driver uses its own error routines which brings error messages to the monitor.

If another error handling is required self-defined handling routines can be written:

extern void far cdecl NB_error (unsigned char Loop, intErrorcode);

Parameter: Loop 1 Loop1,2: Loop2,0: non-defined.

Errorcode: Error code.

The call can also come from an interrupt routine and is therefore to be handled with caution.

(Fast handling were possible, conserve stack usage, no direct or indirect DOS calls, no floppy or hard disc I/Os, do not use mathematical coprocessor/emulator, do not force blocking task changes, etc.. In a multi tasking system only one semaphore should be set and activating herefor a high priority for the error handling).

11.6. Reading and Writing Data

The micro controller of the ND21 is equipped with RAM, ROM and an electrically programmable and erasable memory (EEPROM). The addresses can be found in the Include-File nd21.h.

Functions for reading out of the RAM or ROM.NB_ReadByte():

- NB_ReadWord()
- NB_ReadLong()

Functions for writing in the RAM:

- NB_WriteByte()
- NB_WriteWord()

Function for reading from the EEPROM:

- NB_ReadEEPROM()

Function for writing in the EEPROM:

- NB_WriteEEPROM()
- NB_RAMtoEEPROM()

In a reading routine the PC sends a transmission, the answer from ND21 contains the requested value. The called function must wait for the answer, waiting time depends on the number of devices in the loop.

If a program (or a task) requests a series of data for reading, there is a waiting period between each requested function. For better organization, the following functions are provided:

Functions for time optimized communication:

- NB_StartReadByte()
- NB_StardReadWord()
- NB_Test()
- NB_Result()

The functions NB_StartReadByte() and NBStartReadWord() start the transmission for reading routines. As a result they send back a job number. The job is then stored in an internal buffer.

With the function NB_Test() one can check if the result of a job are already available or not.

With the function NB_Results one can call up results. This

function can be used at anytime, even if the result is not yet available. In this case, the function will wait for the results.

Calling up NB_Test is optional. NB_Result can only be used once for each job.

This function delivers the result of the job and at the same time erases it from the internal bus. When tried a second time no result will be found and an error message will be displayed (incorrect job no.).

To avoid overloading the buffer each result should be called up using the function NB_Result().

11.7. Novobus in Real Time Systems

The Novobus functions send different serial transmissions which can vary in their length. At 38400 bit/s it takes 286 µs to send a byte.

To avoid waiting time, data is buffered in an internal storage (FIFO).

When the buffer is full, the task which calls up a Novobus function must wait. If its a reading function, also the result must be waited on.

In a real-time system different tasks that work with the bus are synchronized by semaphores. Most of the functions have an optional parameter “Semaphore”.

If no semaphore is given the Novobus function waits for a periodical query (polling). This cost however computing time and in a real-time system these task will be blocked and given low priority. This is the reason for the solution using synchronized semaphores.

All reading functions work internally with the function NB_Result(). It first checks if the result of a reading routine is available. If it is not available it waits, with the help of semaphores, in a suspended condition (no usage or waste of computing time).

Since several tasks can simultaneously request Novobus functions it may occur that several semaphores are available. However one semaphore per task is sufficient.

In order to work with semaphores, the Novobus driver requires two functions which are to be defined by the user:

void NB_Signal (void* Semaphore);
Function: sets a semaphore

void NB_Wait (void* Semaphore);
Function: Waits for a semaphore

Example: With RTKernel:
void near* semaphore = (void near*)
RTKCreateSema(Binary,0);

```
extern void far cdecl NB_Signal(void near* semaphore)
{
RTKSignal( (Semaphore) semaphore);
}
extern void far cdecl NB_Wait( void near* semaphore)
{
RTKWait ( ( Semaphore) semaphore);
}
```

11.8. Starting and Stopping Drives

ND21 has the following possible modes:

Reset After being switched on the drives are in a Reset mode. The drive initializes the hardware and the software, and performs a thorough self test. The drive can be reseted with the function NB_ResetH8().

Block In the “regulator blockage” mode the regulator and the end phase will be blocked. There is no current flow and the motor has no torque. The drive can be blocked by a hardware signal (hardware block) or by the bus (software block). The function NB_Sperre() is provided for this function. The software blockage can only be removed through the bus. The hardware blockage can only be removed by a hardware signal. For the enabling, the software as well as the hardware blockage must be removed.

Stop In the “Stop mode” the regulator is enabled, the drive however is stopped. Independent of the given spin speed nominal value, ND21 will stop the motor. If the motor is currently spinning the ND21 will stop it with the predefined ramp. The drive can be stopped by a hardware signal (hardware stop) or by the bus (software stop).

In the Novobus driver the function NB_Stop() is provided. A software stop can only be removed through the bus. A hardware stop can only be removed by a hardware signal. To reinstate the operating mode the software and hardware stops must be removed.

Go In the operation mode “Go”, the drive will spin according to the nominal value. Herefore the signals “enable “ and “Start” must be active. The function NB_Go() removes the software block and the software stop and starts the drive unit.

Error With certain errors the drive goes into an error mode (Error). It registers through the 7-segment display a 3 digit error code. It can also be read out by the function NB_ReadWort(). In an error mode the regulator is blocked. Errors can be confirmed with the function NB_DeleteError().

11.9. Operating the Position Controller

The ND21 is equipped with an integrated one axis positioning controller (PS). The positioning controller can be controlled by a PC controller via Novobus or by signals from an external controller (PLC).

The servo converter ND21 works with resolver loop messages. The resolver delivers the mechanical rotor angle. This value can be electronically moved. The position regulator from ND21 works with a 4 byte resolution. The lower 2 bytes show the angle deviation from the zero point and the upper 2 bytes the distance in rotation. If the motor is not equipped with a multi turn resolver system, the drive will not be able to identify the zero point when it is switched on.

Initialization Three functions are provided for the initialization of the positioning controller. The function NB_InitPS() initializes without changing the actual position value. The function NB_IstwertSetzen() initializes the positioning controller and sets the position actual value to the given value. The function NB_Referenzfahrt() initializes the positioning controller with a referencing drive.

Positioning For the positioning routine the new target position must be provided to the drive using the function NB_WritePS(). Afterwards the drive can be started with the function NB_StartPS().

12. Library Functions

12.1. Bus Open, Close and Status

unsigned int InitNovobus1	(unsigned int	*COMAddress*,
	unsigned int	*COMIRQ*,
	unsigned char	*RingSize*,
	NBSEMAPHORE	*Semaphore*);

Function:	Initializes and starts the Novobus with a loop.
Parameter:	*COMAddress:* Basisaddress of the communication module (normally: COM1 = 0x3f8, COM2 = 0x2f8).
COMIRQ:	Interrupt-Request Occupation. (normally: COM1 = 4, COM2 = 3).
RingSize:	Size of the loop, Number of drives in the loop (1...250).
Semaphore:	For task synchronization in multi tasking systems (optional).
Result:	Unequal to 0 means an error during initialization.

unsigned int InitNovobus2	(unsigned int	*COMAddress1*,
	unsigned int	*COMIRQ1*,
	unsigned char	*RingSize1*,
	unsigned int	*COMAddress2*,
	unsigned int	*COMIRQ2*,
	unsigned char	*RingSize2*,
	NBSEMAPHORE	*Semaphore*);

Function:	Initializes and starts the Novobus with 2 loops.
Parameter:	Same as InitNovobus().
Result:	Same as InitNovobus().

void CloseNovobus (NBSEMAPHORE *Semaphore*);

Function: Closes the Novobus.

Parameter: *Semaphore, for* task synchronisation in multitasking systems (optional).

unsigned int NB_Status (void)

Function: Provides information about the status to the bus.

Result: 0 = inactive

1 = one loop active.

2 = two loops active.

unsigned int NB_BusError (void);

Function: Provides information about transmission errors.

Result: Number of automatically corrected errors.

12.2. Writing Data

void NB_WriteByte (unsigned int *Axis*
unsigned int, *Address*,
unsigned char *Data*,
NBSEMAPHORE *Semaphore*);

Function: Writes a byte in the RAM of the ND21.

Parameter:

Axis:	Address of the drive. Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
Address:	Address of the storage unit. 0xfd80-0xff7f: RAM
Data:	New contents of the storage unit.
Semaphore:	Optional

void NB_WriteWord	(unsigned int	*Axis*,
	unsigned int	*Address*,
	unsigned int	*Data*,
	NBSEMAPHORE	*Semaphore*);

Function: Writes a word into the RAM of ND21.

Parameter:	*Axis:*	Address of the drive. Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Address:	Adress of the memory cell. 0xfd80-0xff7f: RAM
	Data:	New contents of the storage unit.
	Semaphore:	Optional

12.3. Reading Data

unsigned char NB_ReadByte	(unsigned int	*Axis*,
	unsigned int	*Address*,
	NBSEMAPHORE	*Semaphore*);

Function: Reads a byte from the RAM of ND21.

Parameter:	*Axis:*	Addresse of the drives.Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Address:	Address of the storage unit (RAM address). 0x2f00-0x2f4f: ROM-Info 0xfd80-0xff7f: RAM 0xff80-0xffff: On-chip Registerfeld
	Semaphore:	Optional

Result: Contents of the storage unit.

unsigned int NB_ReadWord (unsigned int *Axis*,
unsigned int *Address*,
NBSEMAPHORE *Semaphore*);

Function: Reads a word out of the memory cell of ND21.

Parameter:		
	Axis:	Address of the drives. Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Address:	Address of the memory cell. 0x2f00-0x2f4f: ROM-Info 0xfd80-0xff7f: RAM
	Semaphore:	Optional

Result: Contents of the storage unit.

unsigned int NB_ReadLong (unsigned int *Axis*,
unsigned int *Address*,
NBSEMAPHORE *Semaphore*);

Function: Reads four bytes out of the ND21.

Parameter:		
	Axis:	Address of the drive. Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Address:	Address of the memory cell. 0x2f00-0x2f4f: ROM-Info 0xfd80-0xff7f: RAM
	Semaphore:	Optional

Result: Contents of the storage unit.

12.4. Functions for Time Optimized Communication

NBORDER NB_StartReadByte	(unsigned int	*Axis*,
	unsigned int	*Address*,
	NBSEMAPHORE	*Semaphore*);

Function: Starts a reading routine

Parameter:	*Axis:*	Addres of the drive. Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Address:	Address of the memory cell. 0x2f00-0x2f4f: ROM-Info 0xfd80-0xff7f: RAM 0xff80-0xffff: On-chip register field.
	Semaphore:	Optional

Result: Job number.

NBORDER NB_StartReadWord	(unsigned int	*Axis*,
	unsigned int	*Address*,
	NBSEMAPHORE	*Semaphore*);

Function: Starts a read routine.

Parameter:	*Axis:*	Address of the drives. Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Address:	Address of the memory cell. 0x2f00-0x2f4f: ROM-Info 0xfd80-0xff7f: RAM
	Semaphore:	Optional

Result: Job number.

unsigned int NB_Test (NBORDER *Job*);

Function:	Checks if a job has been completed.
Parameter:	*Job*: Job number.
Result:	0: In process 1: Results available.

unsigned int NB_Result (NBORDER *Job*, NBSEMAPHORE *Semaphore*);

Function:	Delivers the results of a job and erases the job.
Parameter:	*Job:* Job number. *Semaphore:* Optional.
Result:	Respective the job.

12.5. Reading and Writing EEPROM

unsigned char NB_ReadEEPROM (unsigned int *Axis*, unsigned char *Address*, NBSEMAPHORE *Semaphore*)

Function:	Reads a byte out of the EEPROM memory.
Parameter:	*Axis:* Address of the drives. Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9 *Address:* Address of the memory cell. 0x00-0x7f: EEPROM *Semaphore:* Optional
Result:	Contents of the memory cell.

void NB_WriteEEPROM	(unsigned int	*Axis*,	
	unsigned char	*Address*,	
	NBSEMAPHORE	*Semaphore*)	
Function:	Reads a byte to the ROM of the ND21		
Parameter:	*Axis:*	Address of the drives. Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9	
	Address:	Address of the memory cell. 0x00-0x7f: EEPROM	
	Semaphore:	Optional	
Result:	Contents of the memory cell.		

void NB_RAM to EEPROM	(unsigned int	*Axis*,	
	NBSEMAPHORE	*Semaphore*);	
Function:	Copies the regulator parameter out of RAM into EEPROM		
RAM-range:	0xff60-0xff7f (32 Byte)		
EEPROM-range:	0x20 - 0x3f (32 Byte)		
Parameter:	*Axis:*	Address of the drives. Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9	
	Address:	Address of the memory cell. 0x00-0x7f: EEPROM	
	Semaphore:	Optional	

12.6. Start / Stop ND21

void NB_Go (unsigned int *Axis*,
NBSEMAPHORE *Semaphore*);

Function: Starts ND21 (deactivates software-block and software-Stop).

Parameter:	*Axis:*	Address of the drive Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Semaphore:	Optional

void NB_Stop

Function: Starts ND21 (deactivates software-block and software-stop).

Parameter:	*Axis:*	Address of the drive Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Semaphore:	Optional

Function: Stops ND21 (software-stop).

Parameter:	*Axis:*	Address of the drive Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Semaphore:	Optional

void NB_Sperre (unsigned int *Axis*,
NBSEMAPHORE *Semaphore*)

Function: Blocks the regulator and the end phase (software block).

Parameter:	*Axis:*	Address of the drive Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Semaphore:	Optional

12.7. Reset ND21

void NB_DeleteError (unsigned int *Axis*,
NBSEMAPHORE *Semaphore*);

Function: Resets the error message from ND21.

Parameter:	*Axis:*	Address of the drive. Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Semaphore:	Optional

void NB_ResetH8 (unsigned int *Axis*,
NBSEMAPHORE *Semaphore*);

Function: Sets the micro controller (H8) of the ND21 and therefore resets the drive (software-reset).

Parameter:	*Axis:*	Address of the drive. Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Semaphore:	Optional

12.8. Setting Digital Outputs

void NB_IO (unsigned int *Axis*
unsigned int *Code*,
NBSEMAPHORE *Semaphore*)

Function: Sets GPO1 or GPO2 to 1 or 0

Parameter:	*Axis:*	Addresse of the drive. Loop 1: 0x0000-0x00f9 Loop 2: 0x0100-0x01f9
	Code	Set GPO1 H'00 Set GPO2 H'02 Reset GPO2 H'03
	Semaphore:	Optional

13. Positioning Controller

13.1. The ND21 - Positioning Controller (PC1.0)

The ND21 contains a software-module for a one-axe-positioning control (PS) as option. The positioning control can be realized without additional hardware.

The positioning control of ND21 works with a trapezoidal shaped velocity curve. Acceleration (ramp) and drive velocity are programmable. (The track is not limited but the drive of a positioning process with PC1.0 must not be longer than 26 seconds). The positioning control can be controlled by a controller (e.g. PC) via NOVOBUS as well as by a external control (PLC) by hardware-signals.

If a controller is available in the system, new target positions can be transmitted by its serial interface (RS232 or RS485) **(see chapter 11. The Novobus Driver)**. The positioning process can be started by the controller as well as by the PLC.

The ND21 can also realize positioning procedures without controller. Therefore a integrated cycle control (AS) is available. Up to 40 positions can be stored in the ND21. The programming can be realized by the setting up with a PC/laptop. The PLC also can program the cycle control on-line. The new target position can be transmitted either bit serial or by a teach-in process.

13.1.1. Referencing Drive

The servo converter ND21 works with resolver loop messages. The resolver delivers the mechanical rotor angle. If the motor is not equipped with a multi turn resolver system, the drive identifies only the motor angle but will not be able to identify the zero point when it is switched on.

A PC can realize the reference routine by the NOVOBUS function *NB_Referenzfahrt().* If the controller identifies the absolute actual position by a different way, the position information can be transmitted directly and the positioning control can be initialized without a referencing drive. Therefore the function *NB_IstwertSetzen()* is available.

The PLC can also start a referencing drive without a controller. Therefore it has to choose the function *„referencing drive“* out of the four mode-switches (five, if GPIN8 is used) and start with the *„Start“*-signal.

Important steps:

1. Driving with the pre-programmed direction and torque (*RefV1*) as long as GPIN4=Low.

2. If GPIN4=High, blocking with the pre-programmed ramp (*nRampe-*).

3. Slower drive with the pre-programmed velocity in the opposite direction (*RefV2*) as long as GPIN4=High.

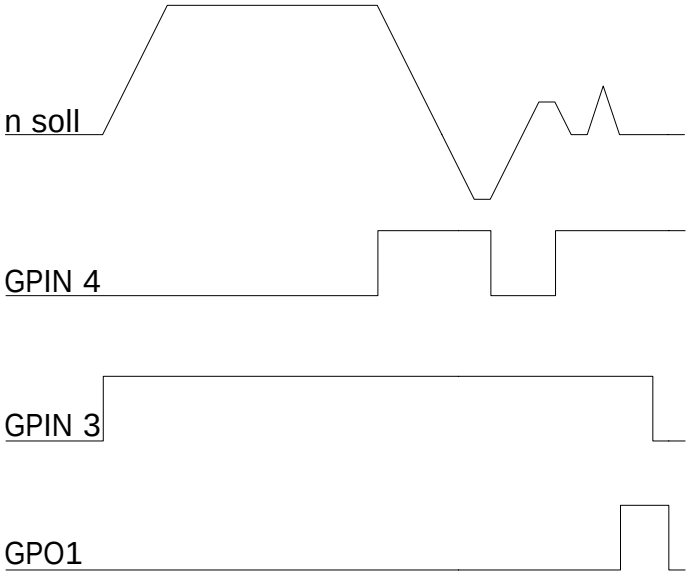
4. If GPIN4=Low, blocking.

5. Slow drive in the first direction until GPIN4=High.

6. Positioning to the next position where the rotor angle is 180^0 (position actual value = H'8000). This position can be shifted with the parameter *RefLage* (H'FD9E). To guarantee the reproduction of the referencing procedure, the referencing switch has to switch very closely to the referring position.

7. Setting the actual position value. The referencing point is according to the 4 Byte actual position value (*RefUmdr0, RefLage0*).

8. If the cycle control is active, ND21 reports with GPO1=High, that the referencing drive has finished.



13.1.2. The Positioning Routine

A positioning routine starts with a new target position of ND21. This can be transmitted by NOVOBUS or bit serial from the PLC or even can be read out of the ND21 EEPROM.

As soon as the new target position is set, the ND21 positioning calculation can be started. The calculation with PS1.0 only can be made in a stand still position and in a released controller condition.

If ND21 is ready with the internal calculation the positioning routine can be started.

As soon as the positioning is ready, it can be reported by digital signals of the PLC be sent via Novobus.

The digital output GPO2 can be programmed for a "In Position"-report.

Velocity and ramps are programmable.

In the drive block condition the drive does not have a stall torque and the motor can run with seperate forces. Therewith the start position is not correct any more and the drive block cancels all results of the positioning controller.

For the positioning the drive has to be started again. If the calculation was started in the condition of drive block , the drive waits until the controller is released again. Afterwards the calculation can be started.

The positioning routine can be stopped with Stop or with the drive block. On Stop the drive stops with the pre programmed ramp, on drive block it will not stopp electriacally. The controller and the PLC can activate stop or drive block.

PLC stops by the following ND21 inputs:

GPIN3=Low (*Stop*) and

GPIN5=Low (*Reglersperre*).

13.1.3. Configuration

There are different drive concepts:

1.: ND21 and PLC.

2.: ND21 and controller (PC).

3.: ND21 and controller (PC) and PLC.

13.1.3.1. ND21 and PLC

In this case a PLC is available to control the cycle of the machine. To that belongs the controller by the ND21 positioning control. The PLC can fall back to one set of 23 functions by digital signals:

- Start positioning,
- Select target position,
- Activate inching operation,
- Start referencing,
- Transmit target positions,
- Perform teach-in,
- Read and quit errors,
- Re read actual position,
- Program the positioning velocity,
- Program inching velocity,
- Program ramp,
- Program currency limit,
- Switch over to analogue nominal value.

13.1.3.2. ND21 and Controller

In this case only little digital signals have to be connected to ND21 (Start, Release, Limit Switch, Referencing Switch).

The ND21 positioning controller is regulated from the controller via NOVOBUS. Therefore special commands are available.

13.1.3.3. ND21 and Controller and PLC

This configuration can be found on machines which requires a quick real time regulation with PLC but at the same time have a great expanse in data processing e.g. visualisation.

In this case the controller find out the new target position to transmit it via NOVOBUS to ND21 and start the calculation of the positioning.

The PLC starts the positioning. With the command *WaitPS* a synchronization between PLC and controller can be realized.

The PLC selects the function and starts with the signal *Start* (GPIN3). The ND21 reports with the digital output signal GPO1=1, if it has received and worked up a new target position from the controller. Afterwards the PLC can start the positioning procedure with the function *StartPS*.

13.1.4. Programming of ND21

Before the positioning controller can be used the current regulator, the spin speed regulator and the position regulator are to be adjusted. The ramps, the current and the spin speed limits are to be set **(see manual Setup and Parameter Setting of ND21).**

To start the cycle control it has to be switched on (bit 3 in HwVersion=1). The positioning controller can only work correctly if the overlayed position regulator is activated (bit 6 in SwVersion=1) and the nominal value is programmed as digital nominal value (bit 1 and bit 0 in SwVersion=1).

The output GPO2 can be programmed as desired:

HwVersion

Bit 6	Bit 5	GPO2
0	0	Programmed by the bus.
0	1	Current limit (motor blocked).
1	0	In-Position (PEH).
1	1	Motor idling.

In Position: With the parameter *Window* (H' FFB7) the InPosition window is set. The in-position monitoring will be switched to GPO1 if the actual position does not differ from the target position less than 4 x *Window*.

Operation mode Stepping:

The velocity is fixed with *Vtippen* (2 Byte, memory address H' FD94). After switching on, the ND21 initializes the value of the EEPROM.

EEPROM-Address: *EE_Vtippen* (H' 50, H' 51).

If bit 0 in *SwVersion2* is 0, the current limit (maximum torque) is given with the analogue input.

Operation Mode Referencing

Reference-velocities are programmable:

1. Velocity:

RefV1 (2 Byte, H' FD8C)/*EE_RefV1* (H' 48, H' 49)

2. Velocity:

RefV2 (2 Byte, H' FD8E)/*EE_RefV2* (H' 4A, H' 4B)

The velocities have to be programmed with prefixes (two's complement), and have to point in the opposite direction (different prefix).

After the referencing the actual value is to be set:

RefUmdr0 (2 Byte, H' FD90)/*EE_RefUmdr0* (H' 4C, H' 4D)

RefLage0 (2 Byte, H' FD92)/*EE_RefLage0* (H' 4E, H' 4F)

The motor zero position can be displaced with *RefLage* (2 Byte, H' FD9E)/*EE_RefLage* (H' 54, H' 55).

Operation Mode Positioning

The positioning velocity can be programmed:

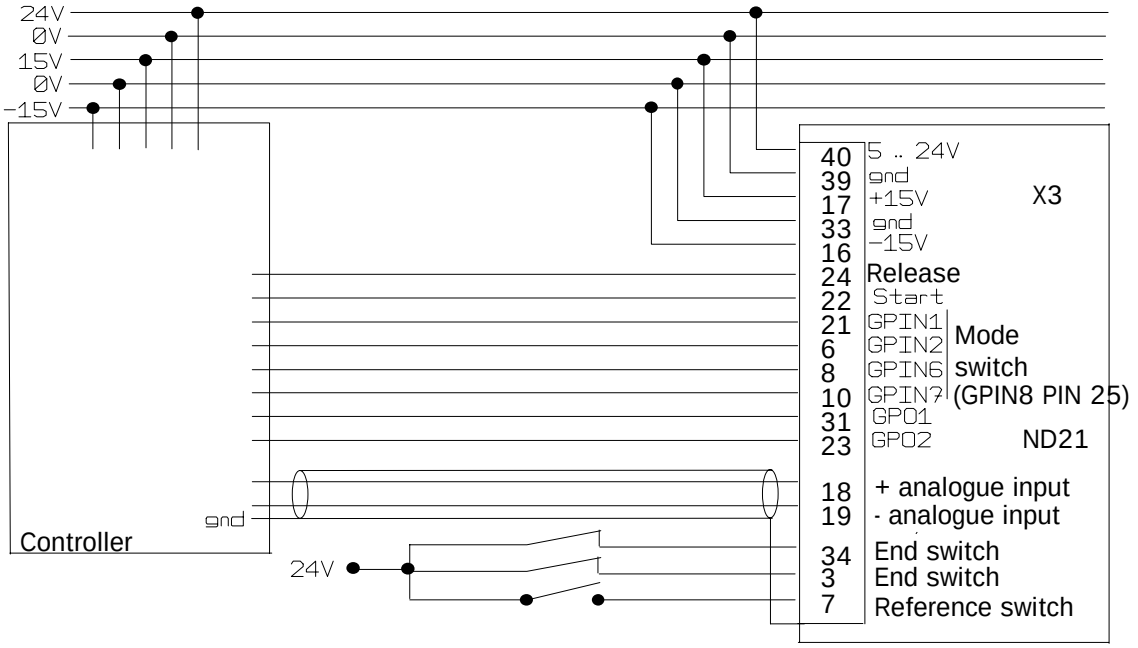
ps_v0 (2 Byte, H' FF4C)

This velocity will be initialized with the maximum velocity.

nMax (2 Byte, H' FF68)

13.1.5. Controlling via PLC (Cycle Control (AS 1.0))

13.1.5.1. Connection Diagram



Controlling of the ND21 cycle control:

Note: GPO1 an GPO2 are open collector outputs with 6.8 kohm an internal pull up resistance.

Note: Every connection should be as short as possible. The cable to the analogue input must be shielded.

Note: The 24V and the +/- 15V power supply must be earthed properly.

Note: The analogue input and the +/- 15V power supply are only required if an analogue signal should be performed (while stepping a spin speed limit).

13.1.6. Function Mode

ND21 integrates a cycle control which enables the ND21 to store 40 positions. These 40 positions can be used by the ND21 positioning controller. The positions can be determined in a tech-in process or can be transmitted bitserial from the PLC. There are functions for: Manual operation (stepping), referencing, positioning selection, start of the positioning procedure, etc.

A function is always selected thus from the signals GPIN (8),7,6,2,1 the function code and afterwards on the start input (GPIN3) an increasing flank is set. With GPO1=*Low* the cycle control monitores that it is ready to start a new function. GPO1 will be High if the function is done.

The start input (GPIN3) has to be *High* as long as GPO1 with *High* monitores the end of the function. Previous *Low* of the start input (GPIN3) means the end of the function.

Example referencing:

1. Setting the function code for referencing:

   GPIN1 = High

   GPIN2 = High

   GPIN6 = *Low*

   GPIN7 = *Low*

   (GPIN8 = *Low*, if necessary)

2. Increasing flank *Low* - High on the start input GPIN3.

3. ND21 performes the referencing. The PLC keeps the start input GPIN3 *High* until GPO1 will be *High*.

4. ND21 monitores with GPO1 = High, that means the referencing procedure has been finished.

5. The PLC switches the start input (GPIN3) to *Low* so that GPO1 also will below and herewith monitores the PLC that the ND21 is ready for further functions.

13.1.7. Signals and Functions

13.1.7.1. Inputs

GPIN8 (X3, 25): Additional Mode-Switch

GPIN7 (X3.10): Mode-Switch

GPIN6 (X3.8): Mode-Switch

GPIN2 (X3.6): Mode-Switch

GPIN1 (X3.21): Mode-Switch

GPIN3 (X3.22): Start-Signal

GPIN4 (X3.7): Reference Switch

GPIN5 (X3.24): Driver Release
Low: Drive block (power stage blocked)
High: Release

13.1.7.2. Outputs

GPO1 (X3.31): Order executed

GPO2 (X3.23): programmable:

- Spin speed limit (motor blocked)
- In-position
- Motor idling

13.1.7.3. Mode-Switch

The desired function can be selected by the mode switches.

(GPIN8= *Low*, if applied)

GPIN7	GPIN6	GPIN2	GPIN1	Function
Low	*Low*	*Low*	*Low*	StartPS
Low	*Low*	*Low*	*High*	Tippen-
Low	*Low*	*High*	*Low*	Tippen+
Low	*Low*	*High*	*High*	Referenzfahrt
Low	*High*	*Low*	*Low*	RdEEWrPS
Low	*High*	*Low*	*High*	ClrPZ
Low	*High*	*High*	*Low*	IncPZ
Low	*High*	*High*	*High*	DecPZ
High	*Low*	*Low*	*Low*	Teach In
High	*Low*	*Low*	*High*	WrEE
High	*Low*	*High*	*Low*	Wr0
High	*Low*	*High*	*High*	Wr1
High	*High*	*Low*	*Low*	WaitPS
High	*High*	*Low*	*High*	RdError
High	*High*	*High*	*Low*	RdPos
High	*High*	*High*	*High*	WrPS

13.1.7.4. PLC Functions

ClrPZ Sets the position pointer to zero (PZ = 0).

IncPZ Increases the PZ with one (PZ++).

DecPZ Decreases the PZ with one(PZ--).

Wr0, Wr1 With *Wr0* or *Wr1* a target position (32 Bit) can be sent bitwise (first job: MSB, last one: LSB). During the transmission no PLC function can be executed. The target position can either be loaded into the positioning controller *(WrPS)* or be stored in the EEPROM (*WrEE*).

WrPS With this function the bitwise transmitted position can be loaded into the positioning controller (PS).

WrEE With this function the bitwise given function can be stored in the EEPROM. Afterwards PZ will be incremented automatically.

RdEEWrPS Reads the actual target position out of the memory (EEPROM), loads it into the positioning controller (PS), starts the calculation and increases the position pointer (PZ) with one (auto-increment).

Note: If the regulator is blocked the drive is waiting for its release afterwards the calculation can be started.

StartPS Starts the positioning procedure. The function will be performed if previously a new target position with *WrPS* or *RdEEWrPS* has been loaded into the positioning controller (PS).

Note: With GPIN3=*Low* (stop) or GPIN5=*Low* (block) the positioning controller can be stopped.

Tippen+, Tippen- Manual operation starts the motor with the pre programmed velocity (Vtippen). GPO1=High monitores to the ND21 that it has startet. With GPIN3=*Low* (Stop) the motor can be stopped.

Referencing With the pre programmed direction and velocity, ND21 surches for the referencing point. The referencing switch has to be connected with the input GPIN4.

Note: With GPIN3=*Low* (Stop) or GPIN5=*Low* (Sperre) the referencing switch can previously be stopped.

Teach-In The actual position of the drive will be stored into the EEPROM to the actual memory cell (PZ), afterwards the positioning marker (PZ) will be incremented.

RdPos With this function the position of the drive (actual value) can be read bitwise (32 bit, first LSB). If this is not necessary it can be switched over to the analogue nominal value by RdPos. Therefore bit 4 in HwVersion2 must be set to 0.

RdError With this function the 12-bit-errorcode can be read. The transmitted bits will be monitored in GPO1. In the correct operation condition the word has the value zero. With the 13th read command an error message can be quitted.

WaitPS This function supports the synchronization of the controller and the regulator (PLC). If the controller sends a new target position via NOVOBUS and the positioning procedure has been started from the PLC the PLC must issue a *WaitPS*. With GPO1=High the ND21 reports that it has found the new target position and the calculation has been finished.

13.1.7.5. Additional Mode-Switch GPIN8

The command range of the cycle control has been extended by 8 additional functions. For the drive GPIN9 (X3 pin 25) is to be used. To use GPIN8 as an additional mode switch, bit 1 in parameter SwVersion 2 and bit 7 in HwVersion is to be set on 0. Afterwards the functions can be used.

Function Mode ND21 always receives 16 bit parameters.The single bit is written with the functions WrO Param and Wr1 Param, bitserial to the ND21 into the parameter buffer.

With the functions Wr nMax, Wr Vtippen, Wr nRampe and Wr i the parameter buffer can be copied into the desired memory cell.

GPIN8	GPIN7	GPIN6	GPIN2	GPIN1	Function
High	*Low*	*High*	*High*	*High*	*Wr0 Param*
High	*Low*	*Low*	*High*	*High*	*Wr1 Param*
High	*High*	*High*	*High*	*Low*	*Wr nMax*
High	*High*	*Low*	*High*	*Low*	*Wr Vtippen*
High	*Low*	*High*	*High*	*Low*	*Wr nRampe*
High	*Low*	*Low*	*High*	*Low*	*Wr i*
High	*High*	*High*	*High*	*High*	*InitPS*
High	*High*	*Low*	*High*	*High*	*Leitungstest*

Wr0 Param A 0-bit is set into the parameter buffer.

Wr1 Param A 1-bit is set into the parameter buffer.

Wr nMax Copies the parameter buffer to nMax. nMax means the velocity for the positioning procedure.

Wr Vtippen Copies the parameter buffer to Vtippen. Vtippen means the velocity for the stepping operation.

Wr nRampe Copies nRampe- (braking ramp) and nRampe+ (acceleration ramp).

Wr i Copies the first 8 bit of the parameter buffer to imax and the second 8 bit to i0. Therewith imax is the peak current and i0 means the motor rated current.

Init PS Initializes the positioning controller after the switching off.

Power Test Causes the error message 0x602. This error message can be read back bitserial with the function Rderror.

13.1.8. Activation of the Cycle Control

Bit 3 in Byte *HwVersion* (H'FF60) is to be set on 1.

13.1.9. Data Administration of the Position

In the non-volatile memory of ND21 (EEPROM), up to 40 positions can be set. All 4 byte positions are stored from H'60 of the EEPROM-Address:

PZ	1. byte	2.byte	3. byte	4. byte
0	H'60	H'61	H'62	H'63
1	H'64	H'65	H'66	H'67
2	H'68	H'69	H'6A	H'6B
3	H'6C	H'6D	H'6E	H'6F
4	H'70	H'71	H'72	H'73
5	H'74	H'75	H'76	H'77
6	H'78	H'79	H'7A	H'7B

7	H'7C	H'7D	H'7E	H'7F
8	H'80	H'81	H'82	H'83
9	H'84	H'85	H'86	H'87
10	H'88	H'89	H'8A	H'8B
11	H'8C	H'8D	H'8E	H'8F
12	H'90	H'91	H'92	H'93
13	H'94	H'95	H'96	H'97
14	H'98	H'99	H'9A	H'9B
15	H'9C	H'9D	H'9E	H'9F
16	H'A0	H'A1	H'A2	H'A3
17	H'A4	H'A5	H'A6	H'A7
18	H'A8	H'A9	H'AA	H'AB
19	H'AC	H'AD	H'AE	H'AF
20	H'B0	H'B1	H'B2	H'B3
21	H'B4	H'B5	H'B6	H'B7
22	H'B8	H'B9	H'BA	H'BB
23	H'BC	H'BD	H'BE	H'BF
24	H'C0	H'C1	H'C2	H'C3
25	H'C4	H'C5	H'C6	H'C7
26	H'C8	H'C9	H'CA	H'CB
27	H'CC	H'CD	H'CE	H'CF
28	H'D0	H'D1	H'D2	H'D3
29	H'D4	H'D5	H'D6	H'D7
30	H'D8	H'D9	H'DA	H'DB
31	H'DC	H'DD	H'DE	H'DF
32	H'E0	H'E1	H'E2	H'E3
33	H'E4	H'E5	H'E6	H'E7

34	H'E8	H'E9	H'EA	H'EB
35	H'EC	H'ED	H'EE	H'EF
36	H'D0	H'D1	H'D2	H'D3
37	H'D4	H'D5	H'D6	H'D7
38	H'D8	H'D9	H'DA	H'DB
39	H'DC	H'DD	H'DE	H'DF

Each position can be programmed when setting up with a PC (Laptop). The positions can be programmed bitwise or by teach-in functions via PLC.

Each position is given in 4 bytes (32 bits = 1 mode bit + 31 bit position):

Bit31 (MSB)	Bit28	1. byte
Bit27	Bit 16	2. byte
Bit15	Bit 8	3. byte
Bit7	Bit 0 (LSB)	4. byte

Bit 31 determines the relative or the absolute position. If bit 31=0, the positions are absolute (distance from the zero point). If bit 31=1, the positions are relative (path). In the teach-in-process only absolute positions are stored.

Bit 30...bit 16 contain whole revolutions.

(-16384...+16383), bit15...bit 0 contains the angle of one revolution. ND21 works with a 16 bit angle resolution, that means 65536 impulse/revolution.

Bit 30...bit 0 contain the position in the double complement format. The double complement is formed out of 31 bit. A bytewise or wordwise double complement building leads to wrong results.

Examples **1. Target position: 10,25 loops from zero point into the positive direction.**

absolute:	Bit31 = 0
10 complete loops:	decimal 10 = H'000A
Bits 30 - 16:	B' 000 0000 0000 1010
0,25 loops (90°):	0.25 · 65536
	= decimal 16384
	= H'4000
Bits 15 - 0:	B' 0100 0000 0000 0000
In hexadecimale form:	1. byte = H'00
	2. byte = H'0A,
	3. byte = H'40
	4. byte = H'00
summarized:	H'000A 4000

2. Target position: -10,25 loops from the zero point.

absolute:	Bit31 = 0
31 bit-double complement of	H'000A 4000
	= H'7FF5 C000

1. byte = H'7F

2. byte = H'F5,

3. byte = H'C0,

4. byte = H'00

3. Distance: 50º

relative data:	Bit 31 = 1,
no complete revolution (0)	Bits 30 - 16 = H'0000,
50 º corresponds to	$65536 \cdot \frac{50}{360}$
= decimal 9102	Bits 15 - 0 = H'238E.
1. byte = H'80 2. byte = H'00 3. byte = H'23 4. byte = H'8E	
summarized:	H'8000 238E

4. Distance: -50º

relative data:	Bit31 = 1
Double complement of	H'0000 238E in 31 Bit
	H'7FFF DC72,
summarized	H'FFFF DC72

5. Further examples:

Target = 6,5 rev. from the zero point:	H'0006 8000
Target = -6,5 rev. from the zero point:	H'7FF9 8000
Target = 100 rev. from the zero point:	H'0064 0000
Target = -100 rev. from the zero point:	H'7F9C 0000
Distance = 6,5 revolutions:	H'8006 8000
Distance = -6,5 revolutions:	H'FFF9 8000
Distance = 100 revolutions:	H'8064 0000
Distance = 100 revolutions:	H'FF9C 0000

13.1.9.1. Positioning Marker (PZ)

The positioning marker (PZ) marks the actual target position (0 <= PZ < 40).

The PLC can modifiy the marker with the functions *ClrPZ* (setting back), *IncPZ* (incrementing) und *DecPZ* (decrementing).

The command *RdEEWrPS* automatically increments the positioning marker.

13.1.9.2. Positioning Data Programmed with PLC

First the positioning marker is to be set on the desired memory position in the EEPROM.

The target position can be transmitted bitserial with the commands *Wr0* and *Wr1*. Starting with bit 31 *Wr0* transmittes a 0 bit and *Wr1* a 1 bit. After 32 commands the whole position is transmitted. It can be stored with the command *WrEE* into the EEPROM. With the command *WrPS* this position can be transmitted directly to the positioning calculation. The transmission of the target position must not be interrupted by other commands.

13.1.9.3. Programming Position Data with Teach-In

First the positioning marker is to be set on the desired memory position in the EEPROM.

Setting the drive into the correct position with *Tippen+* and *Tippen-*. The transmission into the EEPROM (where the positioning marker shows) is made by the command t*each-in*.

13.1.9.4. Programming Positions with PC

In the service menu (sub-menu EEPROM) the target position is to fill in, as calculated above, with the function write byte.

13.1.9.5. Read the Actual Position

With the command *RdPos* the 32 bit actual position can be read bitserial. Each command *RdPos* transmitts a bit to GPO1. Bit 0 is transmitted firts. After 32 commands the transmission is completed. The transmission of the actual value must not be interrupted by other commands.

With RdPos it can be switched over to analogue nominal values. Therefore bit 4 in HwVersion2 is to be set on 0.

13.1.10. Referencing

Referencing it started with the command *Referenzfahrt*. ND21 reports with GPO1 = 1, when the referencing is finished.

13.1.11. Positioning

The positioning marker is to be set on the desired position. With *RdEEWrPS* the position can be read out of the EEPROM and the positioning calculation can be started. The positioning can be started with *StartPS*.

The drive reports GPO1=High if the positioning calculation has been finished.

13.1.12. Stepping

With *Tippen+* or *Tippen-* the motor with the pre programmed stepping velocity can be run. The motion continous during the start signal (GPIn 3).

13.1.13. Completing the Remaining Track

After the drive has been stopped with stop or block, the positioning can be continued by the following instructions:

- Release drive.

- Reload the positioning process with WrPS.

- Start the positioning process with StartPS.

13.2. Controlling via NOVOBUS

13.2.1. Initialization of the Positioning Controller

There are four ways to initialize the positioning control:

- Referencing
- Setting Actual Value
- InitPS
- Automatically initialization after switching on. Therefore bit 1 in HwVersion2 is to be seto on 0.

13.2.1.1. Referencing

There is a special function in the NOVOBUS driver:

void NB_Referenzfahrt (unsigned int *Axis*,
NBSEMAPHORE *Semaphore*);

Function: Initializes the positioning controller (PS) with referencing

Parameter:	*Axis:*	Addressof the drive, Ring 1: 0x0000-0x00f9 Ring 2: 0x0100-0x01f9
	Code	Set GPO1 H'00 Set GPO2 H'02 Reset GPO2 H'03
	Semaphore:	Optional

NB_Referenzfahrt is a function which performes several actions in the NOVOBUS:

1. ps_v0 (H'FF4C) is read and intermediately stored.
2. Status (H'FF00) is read and tested:
   whether the reset is still active,
   and no error occures.
3. If the reset is not active any more and no error is located ps_status (H'FF43) is to be set on 0.

4. Both 5 bits of Flags2 (H'FF57) are to be set on 0.

5. nsoll (H'FF08) is set on 0.

6. Bit 7 in Flags2 (H'FF57) is set on 1 to start referencing.

7. Flags2 (H'FF57) bits 5, 6 and 7 are controlled. As long as not all 3 its are null, the referencing is active. At the same time bit 5 and 7 are controlled in Flags (H'FF56), for that the referencing has not been interrupted.

8. As soon as the referencing process has been finished, ps_status (H'FF43) will be controlled until the drive monitores in-position with ps_status = H'01. Therewith stop (bit5) and block (bit 7) in Flags are controlled.

9. ps_v0 is read back.

13.2.1.2. Setting Actual Value

There is a special function in the NOVOBUS driver:

void NB_IstwertSetzen	(unsigned int	*Axis*,
	unsigned long	*Position,*
	NBSEMAPHORE	*Semaphore*);

Function: Initializes the positioning controller (PS) without referencing and monitores the new value for the actual position.

Parameter:	*Axis:*	Addresse of the drive, Ring 1: 0x0000-0x00f9 Ring 2: 0x0100-0x01f9
	Position	The actual rotor position corresponding to the latest position actual value.
	Semaphore:	Optional

NB_IstwertSetzen is a function which performes several actions in the NOVOBUS:

1. Istposition (H'FF16 4 Byte) and dLage (H'FF74) are read:
   ....Lage = NB_ReadLong (drive, 0xFF16)
   ....dlage = NB_ReadWord (drive, 0xFF74)

2. Deduct the rotor position shift from the actual position:
   Lage = Lage - dlage

3. Calculation of the new rotor position shift:
   dlage = Istwert - Lage

4. Asking whether the drive is blocked. If not, it is to be blocked.

5. New dlage in ND21 (H'FF74).

6. Actual value revolution in umdrist (H'FF16).

7. If the drive has not been blocked on 4., it is to be released now.

8. The new actual position (H'FF16 4 byte) must be read Lage = NB_ReadLong (drive =xFF16) and be compared with the actual value.

   delta = Istwert - Lage
   If the deviation is smaller than 256 Increments, continue with 9., otherwise repeat from 2.

9. Bit3 in Flags2 (H'FF57) must be set on 0.

10. H'01 in ps_status (H'FF43).

13.2.1.3. Init PS

There is a special function in the NOVOBUS driver:

void NB_InitPS	(unsigned int	*Axis*,
	NBSEMAPHORE	*Semaphore*);

Function: Initializes the positioning controller (PS) without referencing and without alteration of the position actual value.

Parameter:	*Axis:*	Address of the drive, Ring 1: 0x0000-0x00f9 Ring 2: 0x0100-0x01f9
	Semaphore:	Optional

NB_InitPS as follows:

1. Bit 3 in Flags2 (H'FF57) is to be set on0.
2. Bit 6 in ps_status (H'FF43) is to be set on0.
3. ps_status = H'01 ? If yes, then ready. If no, go to 4.
4. If ps_umdrehung = H'FF then ps_status is to be set on H'01, ready. If ps_umdrehung <> H'FF then go to 5.
5. Wait until ps_status bit 5 or bit 1 will be 1 or ps_status = 0.
6. ps_status is to be set on H'01.

13.2.2. Transmitting the New Target and Starting the Positioning Calculation

void NB_WritePS	(unsigned int	*Axis*,
	unsigned long	*Position,*
	NBSEMAPHORE	*Semaphore*);

Function: Sends the new target position of the positioning controller and starts the calculation.

Parameter:	*Axis:*	Address of the drive, RIng 1: 0x0000-0x00f9 Ring 2: 0x0100-0x01f9
	Position	New target position.
	Semaphore:	Optional.

NB_WritePS as follows:

1. Controlling Flags2 (H'FF57) bit 3 and wait until it is null.

2. Target setting: upper 2 bytes in ps_positionH (H'FF44)
lower 2 bytes in ps_positionL (H'FF46)

3. Flags2 bit3 is to be set on 1.

13.2.3. Starting Positioning Process

void NB_StartPS (unsigned int *Axis*,
NBSEMAPHORE *Semaphore*);

Function: Starts the positioning procedure (software-start).

Parameter: *Axis:* Address of the drive,
Ring 1: 0x0000-0x00f9
Ring 2: 0x0100-0x01f9

Semaphore: Optional

NB_StartPS as follows:

1. Waiting until bit 5 of ps_status (H'FF43) willl be 1.

2. Bit 4 from ps_status is to be set on 1.

13.2.4. End of the Positioning Process

The positioning procedure is finshed, if ps_status has the value H'01. This can be read with the function NB_ReadByte.

If GPO2 has been programmed for the in-position, the In Position-report in H'FFB7 can be read with bit 5 = 1.

13.3. Internal Organisation of the Positioning Controller

13.3.1. Status

The positioning controller is controlled by the memory cell ps_status address H'FF43. The bits have the following meaning:

Bit7:	Direction of the positioning process
Bit6:	1: starts the PS-calculation
Bit5:	1: reports "PS-Rechnung fertig" (PS-calculation finished)
Bit4:	1: starts the positioning process
Bit3:	1: Phase "Beschleunigen und Konstantfahren" (Acceleration and constant drive)
Bit2:	1: Phase "Bremsen" (braking)
Bit1:	1: Phase "Zielbremsen" (target braking)
Bit0:	1: reports "Positionierung abgeschlossen" (positioning procedure finished)

13.3.2. Organisation of the Positioning Controller

13.3.2.1. Transmitting the Target Position

The absolute target position with 32 bit (16 Bit revolutions and 16 bit rotor position) will be set in ps_positionH (H'FF44) and ps_positionL (H'FF46).

13.3.2.2. Starting Calculation

Therefore bit 3 of Flags2 (H'FF57) is to be set on 1.

ND21 calculates and registers the direction of the positioning procedure in bit 7 of ps_status. Additionally the amount of the required motor revolutions and the impulses are caluculated and registered in ps_umdrehung (H'FF48) and ps_impulse (H'FF4A).

Then bit 6 in ps_status is to be set on 1. Thereby the calculation of the positioning process will be started.

The calculation of the positioning process determines the values ps_v0 (H'FF4C), ps_k0 (H'FF4E), ps_v1 (H'FF51), ps_k1 (H'FF52) and ps_delta (H'FF53). These values determine which time and which velocity it takes to stop, when reaching the target.

As soon as the calculation is finished, bit 5 in ps_status is to be set on 1.

13.3.2.3. Starting Positioning

After bit 5 in ps_status has monitored the end of the calculation, the positioning process can be started with ps_status bit 4 = 1. The run of the positioning process can be followed with ps_status.

During the acceleration and the constant drive bit 3 is set.

During braking bit 2 is set.

During target braking bit 1 is set.

As soon as the positioning process has been finished, bit 0 is set.

At the same time bit 3 Flags2 will be set back to monitor, that a new calculation can be started.

13.3.2.4. Relative Positioning

For the relative positioning it is sufficiently to register the number of revolutions in ps_umdrehungen, the amount of the motor pulses in ps_impulse, and the direction in bit 7 of ps_status. Bit 6 of ps_status is to be set on 1 to start the positioning calculation. Waiting until bit 5 of ps_status is set on 1. Then starting the positioning process with ps_status bit 4=1.

13.4. Error Sources

Problem	Reason
PC-calculation can not be started.	Drive blocked.
Error 600 positioning controller-overflow.	Positioning time > 26 sec, shorten the positioning distance or increase velocity.
In-position-report does not appear on GPO2.	Checking programming of HwVersion 2, checking adjustment window.
Referencing stops occasionally one revolution further.	Adjusting referencing switch.
Error 601.	Positioning control not released.

13.5. Positioning Control with Sinus^2-Curve

13.5.1. General

The positioning process with a trapezoidal shaped velocity-ramp results in high acceleration jumps. They affect disturbingly vibration systems in its motion cycle. At the same time the mechanics are loaded above the average. That again can decrease the service life of the plant. To avoid that, ND21 delivers possibilities to increase the velocity with a sinus^2-function (S-ramp).

13.5.2. Requirements

The pre calculations for the S-ramp does not take place in the ND21 but in a PC. Therefore it must be integrated in the system. Furthermore it can undertake all tasks, like monitoring drive information or transmitting control commands, etc.

Additionally to the PC, the following instructions are to be performed:

- ND 21 main processor H8 from version V3.1.
- ND 21 coprocessor Novochip 2204.
- Novobus-driver NOVOBUS.LIB from V3.1.
- ND21-parameter: Bit 2 in SwVersion=0.

13.5.3. Description of the Functions

The programming of the S-ramp-parameters, like ramp gradient, drive velocity, distance and drive direction is made via IBM compatible PC. This PC is connected with the ND21 via RS232 or RS485 interface. The transmission rate is 38400 Baud.

The NOVOBUS-driver (NOVOBUS.LIB) as software library is provided free of charge. It contains from version 3.1 three special commands for the programming of the positioning controller with S-ramp.

SK_WritePS With this command the NOVOBUS-driver receives the latest position and velocity data. The drive calculates from that several points of support for the S-ramp and sends it to the addressed drive. Between the points of support a constant acceleration proceeds.

The maximum duration of the positioning process is limited to 120 minutes, that means after that time the programmed distance has to be covered. Please note, when programming the distance: No positions are given but distances (incremental).

An absolute positioning like the trapezoidal shaped velocity procedure is not integrated with the S-ramp at the time. The drive direction is set with the command SK_StartPS.

SK_StartPS With this command starts the positioning process in the ND21. Additionally the drive direction is stated. The advantage is, that with same distances the points of support are not to be calculated again. It is suffiecient to send the command SK_StartPS again.

BYTE SK_StatusPS This function delivers the status of the positioning with S-ramp. Several conditions are possible.

Ready The transmission of the distance and velocity information from the PC to the ND 21 is finished. The positioning can now be activated by the command SK_StartPS.

Walt This condition occurs when there is a hardware or software-stop before starting. ND 21 then is waiting for a hardware or software-start.

Stopped If the drive is stopped or blocked during a positioning procedure, the status changes into the condition *stopped*. The following distances are done from this position.

Doing The positioning process is active at the moment.

Done After reaching the target position the status *done* is sent.

13.5.4. Syntax of the Command for Positioning with S-Ramp

The following commands declare the additionally commands which are required for the positioning with a sinus^2-curve and which are integrated in the driver-version 3.1.

unsigned char SK_WritePS	(unsigned int	*Achse*,
	unsigned long	*Weg*
	unsigned int	Upm,
	unsigned char	*Rampe*,
	NBSEMAPHORE	*Semaphore*);

Explanation: The function sends a new distance (incremental) for the positioning with sinus^2-curve. The required pre calculations are done in the PC.

Parameter:	*Address:*	Novobus-address.
	Weg:	Distance in impulses (0x10000 corresponding to 1 revolution).
	Upm:	Size of the loop, number of drives in the loop (1...250)
	Rampe:	Acceleration time in ms.
	Semaphore:	Optional

Result: = 0:Command executed successfully.
<> 0: Error during operation.

unsigned char SK_StartPS	(unsigned int	*Achse*,
	char	*Richtung,*

NBSEMAPHORE *Semaphore*);

Explanation: The funciton starts the positioning process with sinus^2-curve. As distance the latest transmitted distance (with SK_WritePS) is relevant.

Parameter:
- *Address:* Novobus-address.
- *Richtung:* '+' or '-'
- *Semaphore:* Optional

Result:
= 0:Command executed successfully.
<> 0: Error during operation.

unsigned char SK_StatusPS (unsigned int *Achse*,

NBSEMAPHORE *Semaphore*);

Explanation: The function delivers the status of the positioning with sinus^2-curve.

Parameter:
- *Address:* Novobus-address.
- *Semaphore:* Optional

Result:
= 0: ready (transmission of the information PC - ND21 finished)
= 1: wait (waiting for a hardware or software-start)
= 2: doing (positioning in operation)
= 3: done (positioning finished)
= 0xFF: stop/block (drive stops).

14. Setup and Output Capabilities of the Novodrive

In this chapter you will learn of the setup capabilities by the Novodrive. In addition, explanations of readable parameters are provided.

Parameter settings affecting the control circuit, can only be made when you have a complete understanding of the respective parameter and its properties.

There are two ways of changing the Novodrive settings:

- Via setup software
- Via NOVOBUS

14.1. Requirements

With the setup software, several parameters can be comfortably edited by the way of menu settings. If this possibilities are not sufficient for your application, please find the necessary information in this chapter.

All parameters can be changed or read by the NOVOBUS. The NOVOBUS driver software makes the required write and read-commands available.

With the setup software you can perform all settings and read the inidividual parameter.

To set and read the parameters with the setup software, a IBM compatible PC and a bus cable is required.

14.2. Novodrive Memory

Your Novodrive has 5 different memories:

- RAM-memory with 512 Byte, Addresses: FD80h - FF7Fh.
- Memory in ASIC.
- EEPROM with 256 Bytes, Addresses: 00h - FFh.
- The micro controller’s memory, Addresses: FF90h - FFA7h
- FFB0h - FFFFh.

EEPROM-memory, Addresses: 0000h - 3FFFh.

See chapter 14.2.1. RAM Memory.

RAM-Memory A change in RAM directly affects the circuit. When the ND21 is switched off and on, changes will be lost in the RAM if they were not previously stored in the EEPROM of the ND21.

The RAM-memory of the ND21 is readable by the way of the service menu using the function RAM-Monitor. Changes are made with the function Write-RAM **(see Manual Setup and Parameter Setting of ND21, chapter 1.4.11.)**

ASIC The ASIC parameter involves the setting of the PI current regulators. Changes in these parameters directly affect the circuit. The ND21 setup software always automatically stores this in the EEPROM.

Changes are made in the service menu using the function ASIC **(see Manual Setup and Parameter Setting of ND21, chapter 1.4.11.).**

EEPROM Memory The ND21 is equipped with a serial EEPROM with 256 byte. The EEPROM is non-volatile, meaning it does not loose its data by switching off of power supply.

Changes are made in the service menu EEPROM **(**RAM **(see Manual Setup and Parameter Setting of ND21, chapter 1.4.11.).**

HS Register Field This contains the addresses for controlling the µcontroller periphery.

EPROM Memory The EPROM memory contains the ND21 firmware, non-changeable parameters and tables.

14.2.1. RAM Memory

Address	Description	Bytes	Meaning
FD80	PoleJump	2	Jump address for motor polarity
FD82	errorcode	2	Error code
FD84	i2	4	
FD88	EEPROMbuffer	2	Internal EEPROM program
FD8A	EEPROM control	1	Internal EEPROM program
FD8B	reserviert	1	

FD8C	RefV1	2	1. Reference speed
FD8E	RefV2	2	2. Reference speed
FD90	RefUmdr0	2	After a referencing run the rotation counter will be set to this value
FD92	RefLage0	2	After a referencing run the rotor position will be set to this value
FD94	Vtippen	2	Speed stepping
FD96	iOffset	2	Internal currency measurement
FD98	dRESOLVER	2	Resolver adjustment
FD9A	PZ	1	Position marker in the cycle control
FD9B	BZ	1	Bit marker in the cycle control
FD9C	AS_Command	1	Command in the cycle control
FD9D	reserviert	1	
FD9E	RefLage	2	Displacing the zero point during referencing run
FDA0			
- FE9F	scope_puffer	256	Internal memory oscilloscope
FEA0			
- FEFF	stack	96	Internal program stack
FF00	Status	1	Status byte
FF01	mmax	1	Max. torque
FF02	msoll	2	Nominal inertia
FF04	iasoll	1	Nominal current phase A
FF05	ibsoll	1	Nominal current phase B
FF06	iaist	1	Actual current phase A
FF07	ibist	1	Actual current phase B
FF08	nsoll	2	Nominal spin speed

FF0A	nsollrampe	2	Nominal spin speed after ramp generator
FF0C	nist	2	Actual spin speed
FF0E	nIntegrator	2	Integrator for speed regulator
FF10	emk	1	Motor electro-magnetic-field
FF11	lagesollL	1	
FF12	umdrsoll	2	Nominal value for rotation
FF14	lagesoll	2	Nominal value for rotor position
FF16	umdrist	2	Actual value for rotation
FF18	lageist	2	Actual value for rotor position
FF1A	lage0	2	Old actual value for rotor position
FF1C	nsollAnalog	2	Reserved
FF1E	icra	2	Input Capture register A
FF20	icra0	2	Input Captuer register A old
FF22	VCOoffset	2	Offset analogue input
FF24	adcsr	1	A/D-Control-Status-Register
FF25	reserviert	1	
FF26	TempKK	1	Cooler temperature
FF27	TempMot	1	Motor temperature
FF28	nsollDigital	2	Reserved
FF2A	Novobus	1	Communication bytes
FF2B	Command	1	
FF2C	ParamBuffer	4	
FF30	ChecksumIn	1	
FF31	ChecksumOut	1	
FF32	DataIn	1	Data channel input address
FF33	DataInBuffer	1	Data channel input bytes

FF34	DataOut	1	Data channel output address
FF35	DataOutBuffer	1	Data channel output bytes
FF36	BusAddr	1	
FF37	reserviert	1	
FF38	scope_status	1	Oscilloscope status
FF39	scope_timer0	1	Time base
FF3A	scope_counter	2	Trigger decelerate
FF3C	scope_delay	1	Decelerate pointer
FF3D	scope_pointer	1	Internal
FF3E	scope_trigger	1	Address trigger source
FF3F	scope_level	1	Trigger threshold
FF40	scope_signal1	1	Address signal 1
FF41	scope_signal2	1	Address signal 2
FF42	reserviert	1	
FF43	ps_status	1	Status positioning control
FF44	ps_positionH	2	Target rotation
FF46	ps_positionL	2	Target rotation
FF48	ps_umdrehung	2	Rotation counter position
FF4A	ps_impuls	2	Impulse counter position
FF4C	ps_v0	2	First speed
FF4E	ps_k0	2	Duration of first velocity
FF50	ps_v1	2	Second speed
FF52	ps_k1	2	Duration of second speed
FF53	ps_delta	1	Remaining path
FF54	reserviert	1	
FF56	Flags	1	ND21 status byte
FF57	Flags2	1	ND21 status byte

FF58	Cnt2	1	Internal counter
FF59	reserviert	1	
FF5A	uP	2	µP utilization
FF5C	RealTimeClock	2	Time incrementor
FF5E	CntBlink	1	Cycle counter display
FF5F	CntRevers	1	Reverse cycle counter
FF60	HwVersion	1	ND21 configuration byte
FF61	HwVersion2	1	ND21 configuration byte
FF62	SwVersion	1	ND21 configuration byte
FF63	SwVersion2	1	ND21 configuration byte
FF64	MaxTempKK	1	Cooler temperature limit
FF65	MaxTempMot	1	Motor-Temperaturgrenze
FF66	imax	1	Peak current
FF67	i0	1	Rated motor current
FF68	nMax	2	Maximum speed
FF6A	reserviert	2	
FF6C	nRampe+	2	Acceleration ramp
FF6E	nRampe-	2	Braking ramp
FF70	nKp	1	P-part speed regulator
FF71	nKi	1	I-part speed regulator
FF72	LKp	1	P- part speed regulator
FF73	LKd	1	D- part speed regulator
FF74	dLage	2	Zero point positioning
FF76	Slipping error	2	Slipping error only with position regulation
FF78	VCOlin	1	Graduation analogue and frequence input
FF79	nFilter	1	Tacho filter

FF7A	emk0	1	EMK Precontroller for electro-magnetic field
FF7B	Window	1	Window for in-position or speed 0
FF7C	AnOut	1	Selection byte, analogue output
FF7D	AnOutOffset	1	Offset calibration, analogue output
FF7E	ReversTakt	1	Cycle for test mode
FF7F	VCOlinL	1	Fine scaling of frequency input

14.2.2. EEPROM Memory

00	EE_LA	Reserved
01; 02; 03	EE_Seriennummer	Serial number
04		Reserved
05; 06	EE_BetriebStd	Operating hours
07	reserviert	
08; 09	EE_SperreStd	Block residual hours
0A; 0B	EE_InitDatum	Date EEPROM initialize
0C	EE_Version	Configuration byte
0D; 0E; 0F	reserviert	00
10 - 1E	reserviert für	Error history
1F	EE_parity	EEPROM test byte
20 - 3F	EE_RAM	RAM-mirror corresponding to RAM FF60 - FF7F, (parameter range)
40	EE_a_param3	ASIC setting
41	EE_a_mpr1	ASIC setting
42	EE_a_mpr2	ASIC setting
43	EE_a_mpr98	ASIC adjustment

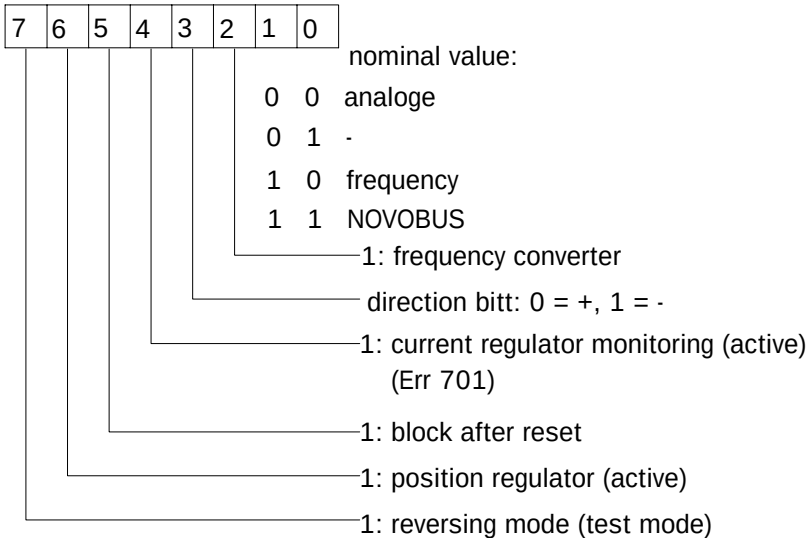
44; 45	EE_impulszahl	Encoder emulation
46	EE_pole	Motor pole counter
47	EE_selftest	Configuration byte
48; 49	EE_RefV1	Referencing speed
4A; 4B	EE_RefV2	Referencing speed
4C; 4D	EE_RefUmdr0	Reference point coordinate
4E; 4F	EE_RefLage0	Reference point coordinate
50; 51	EE_Vtippen	Stepping speed
52; 53	EE_dRESOLVER	Resolver adjustment
54; 55	EE_RefLage	Zero point positioning
56	EE_nd2204	ASIC setting
H'57	reserviert	
58; 59	EE_ProgDatum	Date parameter change
5A; 5B	EE_ProgName	
5C - 5F	EE_Kunde	Reserved for customer (4 Byte)
60 - FF	EE_Ablauf	Cycle control (160 Byte)

14.3. Configuration

14.3.1. The Byte SwVersion

RAM-parameter Address: H'FF62
EPROM: H'22

The byte is a read and write byte.



Remarks:

Bit 0 and 1:	Determines where the speed regulator of the ND21 receives ist nominal value.
Bit 2:	1 is the frequency converter mode. 0 is the servo converter mode.
Bit 3:	With this bit the spin direction of the motor can be changed. If the motor is supposed to be running in the positive direction, however is running negative, this can be corrected by changing bit 3.
Bit 4:	With 1 an additional monitoring function for the current regulator can be activated.
Bit 5:	With 1 the ND21 will first become active (after being switched on) when enabling by the way of the NOVOBUS occurs. With 0 the ND21 will become active (after being switched on) when voltage is applied to the regulator blockage input.

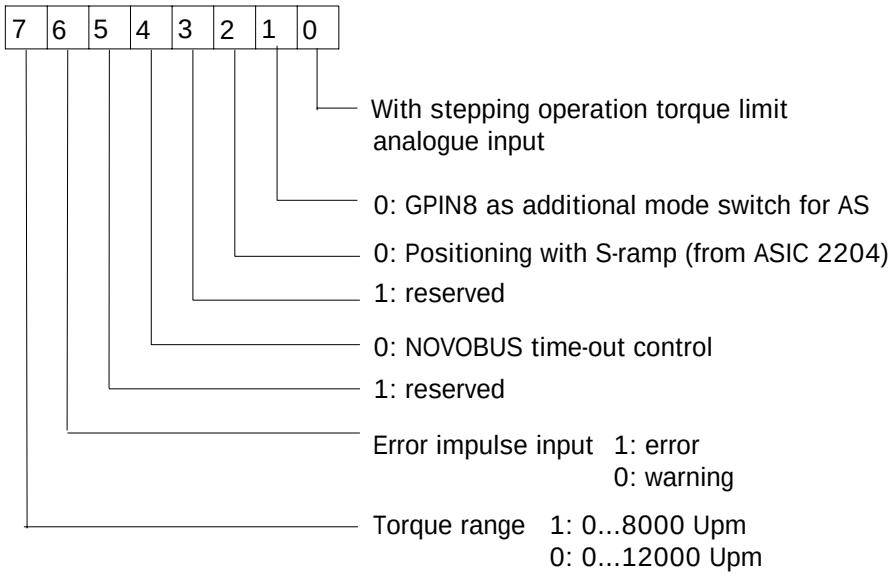
Bit 6: With 1 the position regulator will be activated. This is also possible when working with spin speed nominal values. In this case the ND21 calculates the position nominal values from the spin speed nominal values. Herewith a higher rigidity of the motor is attained.

Bit 7: With 1 and internal test mode is switched on, so that the ND21 reverses the motor with the set spin speed. This operating mode is especially useful in optimizing the regulator parameter.

14.3.2. The Byte SwVersion2

RAM-parameter Address: H'FF63
EEPROM: H'23

The byte is a read and write byte.



Remarks:

Bit0: With 0 the reading of the maximum torque from the analogue input can be programmed.

Bit 1-5: Reserved, to be set on 1.

Bit 6: If 1 is programmed here, it will lead to an error on the impulse input which will switch off the power stage activating an error message. With 0 a warning will be activated without switching off the power stage.

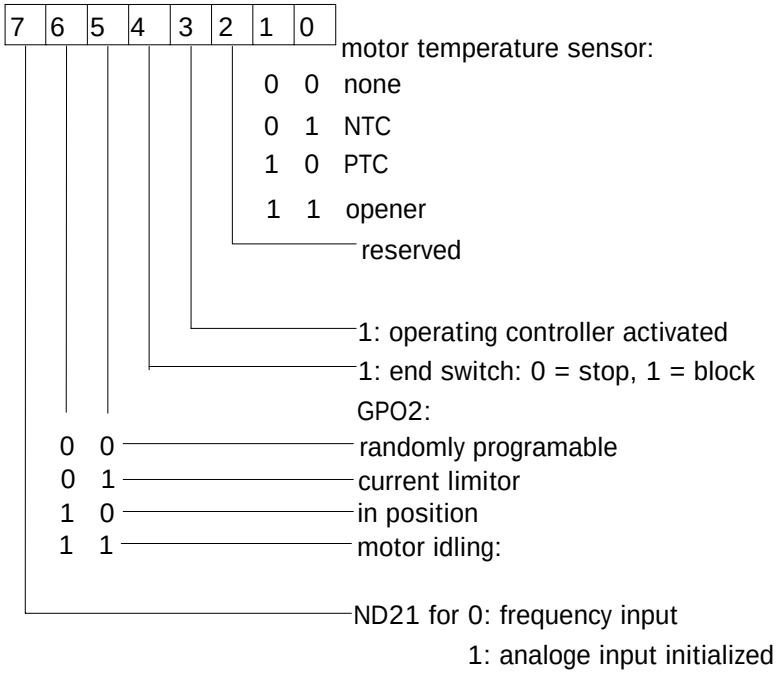
Bit 7: Adjustment of the ND21 speed range.

14.3.3. The Byte HwVersion

RAM-parameter Address: H'FF60
EEPROM: H'20

This byte is a read and a write byte. The byte has been modified with the updating of ND21.

New version after H8 version 01.07.93.

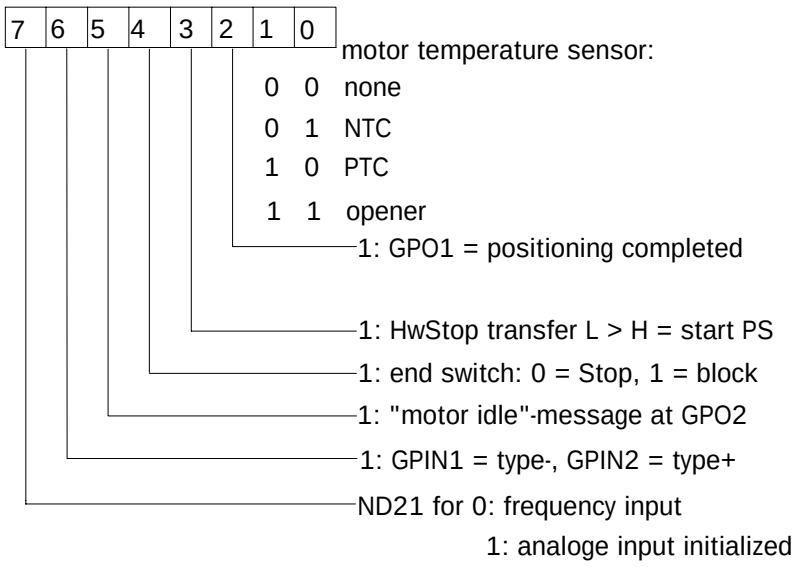


Remarks:

Bit 0 and 1:	Indicates the ND21 which temperature sensor is in the motor.
Bit 2:	1: Reserved.
Bit 3:	With 1 the built-in cycle control will be activated.
Bit 4:	Determines the reaction of the activated end switch. With 0 the drive will stop, with 1 the regulator will be blocked.
Bit 5 and 6:	These two bits determine which message will be switched to the GPO2 output.
Bit 7:	Determines the initialization of the 16 bit counter in ND21. With 1 the ND21 will be initialized for the analogue input and with 0 the frequency input.

After HwVersion was changed, a resetting of the motor must be made with [R] in the service menu, to allow all changes to become active (previously stored in EEPROM).

Old Version: Up unitl H8 version 30.06.93:



Remarks:

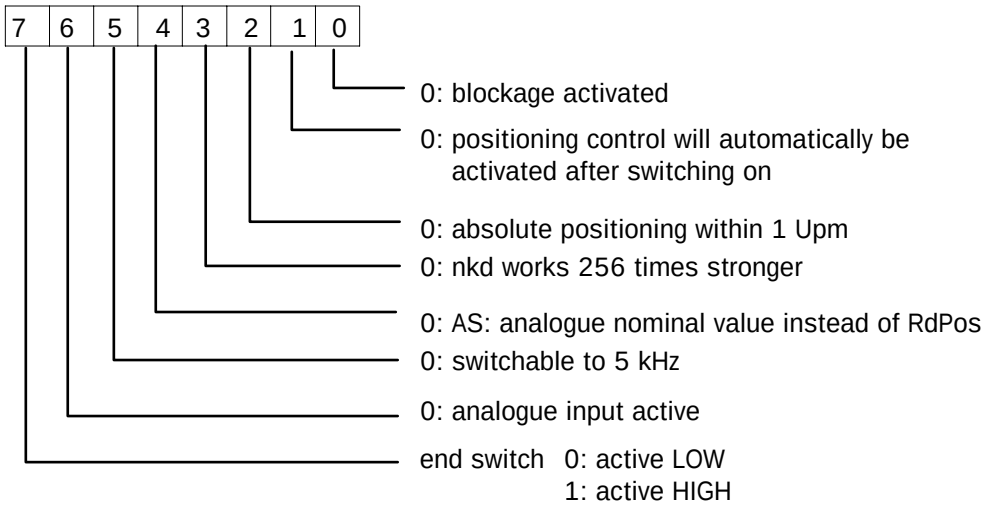
Bit 0 and 1:	Indicates the ND21 which temperature sensor is in the motor.
Bit 2:	With 1 the output GPO1 for the message „position reached“ will be used in the position controller. If bit 2 = 0, this output can be used for other functions.
Bit 3:	If this bit is set, the positioning procedure will be activated by an increasing flank on the Start-Stop input (GPO3).
Bit 4:	Determines the reaction of the activated end switch. With 0 the drive will stop and with 1 the regulator will be blocked.
Bit 5:	If this bit is set, and the motor is idling, the ND21 will send an „Idle“ message to the output GPO2.
Bit 6:	If this bit is set, the inputs GPIN1 and GPIN2 will be used for the step function of the positioning controller.
Bit 7:	Determines the initialization of the 16 bit counter in ND21. With 1 the ND21 will be initialized for the analogue input and with 0 for the frequency input.

After the HWVersion was changed a resetting of the motor must be made in the service menu [R] to allow all changes to become active (previously stored in EEPROM).

14.3.4. The Byte HwVersion2

RAM-parameter Address: H'FF61
EEPROM: H'21

The byte is a read and write byte.



Remarks:

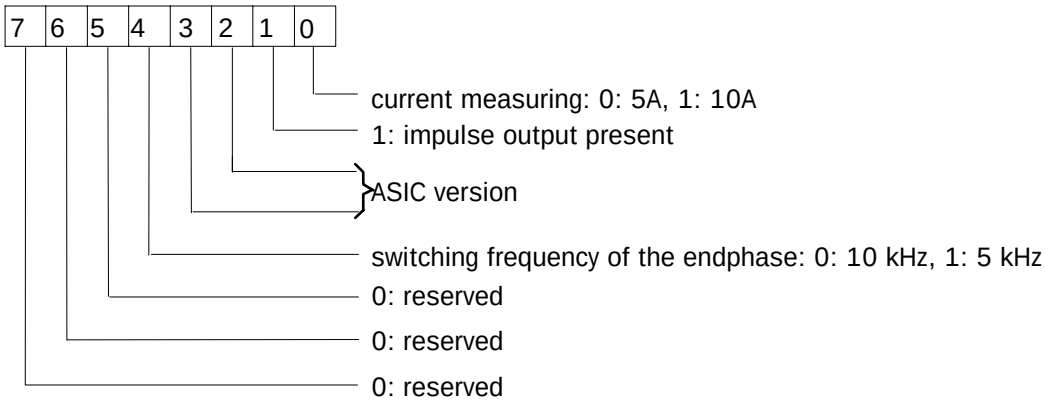
Bit 0 - 4:	Reserved for future expansions, set all to 1.
Bit 5:	With = the switching frequency of the output transformer can be reduced to 5 kHz.
Bit 6:	If not set, the analogue output will be activated. In this case, and at this time, GPO1 can not be used as digital output.
Bit 7:	Sets the active level of the end switch.

Note: The functions of bit 0 to 4 are available from H8-version 3.3 of 30.12.94.

14.3.5. The Byte EEVersion

EEPROM-parameter EEPROM: H'0C

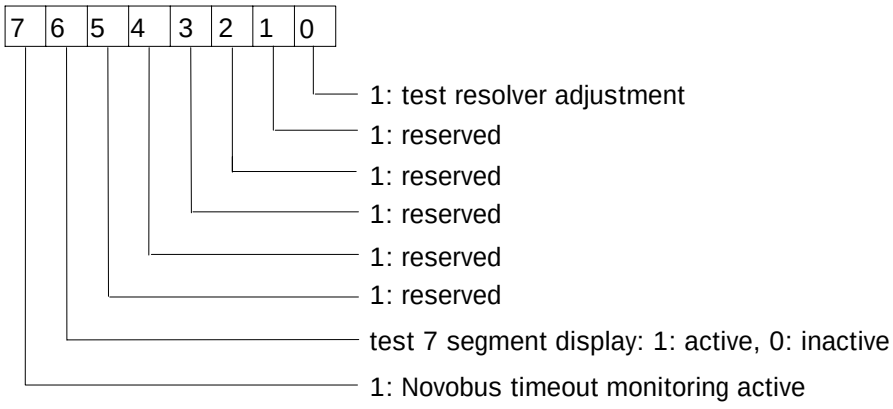
This byte is a read only byte.



14.3.6. The Byte EESelftest

EEPROM-parameter EEPROM: H'47

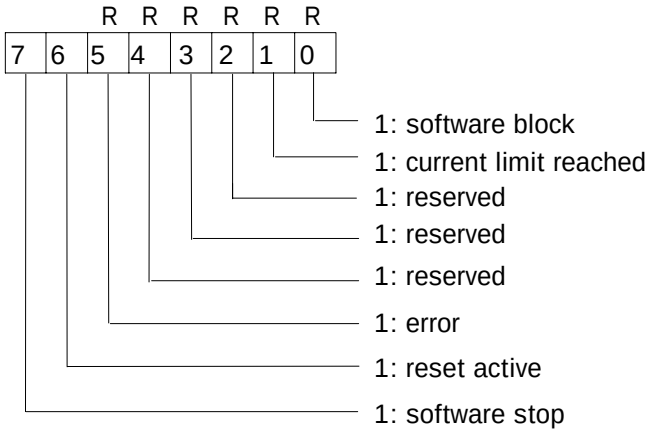
This is a read and write byte.



14.4. ND21 Status

14.4.1. The Byte Status

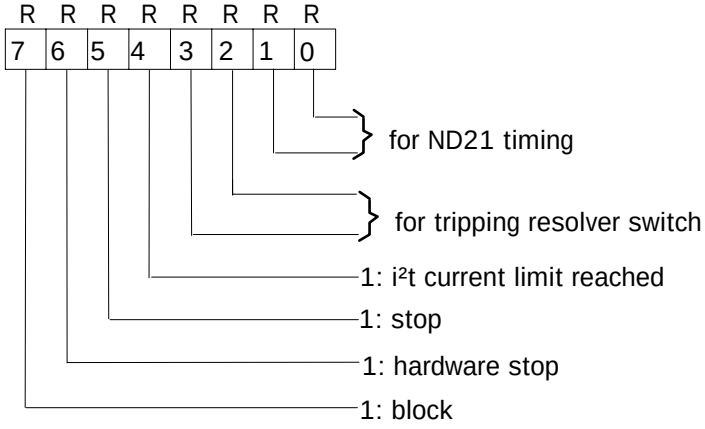
RAM cell address H'FF00



Remarks: R: These bites are read only bites.

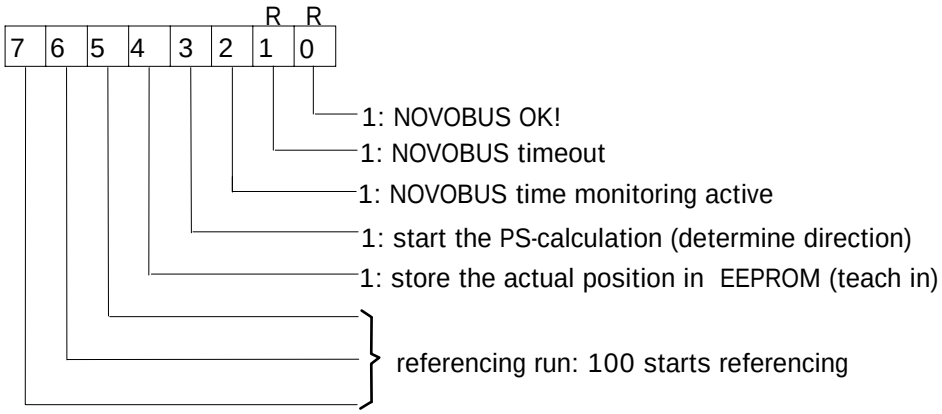
14.4.2. The Byte Flags

RAM cell address H'FF56



The byte flag is a read only byte.

14.4.3. The Byte Flags2



Remarks: R: These bites are read only bites.

14.5. Current Actual, Nominal and Limit Values

14.5.1. Currency

14.5.1.1. Actual and Nominal Values

These values are read only values.

Motor current-actual values:	iaist:	Address H'FF06
	ibist:	Address H'FF07
Motor current-nominal values:	iasoll:	Address H'FF04
	ibsoll:	Address H'FF05

Data format: Byte, double complemented.

Scale	**5605**	**5610**	**5615**	**5620**
A_{eff} per bit	0,122	0,244	0,355	0,488

14.5.1.2. Current Limit Values

These values are set by the user.

Peak current:	imax:	Address H'FF66 EEPROM H'26
Rated current:	i0:	Address H'FF67 EEPROM H'27

Data format: Byte

Scale	**5605**	**5610**	**5615**	**5620**
A_{eff} pro Bit	0,086	0,172	0,251	0,345
Maximum	H'74	H'74	H'52	H'74

14.5.2. Inertia

These values are read only bytes.

Inertia nominal value:

msoll: Address H'FF02

Date format: Word, double complemented.

Inertia limit:

mmax: Address H'FF01

Data format: Byte, double complemented.

Scale:	**5605**	**5610**	**5615**	**5620**
mmax A_{eff} per bit	0,086	0,172	0,251	0,345
msoll mA_{eff} per bit	0,336	0,671	0,980	1,347

14.5.3. Spin Speed

14.5.3.1. Actual and Nominal Value

The values nist and nsollrampe are read only values.

Spin speed actual value:	nist	Address: H'FF0C
Spin speed nominal value:	nsoll	Address: H'FF08
Spin speed nominal value after ramp:	nsollrampe	Address: H'FF0A

Data format: Word, double complement.

Scale: 0,264909525 rpm per bit for SwVersion2 bit7=1
0,52981905 rpm per bit for SwVersion2 bit7 = 0

14.5.3.2. Limit Values

This value is programmed by the user.

Maximum speed:

nMax: Address: H'FF68
EEPROM: H'28

Data format: Word

Scale: See nominal and actual values.

14.5.4. Ramping

Ramps are programmed by the user.

Acceleration ramp:	nRampe+	Address: H'FF6C EEPROM: H'2C
Braking ramp:	nRampe-	Address: H'FF6E EEPROM: H'2E

Data forma: Word

Scale: 64,18332625 rad/s² pro Bit für SwVersion2 Bit 7 = 1
128,3666525 rad/s² pro Bit für SwVersion2 Bit 7 = 0

14.5.5. Rotation

Nominal values can be programmed by the user.

Rotor position-nominal value	lagesoll	Address: H'FF14
Rotation-nominal value	umdrsoll	Address: H'FF12
Rotor position-nominal value	lageist	Address: H'FF18
Rotation position-actual value	umrdist	Address: H'FF16

Data format: umdrsoll and lagesoll as well as umdrist and lageist each one builds together a 32 bit double word. Double complemented.

14.5.6. Slipping Errors

By operation with position regulators, slipping monitoring can be used. When slipping occurs the endphase will be blocked and an error message H'700 will be generated.

Slipping error Address: H'FF76
EEPROM: H'36, 37

Data form: Word

Scale: Deactivates if H'8000 is listed 1 bit means 0.024543692 rad.

14.5.7. Temperatures

14.5.7.1. Cooler Temperatures

These values are read only values.

Cooler temperature	tempKK	Address: H'FF26
Over temperature threshold	MaxTempKK	Address: H'FF64 EEPROM: H'24

Data format: Byte

Scale: $$temp1 = \frac{tempKK}{256 - tempKK} \cdot 0{,}27$$

Herewith: Cooler temperature (°C]:

$$T = \frac{1}{\frac{\ln(temp1)}{3600} + \frac{1}{298}} - 273$$

14.5.7.2. Motor Temperature

The motor temperature (actual value) is a read only value. The motor temperature limit value can be read and written.

Motor temperature:	tempMot	Address: H'FF27
Temperature threshold:	MaxTempMot	Address: H'FF65 EEPROM: H'25

Data format: Byte

Scale: The resistance value of the motor temperature sensor is calculated as follows:

$$R = \frac{tempMot \cdot 1227067 - 7180800}{-1 \cdot tempMot \cdot 1298{,}5 + 326400}$$

The motor temperature can be calculated from the characteristics of the sensor.

14.6. Regulator Parameter

All regulator parameters can be changed by the user.

14.6.1. Current Regulator

ND21 works with two PI current regulators.

P-part ikp:	a_mpr1	Address: H'FF80 EEPROM: H'41
	a_mpr98	Address: H'FF82 EEPROM: H'43

Data format: 10 bit without prefix. Lower 8 bit in a_mpr1, upper 2 bits: Bit 7 and 6 of _mpr98.

I-part iki: a_mpr2 EEPROM: H'42

a_mpr98 EEPROM: H'43

Data format: 10 bit without prefix. Lower 8 bit in a_mpr2, upper 2 bits: Bit 5 and 4 of _mpr98

Changes in EEPROM at a_mpr1, a_mpr2 and a_mpr98 only become active after reset.

The current regulator parameter may only be changed in EEPROM (or with the ND21 setup software). They become active after a reset. Should it be necessary to change these parameters during operation, please contact Novotron for detailed information.

14.6.2. EMF-Compensation

The EMF (electro-magnetic-field) compensation performs a pre-controlling of the voltage to compensate the opposing EMF of the motor.

EMF-compensation: emk0 Address: H'FF7A
EEPROM: H'3A

Data format: Byte, no prefixes.

Scale: emk0 = 0.38636 · (voltage radiant of the motors in V/1000Upm)

14.6.3. Tacho Filter

With the tacho filter it is possible to filter the actual spin speed value.

Tacho filter: nFilter Address: H'FF79
EEPROM: H'FF39

Data format: Byte, no prefixes, range: H'00 - H'7F.

Scale: $$\frac{432\mu s}{1-\frac{nFilter}{128}}$$

14.6.4. Spin Speed Regulator

ND21 works with a PI speed regulator.

P-part	nKp	Address: H'FF70 EEPROM: H'30
I-part	nKi	Address: H'FF71 EEPROM: H'31

Data format: Byte, no prefixes, range: H'00 - H'7F.

14.6.5. Position Regulator

ND21 works with a PD regulator.

P-part	LKp	Address: H'FF72 EEPROM: H'32
D-part	LKd	Address: H'FF73 EEPROM: H'33

Data format: Byte, no prefixes, range: H'00 - H'7F.

14.6.6. Resolver Adjustment

To achieve a correct commutation, the resolver mounting position can be moved electronically.

Resolver setting dRESOLVER Address: H'FD98
EEPROM: H'52,53

Data fomat: Word, higher value byte in H'52

Scale: 1 bit means: 1 rotation / 65536

14.6.7. Motor Poles

It is possible to program the number of motor poles in the ND21 (2 to 12).

Motor polel: EE_pole EEPROM: H'46

Data format: Byte, no prefixes, value range: H'02, H'04, H'06, H'08, H'0A, H'0C.

14.7. Signal In- and Ouputs

14.7.1. Digital Signals

Table of the ND21 digital inputs:

Digital input	Memory cell	Bit	Port	X3 Pin	0V =
GPIN1	H'FFB3	4	2	21	"1"
GPIN2	H'FFB3	5	2	6	"1"
GPIN3	H'FFB3	6	2	22	"1"
GPIN4	H'FFB3	7	2	7	"1"
GPIN5	H'FFBE	3	7	24	"1"
GPIN6	H'FFBE	7	7	8	"1"
GPIN7	H'FFBB	0	6	10	"1"
GPIN8	H'FFBB	2	6	25	"1"

End switch: H'FFB3 0,1,2,3 2.

The following codes apply to the end switch:

(0 = 0V, 1 = 5 - 24V) X3 Pin

End switch N	0	1	0	1	3
End switch P	0	0	1	1	34

H'FFB3Bit 0	0	0	1	0
H'FFB3 Bit 1	0	1	1	1
H'FFB3 Bit 2	1	0	0	1
H'FFB3 Bit 3	1	1	1	1

Note: If an internal error is present, the end switch code will be overwritten by an error code.

Addresse digital outputs:

Digital output	Memory cell	Bit	Port	X3 Pin	0V =
GPO1	H'FFB7	7	4	31	"0"
GPO2	H'FFB7	5	4	23	"0"

The signal conditions can be read from the respective memory cells. Special NOVOBUS commands are provided for writing directly to the GPOs.

14.7.1.1. Encoder Emulation

Impulse number EE_impulszahl
EEPROM: H'44; 45

Data format: Word, no prefixes, range H'0001 to H'0400, higher value byte in H'44. The new impulse number is active after resetting.

14.7.2. Analogue Input

ND21 is programmed on analogue input with HwVersion (H'FF60) bit 7 = 1. SwVersion (H'FF62) bits 1,0 = 00 selects the analogue input for nominal value settings. Here it is necessary to save HwVersion on the EEPROM and activate it with a reset.

Nominal value scale VCOlin Address: H'FF78

EEPROM: H'38

Scale: approx. VCOlin = (speed[rpm] at 10V)/60

Offset: VCOoffset

Address: H'FF22

The offset will automatically be calibrated every second.

14.7.3. Frequency Input

ND21 is programmed on frequency input with HwVersion (H'FF60) bit 7 = 0. SwVersion (H'FF62).

Bits 1,0 = 10 selects the frequency input for nominal value settings. It is necessary to save HwVersion on the EEPROM and activate it with a reset. The ramp must be deactivated for frequency settings.

nRampe- (H'FF6E) = nRampe+ (H'FF6C) = H'7FFF.

The position regulator has to be activated:

SwVersion (H'FF62) Bit 6 = 1. nominal value scale:

VCOlin Address: H'FF78
EEPROM: H'38

VCOlinL Address: H'FF7F
EEPROM: H'3F

Scale: VCOlin x 256 + VCOlinL = 33554432 / (Impulse per rotation)

14.7.4. Analogue Output

The ND21 has an 8 bit analogue output. The analogue output can only be used when GPO1 is not used. Activate the analogue output with:

HwVersion2 (H'FF61) Bit 6 = 0.

Signal output:

AnOut Address: H'FF7C
EEPROM: H'3C

A signal address is to be inserted in the range of H'FFXX.

Example: For nist (H'FF0C) the value H'0C.

Offset:

AnOutOffset Address: H'FF7D
EEPROM: H'3D

14.7.5. GPO2

For GPO2 ther are 4 programming possibilities:

HwVersion (H'FF60)	Bit 6	Bit5
Randomly programmable	0	0
Current limit reached	0	1
In position	1	0
Motor idling	1	1

14.7.5.1. Randomly Programmable

GPO2 can be set to 0 or 1 with the NOVOBUS.

14.7.5.2. Current Limit Reached

GPO will be set to 1, when the motor current is limited by the current limiting.

14.7.5.3. In Position

(see chapter 13.1.4. Programming of ND21)

14.7.5.4. Motor Idling

With the parameter window (H'FF7B) a minimum of speed can be programmed. When exceeded GPO1 will be set to 1. Hereby the window will be compared with the lower value byte of nist.

14.8. Drive Information

14.8.1. Serial Number

The serial number can be read out of the memory cell H'01, 02, 03.

Data format: BCD, *High* value byte H'01.

14.8.2. Operating Hours

Operating hours can be read out of the EEPROM.

Aktive time: EEPROM: H'05, H'06.

Data format: Word, high value byte in H'06.

Time blocked: EEPROM: H'08, H'09.

Data format: Word, high value byte in H'09.

14.8.3. Start Up Date

The start up date can be read out of the EEPROM.

EE_InitDatum
EEPROM: H'0A, H'0B

Data format: Word, high value byte H'0A.

Bits 0 - 4: Day

Bits 5 - 8: Month

Bits 9 - 15: Year

The calendar year is achieved by adding 1993 to the calendar year.

14.8.4. Date of the Last Parameter Change

This date can be read out of the EEPROM. This date will be updated in the EEPROM with the save function Save! or ProgParam. A user program for changing EEPROM data should also always actualize this date.

EE_ProgDatum
EEPROM: H'58, H'59.

Data format: Word, high value byte H'58.

Meaning of the bits same as EE_InitDatum.

14.9. Controlling of ND21 via NOVOBUS

With the exception of a few special commands i.e. reset, controlling via NOVOBUS is performed principally by reading and writing memory cells in the RAM of the micro controller H8.

14.10. Operating Modes

Disable: Set bit 0 Status (H'FF00) to 1.

Enable: Set bit 0 Status (H'FF00) to 0.

Stop: Set bit 7 Status (H'FF00) to 1.

Continue: Set bit 7 Status (H'FF00) to 0.

Reset: Run NOVOBUS-command resetH8.

The respective modes can be read from the memory cells: Status (H'FF00), Flags (H'FF56) and Flags2 (H'FF57).

14.10.1. Error Mode

Reques error mode:	Read bit 5 Status (H'FF00).
Error code:	Errorcode, Adresse: H'FD82.
Data format:	12 bit, high valaue byte H'FD82.
Reset error:	Write H'AF in RAM cell error code (H'FD82).

14.10.2. NOVOBUS Timeout

The NOVOBUS time-tout-monitoring is actvated Flags2 (H'FF57) Bit 2 = 1 active. This bit is initialized from the EEPROM with bit 7 with EE_selftest (H'47).

The time-out-monitoring will be activated if no data has been received by the NOVOBUS after at least 10 ms and no later than 20 ms. It makes no difference if the data were determined for this drive or not.

14.10.3. Exchange Nominal and Actual Value

There are two possibilities of defining NOVOBUS nominal values and to read back actual values.

1. Write the nominal values in the parameter channel with the respective NOVOBUS-commands. The actual values can be read out of the respective memory cells **(see chapter 11 NOVOBUS PC Driver).**

2. Faster exchange of nominal and actual values in the process data channel.

In each telegram of the process data channel 2 bytes are received and 2 bytes are sent.

Receive address DataIn Address: H'FF32
Send address DataOut Addresse: H'FF34

Incoming data are written in the RAM memory cell H'FFXX. The address of this cell is written in DataIn.

Example: Incoming data mean the nominal value for spin speed: nsoll has written the address H'FF08 or H'08 in DataIn.

The sent data are taken from the memory cells in the RAM range H'FFXX. The address for these cells are written in DataOut.

Example: The sent data indicating lageist has written the address H'FF18 or H'18 in DataOut.

14.10.4. Positioning Controller

(see chapter 13. The ND21-Positioning Controller (PC1.0))

14.10.5. EEPROM Controller

14.10.5.1. EEPROM Commands

EEPROM commands are written in the following memory cell of ND21:
EEPROMcontrol Address: H'FD8A

Command	Code
Read Byte	H'81
Write Byte	H'82
ProgParam	H'84

14.10.5.2. Read Byte

Write the EEPROM address of the byte in the EEPROMbuffer (H'FD88). Write H'81 in EEORIMcontrol. Wait until bit 5 accepts the value 1 in EEPROMcontrol. Read the resulting byte from EEPROMbuffer. Write H'10 in EEPROMcontrol.

14.10.5.3. Write Byte

Write the EEPROM address of the byte in EEPROMbuffer (H'FD88). Write the data byte in the memory cell with the address H'FD89. Write H'82 in EEPROMcontrol. The writing is completed as soon as bit 4 accepts value 1 in EEPROMcontrol.

14.10.5.4. ProgParam

It is possible herewith to copy the regulator parameter (memory range H'FF60 to H'FF7F) into the EEPROM with only one command. Therefore write H'84 in EEPROMcontrol. The copy procedure is completed as soon as bit 4 in EEPROMcontrol accepts value 1.

14.10.6. Oscilloscope

14.10.6.1. Signal Selection

Signals located in RAM range H'FFxx can be selected. The signal selection is performed by writing the lower value byte of the RAM address as follows:

Signal1 in scope_signal1 H'FF40,

Signal2 in scope_signal2 H'FF41 or

Trigger signal in scope_trigger H'FF3E,

Example: H'0C in the RAM cell H'FF40 designates that Signal1 has been selected for the actual spin speed value (RAM cell H'FF0C).

If only one signal is to be seen, write the same address in scope_signal1 and in scope_signal2.

The contents of the selected memory cells will be recorded. The meaning and the scale of the recording signal are taken from the selected signal.

Example: If with H'06 in RAM cell H'FF40, iaist as Signal1 has been selected:
iaist: Motor current actual value phase A,
Double complemted data format, with ND21 5610 that would be 0,244 A per bit.

If, i.e. a value from H'4C is recorded, this would result in a current value of +18,5A. With H'C8 the value would be 13,7A.

The signal selecting must be made with a blocked recording.

14.10.6.2. Time Basis

The scanning rate of the memory oscilloscope is written in the RAM cell scope_timer0 H'FF39. That means: After each scan period the measured value is stored.

T_{AB} = 108 µs · scope_timer0

A total of 256 measurements will be read.

Example: scope_timer = 0: Every 108 µs a measurement will be stored in the scope buffer. The full scope buffer will then be stretched over a time frame of 256 · 108 µs = 27,648 ms.

14.10.6.3. Trigger Threshold

Die Trigger threshold is written in the RAM cell scope_level H'FF3F with a inverted bit 7.

Example: iaist is to be triggered at -7 A. 7A on ND21 5610:7/0,244=29bit=H'1D. -7 would be: -H'1D=H'E3. Bit 7 inverted: From H'E3, H'63 will be written in RAM cell scope_level.

The trigger threshold can be changed at anytime.

14.10.6.4. Trigger Delay

For the trigger delay the RAM cells scope_delay and scope_counterH'FF3A must be programmed. both RAM cells must be programmed with each new recording. Both values are figured as follows:

The desired delay time must be provided: T_{VZ}

scope_counter H'0100 + T_{VZ} / T_{AB} für T_{VZ} >= 0

scope_delay H'00 für T_{VZ} < 0:

scope_delay $-T_{VZ}$ / T_{AB}

Example: Scan time 108µs, delay time -5ms, that means there are 46 scanning points before and 210 scanning intervals after the triggering of the recording.
scope_delay 46 = H'2E,
scope_counter H'0100 - H'2E = H'00D2

14.10.6.5. Scope Status

The byte scope_status H'FF38 controls the recording:

Bit 7:	Identifies the condition of the oscilloscope.
Bit 7 = 0:	Recording runs.
Bit 7 = 1:	Recording stopps.
Bit 6:	Identifies the trigger status: 0: No triggering occured 1: Triggering occured
Bit 5:	Trigger status to be initialized with 1.
Bit 4:	Trigger flank.
Bit 3 .. 0:	Reserved = 0

At the end of a recording bit 7 will automatically be set to 0. A recording can also be stopped by setting bit 7 to 0.

14.10.6.6. Recording Procedure

When recording is stopped scope_delay and scope_counter are loaded. Afterwards scope_status will be overwritten with H'20 for negative flank and with H'and herewith the recording begins. First the delay time runs down. Next the triggering conditions will be checked.

Bit 5 is set to 0 as soon as the trigger signal is below the triggering threshold for positive flank or above the triggering threshold for negative flank. Bit 5 will return to 1 as soon as the trigger threshold is again passed. This activates the trigger signal. Bit 6 will be set to 1. As soon as triggering has occurred the programmed time in scope_counter will begin to run.

After the time runs down bit 7 in scope_status will be stopped and set to 1. The recorded data is now in scope_puffer and is available for reading. The even addresses in the scope buffer contain data from Signal1 and the odd addresses contain data from Signal2.

The data ranges are as follows:

Beginning: H'FDA0 + scope_pointer + 1
To: H'FE9F
From: H'FDA0
End: H'FDA0 + scope_pointer.

14.10.6.7. Auto Trigger

A recording without trigger can be made by setting bit 7 scope_status to 0 for the duration of a recording. After the recording set scope_status back to 1.

14.11. Error Codes

14.11.1. H8-Port-Error

P10 H'0010

P11 H'0011

P12 H'0012

P13 H'0013

P14 H'0014

P15 H'0015

P16 H'0016

P17 H'0017

P20 H'0020

P21 H'0021

P22 H'0022

P23 H'0023

P24 H'0024

P25 H'0025

P26 H'0026

P27 H'0027

P30 H'0030

P31 H'0031

P32 H'0032

P33 H'0033

P34 H'0034

P35 H'0035

P36 H'0036

P37 H'0037

P40 H'0040

P41 H'0041

P42 H'0042

P43 H'0043

P45 H'0045

P46 H'0046

P47 H'0047

P50 H'0050

P51 H'0051

P52 H'0052

P60 H'0060

P61 H'0061

P62 H'0062

P63 H'0063

P64 H'0064

P65 H'0065

P66 H'0066

P67 H'0067

P70 H'0070

P71 H'0071

P72 H'0072

P73 H'0073

P74 H'0074

P75 H'0075

P76 H'0076

P77 H'0077

P80 H'0080

P81 H'0081

P82 H'0082

P83 H'0083

P84 H'0084

P85 H'0085

P86 H'0086

P90 H'0090

P91 H'0091

P92 H'0092

P93 H'0093

P94 H'0094

P95 H'0095

P96 H'0096

P97 H'0097

14.11.2. EEPROM-Error

EEPROMrdwr	H'0100
EEPROMvoid	H'0101
EEPROMparity	H'0102
EPROMwr	H'0103
EEPROMbus	H'0104

14.11.3. A/D-Error

TempMot	H'0110
TempKK	H'0111
Strom A	H'0112
Strom B	H'0113
BitSplitting	H'0114
BitSplittingA	H'0115
BitSplittingB	H'0116

14.11.4. Error Analogue Input

AnInput0V	H'0120
AnInput10V	H'0121
AnInput12V	H'0122

14.11.5. Resolver-Error

ResolverIC	H'0130
RslvUeberw	H'0131
RAMSt	H'0132
RslvJustage	H'0133
ResolvKabel	H'0134
Rslv	H'0135

14.11.6. Watchdog-Error

WdogHw	H'0140
IntWdog	H'0141

14.11.7. ASIC-Error

ASICfr	H'0150
ASICread0	H'0151
ASICrdwr	H'0152
ASICtype	H'0153
ASICram	H'0154

14.11.8. Modulator-Error

PhaseABC	H'0160
PhaseA	H'0161
PhaseB	H'0162
PhaseC	H'0163

14.11.9. Cycle-Error

PC	H'0201
IT	H'0202
reset	H'0203

14.11.10. H8-Register-Error

SP	H'0210
SYSCR	H'0211
IER	H'0212
ISCR	H'0213
TMR0	H'0214
TMR1	H'0215
FRC	H'0216
CCR	H'0217
RAM	H'0218
R0	H'0220
R1	H'0221
R2	H'0222
R3	H'0223

14.11.11. Motor-Current-Errors

imax	H'0250
ia	H'0251
ib	H'0252
ic	H'0253

14.11.12. PAL-Errors

PAL	H'0300
PALfrgabe	H'0301
5V	H'0302
15V	H'0303
ZkUeberSp	H'0304
ZkUnterSp	H'0305
Wdog	H'0306
Endstufe	H'0307
UeberStrom	H'0308
Resolver	H'0309
EndSchaltP	H'0310
EndSchaltN	H'0311
EndSchalter	'0314

14.11.13. Temperature-Errors

OverTempKK	H'0400
OverTempMot	H'0401
OverTempKK_	H'0410

14.11.14. Serial-Interface-Errors

SCIoverrun	H'0501
SCIparity	H'0502
SCIframing	H'0503
NBsync	H'0504
NBcommand	H'0505
NBparam	H'0506
NBchecksum	H'0507
NBtimeout	H'0508
Bextern	H'0509
NBdata	H'0510

14.11.15. Positioning-Controller-Errors

PS	H'0600
DisPS	H'0601

14.11.16. Regulator-Errors

Schleppfehler	'0700
Strom0	H'0701

14.11.17. Nominal-Value-Error

ImpulsEingang	H'0800

Appendix

A1. Number Formats

A1.1. Number Systems

A single number can be presented in various number systems.

A1.1.1. The Decimal System

The decimal system based on 10, also called powers of 10, is the counting system which is mostly used by people for calculating. The decimal system uses numbers from 0 - 9. These numbers are grouped together for making larger numbers. The number to the right is the smallest digit. It indicates the ones of a number. The next number from the right indicates the tens, the next the hundreds, then thousands etc. Each position is increased by a power of ten. With a 3 digit decimal number the numbers 000 - 999 can be indicated.

A1.1.2 The Binary System

The binary system also known as the two´s system. This is the number system used by all digital calculators. It uses only the digits 0 and 1. To create larger numbers the digits are columened togeteher. The right most number is the lowest digit, which indicates the ones. The next number indicates the twos, the next the fours, then the eights, then sixteens etc. Each position represents an additonal power of twos. A three positon binary number can indicate the numbers 000 - 111. For example 101, 1 ones + 0 twos + 1 fours = 5 in the decimal system.

A1.1.3 The Hexadecimal System

The hexadecimal system is also known as sixteenth's system. Using the binary system large numbers are long and complex to write. For this reason the hexadecimal system is used in its place. Four digits (or bits) in the binary system are the same as one digit in the hexadecimal system. The hexadecimal system uses the digits and characters 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Characters A - F represent the numbers 10 - 15.

To make large numbers, the digits and charachters are grouped together. The number further to the right is the lowest number. It indicates a ones number. The next number names the 16, the next a 256 then 4096, 65536 etc. Each position represents an exponent of 16. A three position hexadecimal number can indicate the numbers 000 - FFF. For example 3E1, 1 ones + 14 16s + 3 256s = 993 in the decimal system.

A1.2. Numbers in Digital Computers

A1.2.1 Bit

Smallest unit. A bit can contain the values 0 and 1.

A1.2.2 Byte

8 bits. Eight digit binary number. The eight bits are numbered for 0 - 7. The lowest bit is bit number 0.

A byte can also be written as 2 position hexadecimal number. With one byte numbers from 0000 0000 upto 1111 1111 (meaning H'00 to H'FF in hexadecimal or 0 - 255 in the decimal system) can be performed.

A1.2.3 Word

16 bit, 2 byte. A word contains the bits 0 to 15. A word is a 16 digit binary number, which indicates a 4 position hexadecimal number. With a word you can indicate the numbers from 0000 0000 0000 0000 to 1111 1111 1111 1111 (meaning H'0000 to H'FFFF Hex or 0 to 65535 decimal).

A1.2.4 Double Word

32 bit, 4 byte. A double word contains the bits 0 to 31. It is a 32 digit binary number, that would be an eight position hexadecimal number. With a double word you can indicate the numbers 0000 0000 0000 0000 0000 0000 0000 0000 to 1111 1111 1111 1111 1111 1111 1111 1111 (meaning, H'00000000 to H'FFFFFFFF Hex or 0 to 4294967295 decimal).

A1.3. Negative Numbers

Up to now only positve numbers with byte, word and double word have been described. To indicate negative numbers you have to define:

For: Byte 0 - 127 positive

128 - 255 negative, number value -128 - -1

Word: 0 - 32767 positive

32768 - 65536 negative, number value -32768 - -1

Double word: 0 - 2147483647 positive

2147483648 - 4294967295 negative,

Number value -2147483647 - -1

This means for exampe with byte:
When you add 0 1 the result is 0000 0001 or 1. When you subtract 0 1 the result is 1111 1111 (H'FF or 255 decimal), according to the above definition -1.

A number is made negative by building the so called two's complement. A complement of a binary number is attained by making a zero out of every one, and a one out of ever zero. The two's complement is achieved, when after building the complement a one is added.

In the example above: 1 = 0000 0001

Complement: 1111 1110 **+ 1:** 1111 1111 = -1

Dezimal 25= x 128 + 0 x 64 + 0 x 32 + 1 x 16 + 1 x 8 + x 4 0 x 2 1 x 1 = 0001 1001

Complement: 1110 0110 **+ 1:** 1110 0111 = - 25

Note: Building of two's complement must always be used on all bits. This means, for numbers that are indicated in a byte, 8 bit complementing must be used, for numbers that are indicated with word 16 bit must be used and for double word 32 bit.