HEIDENHAIN

HEIDENHAIN
Press any key to continue.
Power was off!
ND 287
Software Version : V1.00 ID 646118-01
LANGUAGE
HELP
ENTER

User‘s Manual

Ethernet Module
Ethernet Driver Software

ND 287

English (en)
December 2009

Firmware version

This document describes for the ND 287 the firmware version 646118-04 and later
and the driver software with ID number 730348-01.
Older firmware versions do not support the Ethernet data interface.

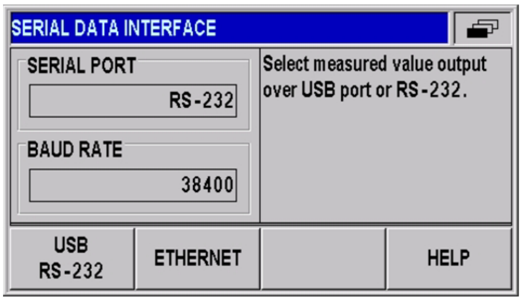
Part 1: Configuring ND 287 for the Ethernet Data Interface

After mounting the optional Ethernet module (see Part 3: Appendix) you can configure the Ethernet interface on the ND 287. The Ethernet data interface can be configured both with the keyboard on the ND 287 and by means of an Ethernet driver on a PC.

Setting up the Ethernet interface

The ND 287 features the following serial interfaces :
RS-232-C/V.24 (X31), USB (UART, X32) and optionally **Ethernet (X26).**
You can use the interfaces to complete the following tasks:

- Sending measured values, error compensation tables or configuration files to a computer.
- Receiving error compensation tables or configuration files from a computer.
- In addition, it allows external operation of the ND 287 via the data interfaces.



SERIAL DATA INTERFACE

SERIAL PORT: RS-232

BAUD RATE: 38400

Select measured value output over USB port or RS-232.

Figure 1: Ethernet data interface

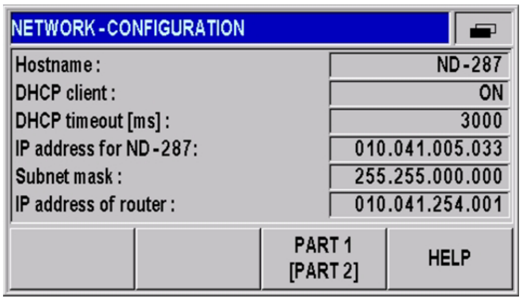
Setting up the interface

- In the *INSTALLATION SETUP* menu, select INTERFACE SETTINGS.
- In the INTERFACE SETUP menu you can use the **ETHERNET** menu to get to the Ethernet settings (soft key appears only if the Ethernet module is connected).

- Host name: This is the name used by the ND 287 in the network. This entry is only required if you are working with DHCP. The default name is "ND-287". If you want to change this name, you can enter a four-digit value with the numeric keys. The first part of the name appears as default as "NC2-" and cannot be changed.

- DHCP client: With the DHCP (Dynamic Host Configuration Protocol), the ND 287 gets its IP address, subnet mask and router address from a DHCP server. The DHCP server identifies the ND 287 by its hostname. You can switch the DHCP on or off with the **ACTIVATE** and **DEACTIVATE** soft keys. You cannot change the IP address, subnet mask and router address when DHCP is active. The default setting is DHCP off.

- DHCP timeout: Here you can use the numeric keys to enter the time allowed for the ND 287 to wait for an answer from the DHCP servers. The default value is 3 seconds (30 000 ms).



NETWORK-CONFIGURATION

Hostname:	ND-287
DHCP client:	ON
DHCP timeout [ms]:	3000
IP address for ND-287:	010.041.005.033
Subnet mask:	255.255.000.000
IP address of router:	010.041.254.001

Figure 2: Ethernet parameters, part 1

- IP address: Use the numeric keys to enter the IP address for the ND 287. The default IP address is 000.000.000.000.

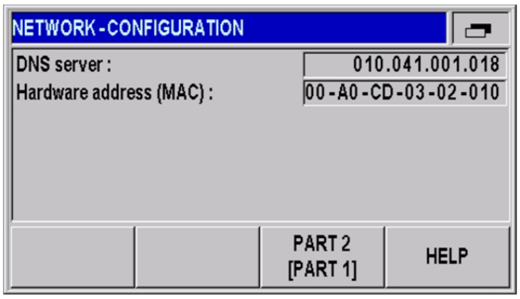
- Subnet mask: Use the numeric keys to enter the subnet mask used by your network.
  The default subnet mask is 255.255.000.000.

- IP address of router: Use the numeric keys to enter the IP address of the router used.
  The default router IP address is 000.000.000.000

- DNS server: This field displays the IP address of the DNS servers used. This address is displayed only when DHCP is active, and it cannot be changed.

- Physical address: This field displays the MAC address of the Ethernet module. This address cannot be changed. Each Ethernet module has its own, unique MAC address.

> ⚠ Any changes you make in the network settings do not become effective until you have switched the position display unit off and back on again.


NETWORK-CONFIGURATION

DNS server : 010.041.001.018

Hardware address (MAC) : 00-A0-CD-03-02-010

Figure 3: Ethernet parameters, part 2

Part 2: Driver Software

1 General Information

Functions are provided for accessing the ND 287 from a software application. This group of functions is supplied as a DLL for Windows systems and as an SO library for Linux. The following operating systems are supported:

- Windows 2000, Windows XP, Windows Vista
- Linux/Unix with kernel 2.6, (i386 systems)

In addition to the libraries, a header file that enables the functions to be integrated into C/C++ programs is also supplied. To create a program, the library must be incorporated into the project.

2 Installation instructions

The stated directories and files refer to the driver software for the ND 287 Ethernet module.

2.1 Windows

For an application to load the DLL, file "ND2.dll" must be copied from the "ND_287\windows\bin" directory to the Windows system directory (e.g. "C:\Windows\system32"). Alternatively, the path for the DLL can be defined in the system. The DLL interface is defined via the two files "ND2.lib" in "ND_287\windows\lib" and "ND2.h“ in "ND_287\windows\include". These must be incorporated into the software project in the development environment (for C/C++ environments). File "ND2.lib" must be copied into the library directory of the development environment or its path entered.

2.2 Linux

For an application to load the SO library, the file "libND2.so" must be copied from the "ND_287/linux/lib" directory to the "usr/local/lib" directory. The library interface is defined via file "ND2.h" in "ND_287/linux/include". This must be copied to "usr/local/include" and must be incorporated into the software project in the development environment. The stated directories are based on the "File System Hierarchy Standard" for Linux operating systems. The "libND2.so" library was compiled for i386 system under kernel 2.6.

3 Overview

3.1 Ethernet interface

The Ethernet (LAN) interface is used for configuring the ND 287 and for transferring the position data packets. TCP communication is used for the configuration whilst UDP communication is used for transferring the position data packets (Soft Realtime). The network setting of the ND 287 can be changed by software commands or directly on the keyboard of the ND 287. Constant values or DHCP can be used to set-up the IP address.

3.2 Operating modes

The ND 287 supports the following operating modes:

- Soft Real Time
- Polling

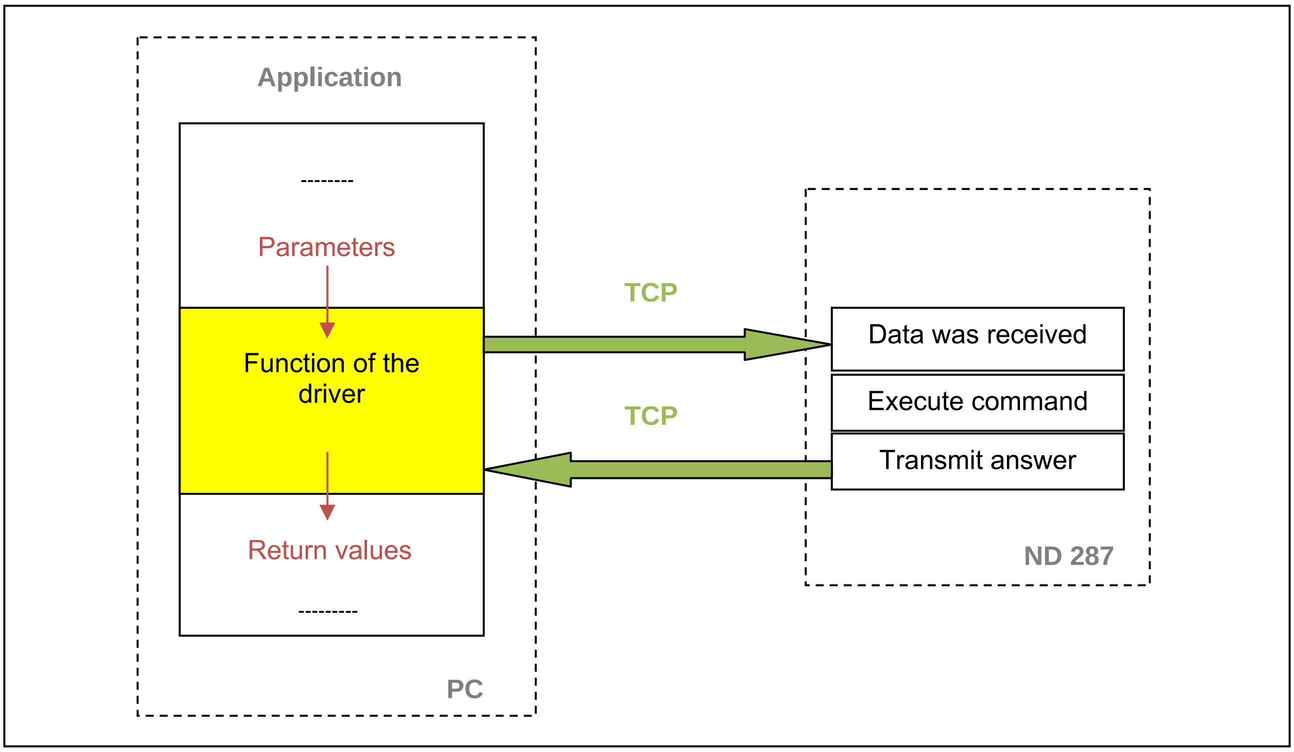
3.2.1 "Polling" operating mode

This operating mode is activated by default after the ND 287 is initialized. The position data are ascertained in the ND 287. As soon as a corresponding command arrives, the ND 287 transmits the data inside the response packet to the customer application.

The diagram below illustrates the sequence of a position poll. A command is sent to the ND 287 from a customer software application on the PC. The ND 287 generates the position data and returns it in a TCP packet. The data is transmitted to the application.

Processing trigger events:

- The time of position value generation is influenced by the software and cannot therefore be determined exactly.
- External trigger inputs are not supported
- Internal periodic trigger sources are not supported



Data Packets in "Polling" Mode:
Depends on the selected function.

The axis functions can be used to access the axes of the ND 287.
The position values can be generated or error messages acknowledged.

It is not necessary to select a trigger source. Triggering occurs implicitly, for example, when the ND2GetPosition() function is called.

3.2.2 "Soft Real-Time" operating mode

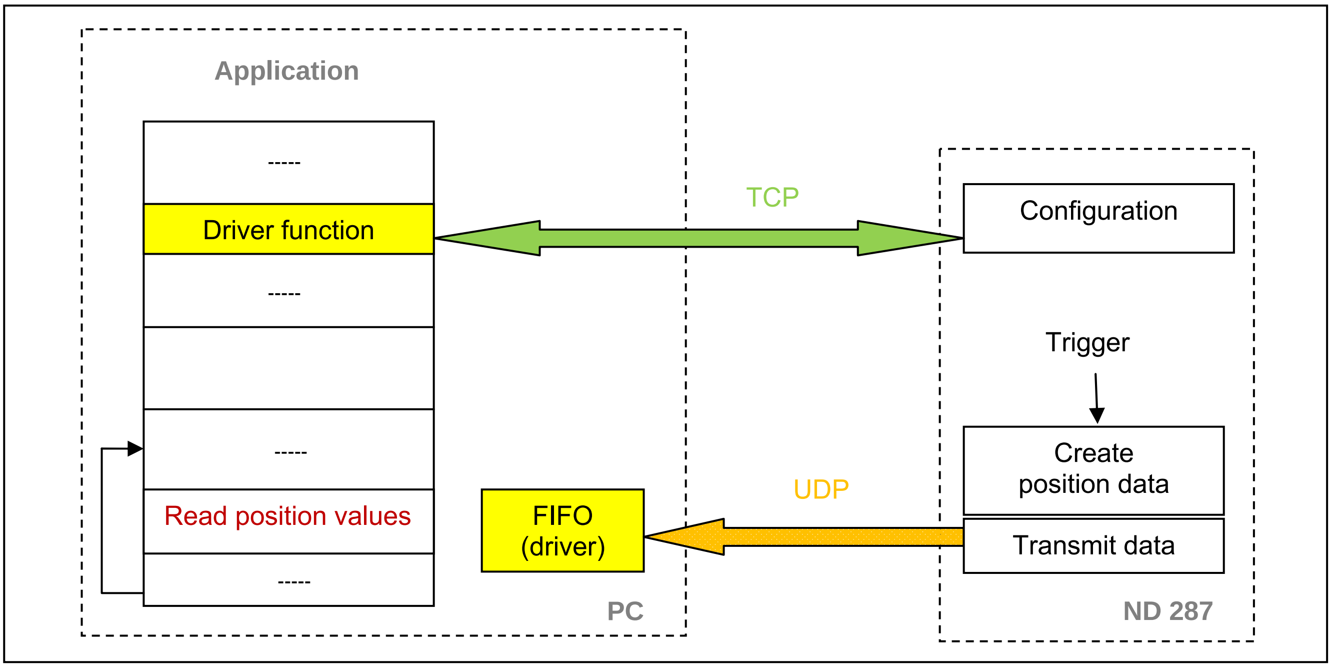
The position data is transported with UDP packets from the ND 287 to the PC. This occurs parallel to the TCP communication via the standard Ethernet interface. The position data is generated when the ND 287 receives a trigger signal. With each trigger event, a data packet is sent to the PC automatically. Here, the packets can be read from a FIFO data structure.

For operation in soft real-time mode, the ND 287 must be configured following the steps listed below.

- Initialization of the ND 287
- Initialization of the axes
- Selection of the operating mode (soft real time)
- Activation of the trigger source

The diagram below illustrates the communication process schematically. The customer software application has to configure the ND 287. The data is then transmitted to the FIFO data structure independently. From here, the application can generate the data within a program loop.

Parallel to the position poll, the status of the ND 287 can be called or error messages cleared.



When the application is closed, the above mentioned steps must be taken in the reverse order. The trigger source must be deactivated first. The operating mode can then be changed or the connection to the ND 287 closed.

Processing trigger events:

- External trigger inputs are supported
- Internal periodic trigger sources are supported
- Software triggers are supported

Position data packets in "Soft Realtime" mode:
Position data packet:

Contents	Encoder Channel no.	Data width (bytes)
Trigger counter	–	2
Status word	1	2
Position value		6
Reserved		4
Reserved		4
Status word	2	2
Position value		6
Reserved		4
Reserved		4
Status word	3	2
Position value		6
Reserved		4
Reserved		4

This table shows only the data contents of the UDP packet. All values displayed above are sent in a UDP packet and copied to the FIFO of the driver. (50 bytes)

The soft real-time mode is activated with the ND2SelectMode() function.
In Soft Real-Time mode, only the error messages from the status word for the position values are reset.

Once Soft Real-Time mode is activated, a trigger source can be selected. The customer software application on the host must continuously read the position data from the receive buffer to prevent an overflow. This can be done using the ND2ReadFIFOData() function.
This function reads one or several entries from the FIFO. Each entry contains data for all axes of the ND 287. The size of an entry can be determined in advance using function ND2SizeOfFIFOEntry(). The individual components of a FIFO entry can be accessed using the functions ND2GetDataFieldPtr().
It is also possible, using the callback mechanism, to register a function that is selected as soon as new data is available in the FIFO.

3.3 Establishing communication

To communicate with the ND 287, a connection must first be established using the ND2Open() function. It is sometimes necessary to determine the IP address beforehand using ND2GetHostIP(). The ND 287 can then be configured using the device functions.
Access to the axes requires handles, which are generated by the ND2GetAxis() function.

At the end of communication, the connection must be closed using the ND2Close() function.

4 Data Types

4.1 Simple data types

ND2_HANDLE	Handle for an ND 287
ND2_AXIS	Handle for an axis of the ND 287 (max. 3 axis handles)
ND2_ERR	Error message
ENCODER_POSITION	Position value (64-bit integer)
TEMPERATURE	Temperature of the temperature sensors (64-bit integer)

4.2 Information for TCP connection

struct `ND2_CONN_INFO`

Component	Description
id	Identification number for the connection
local_ip	Local IP address for this connection
local_port	Local port number for this connection
remote_ip	IP address of the ND 287 for this connection
remote_port	Port number of the ND 287 for this connection

5 Parameters and Return Values

All functions supply a return value of the type ND2_ERR. This labels a function call as successful or reports an error that occurred during execution.

Input values for the functions are transmitted as variables (transfer by value). For return values, a pointer is transferred to a variable that contains the result after the function has been successfully executed (transfer by reference).

6 Status Word

The status word must be interpreted, irrespective of the poll type:

- Incremental position data
- EnDat position data
- Analog files

The status word is transmitted separately for each encoder channel and does not depend on which operating mode is set.
The status word is transmitted by the following commands:

```
ND2_ERR ND2GetPosition()
ND2_ERR ND2GetPositionInfo()
ND2_ERR ND2GetTemperature()
```

It is also transmitted in the Real-Time operating mode.

Bit no.	Incremental encoder	EnDat encoder	Analog sensor
0	1 = Valid position	1 = Valid position	1 = Valid position
1	1 = Signal amplitude error	1 = Signal amplitude error	Reserved
2	Reserved	Reserved	Reserved
3	1 = Frequency exceeded	Reserved	Reserved
4	Reserved	Reserved	Reserved
5	Reserved	Reserved	Reserved
6	Reserved	Reserved	Reserved
7	1 = Lost Trigger	1 = Lost Trigger	1 = Lost Trigger
8	Reserved	1 = Communication error CRC error	Reserved
9	Reserved	1 = Incorrect position value	Reserved
10	Reserved	Reserved	Reserved
11	1 = Error when calculating the coded reference value for distance-coded reference marks	Reserved	Reserved
12	Reserved	Reserved	Reserved
13	Reserved	Reserved	Reserved
14	Reserved	Reserved	Reserved
15	Reserved	Reserved	Reserved

Notes on the error bits

Name	Meaning
Valid position	1 = No error occurred This bit indicates whether the transmitted position is valid or not
Signal amplitude error	1 = Signal amplitude is or was too low (once or more since the last deletion of the error message)
Frequency exceeded	1 = Excessive input signal frequency was detected (once or more since the last deletion of the error message)
CRC error Communication error	1 = CRC error during EnDat data transmission
Faulty position value	1 = The position value ascertained by the EnDat encoder is erroneous
Lost trigger	See "Maximum trigger rate" section
Error when calculating the coded reference value for distance-coded reference marks	1 = Error in calculation of the coded reference value; must be explicitly reset

The error bits are not reset automatically. They have to be reset by a software command `ND2ClearEncoderError()` from a customer software application.
If an error is not reset, it will be re-transmitted with every subsequent position data packet.
With incremental encoders, an error in the position data packet indicates that the position is no longer valid and has lost any reference to reference marks or other measuring channels.
The occurrence of one error can initiate others.

Lost trigger
The "lost trigger" bit indicates that at least one trigger event was processed incorrectly because the period between two trigger events was too short. The "lost trigger" bit can also occur if malfunctions superimpose the trigger line or EMC influences have a negative effect on the transmission. A "lost trigger" does not mean that the position values are false. It merely indicates that trigger events were unable to be processed correctly. The reset must also be done actively by the software command `ND2ClearLostTriggerErrors()`.

Error in reference position for distance-coded reference marks
This bit is set if an error has occurred while the coded reference value for distance-coded reference marks is being calculated. One reason for this could be that the parameter for the distance-coded reference marks is set incorrectly.

7 Auxiliary Functions

7.1 Determining the IP address

The host name of the ND 287 or the IP address (as C-string) is converted to an IP address in "Host Byte Order." The name must be transmitted as a C-string. Examples are "192.168.1.2" or "ND287-1".

Function

```
ND2_ERR ND2GetHostIP    (     const char*     hostname,
                              unsigned long*  ip
                        )
```

Parameters

hostname	Pointer to a C-string containing the IP address or the host name of the ND 287
ip	*[return value]* Pointer to a variable to which the IP address of the ND 287 is saved

Return value
The return value delivers a status for the function call. Possible values are listed below.

ND2_NoError	Function call successful
ND2_HostNotFound	IP address was unable to be determined

8 Device Functions

The device functions always refer to the entire ND 287. No distinction between the axes is possible. With some functions, parameters of all axes are influenced.

All device functions are able to deliver the following error messages as a return value. They can also return further values individually. These are listed separately for each function.

Standard return values

ND2_NoError	Function call successful
ND2_InvalidHandle	The handle on the ND 287 is invalid.
ND2_FuncNotSupp	This function is not supported by ND 287.
ND2_InvalidResponse	Error in data transfer
ND2_AccNotAllowed	Function cannot be performed, as the ND 287 does not permit access.
ND2_ConnReset	Connection was terminated by the ND 287.
ND2_ConnTimeout	Timeout in the data transmission to the ND 287
ND2_ReceiveError	Error while receiving the data
ND2_SendError	Error while sending the data
ND2_OutOfMemory	The system is unable to allocate sufficient memory.

8.1 Opening a connection to the ND 287

A TCP connection is established with the ND 287. In doing so, no settings are changed in the ND 287. If the connection cannot be established, then an error message is returned. The driver must be compatible with the ND 287 firmware to function correctly. This is verified once the connection is established. If necessary, the ND 287 firmware version can be read using this function. To do this, the "ident" parameter must be used to transmit the address of a memory area to which the version number is written as a C-string.

Function

```
ND2_ERR ND2Open    (     unsigned long      ip,
                         ND2_HANDLE*        ND2,
                         long               timeout,
                         char*              ident,
                         unsigned long      len
                   )
```

Parameters

ip	IP address in "Host Byte Order"
ND2	*[return value]* Handle for the ND 287 if the function was closed successfully
timeout	Timeout for following commands in milliseconds
ident	*[return value]* Pointer to the target memory in which the firmware version of the ND 287 is saved as a C string. This memory must be at least 9 bytes. If this parameter is a NULL pointer, the firmware version of the ND 287 is not read.
len	Size of the target memory in bytes (0, where ident = ZERO)

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantInitWinSock	Socket layer of the operating system cannot be initialized (applies only for Windows).
ND2_CantOpenSocket	System resources for connection not available
ND2_OutOfMemory	Insufficient memory available
ND2_IFVersionInv	ND 287 firmware is incompatible with the driver.
ND2_CantConnect	Connection cannot be established (ND 287 may have been switched off or is unavailable)

8.2 Closing a connection to the ND 287

The connection to the ND 287 is closed. The ND2 handle must not be used again. Likewise, all handles on the axes generated from this ND2 handle are invalid. If a special operating mode of the ND 287 has been activated via this handle, polling mode will be activated when the connection is closed. All other settings in the ND 287 are retained.

Function

```
ND2_ERR ND2Close     (     ND2_HANDLE   ND2
                     )
```

Parameters

ND2	ND2 handle

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.3 Polling connection status

The status of the connection to the ND 287 is polled. Whether a connection has already been closed or a communication has error occurred can then be determined. This function does not send data to the ND 287. The status refers to the previous commands.

Function

```
ND2_ERR ND2GetConnectionStatus    (     ND2_HANDLE        ND2,
                                        ND2_CONN_STATUS*  status
                                  )
```

Parameters

ND2	ND2 handle
status	*[return value]* Pointer to the target variable for the status

Component	Description
ND2_CS_Connected	Connection to the ND 287 established
ND2_CS_Closed	No connection to the ND 287
ND2_CS_Timeout	Time exceeded during data transmission
ND2_CS_ConnectionReset	Connection was terminated by the ND 287.
ND2_CS_TransmissionError	Transmission error occurred

Return value
The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.4 Setting up timeout

The timeout for the TCP connection to the ND 287 is reset. This value applies for all subsequent function calls.
The timeout must be at least 100 ms. Lower values are automatically increased to 100.

Function
```
ND2_ERR ND2SetTimeout      (     ND2_HANDLE      ND2,
                                 long  timeout
                           )
```

Parameters

ND2	ND2 handle
timeout	Timeout in milliseconds (>= 100)

Return value
The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_IllegalParameter	The timeout cannot be set

8.5 Requesting handle for axis

The handles for access to the axes of the ND 287 are generated. They are saved in an array, the size of which must also be transferred as a parameter. The number of valid handles is delivered as a return value. For each axis of the ND 287, the function delivers one handle, the maximum being the number accommodated in the array ("size" parameter).
The handles are stored in the array in ascending order, starting with axis 1.
No more than 3 handles are permissible for axes for the ND 287.

The ND 287 can be provided with up to 2 encoder modules, handle for axis 1 and handle for axis 2.
The position values of the axes can be shown coupled (X1+X2; X1-X2; X1(fctX2).

Axis 3 is a virtual axis and is always shown with respect to the position displayed on the LCD of the ND 287.

Function
```
ND2_ERR ND2GetAxis      (     ND2_HANDLE         ND2,
                              ND2_AXIS*          set,
                              unsigned long      size,
                              unsigned long*     len
                        )
```

Parameters

ND2	ND2 handle
set	*[return value]* Pointer to the first element of the handle array
size	Maximum number of entries in the array
len	*[return value]* Number of valid entries in the array

Return value
The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.6 Selecting the operating mode

The operating mode for the ND 287 can be set. Both "polling" and "soft real-time" modes are supported.

Function

```
ND2_ERR ND2SelectMode      (     ND2_HANDLE              ND2,
                                 ND2_OPERATING_MODE      mode
                           )
```

Parameters

ND2	ND2 handle
mode	Operating mode

Mode	Operating mode
ND2_OM_Polling	Polling mode
ND2_OM_SoftRealtime	Soft Real-Time Mode

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantOpenSocket	Internal error (socket error)
ND2_CantStartThread	Internal error (thread error)
ND2_SoftRTEn	Soft real-time mode is already active
ND2_PortNoInv	Internal error (UDP port error)
ND2_DestIPUnreach	Internal error (IP address error)

8.7 Saving network parameters

The parameters for the Ethernet interface of the ND 287 can be set. This means the ND 287 can be adapted to the network. The settings do not take effect until after the next boot process. If the DHCP client is active, the ND 287 tries to obtain an IP address from the DHCP server. If the server fails to respond within the set timeout, the configured IP address will be used.

The network parameters can also be entered through the ND 287 keypad.

Function

```
ND2_ERR ND2SetNetwork      (     ND2_HANDLE         ND2,
                                 unsigned long      ip,
                                 unsigned long      netmask,
                                 unsigned long      gateway,
                                 ND2_MODE           dhcp,
                                 unsigned long      timeout
                           )
```

Parameters

ND2	ND2 handle
ip	IP address of the ND 287 in "Host Byte Order"
netmask	Network mask for the network in "Host Byte Order"
gateway	IP address of the standard gateway in "Host Byte Order"
dhcp	Flag for the DHCP client in the ND 287

dhcp	Description
ND2_MD_Disable	Deactivate DHCP client
ND2_MD_Enable	Activate DHCP client

timeout	Timeout for the DHCP client in seconds

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantSaveCustNW	Network settings cannot be stored
ND2_CantSaveDHCP	DHCP Timeout cannot be saved
ND2_DHCPTimeoutInv	DHCP timeout invalid
ND2_ParamInvalid	Parameters are not a valid network configuration

8.8 Reading network parameters

The parameters for the Ethernet interface can be read. The settings that are output are always the user-defined settings.

Function

```
ND2_ERR ND2GetNetwork     (     ND2_HANDLE          ND2,
                                unsigned long*      ip,
                                unsigned long*      netmask,
                                unsigned long*      gateway,
                                ND2_MODE*           dhcp
                          )
```

Parameters

ND2	ND2 handle	
ip	*[return value]*	Pointer to the variable for the IP address in "Host Byte Order"
netmask	*[return value]*	Pointer to the variable for the network mask in "Host Byte Order"
gateway	*[return value]*	Pointer to the variable for the IP address of the standard gateway in "Host Byte Order"
dhcp	*[return value]*	Pointer to the variable for the flag for the DHCP client

dhcp	Description
ND2_MD_Disable	DHCP client inactive
ND2_MD_Enable	DHCP client active

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_NoCustNetwork	No customized settings present

8.9 Saving the host name

The host name of the ND 287 is saved. The name must be transmitted as a C-string, which can be a maximum of 9 characters long including the zero byte. If it is any longer, the rest will be cut off. If a string with a length of zero or a ZERO pointer is transferred, the ND 287 sets the host name to the standard value on delivery.

Function

```
ND2_ERR ND2SetHostname    (     ND2_HANDLE        ND2,
                                const char*       hostname
                          )
```

Parameters

ND2	ND2 handle
hostname	Pointer to the new host name

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_HostnameInvalid	Hostname is invalid

8.10 Reading the host name

The host name of the ND 287 is read and saved in the target memory as a C-string. The string is up to 9 characters long (incl. the null byte). If the target memory is not big enough to take the entire string, only the first part is copied.

Function

```
ND2_ERR ND2GetHostname    (    ND2_HANDLE        ND2,
                               char* hostname,
                               unsigned long len
                          )
```

Parameters

ND2	ND2 handle
hostname	*[return code]* Pointer to the target memory for the host name
len	Size of the target memory in bytes

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantRdHostname	Host name cannot be read

8.11 Reading the device ID

The device ID of the ND 287 is generated as a C-string. The string is written to the target memory. If the target string does not provide sufficient space for the number, an error will be generated. The number can be up to 16 characters long (incl. zero byte).

Function

```
ND2_ERR ND2GetIdentNumber    (    ND2_HANDLE        ND2,
                                  char*             ident,
                                  unsigned long     len
                             )
```

Parameters

ND2	ND2 handle
ident	*[return value]* Pointer to the target memory for the device ID number
len	Size of the target memory in bytes

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantRdIdent	Device number cannot be read
ND2_BufferTooSmall	Target memory is too small

8.12 Reading MAC address

The MAC address of the ND 287 is output. The address is output in binary format. The target memory must be at least 6 bytes. The first six bytes are always used. The lowest-value byte of the MAC address is copied to the first byte of the target memory. For example for "00:A0:CD:85:00:01".

Offset	Memory Contents
0	0x01
1	0x00
2	0x85
3	0xCD
4	0xA0
5	0x00

Function

```
ND2_ERR ND2GetMAC    (     ND2_HANDLE         ND2,
                           unsigned char*     mac
                     )
```

Parameters

ND2 ND2 handle
mac *[return value]* Pointer to the target memory for the MAC address

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.13 Reading out the firmware version number

The version number of the ND 287 firmware is output as a C-string. For the string, including the zero byte, the target memory must be at least 9 bytes. If the target memory is too small to take the entire string, only the first part is copied.
Example: "64611804", the firmware ID number is 646118. The firmware version is 04.

Function

```
ND2_ERR ND2GetVersion     (     ND2_HANDLE         ND2,
                                char*              ident,
                                unsigned long      len,
                          )
```

Parameters

ND2 ND2 handle
ident *[return value]* Pointer to the target memory for the firmware version number
len Size of the target memory in bytes

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.14 Reading the number of open connections

The number of currently open connections to the ND 287 is generated. This also includes semi-open functions that the remote station has already closed but are still open on the ND 287.

Function

```
ND2_ERR ND2GetNumberOfOpenConnections     (     ND2_HANDLE         ND2,
                                                unsigned long*     cnt
                                          )
```

Parameters

ND2 ND2 handle
cnt *[return value]* Pointer to the variable for the number of open connections

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.15 Reading the connection data

The connection data of all currently open connections to the ND 287 can be read. For each connection, an entry is assigned to the array. However, the maximum is the number specified by the "size" parameter. The number of valid elements in the array is returned by the "cnt" parameter. The contents of the connection data is listed in the "Data types" section.

Function

```
ND2_ERR ND2ConnectionInfo    (     ND2_HANDLE         ND2,
                                   ND2_CONN_INFO*     info,
                                   unsigned long      size,
                                   unsigned long*     cnt
                             )
```

Parameters

ND2	ND2 handle
info	*[return value]* Pointer to the first element in the array for the connection data
size	Size of the "info" array
cnt	*[return value]* Pointer to the variable for the number of valid elements in the array

Return value
The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.16 Terminating the connection

An open connection to the ND 287 can be terminated. It is not possible to close the connection used to call the function. The primary use of this function is to close semi-open connections that have not been terminated properly due, for example, to an error on the host. The ID can be taken from connection data ND2_CONN_INFO (see "Read connection data").

Function

```
ND2_ERR ND2TerminateConnection    (     ND2_HANDLE         ND2,
                                        unsigned long      id
                                  )
```

Parameters

ND2	ND2 handle
id	ID of the terminated connection

Return value
The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantTermConn	The connection cannot be terminated.
ND2_CantTermSelf	The connection cannot terminate itself.
ND2_ParamInvalid	The parameter is not a valid index for a connection

8.17 Setting the timer trigger period duration

The period duration of the timer trigger can be set. To do this, the duration of the period must be stated. The period duration equals the entered value multiplied by 20 ms. The value must be a natural number greater than zero. If the timer trigger is activated, it will initiate a trigger event after each period (position values of both axes are triggered).

Function

```
ND2_ERR ND2SetTimerTriggerPeriod    (     ND2_HANDLE         ND2,
                                          unsigned long      period
                                    )
```

Parameters

ND2	ND2 handle
period	Timer trigger period (>0)

Return value
The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_ParamInvalid	Trigger period invalid

8.18 Reading the timer trigger period duration

The period duration of the timer trigger can be read out. The value issued is a multiple of 20 ms.
Period duration [ms] = period x 20

Function

```
ND2_ERR ND2GetTimerTriggerTicks      (     ND2_HANDLE        ND2,
                                           unsigned long*    period
                                     )
```

Parameters

ND2	ND2 handle
Period	*[return value]* Pointer to the variable for the period duration

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.19 Activating the timer trigger

The timer trigger can be activated or deactivated. The external trigger and the timer trigger cannot be operated at the same time. If the external trigger signal has already been enabled, the ND 287 delivers an error message as soon as an attempt is made to activate the timer trigger.

Function

```
ND2_ERR ND2EnableTimerTrigger     (     ND2_HANDLE      ND2,
                                        ND2_MODE        mode
                                  )
```

Parameters

ND2	ND2 handle
mode	Activate or deactivate the timer trigger

Mode	Description
ND2_MD_Disable	Deactivate the timer trigger
ND2_MD_Enable	Activating the Timer Trigger

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_ExtTrgEn	External trigger is already active

8.20 Clearing the trigger counter

The trigger counter of the ND 287 is set to zero.

Function

```
ND2_ERR ND2ResetTriggerCounter    (     ND2_HANDLE      ND2
                                  )
```

Parameters

ND2	ND2 handle

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.21 Acknowledging the trigger error

The error message for the trigger interface is acknowledged. The trigger error is deleted for all axes of an ND 287.

Function

```
ND2_ERR ND2ClearLostTriggerErrors    (     ND2_HANDLE      ND2
                                     )
```

Parameters

ND2	ND2 handle

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.22 Enabling the external trigger signal

The external trigger signal can be enabled or disabled. The external trigger and the timer trigger cannot be operated at the same time. If the timer trigger is already active, the ND 287 delivers an error message as soon as an attempt is made to activate the external trigger.

Function

```
ND2_ERR ND2EnableExternalTrigger     (     ND2_HANDLE      ND2,
                                           ND2_MODE        mode
                                     )
```

Parameters

ND2	ND2 handle
mode	Activate or deactivate the external trigger

Mode	Description
ND2_MD_Disable	Deactivate external trigger
ND2_MD_Enable	Activate external trigger

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_TimerTrgEn	Timer trigger is already active

8.23 Software trigger

The software trigger generates a trigger event and induces the ND 287 to send data to the remote station. The "source" parameter is reserved for further options and must be set to zero. This function can only be performed in Soft Real-Time mode. Here, both the timer trigger and the external trigger must be inactive.

Function

```
ND2_ERR ND2SoftwareTrigger   (     ND2_HANDLE      ND2,
                                   unsigned long   source
                             )
```

Parameters

ND2	ND2 handle
source	Trigger source (must be 0)

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_TimerTrgEn	Timer trigger is already active
ND2_ExtTrgEn	External trigger is already active

8.24 Reset

The ND 287 performs a reset and reboots. This function has the same effect as pressing the power switch. The connection to the ND 287 is automatically closed (as with ND2Close).

Function

```
ND2_ERR ND2Reset        (     ND2_HANDLE   ND2
                        )
```

Parameters

ND2 ND2 handle

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.25 Identifying ND 287

This function makes it possible to change the color of the LED on the front panel of the ND 287. In normal operation, the LED glows in green. When this function is activated it glows red.
If several devices are arranged side by side, an ND 287 with a certain IP address is easily located.

Function

```
ND2_ERR ND2Identify     (     ND2_HANDLE      ND2,
                              ND2_MODE        mode
                        )
```

Parameters

ND2 ND2 handle
mode Change the color of the LED

Mode	Description
ND2_MD_Enable	LED glows in red.
ND2_MD_Disable	LED glows in green

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_IllegalParameter LED status cannot be changed (parameter is invalid).

8.26 Datum setting

Sets the axis display to the preset value for the datum.
If a coupled position X1:X2 is active, the ND 287 sets X1 to the preset value for the datum and resets X2 to zero.

Function

```
ND2_ERR ND2PresetCounter     (     ND2_HANDLE      ND2
                             )
```

Parameters

ND2 ND2 handle

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantPreset Datum setting is not possible.

8.27 Output an error message

Error message on the ND 287 are displayed in plain language in the message line of the LCD display.
With this function, the error message is output and saved in the target memory as a C-string. The string is up to 40 characters long (incl. the null byte). The output works only if the ND 287 displays an error message.

Function

```
ND2_ERR ND2GetHostname    (     ND2_HANDLE         ND2,
                                char*              hostname,
                                unsigned long      len
                          )
```

Parameters

ND2	ND2 handle
hostname	*[return code]* Pointer to the target memory for the error message
len	Size of the target memory in bytes

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_NoErrorMessage	No error message available

8.28 Deleting an error message

If an error occurs while you are working with ND 287, the message will appear on the LCD display and provide an explanation of what caused the error. The function acknowledges and clears the error message.

Function

```
ND2_ERR ND2ClearError     (     ND2_HANDLE   ND2
                          )
```

Parameters

ND2	ND2 handle

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.29 Deleting an axis error message

Error occurring in the connected encoders are acknowledged and deleted.
The error messages are deleted for all axes of an ND 287 at the same time.

Function

```
ND2_ERR ND2ClearEncoderError    (     ND2_HANDLE   ND2
                                )
```

Parameters

ND2	ND2 handle

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.30 Switching on the keypad lock

This function locks the keypad of the ND 287.
The ND 287 can be operated only by external key commands. The keypad can be unlocked either by transmitting the function **ND2ReleaseKeyPad** or by switching the position display unit off and then back on.

Function

```
ND2_ERR ND2LockKeypad      (       ND2_HANDLE   ND2
                           )
```

Parameters

ND2 ND2 handle

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantLockKeypad Keypad cannot be locked.

8.31 Switching of the keypad lock

This function makes it possible to unlock a locked keypad.

Function

```
ND2_ERR ND2ReleaseKeypad     (     ND2_HANDLE   ND2
                             )
```

Parameters

ND2 ND2 handle

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantReleaseKeypad Keypad cannot be unlocked.

8.32 Switching on REF mode

This function makes it possible to switch off the REF mode of both axes.

Function

```
ND2_ERR ND2Clear_RefStatus   (     ND2_HANDLE   ND2
                             )
```

Parameters

ND2 ND2 handle

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantChangeREFstatus REF mode cannot be switched off.

8.33 Activating the REF mode

This function makes it possible to switch on the REF mode of both axes.
The reference marks now have to be scanned.

Function

```
ND2_ERR ND2Start_Ref    (     ND2_HANDLE   ND2
                        )
```

Parameters

ND2 ND2 handle

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantChangeREFstatus REF mode cannot be activated.

8.34 Canceling REF mode

REF mode for scanning the reference marks can be canceled.
The reference run is then terminated.

Function

```
ND2_ERR ND2Stop_Ref     (     ND2_HANDLE   ND2
                        )
```

Parameters

ND2	ND2 handle

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantChangeREFstatus	REF mode cannot be canceled.

8.35 Starting a measurement series/SPC

depending on the operating modes that are set on the ND 287, either a measurement series or an SPC operation is started.

Function

```
ND2_ERR ND2StartMeasurement     (     ND2_HANDLE   ND2
                                )
```

Parameters

ND2	ND2 handle

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantStartMeasurement	Measurement series or SPC cannot be started.

8.36 Stopping a measurement series/SPC

depending on the operating modes that are set on the ND 287, either a measurement series or an SPC operation is stopped.

Function

```
ND2_ERR ND2StoptMeasurement     (     ND2_HANDLE   ND2
                                )
```

Parameters

ND2	ND2 handle

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantStartMeasurement	Measurement series or SPC cannot be started.

8.37 Switching between the measurement series or SPC modes of operation

With the ND 287 you can record and evaluate measured values as a **series of measurements** or you can conduct **Statistical Process Control (SPC)** operations. With this function you can switch between the two operating modes.

Function

```
ND2_ERR ND2Set_MeasMode    (     ND2_HANDLE   ND2,
                                 ND2_MEAS_MODEmeasmode
                           )
```

Parameters

ND2	ND2 handle
measmode	Operating mode

measmode	Description
ND2_MEAS_SERIE	Operating mode for measurement series
ND2_SPC	Operating mode for SPC

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantchangeMode	Operating mode cannot be set

8.38 Deleting SPC statistics

In the SPC operating mode on the ND 287 you can delete all previously saved measurement data.

Function

```
ND2_ERR ND2Delete_SPC_Values    (     ND2_HANDLE   ND2
                                )
```

Parameters

ND2	ND2 handle

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.39 External operation / keystroke commands

This function makes it possible to operate the ND 287 remotely through your PC.
All keys of the ND 287 can be simulated by transmitting the corresponding keystroke command.

Function

```
ND2_ERR ND2SetKey    (     ND2_HANDLE   ND2,
                           ND2_KEY      key
                     )
```

Parameters

ND2 ND2 handle
ND2_KEY Select a key

key	Description
ND2_KEY_0	Numerical key for 0
ND2_KEY_1	Numerical key for 1
ND2_KEY_2	Numerical key for 2
ND2_KEY_3	Numerical key for 3
ND2_KEY_4	Numerical key for 4
ND2_KEY_5	Numerical key for 5
ND2_KEY_6	Numerical key for 6
ND2_KEY_7	Numerical key for 7
ND2_KEY_8	Numerical key for 8
ND2_KEY_9	Numerical key for 9
ND2_KEY_C	C (Clear) key
ND2_KEY_MINUS	Minus key
ND2_KEY_DP	Decimal point key
ND2_KEY_NAV	Navigation key
ND2_KEY_ENT	Enter key
ND2_KEY_UP	"Cursor Up" key
ND2_KEY_DOWN	"Cursor Down" key
ND2_KEY_SOFT1	"Soft key 1" (left)
ND2_KEY_SOFT2	"Soft key 2"
ND2_KEY_SOFT3	"Soft key 3"
ND2_KEY_SOFT4	"Soft key 4"

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_KeyError Invalid key command or insufficient duration between two keystrokes

8.40 Reading out external inputs

The logical status of the external inputs is read out and saved in the target memory as a C-string. The string is 16 characters long (incl. the null byte).

"0" = LOW status
"1" = HIGH status

Function

```
ND2_ERR ND2ReadInputs     (     ND2_HANDLE        ND2,
                                unsigned char*    level
                          )
```

Parameters

ND2 ND2 handle
level *[return value]* Pointer to the target memory for the inputs

level [bytes]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
External input [pin]		nc	nc	nc	nc	nc	nc	25	22/23	9	8	7	6	5	3	2

For connector assignment see ND 287 operating instructions.

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.41 Reading out external outputs

The logical status of the external outputs is read out and saved in the target memory as a C-string. The string is 16 characters long (incl. the null byte).

"0" = LOW status
"1" = HIGH status

Function

```
ND2_ERR ND2ReadInputs     (    ND2_HANDLE        ND2,
                               unsigned char*    level
                          )
```

Parameters

ND2 ND2 handle
level *[return value]* Pointer to the target memory for the outputs

level [bytes]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
External output [pin]		nc	nc	nc	nc	nc	nc	25	nc	nc	19	18	17	16	15	14

For connector assignment see ND 287 operating instructions.

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.42 Outputting the number of connected encoders

The number of connected encoders on the ND 287 is determined and output.

Possible return values:	0	No encoder connected
	1	One encoder connected
	2	Two encoders connected

Function

```
ND2_ERR ND2GetNumberOfConnectedAxis    (    ND2_HANDLE        ND2,
                                            unsigned long*    axis
                                       )
```

Parameters

ND2 ND2 handle
axis *[return value]* Pointer to the variable for the number of encoders

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.43 Output of status bar

The status of the status bar is read and saved in the target memory as a C-string. The string is 16 characters long (incl. zero byte).

Function

```
ND2_ERR ND2ReadStatusBar     (     ND2_HANDLE        ND2,
                                   unsigned char*    level
                             )
```

Parameters

ND2 ND2 handle

level *[return value]* Pointer to the target memory for the status bar

level [bytes]	Parameters	Description
0	Mode of operation	0 = Actual value 1 = Distance to go
1	Display mode	0 = X1 1 = X2 2 = X1+X2 3 = X1-X2 4 = f(X1,X2)
2	Scale factor	0 = Not activated 1 = Activated
3	Compensation	0 = No compensation 1 = Axis error compensation
4	Stopwatch	0 = Stopped 1 = Stopwatch is running
5	Unit of Measure	0 = mm 1 = inch 2 = GRD 3 = GMS 4 = rad
6	Datum	1 = Datum 1 2 = Datum 2
7	Soft-key page	1 = Page 1 2 = Page 2 3 = Page 3 4 = Keyboard is locked
8-14	nc	

For description of status bar, see ND 287 operating instructions.

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.44 Outputting the status of the status indicators

The status of the status indicators is read and saved in the target memory as a C-string. The string is 16 characters long (incl. the null byte).

Output of the symbols in the status display: "0" = Symbol not active (gray)
„1“ = Symbol active (red)
„2“ = Symbol flashing

Function

```
ND2_ERR ND2ReadStatusIndicator    (     ND2_HANDLE        ND2,
                                        unsigned char*    level
                                  )
```

Parameters

ND2 ND2 handle

level *[return value]* Pointer to the target memory for the status indicators

level [bytes]	Parameters	Description
0	<	Displayed tolerance checking and sorting mode
1	=	Displayed tolerance checking and sorting mode
2	>	Displayed tolerance checking and sorting mode
3	MIN	Display mode for series of measurements
4	ACTL	Display mode for series of measurements
5	MAX	Display mode for series of measurements
6	DIFF	Display mode for series of measurements
7	SET	Datum setting
8	REF	Reference mark evaluation
9	Meas. series / SPC	0 = No measurement 1 = Meas. series/SPC started
10-14	nc	

For description of status indicators, see ND 287 operating instructions.

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.45 Setting the datum

The ND 287 has two datum levels. With this function you can switch between datum 1 and datum 2.

Function

```
ND2_ERR ND2_SetDatum       (    ND2_HANDLE            ND2,
                                ND2_DATUM_SETTING     datumsetting
                           )
```

Parameters

ND2	ND2 handle
datumsetting	Setting the datum

datumsetting	Description
ND2_DATUM1	Setting datum 1
ND2_DATUM2	Setting datum 2

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantChangeMode	Datum cannot be switched

8.46 Setting the unit of measure for length

With this function you can switch the unit of measure between millimeters and inches.

Function

```
ND2_ERR ND2Set_Linear_Units    (    ND2_HANDLE      ND2,
                                    ND2_LINEAR      linear
                               )
```

Parameters

ND2	ND2 handle
linear	Setting the unit of measure

linear	Description
ND2_MM	Unit of measure is "mm"
ND2_INCH	Unit of measure is "inch"

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantChangeMode	Unit of measure cannot be switched

8.47 Setting the unit of measure for angle

With this function you can switch the unit of angular measure between "grad", "dms" and "rad".

Function

```
ND2_ERR ND2Set_Angular_Units    (     ND2_HANDLE      ND2,
                                      ND2_ANGULAR     angular
                                )
```

Parameters

ND2	ND2 handle
angular	Setting the unit of measure

linear	**Description**
ND2_DEG	Unit of measure is the grad
ND2_DMS	Deg . Min . Sec
ND2_RAD	Unit of measure is the radian

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantChangeMode	Unit of measure cannot be switched

8.48 Defining the display for a series of measurements

With this function you can specify which display you want during a running measurement series.

ACTL DISPLAY	- Current measured value is displayed.
MIN DISPLAY	- Minimum value of a series of measurements is displayed.
MAX DISPLAY	- Maximum value of a series of measurements is displayed.
DIFF DISPLAY	- Difference MAX - MIN (i.e. the range) is displayed.

Function

```
ND2_ERR ND2Set_Meas_Display     (     ND2_HANDLE         ND2,
                                      ND2_MEAS_DISPL     meas_displ
                                )
```

Parameters

ND2	ND2 handle
meas_displ	Specify the display mode

meas_displ	**Description**
ND2_ACTL	Actual measured value
ND2_MIN	Minimum measured value
ND2_MAX	Maximum measured value
ND2_DIFF	Difference between MAX and MIN

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantChangeMode	Display cannot be switched.

8.49 Setting the display mode of the axes (only possible with second encoder module)

With this function you can specify which display mode is shown on the screen of the ND 287.

ANZEIGE X1 ; ANZEIGE X2 ; ANZEIGE X1+X2; ANZEIGE X1-X2; ANZEIGE f(X1,X2)

Function

```
ND2_ERR ND2Set_Display_Mode     (     ND2_HANDLE         ND2,
                                      ND2_DISPL_MODE     displ_mode
                                )
```

Parameters

ND2	ND2 handle
displ_mode	Specify the display mode

displ_mode	Description
ND2_X1	X1
ND2_X2	X2
ND2_X1pX2	X1 + X2
ND2_X1mX2	X1 – X2
ND2_X1fctX2	Formula entry

Return value
The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantChangeMode	Display cannot be switched.

8.50 Read out the axis display mode

With this function you can read out the display mode on the screen of the ND 287.

Function

```
ND2_ERR ND2Get_Display_Mode     (     ND2_HANDLE        ND2,
                                      ND2_DISPL_MODE*   displ_mode
                                )
```

Parameters

ND2	ND2 handle
displ_mode	*[return value]* Pointer to the variable of the display mode

displ_mode	Description
ND2_X1	X1
ND2_X2	X2
ND2_X1pX2	X1 + X2
ND2_X1mX2	X1 – X2
ND2_X1fctX2	Formula entry

Return value
The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.51 Sampling the temperature

If an analog thermistor is connected to input X2 and the COMPENSATION encoder type is selected, the temperature is shown in the note line of the ND 287. This temperature can be read out with the aid of this function.
A status word from which potential errors emerge is also transmitted. The Temperature can only be sampled in polling mode.

Function

```
ND2_ERR ND2GetTemperature    (     ND2_HANDLE        ND2,
                                   unsigned short*   status,
                                   TEMPERATURE*      temp
                             )
```

Parameters

ND2	ND2 handle
status	*[return value]* Pointer to the variable for the status word
temp	*[return value]* Pointer to the variable for the temperature

Return value
The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.52 Setting the temperature compensation

If an analog thermistor is connected to input X2 and the COMPENSATION encoder type is selected, the position value of axis X1 can be compensated.
The temperature compensation can be activated or deactivated with the aid of this function.

Function

```
ND2_ERR ND2Enable_TmpCompensation    (     ND2_HANDLE      ND2,
                                           ND2_MODE        mode
                                     )
```

Parameters

ND2	ND2 handle
mode	Activate or deactivate temperature compensation

Mode	Description
ND2_MD_Disable	Deactivate temperature compensation
ND2_MD_Enable	Activate temperature compensation

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_CantChangeMode	Compensation cannot be activated or deactivated
ND2_TmpCmpEn	No thermistor connected

8.53 Outputting the currently displayed axis value as a C string

The axis position shown on the LCD display of the ND 287 is read with additional information and saved in the target memory as a C-string. The string is up to 24 characters long (incl. the null byte).
The format of the output is identical with that used with the RS-232 and USB interface of the ND 287. The position can only be polled in polling mode.

Function

```
ND2_ERR ND2GetDisplayedPosition    (     ND2_HANDLE      ND2,
                                         char*           displ_pos
                                   )
```

Parameters

ND2	ND2 handle
displ_pos	*[return code]* Pointer to the target memory for the position

Example: Data sequence during measured value output

Measured value X = –5.23 mm
The measured value is within the sorting limits (=) and is the current value (A) of a series of measurements.

Measured value output:

–	5.23			=	A	< CR >	<LF>
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

1 +/– Algebraic sign
2 Numerical value with decimal point: 10 characters total. The ND transmits leading zeros as blank spaces.
3 Blank space
4 Measured unit: **blank space** = mm, “= inch, ? = error
5 Sorting status (**<** / **=** / **>**)
? = lower sorting limit > upper sorting limit

6 When series of measurements has been started:
S = MIN, **A** = ACTL, **G** = MAX, **D** = DIFF
If two axes are coupled (optional) and no series of measurements has been started:
1 = X1, **2** = X2, **A** = X1 + X2, **S** = X1 - X2, **F** = f(X1,X2)
7 Carriage return
8 Blank line (Line Feed)

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.54 Outputting currently displayed axis value with additional information

The axis position shown on the display of the ND 287 is read out with additional information.
A status word from which potential position errors emerge is also transmitted. The position can only be polled in polling mode. The additional information is saved in target memory as a C string. The string is up to 24 characters long (incl. the null byte). The additional information includes the status bar, status display and the decimal places of the position value.

Function

```
ND2_ERR ND2GetPositionInfo   (     ND2_HANDLE         ND2,
                                   unsigned short*    status,
                                   ENCODER_POSITION*  pos,
                                   unsigned char*     add_info
                             )
```

Parameters

ND2	ND2 handle
status	*[return value]* Pointer to the variable for the status word
pos	*[return value]* Pointer to the variable for the position
add_info	*[return code]* Pointer to the target memory for the additional information

Add_info [byte]	Parameters	Description
0	Mode of operation	0 = Actual value 1 = Distance to go
1	Display mode	0 = X1 1 = X2 2 = X1+X2 3 = X1-X2 4 = f(X1,X2)
2	Scale factor	0 = Not activated 1 = Activated
3	Compensation	0 = No compensation 1 = Axis error compensation
4	Stopwatch	0 = Stopped 1 = Stopwatch is running
5	Unit of Measure	0 = mm 1 = inch 2 = GRD 3 = GMS 4 = rad
6	Datum	1 = Datum 1 2 = Datum 2
7	Soft-key page	1 = Page 1 2 = Page 2 3 = Page 3 4 = Keyboard is locked
8	<	0 = Symbol not active (gray) 1= Symbol active (red)
9	=	0 = Symbol not active (gray) 1= Symbol active (red)
10	>	0 = Symbol not active (gray) 1= Symbol active (red)
11	MIN	0 = Symbol not active (gray) 1= Symbol active (red)
12	ACTL	0 = Symbol not active (gray) 1= Symbol active (red)
13	MAX	0 = Symbol not active (gray) 1= Symbol active (red)
14	DIFF	0 = Symbol not active (gray) 1= Symbol active (red)
15	SET	0 = Symbol not active (gray) 1= Symbol active (red) 2 = Symbol flashes
16	REF	0 = Symbol not active (gray) 1= Symbol active (red) 2 = Symbol flashes
17	Meas. series / SPC	0 = No measurement 1 = Meas. series/SPC started
18	nc	
19	nc	
20	Decimal places	0 – 6 = Decimal places of the position value
21	nc	
22	nc	

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.55 Outputting an SPC measured value list

The ND 287 makes it possible to output the saved measurement data and measured values for analysis in Statistical Process Control. The internal memory of the ND 287 can evaluate and save up to 1000 measured values. The measured data are saved in target memory as a C string.
The string can be up to 40 000 characters long (incl. the null byte). In addition, the length of the measured data is output in bytes.

Function

```
ND2_ERR ND2Get_SPC_Value     (     ND2_HANDLE         ND2,
                                   unsigned char*     SPC_Values,
                                   unsigned long      size,
                                   unsigned long*     rlen
                             )
```

Parameters

ND2	ND2 handle
SPC_Values	*[return value]* Pointer to the target memory for the SPC data
size	Size of the target memory in bytes
rlen	*[return value]* Pointer to the variable for the number of received bytes

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_NoData	No SPC data available (memory empty)

Example of an SPC measured value list:

```
#
SPC-DATA: X1

SAMPLES        =         100
VAL./SAMPLE    =          10
MAX            =  +  10.0645
MIN            =  +   9.9960
DIFF           =  +   0.0685
AVERAGE        =  +  10.0108
STD. DEV.      =      0.0198

   1 =    +   10.0145
   2 =    +   10.0120
   3 =    +   10.0190
   4 =    +   10.0255
   5 =    +   10.0245
   6 =    +   10.0250
   7 =    +   10.0295
   8 =    +   10.0600
   9 =    +   10.0640
  10 =    +   10.0645

  11 =    +   10.0620
  12 =    +   10.0605
      .
      .
      .
      .
  99 =    +    9.9980
 100 =    +    9.9975
#
```

8.56 Outputting a parameter list

The ND 287 allows you to output the saved parameters as a list.
The parameters are saved in target memory as a C string. The string can be up to 5000 characters long (incl. the null byte).
In addition, the length of the parameter list is output in bytes.
This makes it possible to save customer-specific parameter settings on the PC and send them back to the ND 287 with the command `ND2Set_ParameterList()`.

Function

```
ND2_ERR ND2Get_ParameterList    (     ND2_HANDLE         ND2,
                                      unsigned char*     ParameterList,
                                      unsigned long      size,
                                      unsigned long*     rlen
                                )
```

Parameters

ND2	ND2 handle
ParameterList	*[return code]* Pointer to the target memory for the parameter list
size	Size of the target memory in bytes
rlen	*[return value]* Pointer to the variable for the number of received bytes

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

Detailed information regarding the output structure of the parameter list is available in the ND 287 operating instructions under section:
II – 7 Input and output of parameter list and error compensation table

8.57 Sending a parameter list to the ND 287

A saved parameter list can be sent as a C string to the ND 287.
When receiving parameter lists, the ND 287 expects the same list structure as used for the output through the command `ND2Get_ParameterList()`.
After successfully receiving the parameter list, the ND 287 resets and reboots with the new parameter values. The connection to the ND 287 is automatically closed (as with ND2Close()).

Function

```
ND2_ERR ND2Set_ParameterList    (     ND2_HANDLE         ND2,
                                      unsigned char*     ParameterList,
                                      unsigned long      size
                                )
```

Parameters

ND2	ND2 handle
ParameterList	Pointer to the parameter list
size	Size of the parameter list in bytes

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_InvParamList	Not a valid parameter list

Detailed information regarding the output structure of the parameter list is available in the ND 287 operating instructions under section:
II – 7 Input and output of parameter list and error compensation table

8.58 Reading and converting data from the FIFO

Data packets are copied from the FIFO to the target memory and converted. The "cnt" parameter indicates the number of entries to be copied from the FIFO. If the FIFO contains fewer data records, the entire content of the FIFO is copied. The number of entries actually copied is returned via the "entries" parameter. The function waits until at least one data record has been copied from the FIFO, though no longer than expiration of the timeout. In this case, zero is returned to "entries". Entire elements are always copied from the FIFO. The target memory must be at least big enough to take the indicated number of FIFO entries. All data words are saved in the standard format for 16-bit or 32-bit integers and the position values are converted to the ENCODER_POSITION format.

Function

```
ND2_ERR ND2ReadFIFOData   (    ND2_HANDLE        ND2,
                               void*             data,
                               unsigned long     cnt,
                               unsigned long*    entries,
                               long              timeout
                          )
```

Parameters

ND2	ND2 handle
data	*[return value]* Pointer to target memory
cnt	Number of entries to be read (>=0)
entries	*[return value]* Number of entries to be copied
timeout	Timeout in milliseconds

Timeout	Description
0	Function returns immediately if no data is present
> 0	Function waits for data for x milliseconds
– 1	Function waits indefinitely

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_FIFOEmpty	No data in the FIFO
ND2_ElementSizeInv	Internal error
ND2_FIFOOverflow	FIFO overflow since the function was last called (data lost)

8.59 Reading the size of a FIFO element following conversion

The size of a FIFO element is generated following conversion. This value corresponds to the size of a FIFO entry, which is read using the ND2ReadFIFOData() function.

Function

```
ND2_ERR ND2SizeOfFIFOEntry   (    ND2_HANDLE        ND2,
                                  unsigned long*    size
                             )
```

Parameters

ND2	ND2 handle
size	*[return value]* Pointer to the variable for the size of a FIFO element in bytes

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.60 Access to the contents of a FIFO element with converted data

This function can be used to access individual fields of a FIFO element with converted position data (ENCODER_POSITION format). A FIFO entry contains, for example, the trigger counter, position data, and the status word. These contents are present for all axes of the ND 287.
This function provides a pointer to the relevant field within the data structure and also the field size in bytes. A general selection is made using the "region" parameter. This then allows the axis from which the field is obtained to be selected. The fine selection can be made using the "type" parameter. This indicates the data field on which an axis will be accessed.

Function

```
ND2_ERR ND2GetDataFieldPtr   (     ND2_HANDLE             ND2,
                                   void*                  data,
                                   ND2_                   DataRegion region,
                                   ND2_PositionDataField  type,
                                   void**                 field,
                                   unsigned long*         size
                             )
```

Parameters

ND2 ND2 handle
data Pointer to the data structure (FIFO element)
region Axis of the ND 287

region	Description
ND2_DR_Global	Global data field for trigger counter
ND2_DR_Encoder1	Data for axis 1
ND2_DR_Encoder2	Data for axis 2
ND2_DR_Encoder2	Data for axis 3

type Data element for an axis

Type	Description
ND2_PDF_TriggerCounter	Trigger counter (only in ND2_DR_Global)
ND2_PDF_StatusWord	Status word for position
ND2_PDF_PositionData	Position value
ND2_PDF_Reserved1	Reserved
ND2_PDF_Reserved2	Reserved

field *[return value]* Pointer to the memory address of the element from the data structure
size *[return value]* Size of the element in bytes

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_FieldNotAvail The indicated field cannot be found

8.61 Reading the number of elements in the FIFO

The number of elements currently stored in the FIFO is generated.

Function

```
ND2_ERR ND2FIFOEntryCount    (     ND2_HANDLE        ND2,
                                   unsigned long*    cnt
                             )
```

Parameters

ND2 ND2 handle
cnt *[return value]* Pointer to the variable for the number of FIFO elements

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.62 Clearing the FIFO

The contents of the FIFO are cleared. This command has no effect if polling mode is active.

Function

```
ND2_ERR ND2ClearFIFO    (     ND2_HANDLE   ND2
                        )
```

Parameters

ND2	ND2 handle

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.63 Setting the FIFO size

The size of the FIFO is reestablished. All data in the FIFO is cleared. The size can only be set in polling mode. The FIFO must be at least 2000 bytes large. If the value is smaller, the value 2000 bytes is used internally.

Function

```
ND2_ERR ND2SetFIFOSize    (     ND2_HANDLE        ND2,
                                unsigned long     size
                          )
```

Parameters

ND2	ND2 handle
size	FIFO size in bytes

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_SoftRTEn	Soft Real-Time mode is activated

8.64 Reading the FIFO size

The size of the FIFO in bytes is generated.

Function

```
ND2_ERR ND2GetFIFOSize    (     ND2_HANDLE        ND2,
                                unsigned long*    size
                          )
```

Parameters

ND2	ND2 handle
size	*[return value]* Pointer to the variable for the FIFO size in bytes

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

8.65 Activating the callback mechanism

The callback mechanism is activated or deactivated and the function pointer saved where applicable. The callback function is called if there are at least as many elements saved in the FIFO as there are in the "threshold" parameter. This function will then not be called until new data has been written to the FIFO and then at least "threshold" elements have been saved in the FIFO.

Function

```
ND2_ERR ND2SetDataCallback   (     ND2_HANDLE           ND2,
                                   void*                data,
                                   ND2_MODE             activate,
                                   unsigned long        threshold,
                                   ND2OnDataAvailable   handler
                             )
```

Parameters

ND2 ND2 handle
data Pointer to user data, this pointer is transmitted as a parameter to the callback function
activate Activate or deactivate callback

Activate	Description
ND2_MD_Disable	Deactivate the callback mechanism
ND2_MD_Enable	Activate the callback mechanism

threshold Number of elements in the FIFO from which the callback mechanism is triggered (>0)
handler Pointer to the callback function (ZERO is permitted if activate = 0)

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

Callback Function

The callback function is performed by the driver and runs in a separate thread. The user must attend to any necessary synchronization with the main program. The "ND2" parameter contains the handle on the ND 287 that has triggered the callback. "cnt" indicates the number of elements currently stored in the FIFO. The "data" parameter contains the pointer that was indicated when registering the callback function.

Prototype

```
typedef void (*ND2OnDataAvailable)     (     ND2_HANDLE      ND2,
                                             unsigned long   cnt,
                                             void*           data
                                       )
```

Parameters

ND2 ND2 handle
cnt Number of elements in the FIFO
data Pointer to user data

9 Axis Functions

The axis functions always refer to just one axis on the ND 287. None of the other axes is affected.
No more than 3 handles are permissible for axes for the ND 287.
The ND 287 can be provided with up to 2 encoder modules, axis 1 and axis 2.
The position values of the axes can be shown coupled (X1+X2; X1-X2; X1(fctX2).
Axis 3 is a virtual axis and is always shown with respect to the position displayed on the LCD of the ND 287.

All axis functions are able to deliver the following error messages as a return value. They can also return further values individually. These are listed separately for each function.

Standard return values

ND2_NoError	Function call successful
ND2_InvalidHandle	The handle on the axis of the ND 287 is invalid
ND2_FuncNotSupp	This function is not supported by ND 287.
ND2_InvalidResponse	Error in data transfer
ND2_AccNotAllowed	Function cannot be performed, as the ND 287 does not permit access.
ND2_ConnReset	Connection was terminated by the ND 287.
ND2_ConnTimeout	Timeout in the data transmission to the ND 287
ND2_ReceiveError	Error while receiving the data
ND2_SendError	Error while sending the data
ND2_OutOfMemory	The system is unable to allocate sufficient memory.

9.1 Clearing the counter

Resets the selected datum of the displayed axis to zero.
In an axis coupling, the datums of both axes are set to zero.

Function

```
ND2_ERR ND2ClearCounter    (     ND2_AXIS     axis
                           )
```

Parameters

axis	AXIS handle

Return value

The return value delivers a status for the function call. All potential values are listed for the standard return values.

9.2 Interrogating a position

The current position value of the axis is read out A status word from which potential position errors emerge is also transmitted. The position can only be polled in polling mode.

Function

```
ND2_ERR ND2GetPosition     (     ND2_AXIS           axis,
                                 unsigned short*    status,
                                 ENCODER_POSITION*  pos
                           )
```

Parameters

axis	AXIS handle
status	*[return value]* Pointer to the variable for the status word
pos	*[return value]* Pointer to the variable for the position

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_NotInitialized	Axis is not configured

9.3 Reading out the configuration of an axis

Reading out the configuration data of an axis (possible only for axes 1 and 2).

Function

```
ND2_ERR ND2GetAxisInfo     (     ND2_AXIS                axis,
                                 ND2_TYPE_OF_INTERFACE*  type_interface
                                 ND2_TYPE_OF_ENCODER*    type_encoder
                                 ND2_KIND_OF_REFMARKS*   kind_refmarks
                           )
```

Parameters

axis AXIS handle

type_interface *[return value]* Pointer to the target variable for the interface

type_interface	Description
ND2_NoSignal	No encoder connected or detected
ND2_1Vss	Encoder with 1 Vpp interface
ND2_11uAss	Encoder with 11 µApp interface
ND2_Endat22	Encoder with EnDat 2.2 interface
ND2_Endat21	Encoder with EnDat 2.1 interface
ND2_Analog	Analog encoder

type_encoder *[return value]* Pointer to the target variable for the encoder type

type_encoder	Description
ND2_linear	Linear encoder
ND2_rotary	Angle encoder
ND2_TempComp	Thermistor
ND2_Sensor	Analog sensor

kind_refmarks *[return value]* Pointer to the target variable for the reference marks

kind_refmarks	Description
ND2_none	Encoder without reference mark
ND2_one	Encoder with one reference mark
ND2_DistCoded	Encoder with distance-coded reference marks

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_InvChannel	Invalid axis handle (only axis 1 and axis 2 allowed)

9.4 Verifying reference status

This makes it possible to verify whether the reference points in an axis were scanned and the reference position was saved or the axis is not referenced.

Function

```
ND2_ERR ND2GetRefMode   (     ND2_AXIS     axis,
                              ND2_MODE*    active
                        )
```

Parameters

axis AXIS handle

active *[return value]* Pointer to the variable for the status

Activate	Description
ND2_MD_Disable	Axis is not referenced
ND2_MD_Enable	Axis is referenced

The return value delivers a status for the function call. All potential values are listed for the standard return values.

9.5 Outputting the recorded data of a measurement series as a list

The ND 287 makes it possible to output the saved measurement data and measured values for analysis of a measurement series. The internal memory of the ND 287 can evaluate and save up to 10000 measured values per axis. The measured data are saved in target memory as a C string.
The string can be up to 233 000 characters long (incl. the null byte). In addition, the length of the measured data is output in bytes.

Function

```
ND2_ERR ND2Get_MeasSerie_Value    (    ND2_AXIS           axis,
                                       unsigned char*     SPC_Values,
                                       unsigned long      size,
                                       unsigned long*     rlen
                                  )
```

Parameters

axis	AXIS handle
SPC_Values	*[return value]* Pointer to the target memory for the data of a measurement series
size	Size of the target memory in bytes
rlen	*[return value]* Pointer to the variable for the number of received bytes

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_NoData	No measurement series data available (memory empty)
ND2_InvChannel	Invalid axis handle (only axis 1 and axis 2 allowed)

Example: Measured value list of a measurement series

```
#
MEASUREMENT-DATA: X1

MAX            =  +  0.2515
MIN            =  –  0.0595
DIFF           =  +  0.3110
AVERAGE        =  +  0.0786
STD. DEV.      =     0.1200

   1 =   –  0.0185
   2 =   –  0.0180
   3 =   –  0.0110
   4 =   –  0.0230
   5 =   –  0.0445
   6 =   –  0.0465
   7 =   –  0.0595
   8 =   –  0.0430
   9 =   –  0.0405
  10 =   –  0.0405
  11 =   +  0.2195
  12 =   +  0.2475
  13 =   +  0.2515
  14 =   +  0.2360
  15 =   +  0.2190
  16 =   +  0.2045
  17 =   +  0.2045
  18 =   +  0.1305
  19 =   +  0.1035
  20 =   +  0.1010
#
```

9.6 Outputting a compensation-value table

The ND 287 allows you to output an internally saved compensation-value table as a list.
The compensation-value table is saved in target memory as a C string. The string can be up to 12 000 characters long (incl. the null byte). In addition, the length of the compensation-value table is output in bytes.
This makes it possible to save compensation-value tables on the PC and send them back to the ND 287 with the command `ND2Set_ErrorCompTable()`.

Function

```
ND2_ERR ND2Get_ErrorCompTable     (     ND2_AXIS           axis,
                                        unsigned char*     CompTable,
                                        unsigned long      size,
                                        unsigned long*     rlen
                                  )
```

Parameters

axis	AXIS handle
CompTable	*[return value]* Pointer to the target memory for the compensation-value table
size	Size of the target memory in bytes
rlen	*[return value]* Pointer to the variable for the number of received bytes

Return value
The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_NoData	No compensation-value table available for this axis
ND2_InvChannel	Invalid axis handle (only axis 1 and axis 2 allowed)

Detailed information regarding the output structure of the compensation-value table is available in the ND 287 operating instructions under section:
II – 7 Input and output of parameter list and error compensation table

9.7 Sending a compensation value table to the ND 287

A saved compensation-value table can be sent as a C string to the ND 287.
When receiving compensation-value tables, the ND 287 expects the same list structure as used for the output through the command
`ND2Get_ErrorCompTable()`.

Function

```
ND2_ERR ND2Set_ErrorCompTable   (     ND2_AXIS           axis,
                                      unsigned char*     CompTable,
                                      unsigned long      size
                                )
```

Parameters

axis	AXIS handle
CompTable	Pointer to the compensation-value table
size	Size of the compensation-value table in bytes

Return value
The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_InvCmpTable	No value compensation-value table

Detailed information regarding the output structure of the compensation-value table is available in the ND 287 operating instructions under section:
II – 7 Input and output of parameter list and error compensation table

10 General Functions

All general functions can provide the following error messages as a return value. They can also return further values individually. These are listed separately for each function.

Standard return values

ND2_NoError	Function call successful
ND2_OutOfMemory	The system is unable to allocate sufficient memory.

10.1 Reading the driver ID number

The product number (ID) of the driver is generated as a C-string. The string is saved to the "ident" pointer. The size of the memory for the string must be indicated in bytes via the "len" parameter. If the string, including the final zero byte, is longer than the memory area, an error message will be generated. The target memory must be at least 9 bytes.

Function

```
ND2_ERR ND2GetDriverID    (     char*           ident,
                                unsigned long   len
                          )
```

Parameters

ident	*[return value]* Target memory for the C-string
len	Size of the target memory in bytes

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_BufferTooSmall	Target memory is too small

10.2 Converting error message into text

An error code is converted into a text message and returned as a C-string. A descriptive text and a brief description are defined in the system for all known error codes. The "mnemonic" parameter is used to return a brief description of the error message in text format (approx. 30-40 characters). The "message" parameter contains a more detailed description (approx. 100-150 characters). If a ZERO point is transmitted for either the "mnemonic" or "message" parameters, the function does not copy the corresponding text. If the target memory is too small to take the entire text, only the first part is copied. The string always ends with a zero byte.

Function

```
ND2_ERR ND2GetErrorInfo   (     ND2_ERR            code,
                                char*              mnemonic,
                                unsigned long      mnemlen,
                                char*              message,
                                unsigned long      msglen
                          )
```

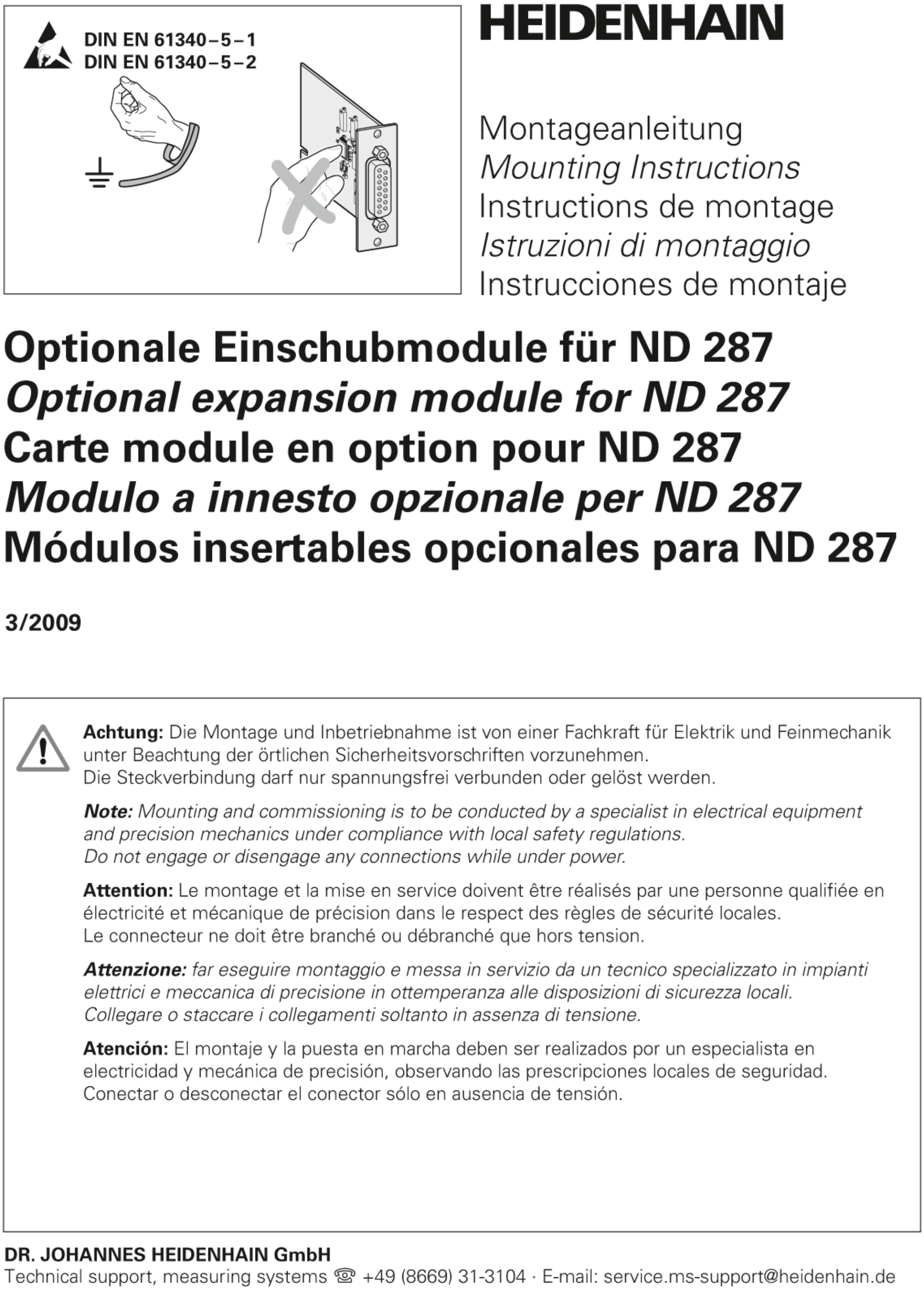
Parameters

code	Error code that is converted into text
mnemonic	*[return value]* Pointer to the target memory for the brief description
mnemlen	Size of the "mnemonic" target memory in bytes
message	*[return value]* Pointer to the target memory for the error text
msglen	Size of the "message" target memory in bytes

Return value

The return value delivers a status for the function call. In addition to the standard return values, the following error messages can also occur.

ND2_IllegalParameter	Invalid error code



DIN EN 61340–5–1
DIN EN 61340–5–2

HEIDENHAIN

Montageanleitung
Mounting Instructions
Instructions de montage
Istruzioni di montaggio
Instrucciones de montaje

Optionale Einschubmodule für ND 287
Optional expansion module for ND 287
Carte module en option pour ND 287
Modulo a innesto opzionale per ND 287
Módulos insertables opcionales para ND 287

3/2009

Achtung: Die Montage und Inbetriebnahme ist von einer Fachkraft für Elektrik und Feinmechanik unter Beachtung der örtlichen Sicherheitsvorschriften vorzunehmen.
Die Steckverbindung darf nur spannungsfrei verbunden oder gelöst werden.

Note: *Mounting and commissioning is to be conducted by a specialist in electrical equipment and precision mechanics under compliance with local safety regulations.*
Do not engage or disengage any connections while under power.

Attention: Le montage et la mise en service doivent être réalisés par une personne qualifiée en électricité et mécanique de précision dans le respect des règles de sécurité locales.
Le connecteur ne doit être branché ou débranché que hors tension.

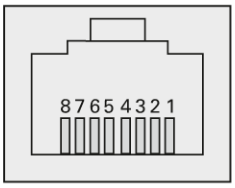
Attenzione: *far eseguire montaggio e messa in servizio da un tecnico specializzato in impianti elettrici e meccanica di precisione in ottemperanza alle disposizioni di sicurezza locali.*
Collegare o staccare i collegamenti soltanto in assenza di tensione.

Atención: El montaje y la puesta en marcha deben ser realizados por un especialista en electricidad y mecánica de precisión, observando las prescripciones locales de seguridad.
Conectar o desconectar el conector sólo en ausencia de tensión.

DR. JOHANNES HEIDENHAIN GmbH
Technical support, measuring systems ☎ +49 (8669) 31-3104 · E-mail: service.ms-support@heidenhain.de

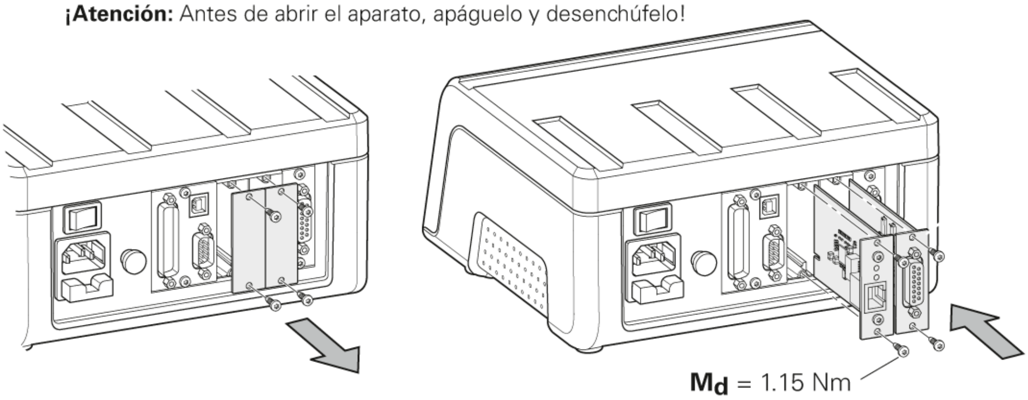
Anschlussbelegung
Pin Layout
Raccordements
Piedinatura
Distribución del conector

Ethernet RJ45



	Serielle Schnittstelle *Serial interface* Interface série *Interfaccia seriale* Interfaz serie
1	D1+ (TX+)
2	D1– (TX–)
3	D2+ (RX+)
6	D2– (RX–)

Achtung: Vor Öffnen des Gerätes, Gerät ausschalten und Netzstecker ziehen!
Caution: *Switch the unit off and disengage the power plug before opening the unit!*
Attention: Avant d'ouvrir l'appareil, le mettre hors-tension et ôter la prise secteur!
Attenzione: *prima dell'apertura, spegnere l'apparecchiatura e estrarre il cavo di rete*
¡Atención: Antes de abrir el aparato, apáguelo y desenchúfelo!



X2
Messgeräte-Modul (15-polig), oder ein Analog-Modul (9-polig).
Encoder module (15-pin), or an analog module (9-pin).
Module pour système de mesure (15 plots) ou un module analogique (9 plots).
Modulo sistemi di misura (15 poli), oppure modulo analogico (9 poli).
Módulo de sistema de medida (15 polos), o un módulo analógico (9 polos).

X26(X27)
Ethernet-Modul
Ethernet module
Module Ethernet
Modulo Ethernet
Módulo Ethernet

HEIDENHAIN

DR. JOHANNES HEIDENHAIN GmbH
Dr.-Johannes-Heidenhain-Straße 5
83301 Traunreut, Germany
☎ +49 8669 31-0
FAX +49 8669 5061
E-mail: info@heidenhain.de

Technical support FAX +49 8669 32-1000
Measuring systems ☎ +49 8669 31-3104
E-mail: service.ms-support@heidenhain.de
TNC support ☎ +49 8669 31-3101
E-mail: service.nc-support@heidenhain.de
NC programming ☎ +49 8669 31-3103
E-mail: service.nc-pgm@heidenhain.de
PLC programming ☎ +49 8669 31-3102
E-mail: service.plc@heidenhain.de
Lathe controls ☎ +49 8669 31-3105
E-mail: service.lathe-support@heidenhain.de

www.heidenhain.de

732 778-20 · Ver00 · pdf · 12/2009