Application Notes

Contents

This document and the cross-references refer to EnDat Specification 297403-09-D-01.

Please also note:

- EnDat 2.2 area on the HEIDENHAIN homepage (www.endat.de or www.heidenhain.de → Fundamentals → Interfaces → EnDat 2.2)
- EnDat seminar documents

Remark:
Changes to the previous version are shown in green (partly only the corresponding heading).

HEIDENHAIN
DR. JOHANNES HEIDENHAIN GmbH
83301 Traunreut, Germany

1 How to Get Started

The following overview is intended to provide support for implementing the EnDat interface in subsequent electronics. For more information, see www.heidenhain.de → Fundamentals → Interfaces → EnDat 2.2. This part of the website is divided into the following sections:

- Implementation of EnDat
- EnDat master
- Functional safety
- Encoder characteristics
- FAQ

Implementation aid for the OEM

The required implementation effort depends on the supported encoder models and the desired functions, e.g. datum shift. You can do a reduced implementation and concentrate on a specific encoder, or be very flexible and cover a broad range of applications.
The EnDat implementation questionnaire is intended as a help to define the OEM's requirements for the EnDat implementation and to prepare a customized implementation recommendation for the OEM. For more information, please refer to your contact person at HEIDENHAIN.

Documentation:

This overview does not replace the detailed documentation on the EnDat interface.
For more information about the available documents, see the "Implementation" section on the web site.

Supported encoder types:

The following encoder types are currently supported by the EnDat 2.2 interface (this information can be read out from the encoder's memory area):

- Incremental linear encoder
- Absolute linear encoder
- Incremental rotary encoder or incremental angle encoder
- Singleturn rotary encoder or singleturn angle encoder
- Multiturn rotary encoder with gears
- Multiturn encoder with battery buffer
- ….

In some cases, parameters must be interpreted differently for the various encoder types (see EnDat Specifications) or EnDat additional data must be processed (e.g. incremental or battery-buffered encoders). The parameters required for operating the encoder and the supported functions are stored in the memory of the encoder. Further information on incremental or battery-buffered encoders can be taken from the EnDat Application Note.
The "EnDat Parameter Overview" chapter contains an overview specifying which EnDat parameters are important for which encoder type.

Encoder functions:

For more information on the encoder functions listed below, refer to the respective chapters:

- OEM memory area
- Datum shift
- Online diagnostics
- Mounting diagnostics

For information on the reduction in the recovery time t_M, see the FAQ section on the web site.

EnDat monitoring functions:
The EnDat interface provides numerous monitoring possibilities, such as for:

- The transmission path
- The encoder
- The integrated memory for OEM data, for example

For more information, see the “EnDat Monitoring Functions” chapter.

Position data formats:
The position data format depends on the encoder type. For more information, see the “Position Data Formats” chapter. The maximum bit width for the position transfer is 48 bit.

Cycle time and readout time:
The readout time is defined as the time it takes from a position request until the position is available in the control. The cycle time is the time required for a complete EnDat communication cycle. The calculation of both values is described in the “Calculation of Readout Time and Cycle Time” chapter. The requirements for the two types of times result from the requirements of the customer’s application. The central time-determining element is the selected clock frequency. The clock frequency must be selected such that the requirements of the customer’s application regarding the cycle time and the readout time are fulfilled. However, the clock frequency should be set no higher than required, and not as high as possible. This is due to the fact that the requirements to be fulfilled by cables and connectors increase with increasing frequencies.

Maximum permissible clock frequency:
The maximum permissible clock frequency depends on the ordering designation and the cables used. In principle, there are three operating modes:

- EnDat01 modes: A maximum of 2 MHz is permissible, the clock frequency must be reduced depending on the cable length; typically, approx. 300 kHz are used to permit a cable length of 150 m.
- EnDat21 modes: Max. 2 MHz with delay compensation.
- EnDat22 modes: Typically 8 .. 16 MHz; the max. clock frequency can be read out from the EnDat 2.2 parameter range, word 25.
  (Note: certain encoders, such as AEF 1323, have other values; see the encoder documentation)

There are two operating modes for encoders with ordering designation EnDat02:

- EnDat01 modes: An adapter cable connecting the incremental signals with the subsequent electronics is used.
- EnDat22 modes: A purely serial adapter cable is used.

Since the introduction of the EnDat 2.2 encoders, the ordering designation has been stored in the EnDat 2.1 parameter range of the encoder in word 40. If you want to adapt the clock frequency of the subsequent electronics flexibly to the connected encoder, the following procedure is recommended:

Is the ordering designation (EnDat 2.1 range, word 40) stored in the encoder?

- No (≙ 0x000 bzw. 0xFFFF)
  - Are words 15+16 “0x0000” stored in the EnDat 2.1 area?
    - Yes → EnDat21 modes
    - No → EnDat01 modes
- Yes
  - “22” → See EnDat22 modes
  - “02” → Are incremental signals available?
    - Yes → EnDat01 modes
    - No → EnDat22 modes

Flexibility and implementation of the EnDat interface:
The information detailed above shows the high flexibility of the EnDat interface. This makes it possible to support a large number of applications. However, beforehand you should think about which range of possibilities should be implemented. You can keep the implementation to a minimum and focus on a specific encoder, or you can go for an implementation that supports the entire scope of the EnDat encoders. Based on the examples (see below) and depending on the requirements for the desired flexibility, the following implementations steps are recommended.

1. Power up and read position
2. Read and write parameters
3. Support of absolute encoder type and effect on reading of parameters and position calculation in degrees or meters
4. Error handling
5. Support of sensors and diagnostics
6. Support of other encoder types: incremental and battery-buffered

Implementation examples:

- Chapter 14: “Example for minimum implementation”
- Chapter 14: “Basic communication with purely serial absolute linear, singleturn and gear-based multiturn encoders from HEIDENHAIN” (the implementation example for the Arduino board is also based on this sequence)
- Chapter 14: “Example for flexible implementation”
- Implementation of other encoder models (e.g. incremental) and further functions, such as datum shift, readout of diagnosis, etc. (see program example for the EIB 74x).

Individual recommendations can be provided upon request (using the form sheet available for this). For more information, please refer to your contact person at HEIDENHAIN.

Sequences and data structures:
This chapter describes the basic mechanisms when accessing data from the encoder memory, as well as the access to EnDat additional data.

Power-up and procedure after switch-on:
See the FAQ section on the website and the “Sequences and Data Structures” chapter.

EnDat master
EnDat masters are available from various providers. See the Web site at “EnDat Master”. The masters are field-proven and therefore their use is highly recommended. In principle, a master can also be implemented based on the EnDat Specifications, but this is not recommended. Please contact HEIDENHAIN beforehand. For more information on the implementation of the mode commands, see the EnDat Specification or the “Notes on the Implementation of EnDat Mode Commands” chapter.

Remark:
For information on the RS-485 transceiver, see the “FAQ” area on the Web site.

Test platform / Testing the EnDat implementation:
HEIDENHAIN provides various tools for supporting customers in conducting tests:

- For the EnDat demo tool software, see the “Implementation” section on the website.
- EIB 74x: The EIB 74x can be used to test more complex sequences; see also the EIB 74x Product Information at www.heidenhain.de. Please contact HEIDENHAIN for information about the availability of specific EnDat demo programs for the EIB 74x.
- For the EnDat error injector, see the corresponding chapter
- See the corresponding chapter for more information

2 Sequences and Data Structures

General information:
The following flowcharts illustrate the basic procedure used to obtain the desired information in EnDat 2.2. In order to make access to parameters or memory information possible in a closed control loop, it was mainly necessary to modify the access to the EEPROM compared to EnDat 2.1. The acknowledgments of MRS codes and addresses were also changed. The following information is provided to aid understanding of the processes:

Mode commands:
See also the “Data Transfer” chapter of the EnDat specification. Please pay particular attention to the timing conditions after Chapter 2.3.2.

Mode commands according to Type 2.1 have been configured for the following typical application:

- Reading or writing of parameters and resetting of errors or warnings occur outside of the closed control loop.
- Absolute position values are not usually read out cyclically via EnDat but are only required during the power-on procedure after the incremental signals (relative position). The control loop is based on the incremental signals. Cross-referencing of absolute to incremental information occurs (if at all) with low cycle time.
- Because of the propagation-delay compensation, a clock frequency of 300 kHz should not be exceeded.
- Encoders with the EnDat21 ordering designation are an exception. These encoders are also used in purely serial mode.

Mode commands according to Type 2.2 have been configured for the following typical application:

- The main difference compared to mode commands according to Type 2.1 is:
  - The position value is also always transferred
  - Additional data can also be transferred
  ➔ All access can occur in the closed control loop.
- Encoder operation is purely serial, meaning an absolute position value is transmitted with each control cycle.

Memory organization (EEPROM):
The memory is organized in word size, i.e. 16 bits.
Different memory areas, such as "operating status" or "parameters of the encoder manufacturer for EnDat 2.1", are defined in EnDat 2.2. The organization of the memory areas is not based on absolute addresses, but on MRS codes (Memory Range Select). These MRS codes are identical for all encoders and are independent of the actual physical organization of the EEPROM. In addition, the address within the MRS area is required to be able to access a specific piece of data.
For example, two mode commands are necessary to access the "supported error messages" data (word 35; see 3.4.20) in the "parameters of the encoder manufacturer for EnDat 2.1“:

1. Selection of the memory area (MRS code "10100101")
2. Reading out the content of address "0x03"

Reading out can be executed with EnDat 2.1 or EnDat 2.2 mode commands.

Note:
Before accessing the memory, the MRS code for the desired range must always be set first. An MRS range remains set until
a) a new MRS range is selected
b) an EnDat 2.1 reset is sent
c) an EnDat 2.2 reset is sent

EEPROM life expectancy:
Due to their physical design, this is because the requirements write accesses and therefore cyclic storage of data is only possible to a limited extent. The manufacturers of the EEPROMs used in the encoders typically guarantee 100,000 write-accesses. This value cannot be guaranteed for the EEPROMs of future encoders. A page orientation of the EEPROM, for example, can radically reduce the number of accesses. For more information, please contact HEIDENHAIN. This must be observed when using EnDat mode commands that trigger write access to the EEPROM. It must also be noted that an EEPROM is not suited as memory for the cyclical storage of data. **One EEPROM access takes max. 12 ms. When a corresponding EnDat mode command has been output, the power supply of the encoder must have stable application for at least 12 ms. This also applies to EnDat 2.2 mode commands that only initiate an EEPROM access but do not wait for the termination ("busy bit").**

The following mode commands initiate write accesses to the EEPROM:

•	EnDat 2.1:	Encoder receive parameter	(011 100)
•	EnDat 2.2:	Encoder send position values and receive parameter	(011 011)
•		Encoder send position values and receive error reset	(101 101)

Application: Resetting error messages
Resetting error messages (EnDat 2.1 or 2.2 mode command) should only then be activated when a set composite error bit F1/F2 shows existing errors. This is also true for the power-on process.

Application: Resetting warnings
- Text: See above: Error -> Warnings.
- The warning "Reference point not reached" is set with incremental encoders until referencing is completed. Clearing this warning, for example following start-up, has no effect as the warning is continually reset until the reference point is reached. The warning can only be deleted with RM bit = 1. For further details see the "Incremental Encoders" chapter.

Additional data 1 or 2:
The additional data is saved in the non-volatile memory of the encoder, and not in the EEPROM. This is necessary to be able to switch between the pieces of additional data within a control cycle. Each additional data was assigned its own MRC code so that only one mode command is necessary for switchover. An address must not be specified. Not all encoders provide the full range of additional data. The available additional data can be read out of the EnDat 2.2 parameter, words 0 and 1.

Example of selection/deselection and rollling output: (see also the EnDat seminar in the "Processes" chapter):

Additional datum 1:
- Rolling transmission of temperature 1 (Temp1: MRS code 0x4C) and temperature 2 (Temp2: MRS code 0x4D)

Additional datum 2:
- Transmission of the operating status error sources (BZFQ: MRS code 0x59)

Mode command 9 (001001): "Encoder transmit position value and receive selection of memory area"

Subsequent electronics → encoder	Encoder→ subsequent electronics
Mode command 9 + MRS code 0x59	Position
Mode command 9 + MRS code 0x4C	Position + BZFQ (0x59)
Mode command 9 + MRS code 0x4D	Position + BZFQ (0x59) + Temp1 (0x4C)
Mode command 9 + MRS code 0x4C	Position + BZFQ (0x59) + Temp2 (0x4D)
Mode command 9 + MRS code 0x4F	Position + BZFQ (0x59) + Temp1 (0x4C)
Mode command 9 + MRS code 0x5F	Position + BZFQ (0x59)
Mode command 9 + MRS code 0x5F	Position

Required processes for accessing the EEPROM:
The EEPROM in the encoder must be accessed for sending or receiving parameters. The access time to the EEPROM is max. 12 ms.

EnDat 2.1:
When the EEPROM is accessed, the system waits until the data is available (i. e. has been read out from the EEPROM) after the desired address has been sent. Then the data is immediately output (see section 2.5 or 2.6).

EnDat 2.2:
When EnDat 2.2 commands are used, all requests are also possible in a closed control loop. Since the control cycles are usually considerably shorter than 12 ms, the access procedure must differ from that in EnDat 2.1.
After the desired address has been transferred, the system does not wait. Only the action "Read from EEPROM" is initiated inside the encoder. The "busy bit" is set during the time the EEPROM takes for reading out the selected memory location.
The busy bit is included in the additional data and must be requested by the user; this means a position value with selected additional data must be requested. These requests are required so that the EEPROM access in the encoder can be handled.
After the busy bit has been reset by the encoder, the contents of the selected address can be read out via additional data 1. Two requests (LSB and MSB) are necessary for this. Also see the "Reading out memory contents" flowchart.
When additional data 1 or 2 is read out, the information is immediately available, because no access to the EEPROM is required. In this case, it is not necessary or reasonable to monitor the busy bit.

Acknowledgements:
Basically, acknowledgments should be requested for all requests made as well as for all requested MRS codes and addresses (read from or write to memory contents).
The acknowledgment in EnDat 2.1 differs from that in EnDat 2.2:

MRS code:
In EnDat 2.1, the MRS code is acknowledged by the encoder immediately after transmission of the MRS code (see 2.4).
In EnDat 2.2, the MRS code is not directly acknowledged. The acknowledgment is requested via additional data 1; see the "Requesting the acknowledgment of an MRS code“ flowchart.

Addresses:
In EnDat 2.1, the address is acknowledged by the encoder immediately after transmission of the address (see 2.4).
In EnDat 2.2, the address is included in additional data 1 when the memory contents are requested.
A 16-bit word is output, which is divided into 8 bits for the address and 8 bits for the LSB or MSB (the memory is organized in words) of the requested memory location; see the "Reading out the memory contents" flowchart.

Note:
When information is saved in the encoder, the saved contents should be checked by reading them out separately.

EnDat memory access during closed-loop operation:
EnDat 2.1 mode commands are always used to access memory if the access takes place outside of closed-loop operation, e.g. during the initialization phase of the subsequent electronics. This is the standard application case and should be used preferentially when no access in the closed control loop is needed.
EnDat 2.2. mode commands are always used when the memory is to be accessed during closed-loop operation. For cyclic requests, the address must be sent again.

General information on the processing of error messages, warnings and error handling in the interface:

Error messages:
The two composite error bits F1 and F2 should be monitored in every query. F1 and F2 are generated independently of each other, and must be evaluated separately from each other. It is recommended that errors and warnings be buffered in the subsequent electronics before they are cleared (for diagnostic purposes at a later date, e.g. error log).
Clearing of errors and warnings is basically possible with EnDat 2.1 and EnDat 2.2 commands. However, not all causes of errors can be reset within one communication cycle with the EnDat 2.2 command. In these cases the error messages can remain set. The following priorities for the clearing of errors are the result (the timing requirements of an application decide when the next priority stage is to be switched to):

- Stage 1: Clearing with EnDat 2.2 mode commands in the control cycle
  Resetting of error causes that can be reset within one control cycle
- Stage 2: Clearing with EnDat 2.1 mode commands
  Resetting of error causes that require the encoder to be re-initialized
- Stage 3: Switching the unit off and on again
  Resetting of error causes that require the encoder to be entirely restarted

Warnings:
The group bit for warnings is transmitted with the additional data. The warning bit should be queried at regular intervals, even if the application does not require the processing of additional data.

Error handling of the interface:
Along with the CRC check there are three different types of errors, which should also be monitored continuously; see Attachment A2.

Also see the “EnDat Monitoring Functions” section.

Notes:

- The queries detailed above are not included in the following flowcharts due to space considerations.
- Errors and warnings should be reset independently of each other; see the sequence “Clearing errors or warnings”.
- As long as the encoder outputs a group error bit, it must be assumed that the values transmitted by the encoder, particularly the position values, are erroneous.
- With incremental encoders certain error states can only be detected after traversing the reference mark(s).

Encoder models or profiles

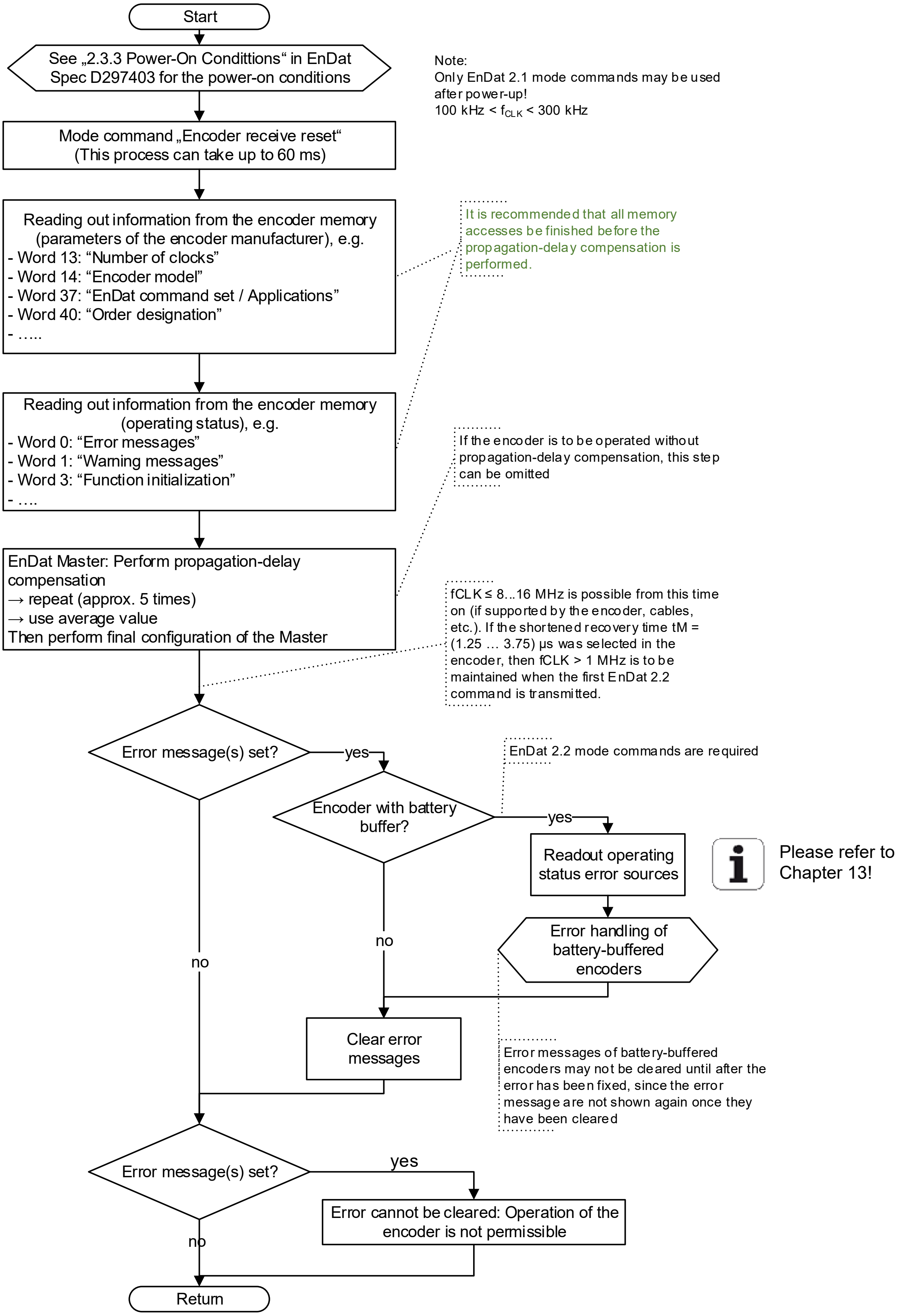
The supported encoder models are stored in “parameters of the encoder manufacturer,” word 14.

In some cases, parameters must be interpreted differently for the various encoder models (see EnDat Specification) or EnDat additional data must be processed (e.g. incremental or battery-buffered encoders).
Refer to the respective chapters for more information on incremental and battery-buffered encoders.

The various profiles must be taken into account for EnDat implementation or for testing the implementation.

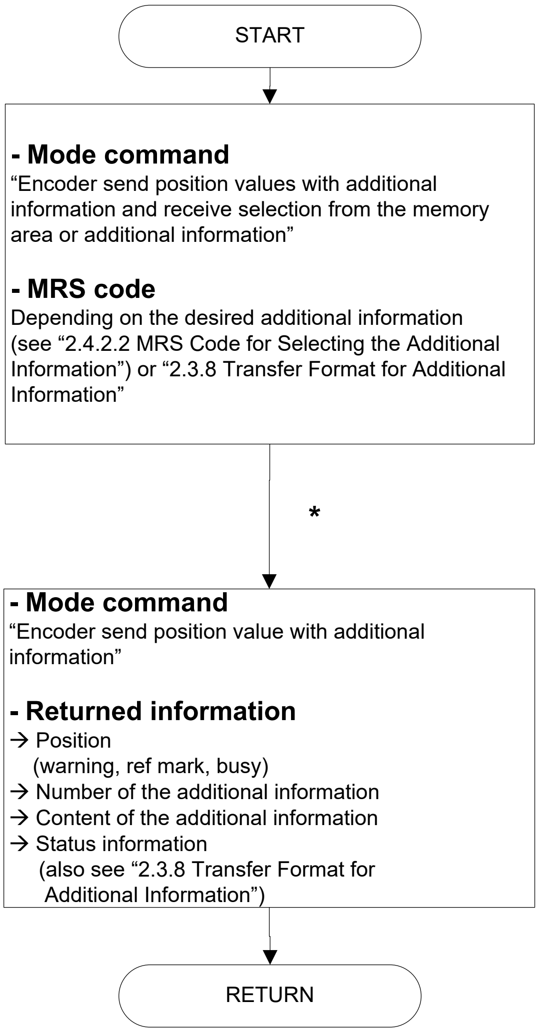
Note:
Further examples on processes are provided in the EnDat Seminar.
Influcene of the EnDat Master
Regarding the permitted and recommended clock frequencies, limitations by the EnDat Master can also occur. Please refer to the documentation for the Master.

Power-on procedure



Selecting additional information

Additional information 1 and/or additional information 2 can be selected. For each additional information the mode command "Encoder send position values with additional information and receive selection from the memory area or additional information" must be given with the corresponding MRS code.
See "3.9.2 Status of Additional Information 1" and "3.9.3 Status of Additional Information 2" for which additional information is supported by the encoder.



** Note:*

Acknowledgment of the MRS code is not useful when selecting additional information.
The information about which additional information is being sent is included in the answer.

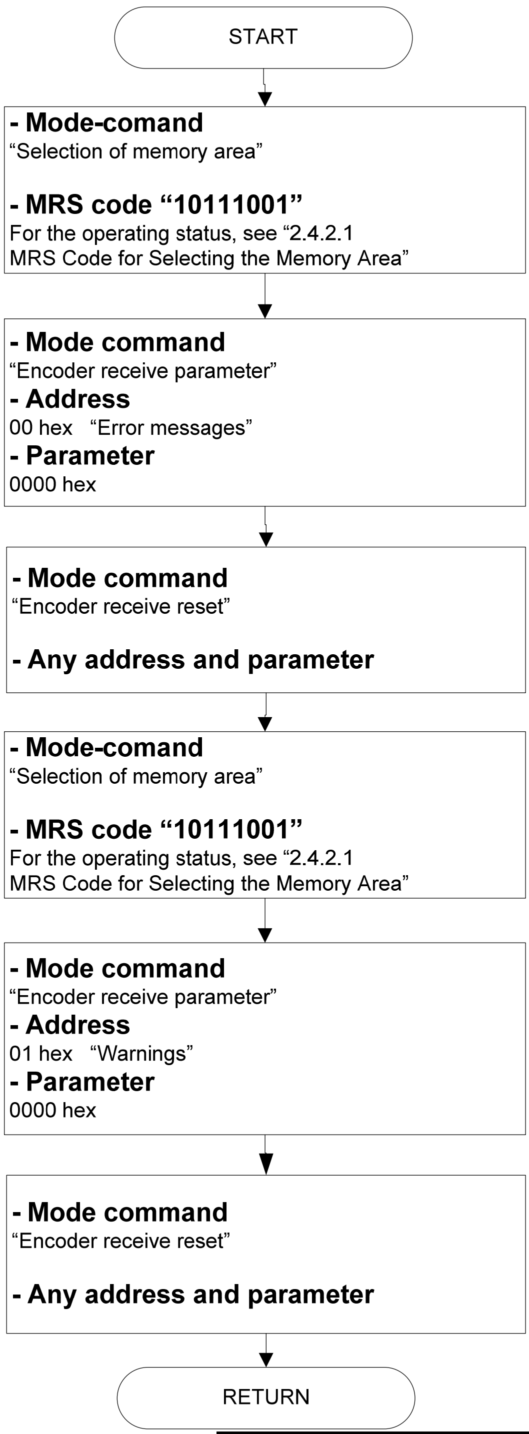
HEIDENHAIN
DR. JOHANNES HEIDENHAIN GmbH
83301 Traunreut, Germany

Clearing errors or warnings

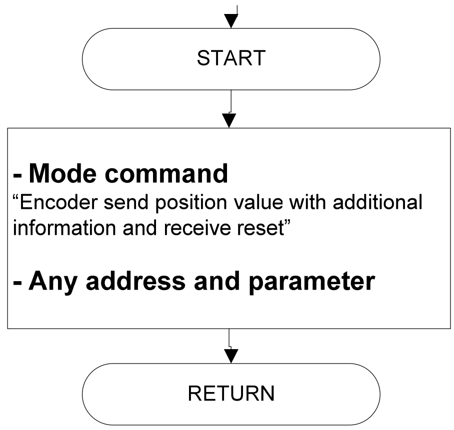
Also see “3.4.1 Error Messages” and 3.4.2 “Warnings”.
To clear errors and warnings, EnDat 2.1 mode commands (see sequence diagram) or the EnDat 2.2 mode command “Encoder send position values with additional information and receive error reset” can be used. This mode command automatically deletes errors and warnings. See also ”2.7.1 Encoder send position values and receive error reset”.
After power-on, errors and warnings must be cleared (see “Sequence for power-on” sequence diagram); EnDat 2.1 mode commands must be used for that. During closed-loop operation both sequences can be used. Please note the general information on the processing of error messages.

Please note:
The service life of the EEPROM is specified as 100,000 access cycles. Depending on the application, it is advisable that you interrogate whether any error or warning messages are pending before performing a general deletion.

Using EnDat 2.1 mode commands

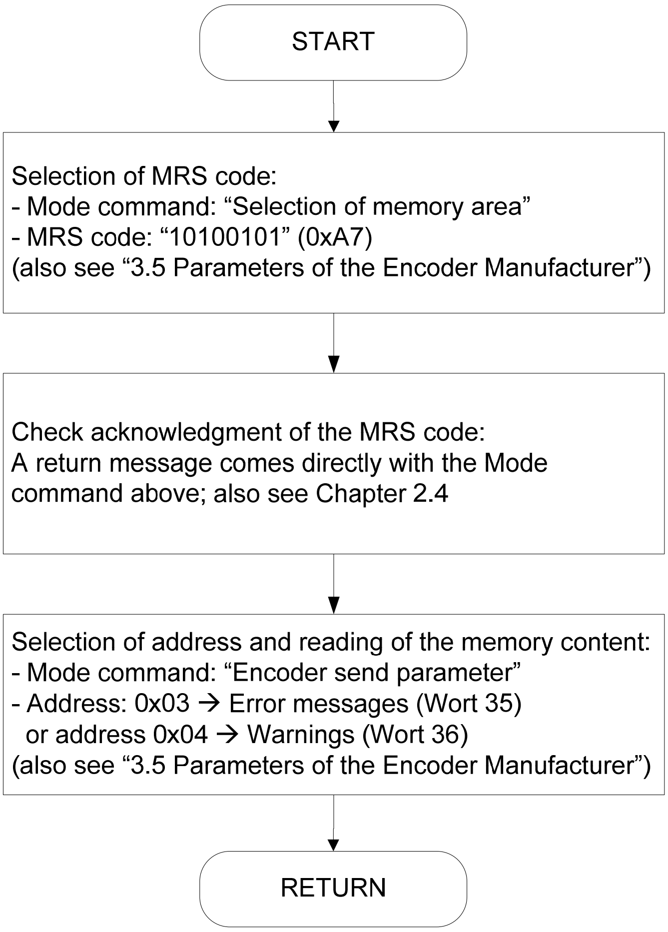


Using the EnDat 2.2 mode command



Memory access by EnDat 2.1:
Supported error messages and warnings

The interrogation of other memory areas, e.g. “Operating parameters”, etc. functions in the same manner.
The MRS code and the address must be adapted accordingly.

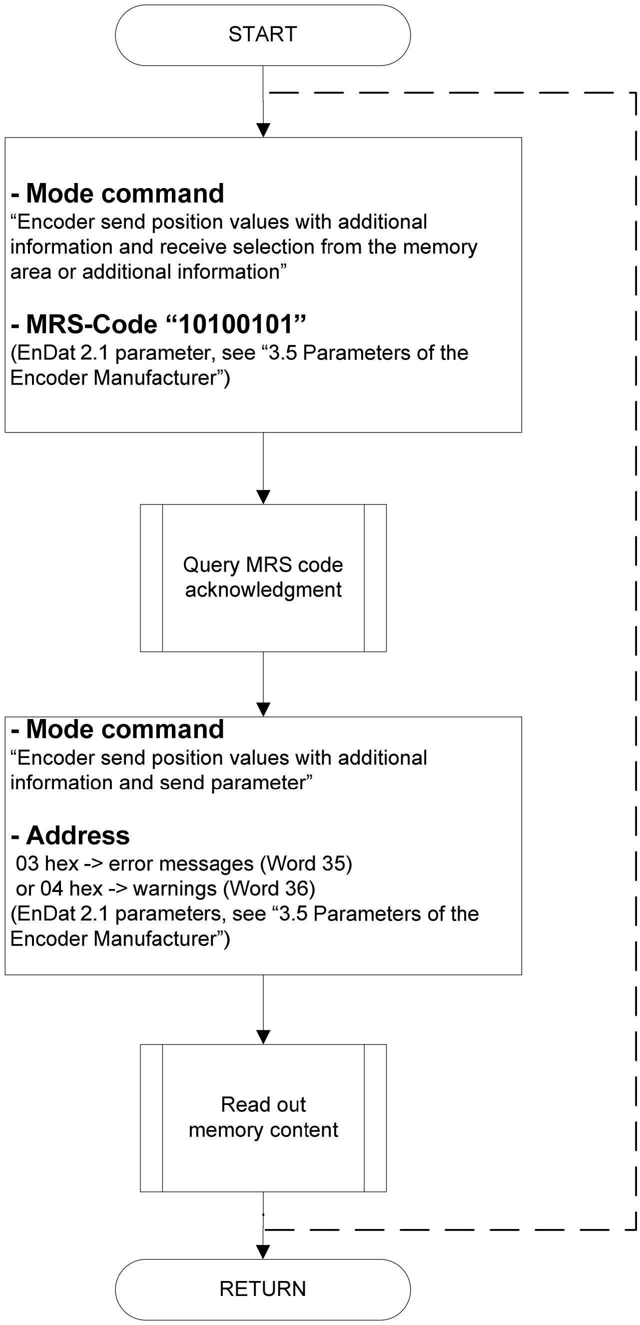


Memory access by EnDat 2.2: Supported error messages and warnings

The query of supported additional information, the diagnostic status, etc., occurs in a similar manner with the appropriate MRS codes (also see “3.9 Parameters of the Encoder Manufacturer for EnDat 2.2”) and addresses.

Also see the sequence for “Reading out parameters of the encoder manufacturer”

EnDat 2.2 memory access should only be used if the memory access must be done in closed loop access.

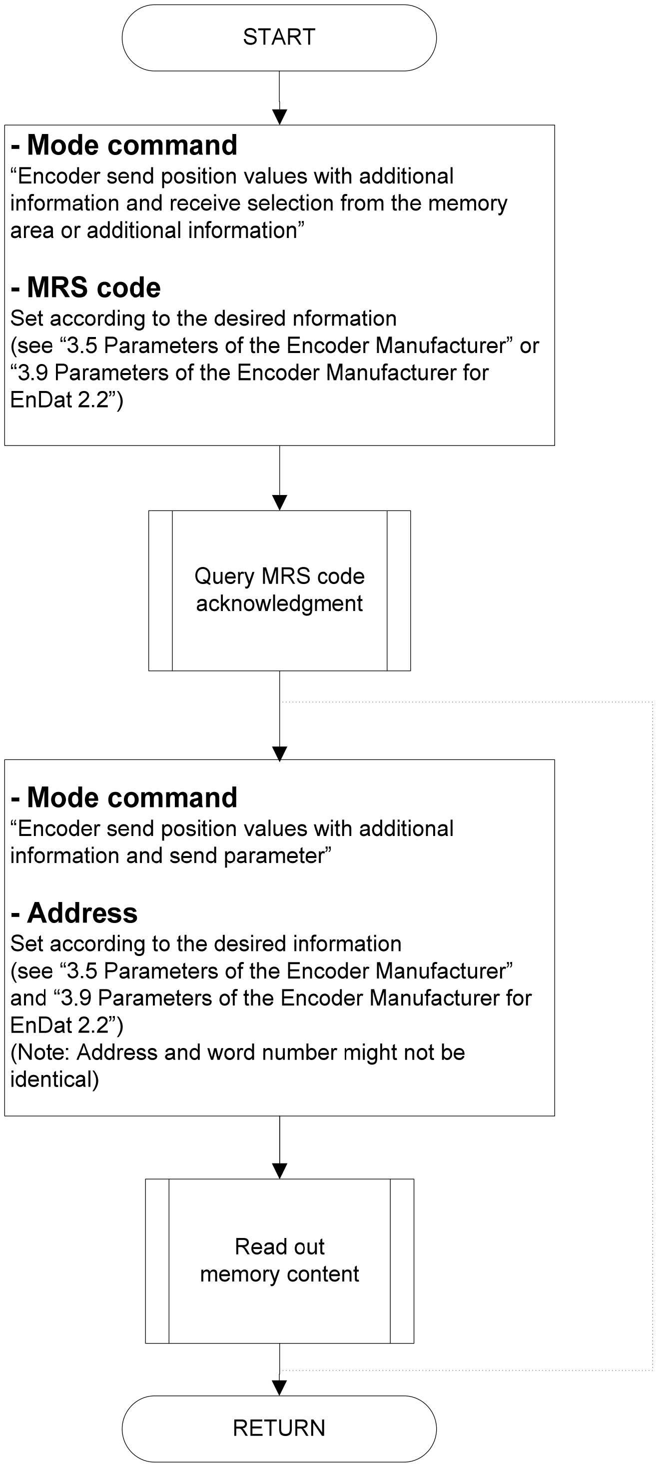


HEIDENHAIN
DR. JOHANNES HEIDENHAIN GmbH
83301 Traunreut, Germany

Memory access by EnDat 2.2: Reading out "Parameters of the encoder manufacturer"

Parameters of the encoder manufacturer for EnDat 2.2 can only be read out with EnDat 2.2 commands.

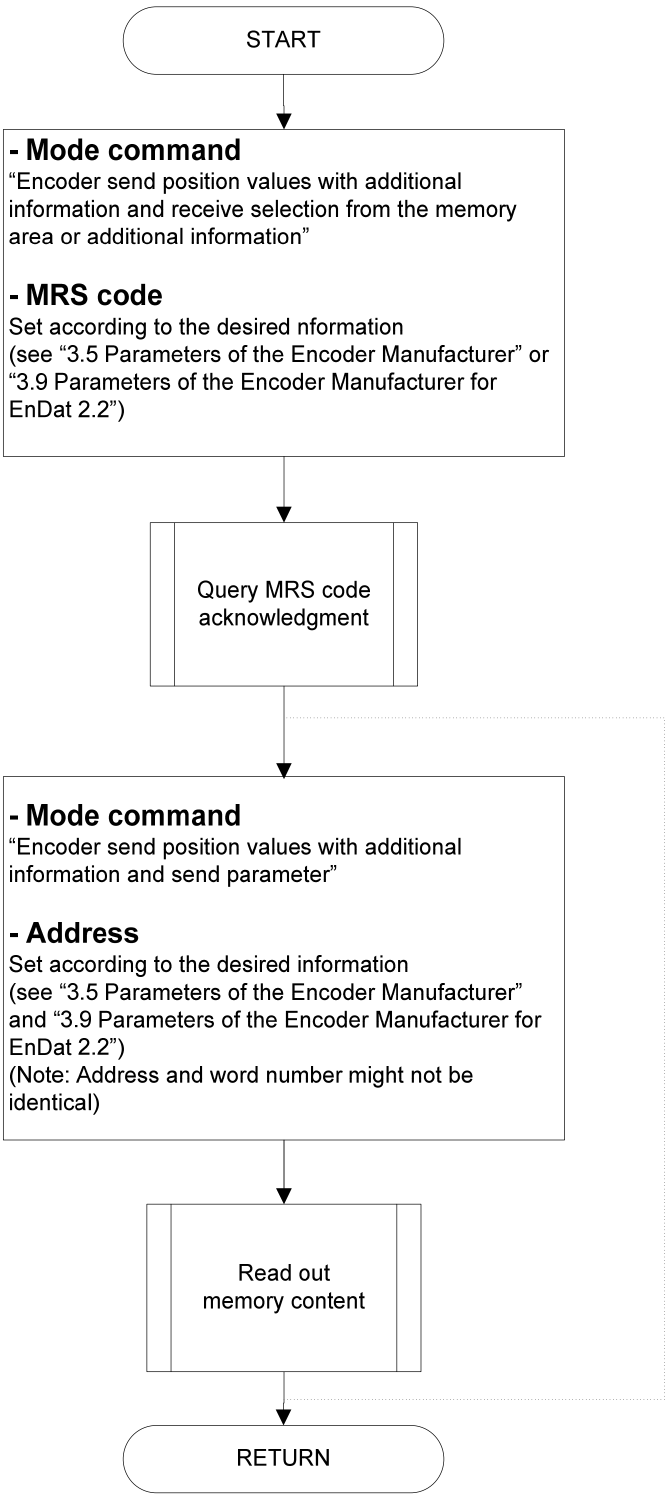
EnDat 2.2 memory access should only be used if the memory access must be done in closed loop access. For a cyclic request the address must be sent again.



Memory access by EnDat 2.2: Reading out “Parameters of the encoder manufacturer”

Parameters of the encoder manufacturer for EnDat 2.2 can only be read out with EnDat 2.2 commands.

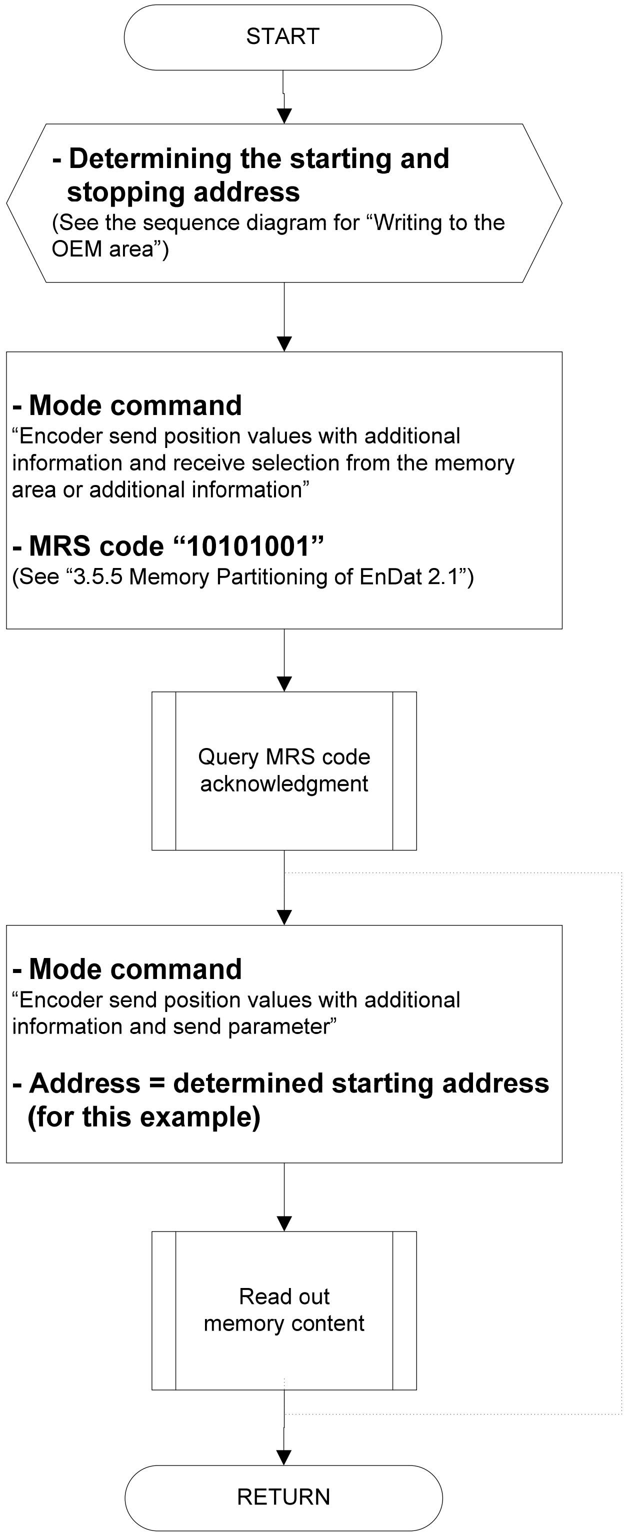
EnDat 2.2 memory access should only be used if the memory access must be done in closed loop access. For a cyclic request the address must be sent again.



Memory access by EnDat 2.2:
Read out OEM area

As an example for the first address (starting address) of OEM area 1.

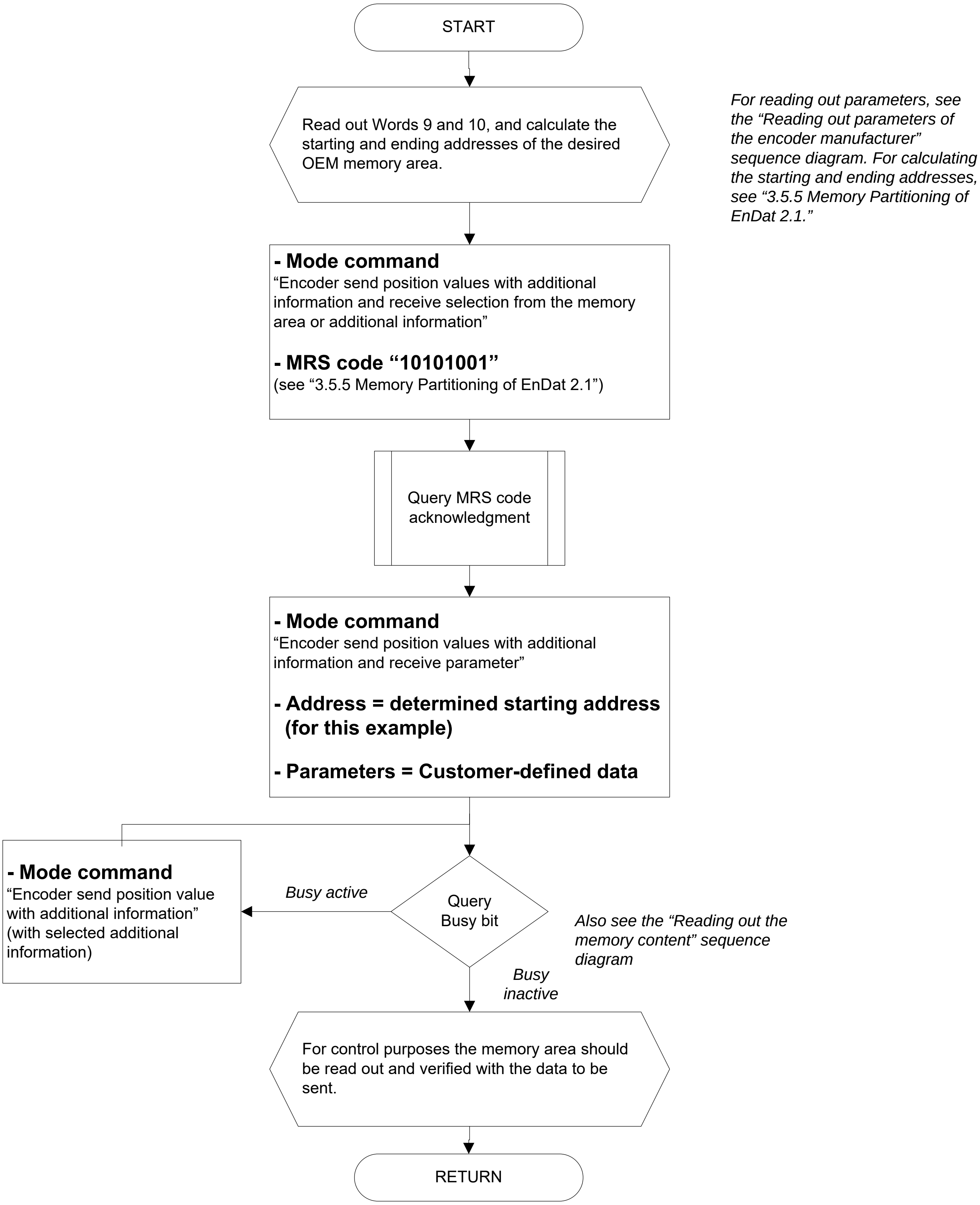
EnDat 2.2 memory access should only be used if the memory access must be done in closed loop access. For a cyclic request the address must be sent again.



Memory access by EnDat 2.2: Writing to the OEM area

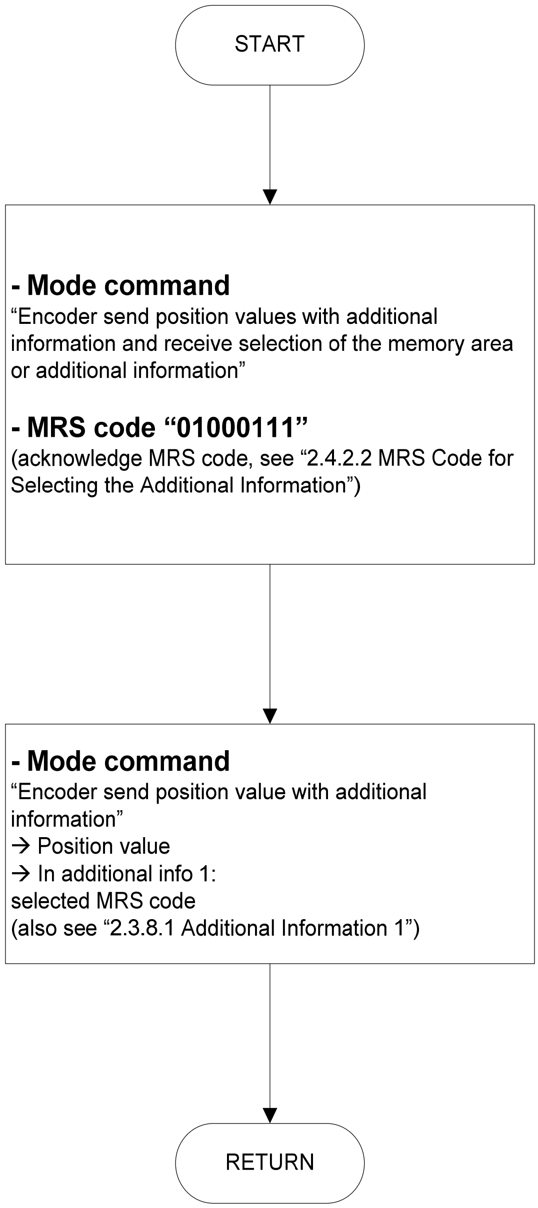
As an example for the first address (starting address) of OEM area 1.

EnDat 2.2 memory access should only be used if the memory access must be done in closed loop access.



Memory access by EnDat 2.2: Request acknowledgment of MRS code

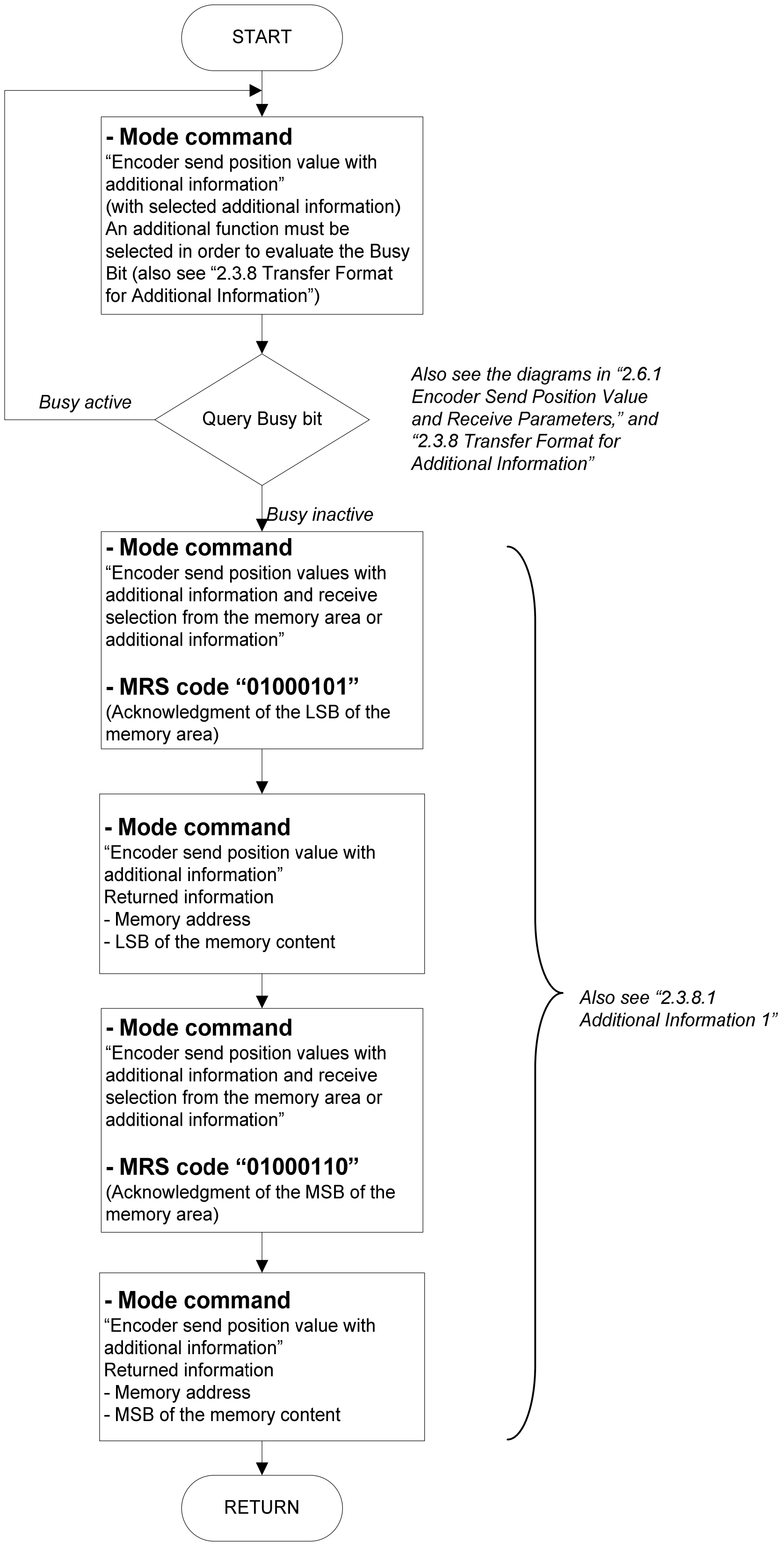
This sequence is a "subprogram" for calling the other sequences.



Memory access by EnDat 2.2: Reading out the memory content

This sequence is a "subprogram" for calling the other sequences. Due to the time for accessing the EEPROM, which can take up to 12 ms, the following sequence is necessary for querying the memory content in the running control cycle.

As an alternative, here the query is possible with the mode command "Encoder send parameter," if processing does not occur in the closed control loop. Maximum access time of the EEPROM = 12 ms.



3 Calculation of Readout Time and Cycle Time

The calculation is based on the EnDat master basic. It is assumed that position-value formation is triggered by hardware strobe via the input “/STR”. As this signal is processed asynchronously, the resulting delay time is very short (approx. 12 ns) and is not taken into account in the calculation described here. The time from the end of the transmission of the last CRC bit until the corresponding flag ("Rcv-Reg1") is set in the status register is also not taken into account.
If you have implemented the EnDat master yourself, you might have to adapt the calculation.

Readout time:
The time it takes from a position request until the position is available in the control is referred to as “readout time” in the following. This is the time period from sending the falling edge on the clock line, which triggers the saving of the position value in the encoder, to the time at which the position, including the CRC, is available in the subsequent electronics.
The readout time is influenced significantly by the EnDat clock frequency f_{CLK} used.

$t_{Readout} = t_{CAL} + t_{TD1} + t_{Cable} + t_{Position}$

t_{CAL}	Calculation time of the encoder. Time t_{CAL} is the smallest time duration after which the position value can be read by the encoder, and defines the time duration from the first falling edge until the earliest possible time at which the start bit can be output by the encoder (see remark below). For clock frequencies < 8 MHz, the value $t_{Sync,CAL} = 4*(1/f_{CLK} – 1/8\ MHz)$ must be added to t_{CAL} for synchronization to the start bit.
t_{TD1}	Delay time of the transceivers in the control (transceiver delay, the duration of the two transceivers for clock and data must be taken into account once per transceiver). Approx. 50 ns per transceiver ➔ typ. 0.1 µs
t_{Cable}	Cable propagation time approx. (6 .. 10) ns/m; the worst-case figure of 10 ns/m should be taken into account in the calculation.
$t_{Position}$	Number of bits for position transfer, depending on the encoder. Eight additional clock pulses (start bit + 2 error bits + 5-bit CRC) have to be taken into account.

Caution:
Due to the many combinations of EnDat mode commands and clock frequencies that customers can set, the formula above only applies to the typical application stated below for encoders with ordering designation EnDat22:

- EnDat 2.2 mode commands and clock frequencies in the range 4 .. 16 MHz

Other applications may require slight deviations from the formula above. For the applications, please refer to your contact person at HEIDENHAIN for more information.

Remark:
At lower clock frequencies the most important fact is that in many cases at least 14.5 clock pulses are necessary until the start bit is transmitted, and depending on the clock frequency selected this can be longer than the time t_{CAL}. It must therefore be checked which time is longer: t_{CAL} or the time 14.5 * 1 / (f_{CLK}). The greater of the two values is then definitive for the time t_{CAL} in the formula above. Example:
Indicated for the encoder: t_{CAL} = 7 µs
Clock frequency: 1 MHz

⇨ The time 14.5*(1/1 MHz) = 14.5 µs is greater, and is therefore to be used as t_{CAL} in the formula above.

Calculation time t_{CAL} for encoders with ordering designation EnDat22:

Constraints for the specification of t_{CAL}:

- Clock frequency f_{CLK} = 8 MHz
- Cable propagation times are not taken into account
- Driver delays of the encoder are taken into account
- Recovery time III: t_{ST} = 2 µs
- Synchronization to start bit $t_{Sync,CAL}$ = 3 * (1/ 8 MHz) is taken into account

Cycle time:
Besides the readout time, the attainable cycle time for cyclic requests to the encoder by the subsequent electronics is important. The attainable cycle time t_{Cycle} for EnDat encoders depends on the following factors:

$t_{Cycle} = t_{Readout} + t_{AI} + t_M + t_R + t_{ASB} + t_{ST} + t_{TD} + t_{Cable}$

<table>
<tr><td>$t_{Readout}$</td><td colspan="2">For readout time, see above.</td></tr>
<tr><td>t_{AI}</td><td colspan="2">Time t_{AI} depends on how many pieces of additional data need to be transmitted. They are selected by the subsequent electronics. Up to two pieces of additional data are possible. The transmission of each piece of additional data requires thirty clock pulses.</td></tr>
<tr><td>t_M</td><td colspan="2">Adjustment of the recovery time I (1.25 to 3.75 µs or 10 to 30 µs)
The respective worst-case figure of 3.75 µs or 30 µs must be taken into account</td></tr>
<tr><td>t_R</td><td colspan="2">Recovery time II; waiting time until a new position request can be sent: 5 µs</td></tr>
<tr><td rowspan="5">t_{ASB}</td><td colspan="2">When using EnDat 2.2 mode commands with transmission supplement (Additional Send Block)
Applies to all EnDat 2.2 mode commands numbers 9 to 13: “Encoder send position values with additional data and ….”
These mode commands are needed, for example, when you want to switch between different pieces of additional data in the control cycle.</td></tr>
<tr><td>t_{DATA}</td><td>This transmission supplement requires a total of 32 clock cycles:
Start bit, 8-bit data, 16-bit address, 8 bits for switching processes</td></tr>
<tr><td>t_{ST}</td><td>Recovery time III; recommendation: 2 µs; minimum: $t_{ST} > t_{Cable}$ (for more information, see EnDat Specifications, Chapter 2.3.2)</td></tr>
<tr><td>t_{TD2} ①</td><td>Transceiver delay. The time must be taken into account twice as the transceivers will run on both control and encoder sides. Approx. 50 ns per transceiver ➔ typically 0.2 µs</td></tr>
<tr><td>t_{Cable} ①</td><td>Cable propagation time
approx. (6 .. 10) ns/m; the worst-case figure of 10 ns/m should be taken into account in the calculation.</td></tr>
</table>

① Also see the “Readout time” section.

The time t_D given in the EnDat Specifications corresponds to $t_{TD2} + t_{cable}$.

Please see the EnDat specifications for the corresponding times and the timing of EnDat mode commands.

Remark:
The times specified for readout time and cycle time are approximate values. The exact values depend on the application (EnDat master, transceiver, cable, …) and must be verified.

Example:
Configuration for the following examples:

- Encoder: LC 415 (36-bit position data)
- Calculation time t_{CAL} = 5 µs @ 8 MHz
- Clock frequency: 8 MHz
- 30 m cable length
- t_{ST} = 2 µs
- t_{TD1} = 2 * 50ns = 0.1 µs
- t_{TD2} = 4 * 50ns = 0.2 µs
- t_M = 1.25..3.75 µs; see EnDat specifications. The worst-case figure of 3.75 µs must be used.
- t_R = 0.5 µs; see EnDat specifications.

Application examples:

a) The minimum possible cycle time should be attained.
   - Only the position value is read out.
b) Encoder with activated additional data (e.g. temperature 1)
   - Position value and additional data (set once) are read out.
c) Encoder with Functional Safety (reading out the position 2)
   - Transfer of position value and one piece of additional data
   - EnDat 2.2 mode commands with transmission supplement are used
d) Maximum possible protocol length
   - Transfer of position value and two pieces of additional data
   - EnDat 2.2 mode commands with transmission supplement are used

Application a): Position-value transfer without additional data, without transmission supplement

Readout time:

t_{CAL}	5 µs + 4 * (1/8 MHz – 1/8 MHz) = 5 µs	5 µs
t_{TD1}	0.1 µs	0.1 µs
t_{Cable}	2 * 30 m * 10 ns/m	0.6 µs
$t_{Position}$	(36 + 8) * 0.125 µs	5.5 µs
$t_{Readout}$		11.2 µs

Cycle time:

$t_{Readout}$			11.2 µs
t_{AI}	-		-
t_M	3.75 µs		3.75 µs
t_R	0.5 µs		0.5 µs
t_{ASB}	t_{DATA}	-	-
	t_{ST}	-	-
	t_{TD2}	-	-
	t_{Cable}	-	-
t_{Cycle}			15.45 µs

Application b): Position-value transfer with one piece of additional data, no transmission supplement

$t_{Readout}$			11.2 µs
t_{AI}	30 * 0.125 µs		3.75 µs
t_M	3.75 µs		3.75 µs
t_R	0.5 µs		0.5 µs
t_{ASB}	t_{DATA}	-	
	t_{ST}	-	-
	t_{TD2}	-	-
	t_{Cable}	-	-
t_{Cycle}			19.2 µs

Application c): Position-value transfer with one piece of additional data and transmission supplement

$t_{Readout}$			11.2 µs
t_{AI}	30 * 0.125 µs		3.75 µs
t_M	3.75 µs		3.75 µs
t_R	0.5 µs		0.5 µs
t_{ASB}	t_{DATA}	32 * 0.125 µs	4 µs
	t_{ST}	2 µs	2 µs
	t_{TD2}	0.2 µs	0.2 µs
	t_{Cable}	2 * 30 m * 10 ns/m	0.6 µs
t_{Cycle}			26.0 µs

Application d): Position-value transfer with two pieces of additional data and transmission supplement

$t_{Readout}$	11.425 µs		11.2 µs
t_{AI}	2 * 30 * 0.125 µs		7.5 µs
t_M	3.75 µs		3.75 µs
t_R	0.5 µs		0.5 µs
t_{ASB}	t_{DATA}	32 * 0.125 µs	4 µs
	t_{ST}	2 µs	2 µs
	t_{TD2}	0.2 µs	0.2 µs
	t_{Cable}	2 * 30 m * 10 ns/m	0.6 µs
t_{Cycle}			29.75 µs

4 Position-Data Formats

To convert the position value transferred via the EnDat interface, further parameters as well as the encoder model must be considered:

Encoder model E21W14	Content E21W14	Measuring step or measuring steps per revolution E21W20,21	Scaling factor + measuring steps… E22W27,28,29	Signal period or signal periods per revolution E21W15,16	Number of distinguishable revolutions E21W17
Incremental Linear Encoder	0000 0001 0010 0011	Not equal to 0→ Formula 1	Not filled	Not relevant	-
		"0" → E22W27-29	Measuring steps in pm → Formula 2	Not relevant	-
			Subdivision values → Formula 3 Second power → Formula 4	Must be specified	-
Absolute linear encoder	0100 0110	Not equal to 0→ Formula 5	-	Not relevant	-
Incremental rotary encoder or angle encoder	1000 1001 1010 1011	Not equal to 0→ Formula 6	-	Not relevant	-
		"0" → E22W27-29	Subdivision values → Formula 9 Power of 2 → Formula 10	Must be specified	-
Singleturn encoder	1100	Not equal to 0→ Formula 7	-	Not relevant	-
Multiturn encoder	1110 1101	Not equal to 0→ Formula 8	-	Not relevant	Relevant, when "0xFFFF" then observe E22W34
EIB	1111	Is not considered.			

Explanations:
E21W14: Parameters of encoder manufacturer, word 14
E22W27,28,29: Parameters of encoder manufacturer for EnDat 2.2, words 27 to 29
…

Formula 1:
Relative position [m] = transferred position value * content of E21W20,21 * 1E-9

Formula 2:
Relative position [m] = transferred position value * content of E22W28,29 * 1E-12

Formula 3:
Relative position [m] = transferred position value * (E21W15,16 / E22W28,29) * 1E-9

Formula 4:
Relative position [m] = transferred position value * (E21W15,16 / 2^E22W28,29) * 1E-9

Relative position:
As explained in the EnDat specification and Application Note, the relative position value is transferred. The absolute position value can be calculated with the same formula (after crossing the reference mark) from the "position value 2" in additional data 1. See the EnDat specification or Application Note for further information and concerning the structure of position value 2.

Formula 5:
Absolute position [m] = transferred position value * content of E21W20,21 * 1E-9

Formula 6:
Relative angle value [°] = 360° * (transferred position value / content of E21W20,21)
Formula 7:
Absolute angle value [°] = 360° * (transferred position value / content of E21W20,21)

Formula 8:
The structure of the position value is according to the EnDat specification and the “Transfer format for position values” chapter.
The multiturn part must initially be extracted from the transferred position value. E21W17 or E22W34 is used for this. In practice only values corresponding to 2^x are used. Thus 2^M must be determined according to the depiction in the EnDat specification. M can be determined from parameter E21W17 or E22W34.

Multiturn part of the position:	Depiction of 2^(S+M) to 2^(S+1); Truncation of the singleturn part from the transferred position value and “push to right”
Singleturn part of the position:	Depiction of 2^S to 2^1 “Zeroing” of the higher-value bits

Absolute angle value [°] = 360° * (content of E21W20,21 / singleturn part of the position)

Formula 9:
Relative angle value [°] = 360° * [transferable position value / (E21W15,16 * E22W28,29)]

Formula 10:
Relative angle value [°] = 360° * [transferable position value / (E21W15,16 * 2^E22W28,29)]

The maximum bit width for the position transfer is 48 bits. The further calculation of the position must be suitable for this data width.

5 EnDat01 Encoders: Absolute and Incremental Position Values

Example: ECN 413

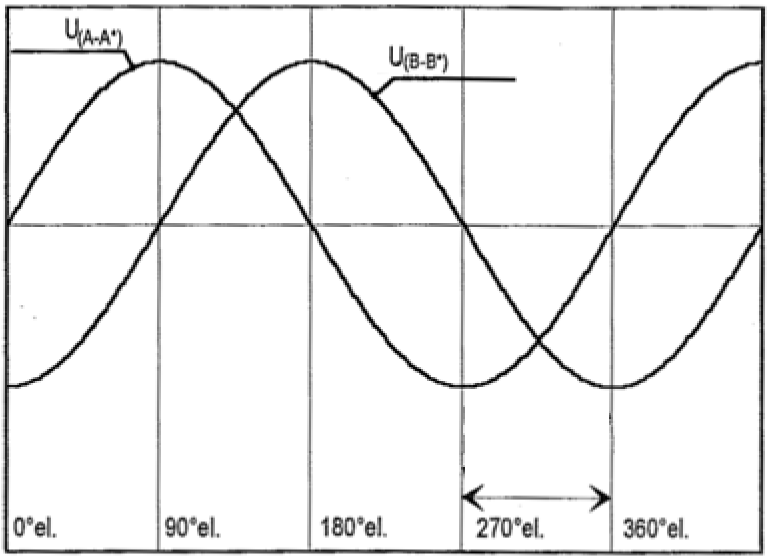
- 2048 signal periods ➔ 11 bits of the position value come from the signal period
- 2 bits come from the interpolation
- ⇨ Bits 1 and 2 → Interpolation
- ⇨ Bits 3 to 13 → Signal periods (absolute position of the signal period with respect to 360°)
- ⇨ Overall: 13 bits make up the singleturn position

ECN 413: 13-bit singleturn position

Binary data

Singleturn [bit]												
13	12	11	10	9	8	7	6	5	4	3	2	1
■	■		■			■			■			■

The two interpolation bits (bits 1 and 2) indicate in which "quadrant" the incremental signals are located. "Quadrant" means the current 90° range of the incremental A and B signals. The following figure shows the location of the quadrants: 0° to 90°, 90° to 180°, 180° to 270° and 270° to 360° with respect to the A and B signals. LSB → bit 1 and LSB-1 → bit 2.



A / B	10	11	01	00	10	
LSB (bit 1)	1	0	1	0	1	8192 (2^{13})
LSB -1 (bit 2)	0	1	1	0	0	4096 (2^{12})
LSB -2 (bit 3)	1	1	1	0	0	2048 (2^{11})

Note:
Please carefully take into account the phase shift of the input amplifier. Depending on the input circuitry, a 180° phase shift (inverting) of the input signals might occur.

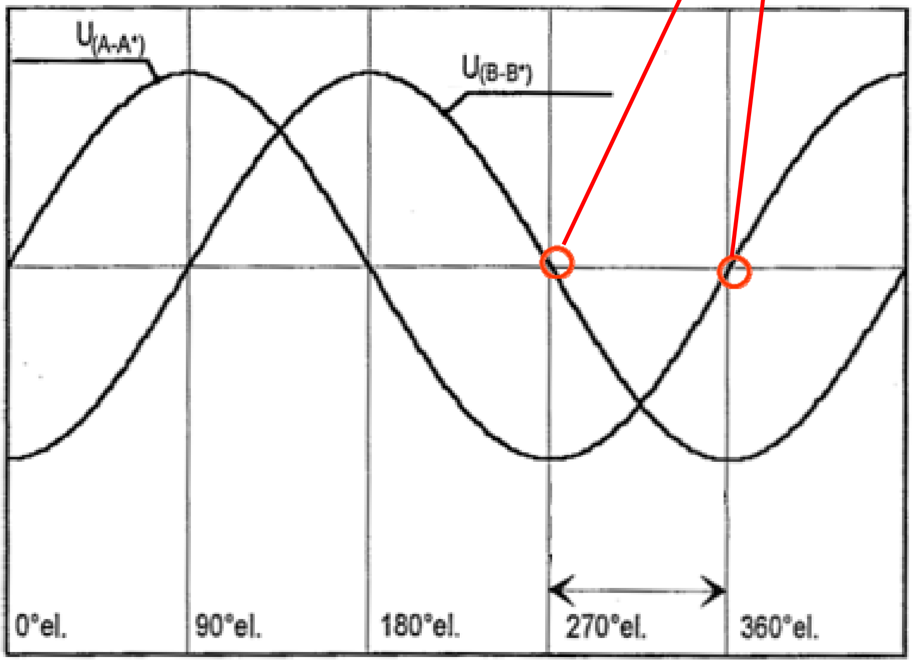
Comment:
If an encoder with 512 lines is used:

- 9 bit of the position come from the signal period and 4 bit come from interpolation

Tolerances of absolute and incremental quadrant information:

Due to the tolerance of the incremental signals (± 15° el.) at the "zero crossovers," the unambiguous quadrant assignment cannot be made with just the absolute position word. If the A and the B signal are evaluated, however, it can be reviewed whether

- the determined incremental signal quadrant is within the allowable tolerance range with respect to the transmitted absolute quadrant.
- the interpolated value can be directly connected to the transmitted absolute position or whether this position (see red marked fields) must first be corrected.



A / B	10	11	01	00	10	
LSB	1	0	1	0	1	8192 (2^{13})
LSB -1	0	1	1	0	0	4096 (2^{12})
LSB -2	1	1	1	0	0	2048 (2^{11})

Note:
Please also take into account the fact that the phase shift between the absolute and incremental position information will increase as the speed increases. This is due to the fact that the data ages between the absolute and incremental signal paths are different. Therefore the encoder indicates in EnDat 2.1 words 33 and 34 the accuracy over speed.

Singleturn position information:
EnDat01 encoders deliver a position value whose resolution is typically not sufficient enough for the controller. To increase the resolution the controller interpolates the A and B signals to get an overall position of higher resolution. Therefore, typically at power-up when the encoder is at a standstill, both values (absolute and incremental) are captured at the same time.

Absolute position:

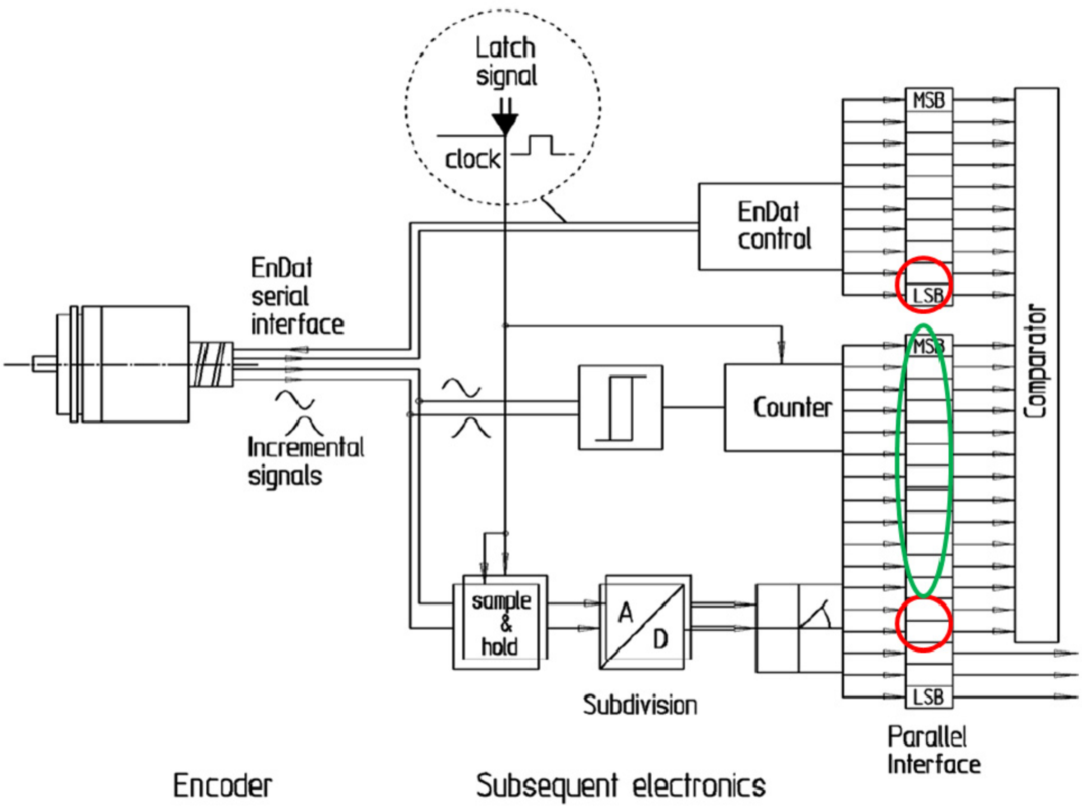
Upper 11 bits: absolute position of the signal period
Lower 2 bits: quadrant information

Incremental position:
From the amplitude values of the A and B signals the controller calculates the current angle within the signal period, i.e. with 10-bit resolution ➔ 10-bit interpolation

Association of absolute and incremental position values:
To get a final value for the position, the absolute and incremental position values need to be associated. Therefore, the shift of the absolute and incremental values with respect to the EnDat "0 position" must be considered. To check for that the absolute position has two interpolation bits to indicate in which quadrant the position is with respect to the incremental signals. So after adapting the phase shift to the "0 position," the lower two bits of the absolute position and the upper two bits of interpolation value must be identical. Please also take the tolerance of the absolute and incremental quadrant information into account (see above).

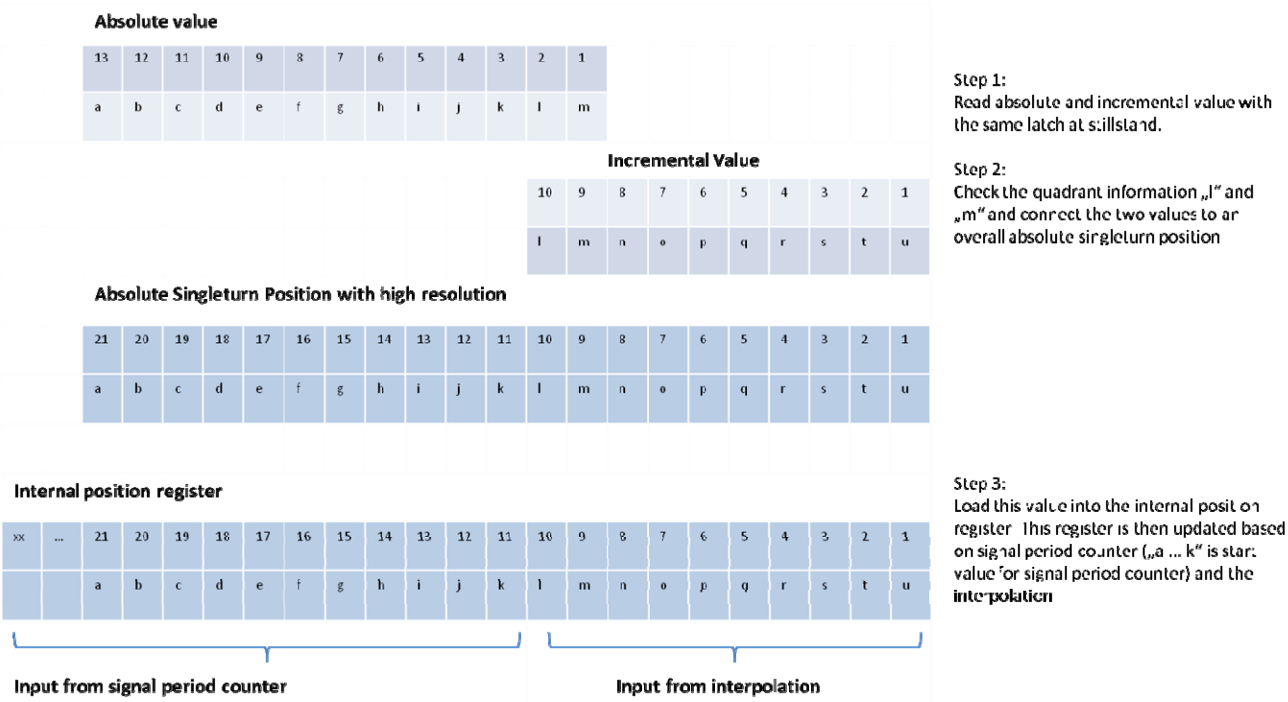
Position register:
Now the position register can be loaded with the position value that was built out of the absolute and incremental values. As the control now typically operates based on the incremental signals, this position register is now updated based on the interpolation values. Since more than one signal period can be crossed between two control cycles, it is necessary to add a counter for the signal periods.

The following figure is taken from the EnDat specification. It shows the basic principle and the corresponding bits which need to be compared → "comparator".
The red circled values are the bits for comparing the absolute and incremental values. The green ellipse shows the signal period counter.



Overview:

The following figure gives an overview of how to combine absolute and incremental position values to an overall internal position register. After the first initialization, the register is updated based on input from the signal period counter and interpolation.



6 EnDat Parameter Overview

See EnDat Technical Information

7 EnDat Monitoring Functions

General Information:
The EnDat interface offers numerous monitoring functions, which are in fact performed by the parent control.

Monitoring or correct encoder operation:

	Interrogation	**Reaction by the encoder**	**Why?**
Group bit error	Transmitted with every position transfer	Safe system condition	Position value no longer reliable
Errors register	Operating status word 0	See above	Detailed information on error source for service and diagnosis
Operating status error sources	Additional datum 2	Optical and inductive devices: Singleturn and multiturn errors are distinguished ➔ Initiation of braking process can be handled flexibly Battery-buffered devices: ➔ See Encoder Specifications for the reaction	Detailed information on the source of the position value error and the exact error sources on battery-buffered devices
Group bit warning	Transmitted with every additional information	No mandatory reaction of the control required	A warning shows that the functional limit has been reached but the device is still supplying valid position values
Warnings register	Operating status word 1	See above	Detailed information on error source for service and diagnosis
Diagnostics	Additional information 1	The functional reserves of the encoder can be displayed	Preventive maintenance Planning of service jobs
Recovery time I or t_M (recovery time)	Used by the encoder to acknowledge a transmission as being valid at the end of this transmission	If recovery time I does not elapse correctly, there may have been problems with the last transmission ➔ Reaction: Set clock line to “high” and wait 30 µs (maximum value of recovery time I)	When recovery time I ends, this always leads to an internal reset of the encoder
Calculation time t_{CAL} and start bit during position value transmissions	See EnDat transmission protocol	A continuous pulse is to be sent until the start bit is output. When the calculation time has elapsed or after 15 pulses at the earliest, the encoder sends the start bit. If the start bit is not sent, this means that transmission of a position value is not possible, either ➔ Clock line to “high” and expiry of max. recovery time I	If the start bit is not sent, this means that no position value can be made available for the transmission that has been initiated; i.e. the encoder must be set to a defined start condition.

Regulation of access to the EEPROM:

	Interrogation	Reaction by the encoder	Why?
Busy bit	Transmitted with every additional information	Waiting until another access to the EEPROM is possible. If the maximum time of 12 ms is exceeded, this indicates a malfunction of the EEPROM ➔ Reaction: Execute Power Off/On; if it occurs repeatedly the encoder must be exchanged	Synchronization of “slow” EEPROM accesses with current position queries

Checking the transmission status:

	Interrogation	Reaction by the encoder	Why?
CRC	Transmitted by the encoder and tested by the control	The data received, such as position and additional information, are impermissible and should not be used for control	The CRC is intended for securing the transfer
Error type I	The encoder sets the data line to “high-impedance”. After pull-up or pull-down on the transceiver, the level is steady.	The clock must be set to “high” and you have to wait for time t_M.	Internal reset of the encoder required
Error Type II	The acknowledgment is returned inverted	Write or read instruction invalid → Repeat	
Error type III	Value 15 or 31 is returned	No mandatory reaction required, but the read/write instruction is invalid	

Please note:
When evaluating an EnDat transmission, you should always check all statuses and derive appropriate measures from them. The CRC, in particular, is a central element of data backup. The CRC of every data transfer should therefore be checked.

Error message(s) set, CRC OK and no EnDat error type:
- There is a malfunction in the encoder

CRC check is erroneous:
- The transmission path is disturbed
- Also occurs with an EnDat error type I
- Error messages can occur

EnDat error type I occurs:
- See the EnDat specification for causes, e.g. mode commands are incorrectly transmitted
- EnDat error messages can occur
- The CRC check is erroneous

EnDat error types II and III occur:
- For causes, see the EnDat Specification
- The selected addresses or MRS codes should be checked

Measures to be taken for safety-relevant applications according to IEC 61508 and EN 13849 are described in the EnDat package of measures (D533095).

Remark:
Depending on the version of the EnDat master, the individual monitoring measures are detected automatically, i.e. the detection has to be carried out via the higher-level application software.

EnDat error messages and warnings

Various error messages and warnings are saved in the "Operating status" memory area of the EnDat encoder. The individual bits of the corresponding EnDat words have a specific meaning and it is recommended that all of these error and warning bits be evaluated and appropriate corrective action be taken. Different bits are supported depending on the encoder. Please also read the EnDat Specification and the EnDat Application Notes. Please also note the documentation for the encoder and the subsequent electronics.

Meaning of error messages and warnings:

- Error message: Position value is not reliable.
- Warning: An internal functional limit of the encoder has been reached

Operating status error sources
The operating status error sources provide a detailed breakdown of the error messages.
The EnDat 2.2 parameter "Support of operating status error sources" includes information about whether and which error messages are supported by the respective encoder. The operating status error sources are output via the additional datum 1. This means that fast access within the control loop is possible via the operating status error sources.

The following error sources are identical to the corresponding error messages in the table (see below) regarding their description and remedy:

- Lighting
- Signal amplitude
- Overvoltage
- Undervoltage
- Overcurrent

The following error sources are identical to the '"Position" item regarding their description and remedy, but also provide additional information on whether the error source refers to position value 1 or position value 2 or to single or multiturn ("S" or "M").

- S Pos1
- S Pos2
- S System
- M Pos1
- M Pos2
- Temperature exceeded

The following error sources refer to battery-buffered devices. For more information, refer to the EnDat Application Notes, "Encoders with battery-buffered revolution counter" chapter:

- M System
- M Power interruption
- M Overflow/Underflow
- M Battery
- S Power interruption

Bit #	Error	Description	Remedy
0	Lighting	An error occurred during the initialization of the encoder or internally in connection with the light source	Restart the encoder (Power off/on). Check the encoder installation. Clean the encoder (if possible)
1	Signal amplitude	The signal amplitudes of the incremental or absolute track have exceeded or fallen below the specified limits.	Check the installation of the encoder Clean the encoder (if possible) Check whether any of the encoder specifications has been exceeded, e.g. the max. traversing speed. [1]
2	Position	An error occurred during the initialization of the encoder or the calculation of the position value.	Restart the encoder (Power off/on). Check the encoder installation. Clean the encoder (if possible) Check whether any of the encoder specifications has been exceeded, e.g. the supply voltage or ambient temperature. [1]
3	Overvoltage	The specified limit for the supply voltage of the encoder has been exceeded.	Check whether any of the encoder specifications has been exceeded, especially the supply voltage. 1) Check the cable
4	Undervoltage	The supply voltage of the encoder has fallen below the specified limit.	Restart the encoder (Power off / on) Check whether any of the encoder specifications has been exceeded, especially the supply voltage. [1] Check the cable
5	Overcurrent	The specified limit for the encoder's current consumption has been exceeded.	Restart the encoder (Power off / on) Check whether any of the encoder specifications has been exceeded, especially the supply voltage. [1] Check the cable
6	Battery	The voltage level of the external battery has fallen below the specified value. This can lead to faulty multiturn information.	Check the voltage level of the battery and change the battery, if required [1] Check the cable

[1] For more information, please refer to the encoder documentation (Mounting Instructions, brochure, Product Information, etc.).

Bit #	Warning	Description	Remedy
0	Frequency collision	An internal warning threshold with regard to the position calculation has been reached.	Check whether any of the encoder specifications has been exceeded, e.g. traversing speed or vibration. 1)
1	Temperature exceeded	The warning threshold for the encoder temperature (according to the operating parameter, word 6) has been reached.	Application-specific reaction is required; usually, the temperature load on the encoder must be reduced.
2	Light-source control reserve	An internal warning threshold with regard to the light source control has been reached.	Clean the encoder (if possible)
3	Battery charge	The voltage level of the external battery has fallen below the threshold.	Check the voltage level of the battery and replace the battery, if necessary, while the power supply of the subsequent electronics is active.
4	Reference point	Reference mark(s) has (have) not been traversed yet (only for incremental encoders)	Traverse the reference mark(s)
5	Cyclic mode	The encoder is in acyclic mode, Closed-Loop operation is not possible.	Restart the encoder (Power off / on)
6	Limit position	A limit position has been reached.	Depending on the encoder and the application, an adequate reaction must be taken, e.g. stopping the drive.
7	Standby	The touch probe is not ready yet.	Check the availability of the touch probe again
8	Diagnostics	A valuation number of the online diagnostics was below the threshold at which servicing of the encoder is recommended.	Check the installation of the encoder Clean the encoder (if possible)

1) For more information, please refer to the encoder documentation (Mounting Instructions, brochure, Product Information, etc.).

8 Online Diagnostics

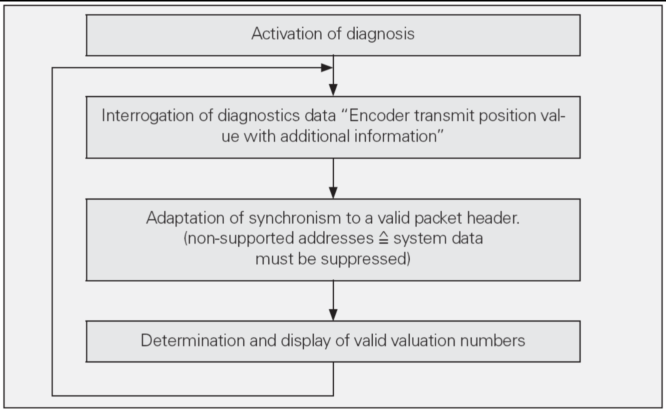
General information:

The EnDat interface makes extensive monitoring and diagnosis of an encoder possible without an additional line. In addition to error and warning messages, online diagnostics in particular enable a high availability of the complete system. Online diagnostics are growing in significance. Decisive points of emphasis are:

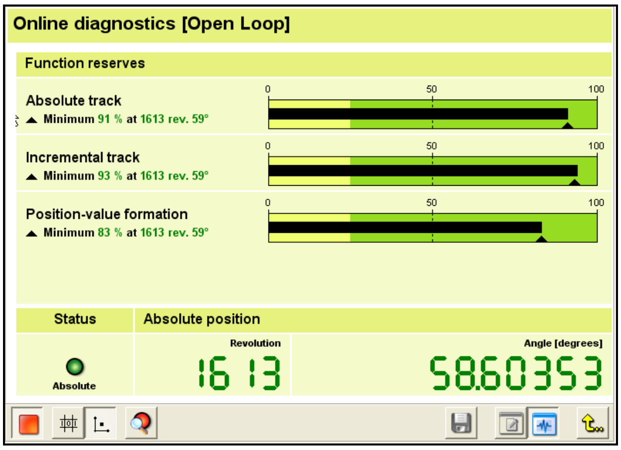
- Machine usage planning
- Support for the service technician on-site
- Simple evaluation of encoder function reserves
- Simplification of trouble-shooting for repair
- Generation of meaningful quality statistics

On encoders with incremental signals, it is possible to use Lissajous figures to analyze the encoder function. Encoders with purely serial interfaces do not provide incremental signals. Encoders with EnDat 2.2 can cyclically output the valuation numbers in order to evaluate the functions of the encoder. The valuation numbers provide the current state of the encoder and indicate the encoder's "function reserves." Their scaling is identical for all HEIDENHAIN encoders. This makes integrated evaluation possible. The valuation numbers supported by the respective encoders are saved in the EnDat 2.2 parameters.

Fundamental process for requesting an online diagnosis and displaying the function reserve:



Flow chart for interrogation of diagnostics data



Transmission of the valuation numbers:
The valuation numbers are transmitted with additional data 1 "(Send) diagnostic values". It permits access to the data in the closed control loop. The data transmitted consists of an 8-bit address and 8 bits of data. The address indicates the valuation number in question and controls the synchronization of the data packages. The 8 bits of data (0 ... 255) contain the actual valuation number.

Valuation number:
Depending on the encoder model, one or more valuation numbers (German abbreviation: BWZ) are supported. The distinction is made based on the address transmitted. The addresses are assigned the following meanings; designations in parentheses refer to incremental encoders:

- Address 1: Evaluation of the incremental or scanning track
- Address 2: Free for future applications
- Address 3: Evaluation of the absolute track / (reference-pulse width)
- Address 4: Evaluation of the position-value formation / (reference-pulse position)
- Addresses 5 and 6: Reserved for touch probes
- Addresses 7 and higher are reserved for future applications

By transmitting several valuation numbers for different functional groups in the encoder, the “function reserve” of the encoder can be determined optimally and also, the service technician on site is supported better.

The designations in parentheses for addresses 3 and 4 are valid for incremental encoders (see also the EnDat Specification: Parameters of the encoder manufacturer, word 14).

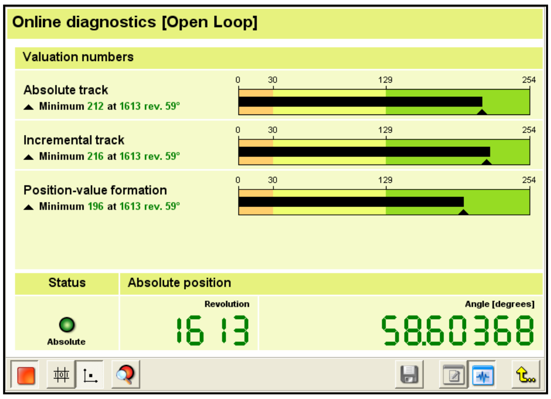
Note on reference pulse width and position:
Because the information on the width and position of the reference mark is not permanently available, the following applies:
- After switching on or reset the measuring instrument, "255" is output
- After crossing the reference mark, the value for position and width thus formed is output until the reference mark is crossed again.

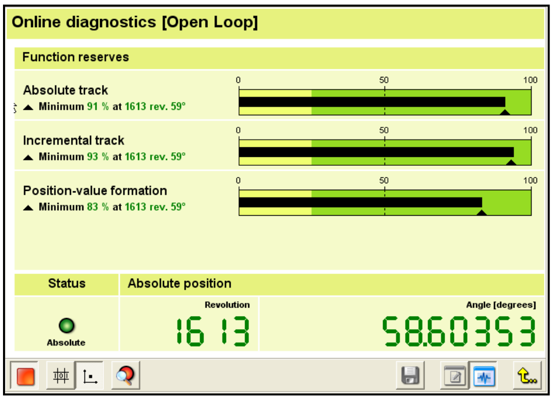
Function reserve:
The OEM is recommended not to display the valuation numbers directly but to convert them into a function reserve. Error messages can occur in the range from 0 … 29, as the tolerance ranges for the diagnostics and the errors can overlap in this range. The range from 231 ... 254 is intended for HEIDENHAIN’s internal purposes and does not increase the function reserve. For this reason, the function reserve should be displayed as follows:

- Valuation number 0 ... 29: Function reserve “0”
- Valuation number 30 ... 230: Function reserve “0 ... 100“
- Valuation number 231 ... 254: Function reserve “100”

This permits the best possible display of the function reserve, as irrelevant information is not shown. The function reserve has the following significance:

- Function reserve 0 ... 25:
  The encoder is working close to the functional limit; servicing is recommended
- Function reserve 26 ... 100:
  The encoder is working distinctly within the functional limits; the value “100” is an ideal state and cannot be guaranteed, even at the time the encoder is shipped.





Configuration of diagnosis and valuation numbers:
The valuation numbers that are supported are specified in the encoder, see word 5 “Parameters of the encoder manufacturer for EnDat 2.2”. The valuation numbers that are transmitted via the interface can be activated individually. They are activated with the operating parameter word 3 “Configuration of diagnosis”. The setting is saved in the EEPROM, which means it must be made only once. It is advisable, however, to always activate all supported valuation numbers also for transmission. Activation of individual valuation numbers can only be recommended for very specific applications. To activate the configuration, either the mode command “Encoder receive reset” must be sent or the device be switched off and on.

Transmission package:
The valuation numbers are always transmitted as a package. A package always consists of a package header and the activated valuation numbers. The data is transmitted in a rolling sequence. To display the valuation numbers, synchronization with a valid package is required first; this means that a valid package header for the online diagnosis has to be awaited first. Valid packages for the online diagnosis have the value $0x00_{hex}$ for the address and $0x3E_{hex}$ for the data (only the lower 6 bits are important!), other contents for the data (and thus the packages) must be ignored. This is necessary in case system-specific packages are transmitted simultaneously with the packages for the online diagnostics. System-specific packages can be activated by the HEIDENHAIN Service department, for example.

Besides these values that are always transmitted, there are further combinations for address and data, showing the following conditions.

- A valuation number can currently not be provided; either because it cannot be determined or because an internal limit for the calculation time was exceeded:
  ➔Cause: Encoder internal
  ➔ Transmission: a) Valid address + data: $0xFF_{hex}$
  b) Address $0xFF_{hex}$ + data: $0x00_{hex}$
  ➔ Error reaction: Ignore data
- No data is available for the valuation numbers
  ➔ Cause: No valuation numbers activated or diagnostics is not supported
  ➔ Transmission: a) Address $0xFF_{hex}$ + data: $0xFF_{hex}$
  b) EnDat error type III
  ➔ Error reaction: Processing of the package canceled and test of whether the valuation numbers were activated.

The distinction between transmission types a) and b) is only important for HEIDENHAIN internal purposes. Both transmission types can be considered alternatives for the evaluation of the valuation numbers.

For the addresses and data that are transmitted via the additional data, the following combinations are possible (shown in binary number format):

Address	Data	Meaning
0000 0000	xx11 1110	Package header; data bits 6 and 7 → HEIDENHAIN internal data
	All; (except xx11 1110)	Package header; system-specific data; ignore package
any. (except 0000 0000 and 111 1111)	xxxx xxxx (except 1111 1111)	Address: Address of valuation number 1 .. 254; Data: Content of valuation number 1 .. 254
	1111 1111	Data for the valuation number could not be determined (ignore content)
1111 1111	0000 0000	Data for the valuation number could not be determined (ignore content)
	1111 111	No data available (alternatively EnDat error type III can occur)

Typical example of the transmission of three valuation numbers, e.g. with a multiturn encoder (binary number format):

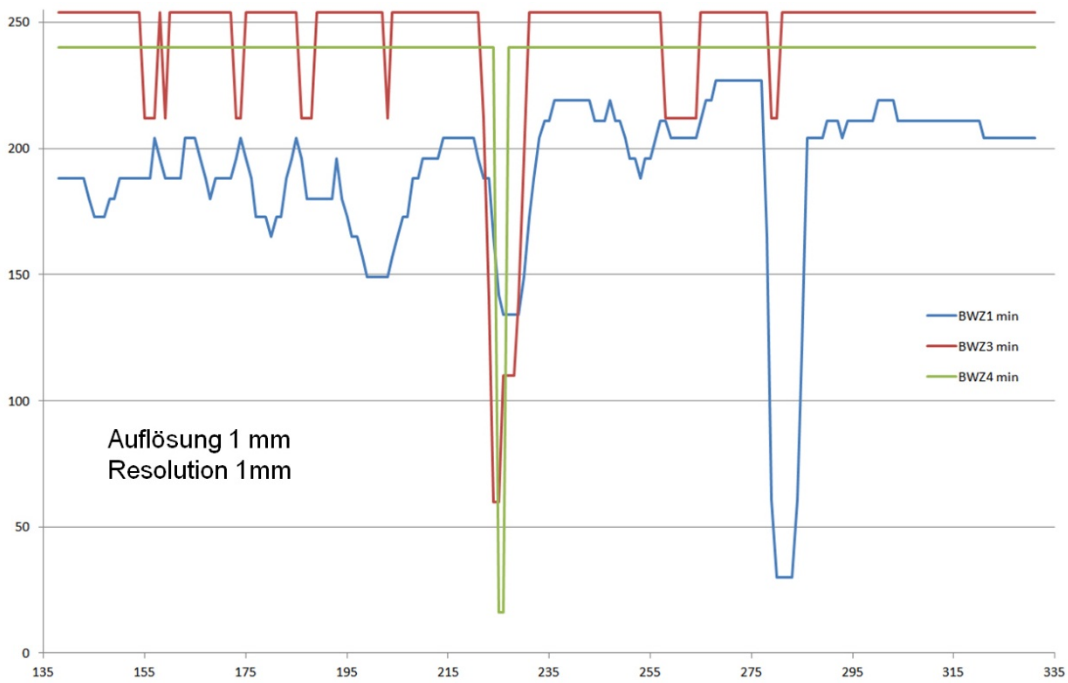
Address	Data	Meaning
0000 0000	1111 1110	Package header (0x3E$_{hex}$)
0000 0001	1010 1010	Valuation number of incremental or scanning track (value: 170)
0000 0011	1110 0000	Valuation number of absolute track (value: 224)
0000 0100	1111 0001	Valuation number of position-value formation (value: 241)
0000 0000	1011 1110	Package header (0x3E$_{hex}$)
0000 0001	1110 1010	Valuation number of incremental or scanning track (value: 234)
0000 0011	1110 1111	Valuation number of absolute track (value: 239)
0000 0100	1010 1001	Valuation number of position-value formation (value: 169)
0000 0000	1111 1110	Package header (0x3E$_{hex}$)
0000 0001	1100 0111	Valuation number of incremental or scanning track (value: 199)
0000 0011	1110 1001	Valuation number of absolute track (value: 233)
0000 0100	0101 0101	Valuation number of position-value formation (value: 85)

Warning bit “Diagnostics”:
If one of the supported valuation numbers is repeatedly only less than or equal to 80, a warning is triggered.

Example of how the valuation numbers appear over the traversed path:
Since the valuation numbers are transmitted together with the position, they can be displayed relative to the traverse path of the encoder. To this intent the minimum value of a valuation number within a certain position interval is to be determined and displayed. As the different valuation numbers react differently to various factors, such as contamination (see example below), all available valuation numbers are to be displayed. As an alternative, the “absolute” minimum of the valuation numbers can also be determined and displayed. Thanks to the large optical scanning surface of HEIDENHAIN encoders, the position interval can also be relatively large:

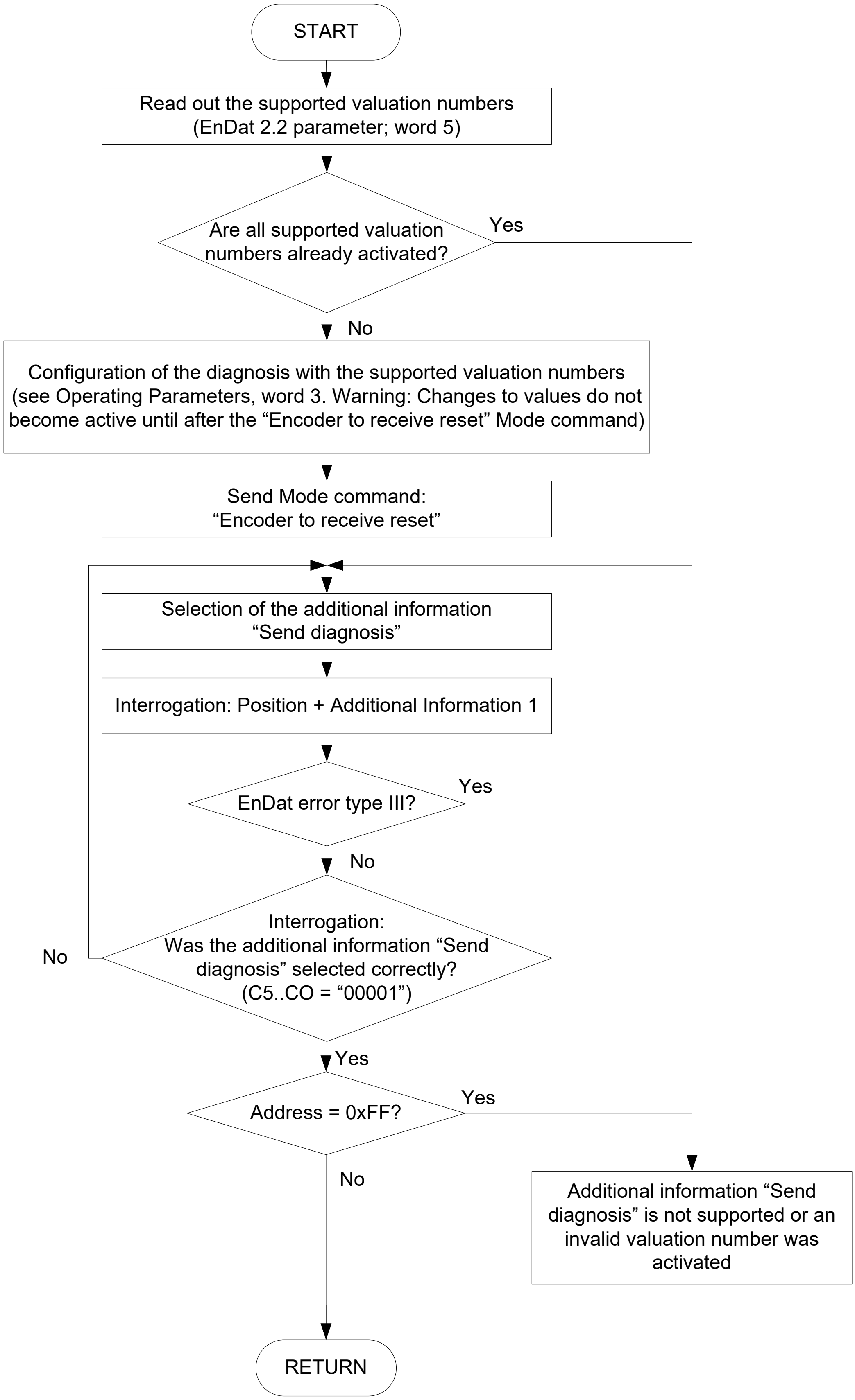
- For linear encoders: 1 mm
- For angle or rotary encoders: 0.5°
- There should be two displays for multiturn encoders:
  ➔ Singleturn display over 360° with a resolution of 0.5°
  ➔ Multiturn display: Minimum of the valuation number within one revolution

Example for a linear encoder that was intentionally and extremely contaminated:



Configuration of the Diagnosis and Valuation Numbers

Activation is only necessary once. It then remains stored in the encoder.
Activation of all supported valuation numbers is recommended.

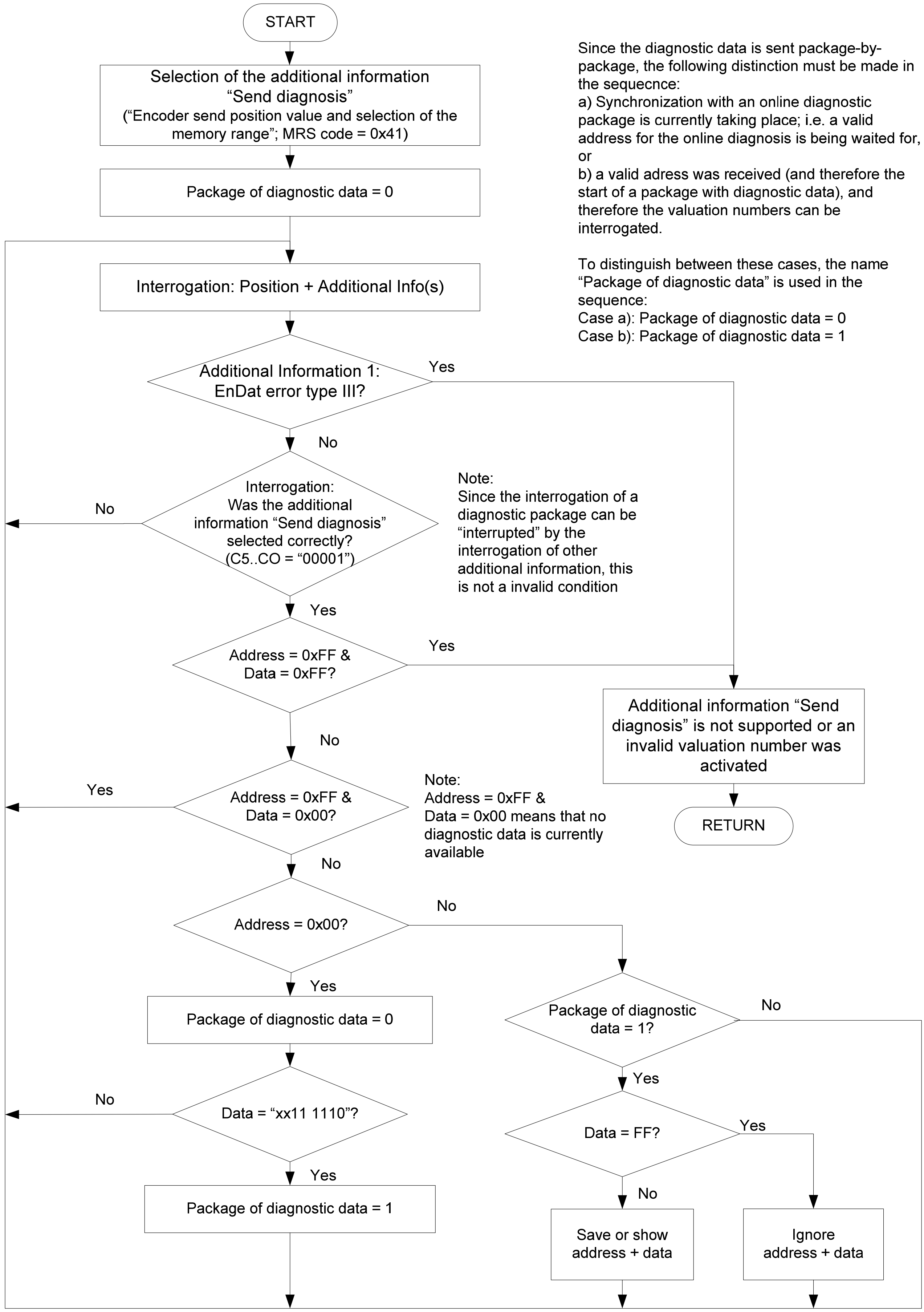


HEIDENHAIN
DR. JOHANNES HEIDENHAIN GmbH
83301 Traunreut, Germany

Interrogation of the diagnostic data and valuation numbers

Since the diagnostic data is sent package-by-package, the following distinction must be made in the sequecnce:
a) Synchronization with an online diagnostic package is currently taking place; i.e. a valid address for the online diagnosis is being waited for, or
b) a valid adress was received (and therefore the start of a package with diagnostic data), and therefore the valuation numbers can be interrogated.

To distinguish between these cases, the name "Package of diagnostic data" is used in the sequence:
Case a): Package of diagnostic data = 0
Case b): Package of diagnostic data = 1



9 Parameters for mounting the encoder (mounting parameters)

General information:
With the help of the mounting parameters, information is output that reflects or evaluates the correct mounting of the encoder. The mounting parameters are part of the diagnostics (output via EnDat additional datum 1) and are output by means of the so-called system-specific data. The mounting parameters can also be switched on and off (see the chapter entitled "Enabling mounting parameters"). The transmitted values for the mounting parameters are signed 16-bit values. The header, which separates the contents of the diagnostics, has the identifier 0x2F for the mounting parameters. If, in the future, further contents are transmitted via the system-specific data, then these contents will be provided with a different identifier.

Enabling mounting parameters
Support for the system-specific data can be requested via the "Parameters of the encoder manufacturer" for EnDat 2.2, word 5, bit 15. By means of the operating parameters word 3, bit 15, the system-specific data can then be enabled. Default: The system-specific data are not enabled.

If the system-specific data from the encoder are not supported, then the corresponding enable bit is ignored. No data are output.

The mounting parameters
Each mounting parameter has a length of 16 bits and is signed (two's complement representation); see also the remarks below. It is transmitted by means of two transfer units. The first transfer unit contains the low data byte of the parameter and has an even address. The second transfer unit has the subsequent, uneven address and contains the high data byte. The values of a parameter thus range from -32767 to +32767. If the parameter cannot be calculated or made available for some reason, the value -32768 is output.

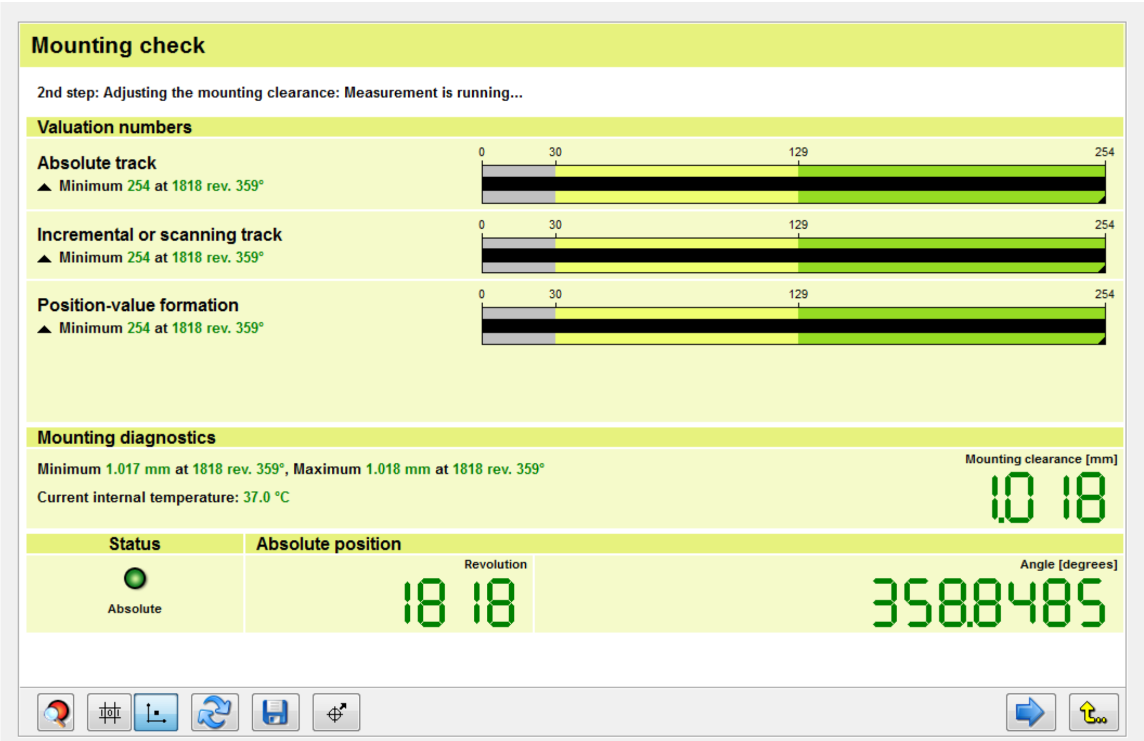
From the list of defined mounting parameters, an encoder only sends the parameters it can support or calculate. There is no parameter 0 because the address 0x00 is already assigned to the header. Parameters 1 to 7 (addresses 0x02 to 0x0F) are reserved.

Address	Abbreviation	Unit	Function
0x01			(Reserved)
:			
0x0F			
0x10	DISTANCE_L	µm	Mounting parameter 8
0x11	DISTANCE_H	µm	Mounting clearance in µm (-32767 µm to +32767 µm)
0x12			Mounting parameter 9
0x13			
0x14			Mounting parameter 10
0x15			
0x16			Mounting parameter 11
0x17			
...			...

Example:
Data contents of the diagnostics with activated valuation numbers and activated system-specific data

…..	
Address 0x03	Valuation number for absolute track
Address 0x04	Valuation number for position-value formation
Header for mounting parameters (0x2F)	
Address 0x10	Mounting clearance (low byte)
Address 0x11	Mounting clearance (high byte)
Header for standard diagnostics (0x3E)	
Address 0x01	Valuation number for incremental or absolute track
Address 0x03	Valuation number for absolute track
Address 0x04	Valuation number for position-value formation
Header for mounting parameters (0x2F)	
Address 0x10	Mounting clearance (low byte)
Address 0x11	Mounting clearance (high byte)
Header for standard diagnostics (0x3E)	
Address 0x01	Valuation number for incremental or absolute track
Address 0x03	Valuation number for absolute track
Address 0x04	Valuation number for position-value formation
Header for mounting parameters (0x2F)	
Address 0x10	Mounting clearance (low byte)
…..	

Example of the display of mounting clearance and valuation numbers through the ATS software. The “Minimum” and “Maximum” evaluations permit, among other things, the misalignment of the encoder rotor to be evaluated.



The following pages show the activation procedures for the system-specific data (mounting interface) and the interrogation of the diagnostics.

Remark: Numeric notation as a two's complement

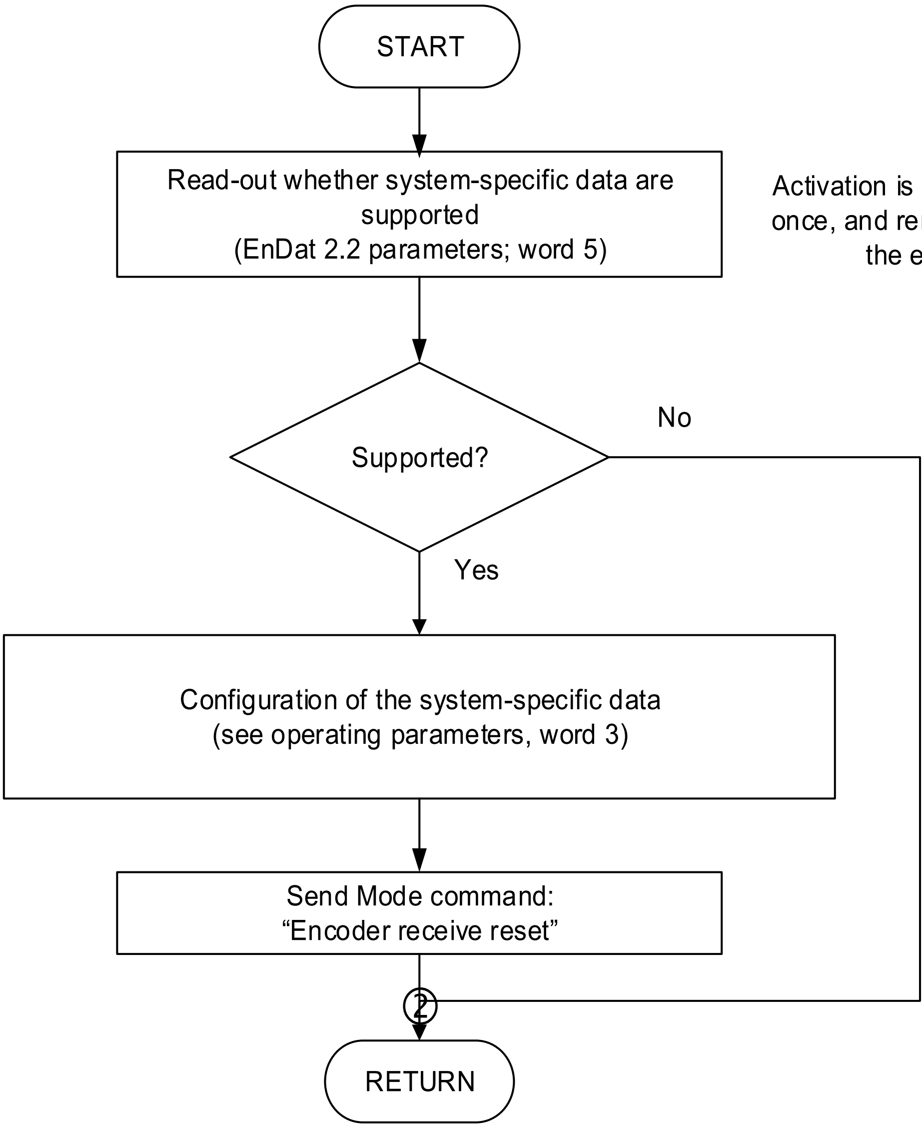
Value ranges:	0x0000 .. 0x7FFF	➔	"0" … "32767"
	0x8000	➔	Error code
	0x8001 .. 0xFFFF	➔	"-32767" .. "-1"

A decision can be made based on bit 15 (MSB):

- Bit 15 = 0 ➔ Positive number, no conversion necessary
- Bit 15 = 1 ➔ Subtract "1" and then invert the number

Examples:
Value 0x0123 ➔ Positive number with the decimal value "291"
Value 0x9876 ➔ Negative number with the decimal value "-26506"

Parameters for mounting the encoder

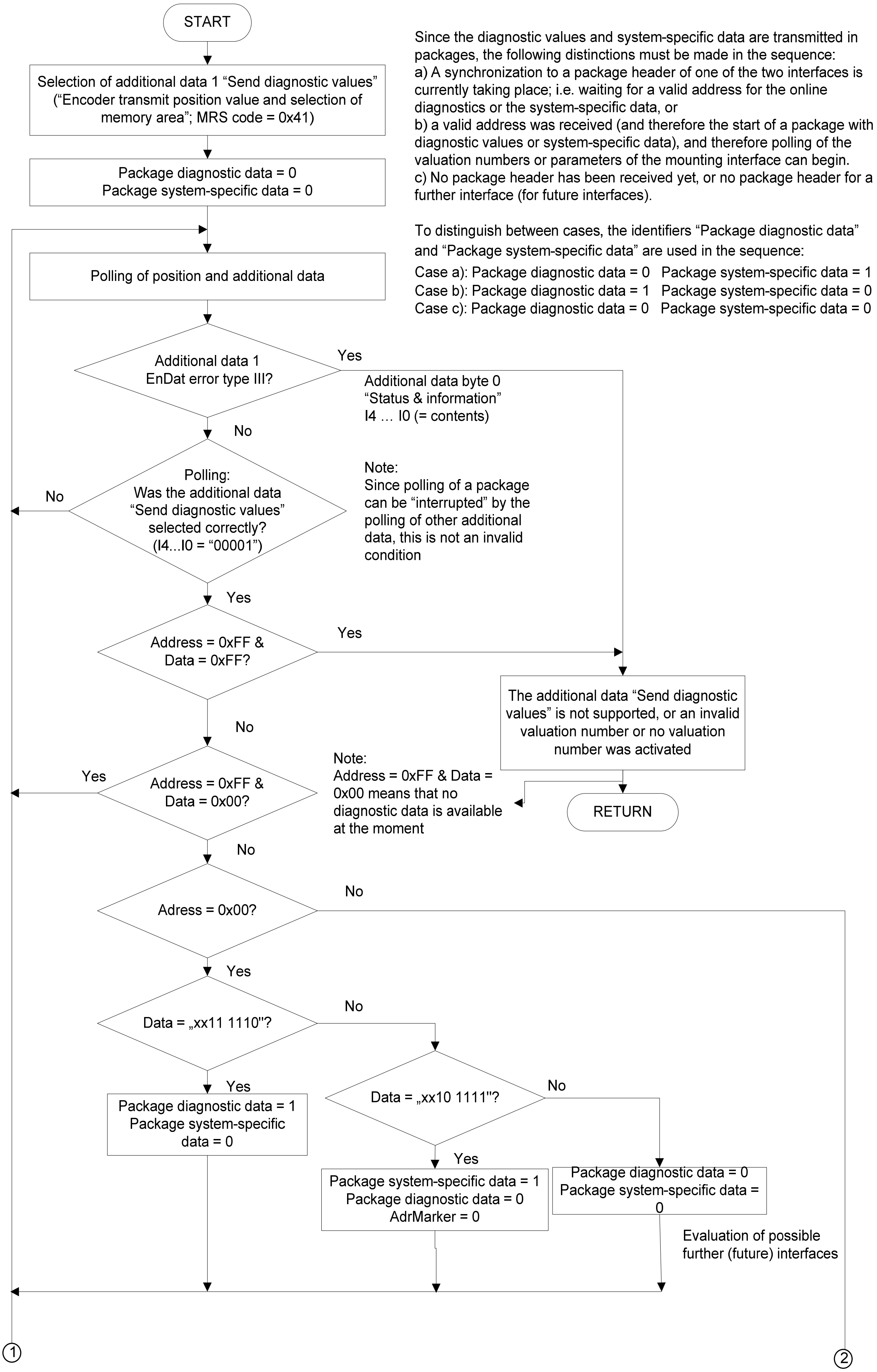


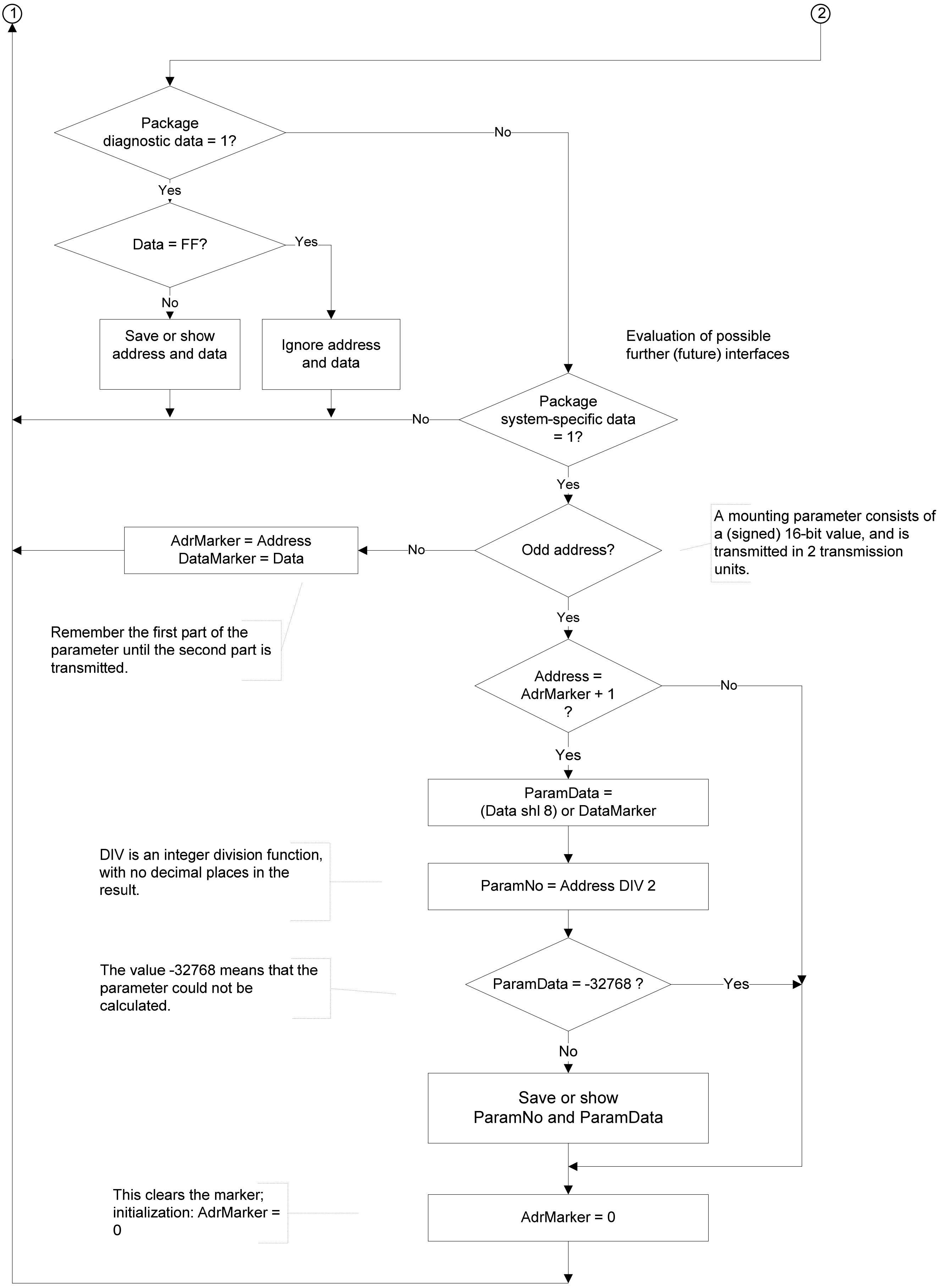
Activation is only necessary once, and remains stored in the encoder

Caution: Changes to the values do not become active until after the mode command "Encoder receive reset".

Since the diagnostic values and system-specific data are transmitted in packages, the following distinctions must be made in the sequence:
a) A synchronization to a package header of one of the two interfaces is currently taking place; i.e. waiting for a valid address for the online diagnostics or the system-specific data, or
b) a valid address was received (and therefore the start of a package with diagnostic values or system-specific data), and therefore polling of the valuation numbers or parameters of the mounting interface can begin.
c) No package header has been received yet, or no package header for a further interface (for future interfaces).

To distinguish between cases, the identifiers "Package diagnostic data" and "Package system-specific data" are used in the sequence:
Case a): Package diagnostic data = 0   Package system-specific data = 1
Case b): Package diagnostic data = 1   Package system-specific data = 0
Case c): Package diagnostic data = 0   Package system-specific data = 0





1
2
Package
diagnostic data = 1?
No
Yes
Data = FF?
Yes
No
Save or show
address and data
Ignore address
and data
Evaluation of possible
further (future) interfaces
Package
system-specific data
= 1?
No
Yes
AdrMarker = Address
DataMarker = Data
No
Odd address?
A mounting parameter consists of
a (signed) 16-bit value, and is
transmitted in 2 transmission
units.
Remember the first part of the
parameter until the second part is
transmitted.
Yes
Address =
AdrMarker + 1
?
No
Yes
ParamData =
(Data shl 8) or DataMarker
DIV is an integer division function,
with no decimal places in the
result.
ParamNo = Address DIV 2
The value -32768 means that the
parameter could not be
calculated.
ParamData = -32768 ?
Yes
No
Save or show
ParamNo and ParamData
This clears the marker;
initialization: AdrMarker =
0
AdrMarker = 0

10 EnDat Datum Shift

The datum shift is used to permanently move the datum of the encoder (the offset is stored in the EEPROM). The new set datum can be assigned a write protection which, however, can no longer be canceled.

General information:

- The datum shift is mainly used for rotary encoders. The main application is "electronic zeroing" of the position. This greatly simplifies mounting a rotary encoder to a motor, for example.
- When using the datum shift for linear encoders, please remember that the position value is output without an algebraic sign; see Chapter 1 or Chapter 3.5.23 of the EnDat Specification.
- After writing words 0, 1 and 2 of the operating parameters, a reset is required for the datum shift to become effective (also see the Specification).
- The value is saved as a two's complement in words 0, 1 and 2 (48-bit width). This simplifies calculation for the encoder. In this case, it requires only a simple addition of position value and datum shift in the encoder.
- In the factory default setting of the encoder, the value for the datum shift defined by the encoder manufacturer is stored in words 22 and 23 of the EnDat parameters as well as in the operating parameters. This can be used both to check whether the datum has already been shifted and to reset a datum shift that has been performed. If the datum shift is reset, the original values of the encoder manufacturer must be written in the operating parameter area.

EnDat-compliant datum shift or datum shift for purely serial operation:
When setting the datum, you must also determine whether the datum position is to be assigned to the signal period. The assignment is necessary if the analog1 V_{PP} signals are also used in the customer's subsequent electronics. In this case, the datum shift must be a whole-number multiple of the signal period to ensure that the relationship between the relative value and the absolute value is maintained. In purely serial operation this assignment must not be taken into account. Also see Chapter 3.6.1 of the EnDat Specification.

EnDat parameters used:
The following table shows the parameters required for the datum shift and the location where they are stored:

Parameter	Parameters of the encoder manufacturer
Encoder model	Word 14
Signal period or signal periods per revolution	Words 15 and 16
Measuring step or measuring steps per revolution	Words 20 and 21

Datum shift applications:

a) Datum is shifted to a certain position value:
In this case, the desired position (target position) must be subtracted from the current position.
The resulting value is used for the algorithm of the datum shift. See the attached flowchart:
"Position" = current position - target position

The target position "0" (i.e. zeroing) is a special case. See the attached flowchart: "Position" = current position - target position ➔ "Position" = current position

b) Datum is shifted by a certain value:
The value for the datum shift (a multiple of the measuring step) is to be used for the calculation. The algebraic sign is to be considered and depends on the direction of the datum shift. The value for the datum shift is then used in the calculation according to the appended flowchart.

Note on the use of the "User datum shift" operating parameter in linear encoders.

Requirement in connection with the position value comparison:
Within the entire measuring range, the highest-value bit of the position value must be 0.

This means that a user datum shift (NPV) is permissible within the following limits:
$0 \leq NPV \leq [\text{meas. step} \times (2^{number\_bits - 1}) - \text{meas. length} - 0.05]m$

Example:
Example: Measuring length = 3 m; number_bits = 32; measuring step = 5 nm
$0 \leq NPV \leq [5 \times 10^{-9} \times 2^{(32-1)} - 3 - 0.05]$ m
$0 \leq NPV \leq [10.74 - 3 - 0.05]$ m
$0 \leq NPV \leq 7.69$ m

However, the chosen datum shift should always be as small as possible – for application with a linear motor, for example, it should be less than the motor`s pole pair spacing.

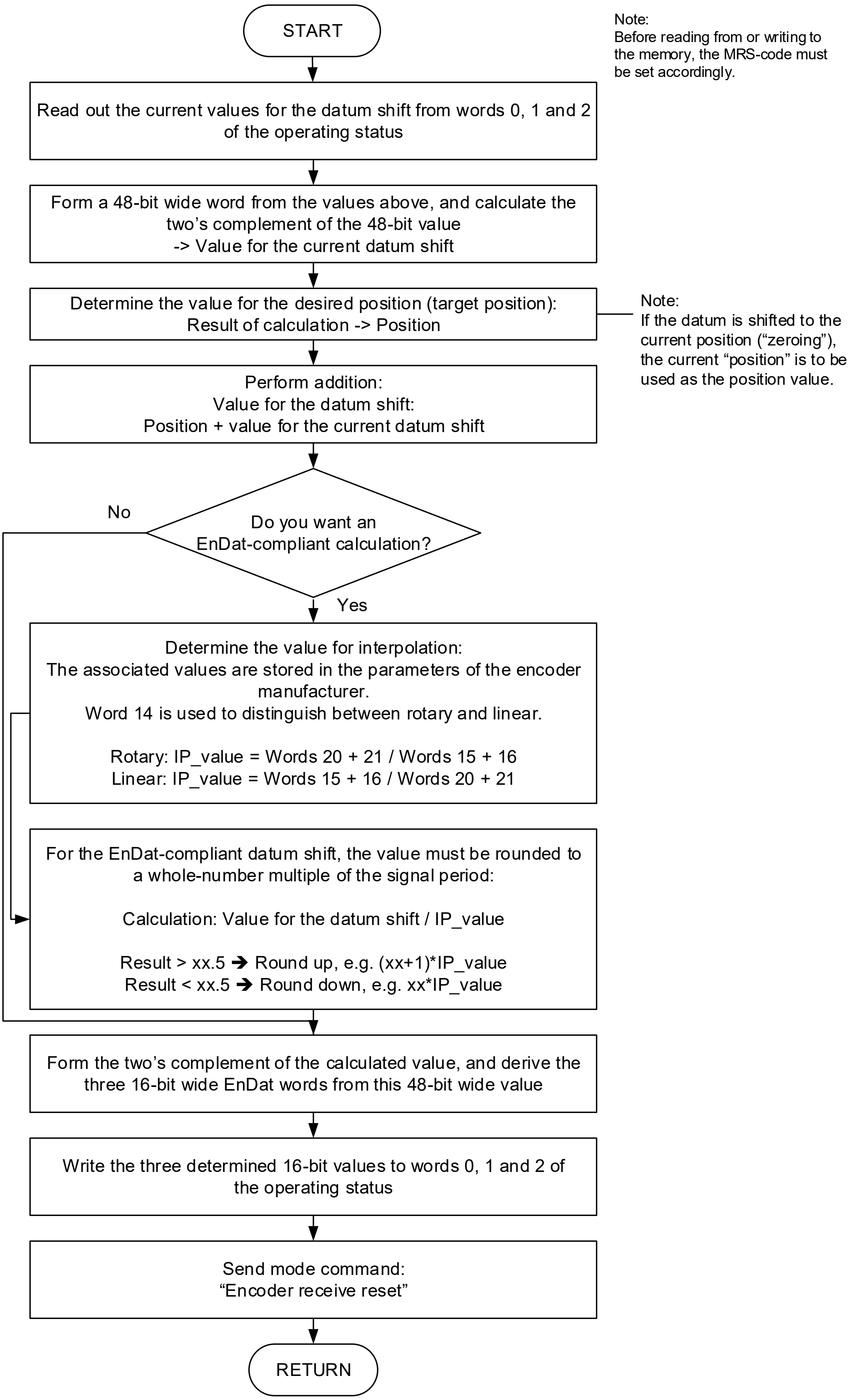
Note on the use of singleturn encoders with a non-binary line count (only for incremental encoders: EIB used in combination with an encoder)

The content of the 48-bit word (words 0, 1, and 2) for the datum shift may only be within the range of 0 … measuring steps per revolution (content of words 20+21).
Any other values are impermissible.

Contrary to the algorithm described, it must be checked at the end of the calculation whether the value for the datum shift is within the above-stated range. If necessary, it must be corrected.

The following results for zeroing the position:
Datum shift = (content of words 20+21) – current position

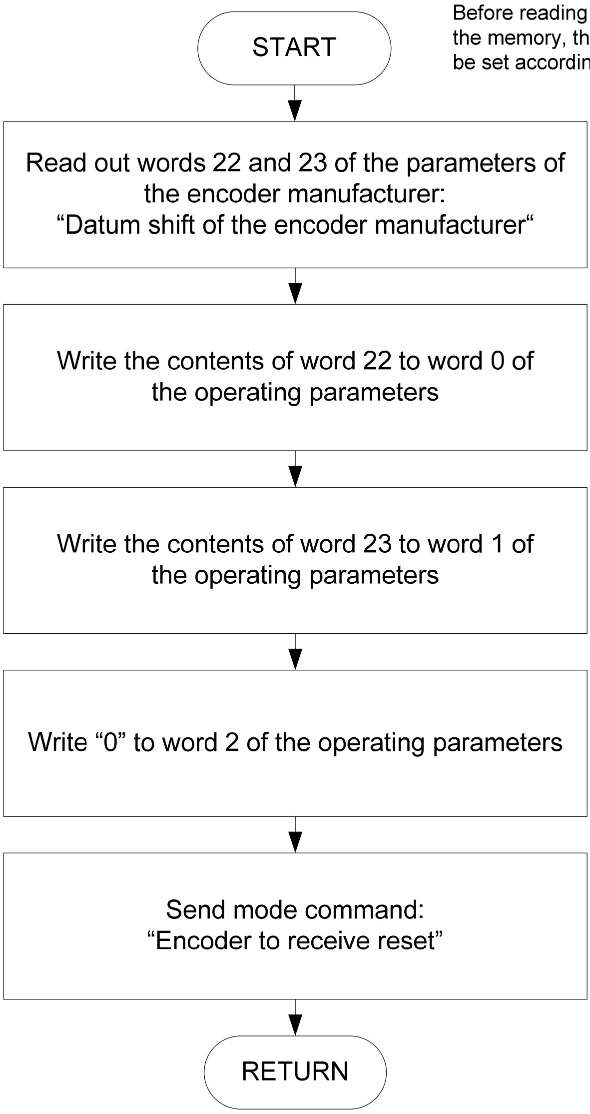
Setting the datum shift



Resetting the datum shift

Resetting the datum shift to the factory default setting of the encoder

Note:
Before reading from or writing to the memory, the MRS-code must be set accordingly.



11 Incremental Encoders

Referencing incremental encoders with EnDat 2.2:

The EnDat 2.2 interface makes it possible to connect incremental encoders. Unlike absolute encoders, referencing is necessary with incremental encoders. Referencing enables an absolute reference of the transmitted relative position. Referencing is implemented by crossing the reference mark(s):

- With systems having one reference mark, this must be crossed
- With systems having distance-coded reference marks, two adjacent reference marks must be crossed

EnDat 2.2 position values with incremental encoders:

- The transferred position value corresponds to the relative position (position value 1).
- The absolute position value can be read from additional data 1 (position value 2). Position value 2 has to be read distributed across three requests (LSB, center part, MSB). 16 bits are transferred for each request to give a position width of 48 bits for the absolute position. The incremental and absolute values are synchronized from position 2 by querying the LSB value. Querying the LSB value means that all 48 bits are frozen, enabling a reference to the relative position for this request to be established.
- Alternative to the methods specified above for requesting position value 2 when the encoder supports additional data 2 "Asynchronous position value / position value 2":
  - Activate additional data 1 "Send position value 2 word 1 LSB"
  - Activate additional data 2 "Position value asynchronous, word 2"

  In this way a position value 2 can be read out with 32 bits. This position value 2 is synchronized with a position value 1 for each request.
- The RM bit (transferred with the additional data) shows whether the system is referenced.
- In addition, with the WRN bit a warning is output if the reference mark was not crossed. After switch-on, this warning is thus usually set. This must be observed in Power-on behavior: “Clearing errors or warnings” (EEPROM life expectancy!).
- After switch-on the RM bit is set to “0”, position value 1 corresponds to position value 2 and both contain the relative position value. If renewed referencing is implemented (see “Re-referencing”), position values 1 and 2 do not change but continue to count from their current value.

In principle this gives the following process for referencing the incremental encoder via the EnDat interface (EnDat 2.2 mode commands must be used):

- Selection of any additional data to request the RM bit.
  Recommendation: “Send position value 2, word 1, LSB”
- If reference mark(s) were crossed, then the RM bit is set from 0 to 1 and position value 2 is set to the absolute value. Position value 1 however continues to count relatively. With the RM bit on "1", the warning "Reference point not reached" can also be reset (the corresponding EnDat 2.2 mode command should be sent).
- The absolute position value can be read out from additional data 1.

See also the software flow for "Referencing incremental encoders with EnDat 2.2".

See also the alternative software flow with support of additional data 2 "Position value asynchronous / position value 2".

Note:
On incremental linear encoders, e.g. LIP 200 or EIB x92, the value “0” is saved as default in the EnDat 2.1 parameters, words 20+21 (see EnDat Specification, Chapter 3.5.12) → The specifications for the cited EnDat 2.2 parameters are to be taken into account.

Temporal behavior:
See EnDat specification, chapter 2.3.4.3, "Incremental encoders".

Remarks:
Pos1: Overflow occurs at position 2^(number of clock pulses to position value transfer)
Pos2: Rotary systems: After referencing, overflow occurs at the position "Number of measuring steps per revolution". This must be specifically observed with non-binary line counts.

Behavior in the event of an error (error bit F1 or F2 active):

A set error bit F1 or F2 shows that the transferred position values 1 and 2 are no longer valid. This also shows that referencing for the system is no longer valid although the RM bit is still set. Specific measures must therefore be taken if error bit F1 or F2 is set.

The system must at first be transferred to safe condition. Depending on the system and the encoder used, further error handling is possible before the system is shut down. This error handling can also include a "re-referencing" of the system.

After the RM bit has been set to "0" (re-referencing), the error messages and warnings can be reset with the EnDat 2.2 mode command "Encoder send position values with additional data and receive error reset".

See also the software flow for "Error handling (F1 or F2) with incremental encoders".

Re-referencing an incremental encoder:

Re-referencing an incremental encoder means that the RM bit and the referencing can be reset via the EnDat interface. This means that the process for referencing an encoder can be repeated, for example if error bit F1 or F2 is set. If the re-referencing is executed by the control, the encoder has the following settings:

- RM bit = “0”
- The warning “Reference mark not traversed” is output
- Position values 1 and 2 will not change

Whether the incremental encoder supports the functionality of the re-referencing can be read out from the "Parameters of the encoder manufacturer for EnDat 2.2", word 46.

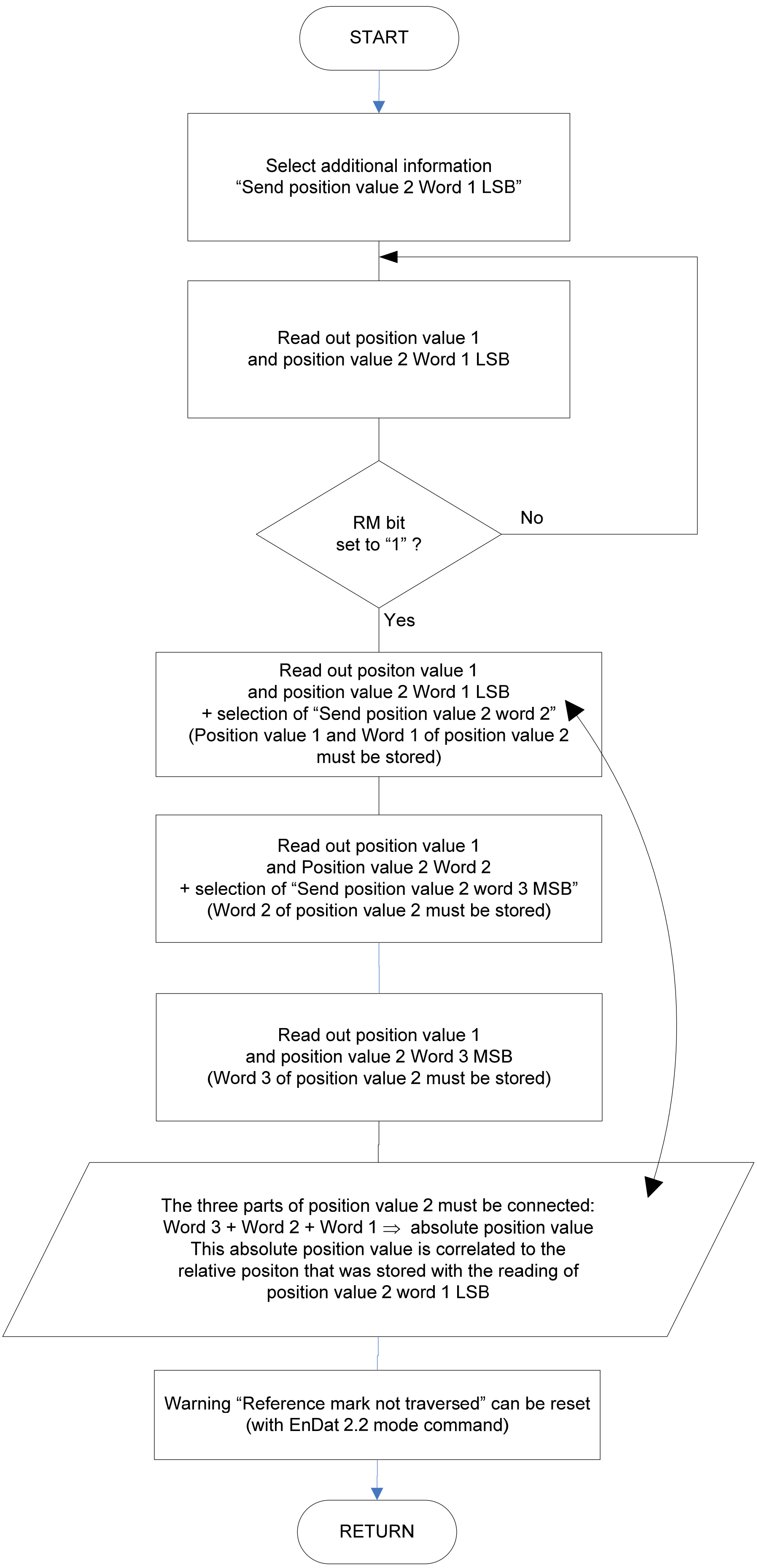
If the encoder supports re-referencing, one of the following methods can be applied for this:

a. Send the “Encoder receive reset” mode command (EnDat 2.1 mode command).
   Remark:
   The “Encoder receive reset” command is also required for the clearing of errors and warnings as part of the “Recommended procedure after power-on” sequence. This means that the momentary reference is also cleared as part of the power-on procedure.
   If errors and warnings are to be cleared during operation, without resetting the reference, then the “Encoder send position value and receive error reset” EnDat 2.2 command must be used.
   or
b. Set the RM bit to "0" by writing the value 0x0001(hex) to word 11 of the memory area "Operating Parameter 2".

See also the software flow for "Re-referencing of incremental encoders."

Remark:
The "re-referencing" feature is normally not supported. Please contact HEIDENHAIN if this feature is required.

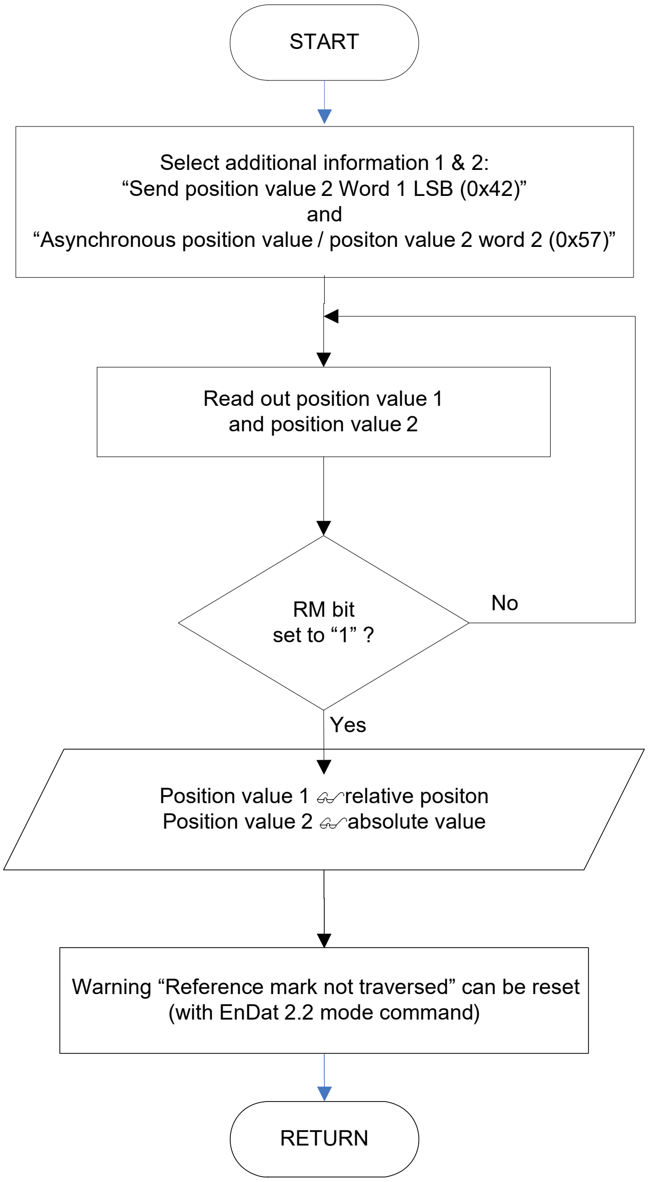
Referencing incremental encoders with EnDat 2.2



HEIDENHAIN
DR. JOHANNES HEIDENHAIN GmbH
83301 Traunreut, Germany

Referencing incremental encoders with EnDat 2.2
Alternative if encoder supports additional information 2 “Asynchronous position”

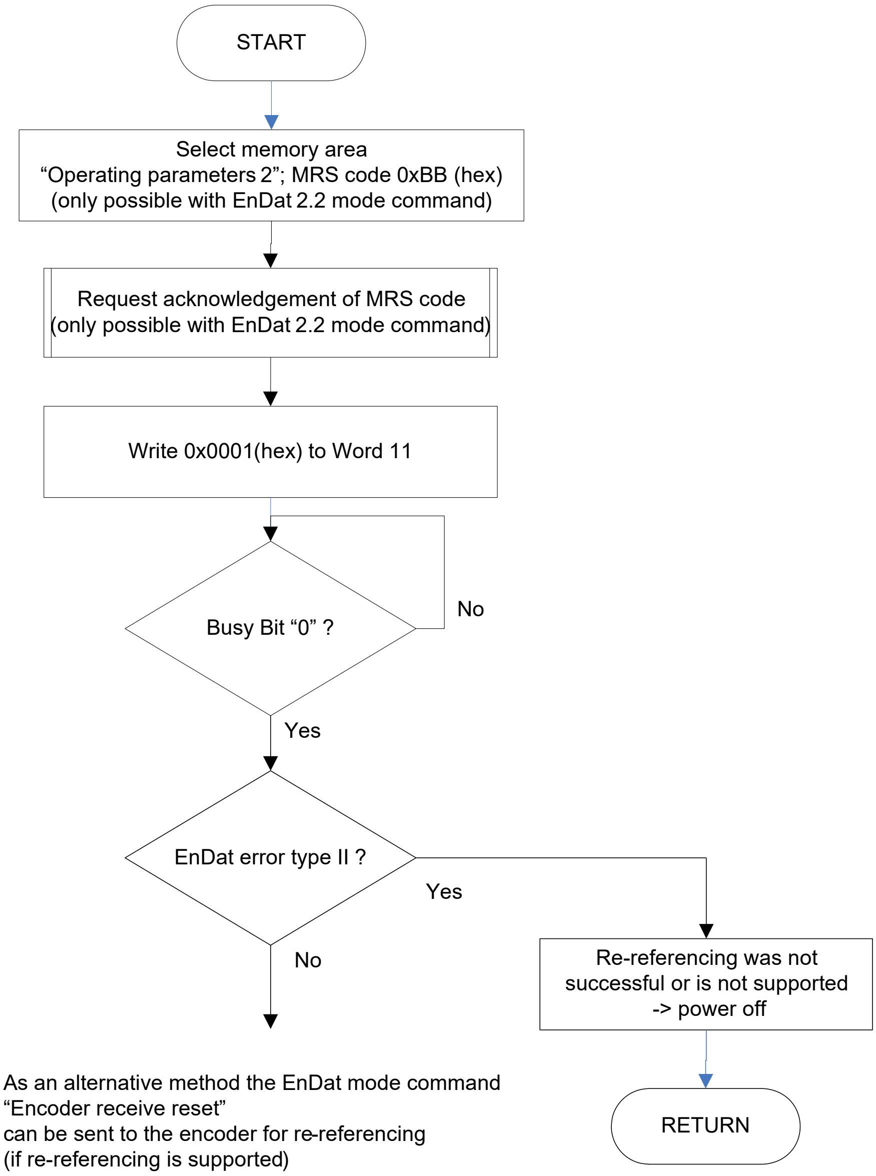
Can only be used if code length of the encoder is equal to or less than 32 bits.
„Caution: only for special applications;
Please have consult with HEIDENHAIN“



HEIDENHAIN
DR. JOHANNES HEIDENHAIN GmbH
83301 Traunreut, Germany

Re-referencing incremental encoders

Re-referencing can only be done if this is supported by the incremental encoder. This can be read out from the “Parameters of the encoder manufacturer for EnDat 2.2”, word 46

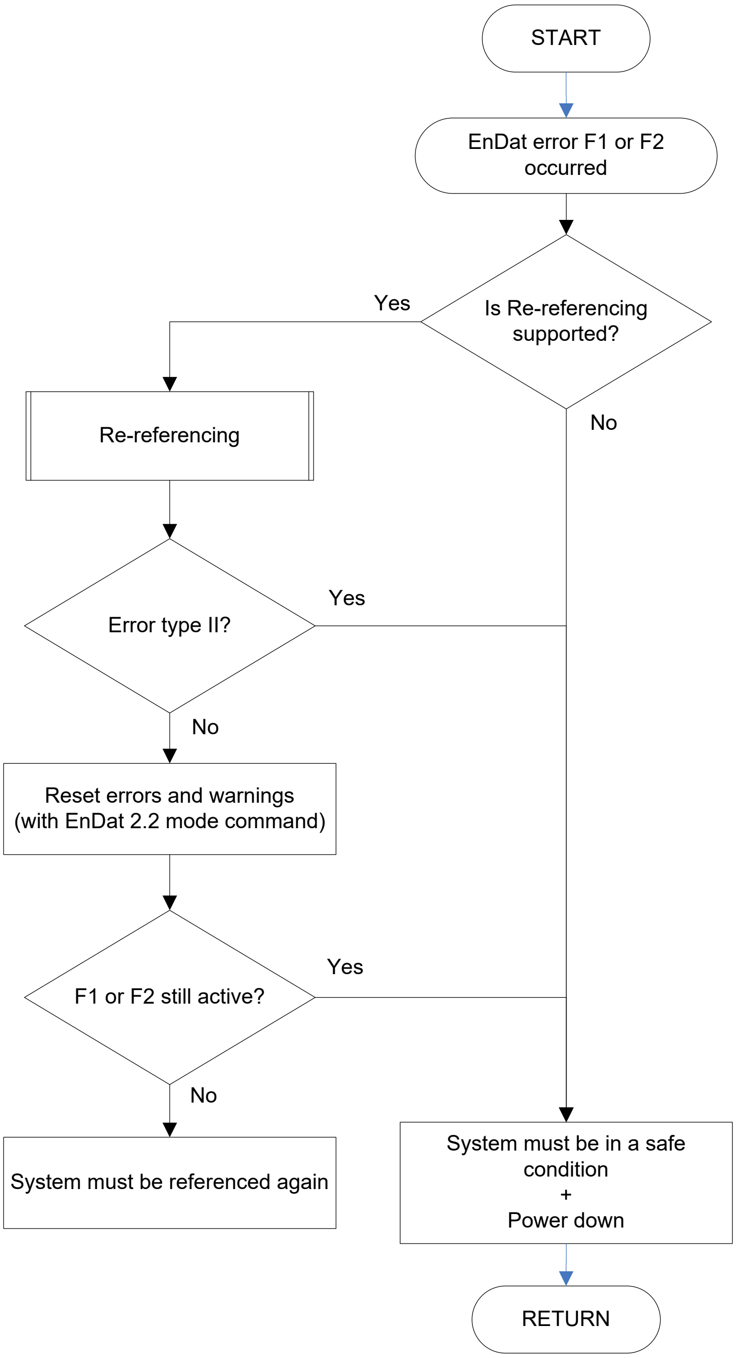


As an alternative method the EnDat mode command
“Encoder receive reset”
can be sent to the encoder for re-referencing
(if re-referencing is supported)

Error handling (F1 or F2) with incremental encoders

If error F1 or F2 occurs, the system must be forced to a safe system status before further measures are taken.
If re-referencing cannot be performed successfully, the system must be powered down and up again to start the referencing procedure again.

Re-referencing can only be done if this is supported by the incremental encoder. This can be read out from the "Parameters of the encoder manufacturer for EnDat 2.2", word 46



12 OEM Memory Area

General Information:
In principle the memory is freely programmable. HEIDENHAIN does not have requirements for the contents of the programming. The memory is divided into four areas. These areas can be used either by the OEM (parameters of the OEM areas 1..4) or by the encoder manufacturer for compensation values (compensation values areas 1..4). The content of the compensation-value areas is not of interest to the user. The EnDat 2.1 parameters (words 9-12, interrogation of words 9 and 10 suffices) contain information on whether the OEM area is supported, and which addresses within an available area can be addressed. Different families of encoders support different OEM memory areas and different address areas. Therefore, the assignment of the OEM areas must be read out for each and every encoder. For this reason, the subsequent electronics should form addresses relative to the determined values, and not use absolute addresses. The programming must be adapted to the individual encoder.

MRS code for the OEM areas:
The OEM memory is divided into four areas. Each area can be addressed via an individual MRS code:

- OEM 1: “10101001” = 0xA9
- OEM 2: “10101011” = 0xAB
- OEM 3: “10101101” = 0xAD
- OEM 4: “10101111” = 0xAF

Reading out the start and stop addresses of the OEM memory areas:
Each of the four OEM memory areas has its own start and stop addresses. These start and stop addresses define the usable range of the respective OEM memory area, and must be read out separately for each encoder. These start and stop addresses must be taken into consideration when saving or reading, i.e. reading and writing must occur relative to these addresses.
Also see "Parameters of the encoder manufacturer", words 9 and 10.

Example:

Content of word 9 ➔ 0x5053 ➔	OEM 1: 64.. 255 (starting address .. stop address; decimal) OEM 2: 0 .. 255
Content of word 10 ➔ 0xFF50 ➔	OEM 3: 0 .. 255 OEM 4: Not available

Reading or writing of OEM memory areas:
The OEM memory areas are accessed via MRS code and the address. EnDat 2.1 and EnDat 2.2 memory accesses are possible. For more information, refer to the "Sequences and Data Structures" chapter.

13 Rotary Encoders with Battery-buffered Revolution Counter

13.1 Contents

This document refers to rotary encoders with battery-buffered revolution counter. When implementing the interface, please pay particular attention to the following notes:

- The revolution counter usually has a resolution (multiturn resolution) of at least 65536 revolutions. Therefore, the read-out procedure is different than with all other encoders and must be fully and properly implemented as defined in the EnDat Specification D297403, section 3.5.9.
- In some cases, warnings or error messages (e.g. battery charge, power interruption) require actions by the operator or customer in order to remedy their causes. It is important to make sure that these cases are indicated to the operator in plain text.
- The standard versions of the battery-buffered rotary encoders do not support the *Overflow/Underflow* error message.

Abbreviations used: M – Multiturn, S – Singleturn

13.2 Interface hardware

Bidirectional data transfer (measured values or parameters) between position encoders and subsequent electronics is performed with transceiver components in accordance with RS-485 (differential signals), synchronous to the clock signal (CLOCK) produced by the subsequent electronics.

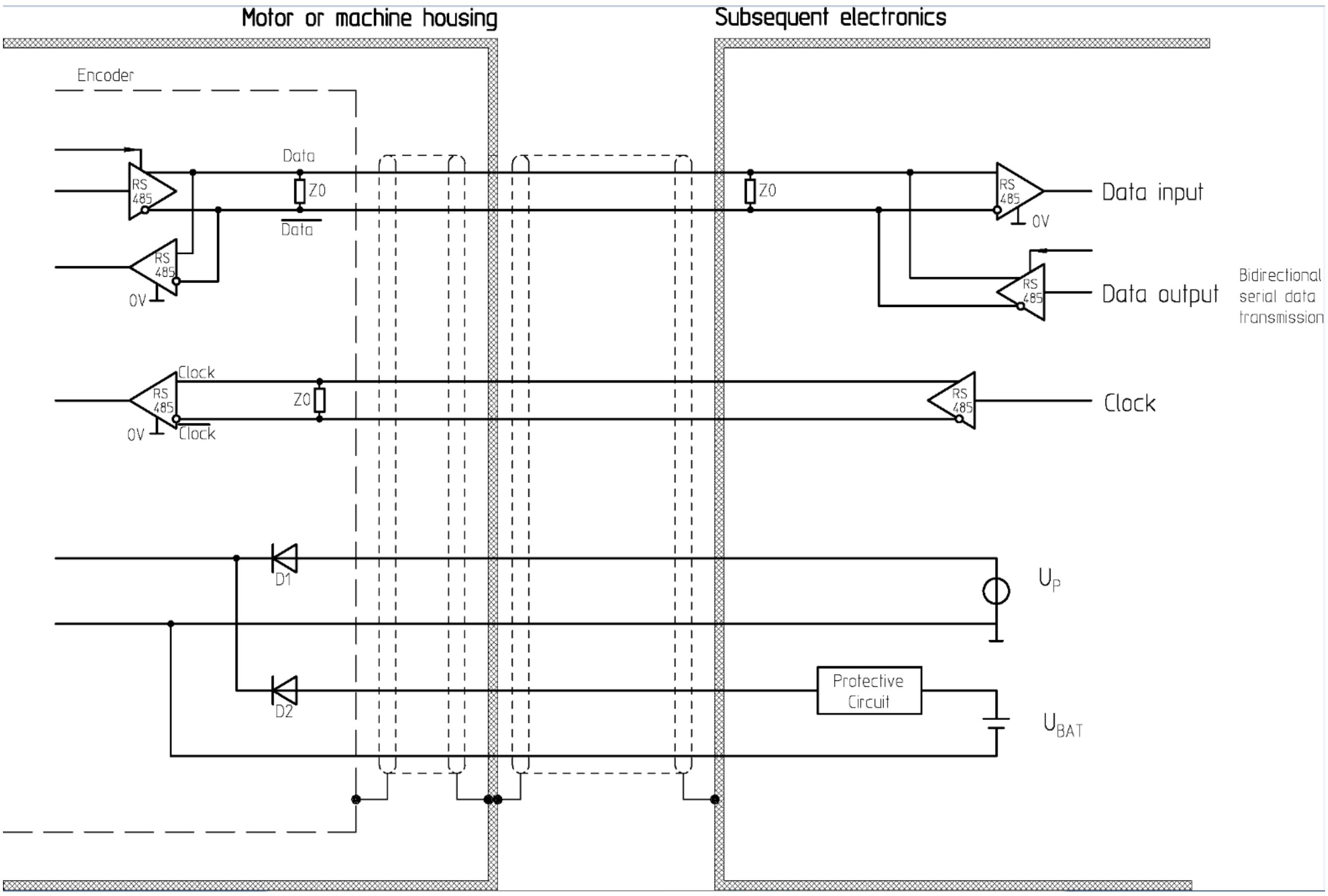


Figure 1: Interface hardware

Z_0 = 120 Ohm
RS 485 transceiver

If the application requires compliance with DIN EN 60086-4 or UL 1642, an appropriate protective circuit is required for protection against wiring errors.

13.3 Operating modes

Rotary encoders with battery-buffered revolution counters differentiate between two operating modes, the *normal operating mode* (the encoder is operated via the main power supply U_P) and the *battery-buffered operating mode* (the encoder is operated via the external buffer battery U_{BAT} if no sufficient main power supply is available).

Depending on the voltage of the main power supply, the encoder switches automatically between these two operating modes.

a) U_P within defined limit values: The encoder switches to / runs in *normal operating mode*

b) U_P deactivated: The encoder switches to / runs in *battery-buffered operating mode*

In *battery-buffered operating mode* the complete internal electronics are switched off except the parts required for multiturn scanning. The encoder thus transfers no data via the interface and does not respond to commands from the subsequent electronics. When the encoder is switched back to *normal operating mode*, complete communication is resumed.

13.4 Counting range of revolution counter

Table 1: Counting range of the revolution counter (with and without Overflow/Underflow error message)

Parameter	Standard version	Available special versions
Number of distinguishable revolutions (E21W17 or E22W34)	65536	-
Support of *Multiturn counter reset* (E22W02)	Supported	-
Reset position of the revolution counter [# revolutions]	0	-
Overflow/Underflow error message if count exceeds ±32768 revolutions	Not supported	Supported

Abbreviations:
E21W17: “Parameters of the encoder manufacturer” memory area, word 17
E22W02: “Parameters of the encoder manufacturer for EnDat 2.2” memory area, word 02
E22W34: “Parameters of the encoder manufacturer for EnDat 2.2” memory area, word 34

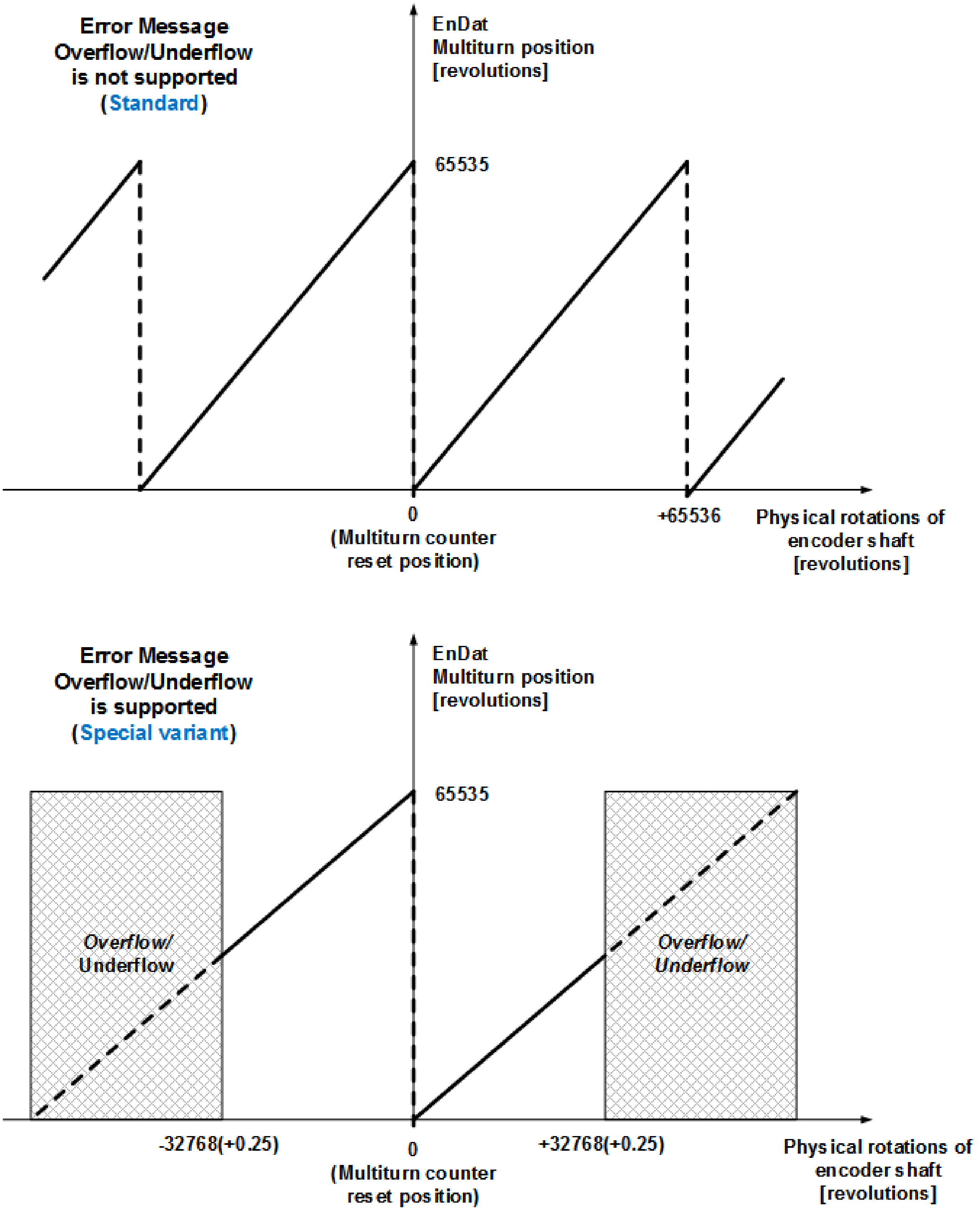


Figure 2: Counting range of the encoder and overflow limits (without EnDat datum shift)

13.5 Error messages and their handling

Various forms of support must be implemented depending on the type of error messages generated.

⚠ **A general deleting of error messages during power-up of the encoder (U_P) without reading these out and evaluating them is thus impermissible. This applies particularly if an *M Power interruption* error message is detected during power-up of the encoder.**

(See Chapter 2 *"Sequences and Data Structures"*)

Detected errors can be read out with various levels of granularity within the EnDat interface:

- **Low granularity:**
    - Group message F1 (error message 1) simply signals that an error has occurred (is a part of all transmitted EnDat positions)
    - Group message WRN (warning) simply signals that a warning has occurred (is a part of all additional data with an EnDat 2.2 position transfer)
- **Middle granularity:**
    - Individually listed error messages in word 0 of the memory area *"Operating status"*
    - Individually listed warnings in word 1 of the memory area *"Operating status"*
- **High granularity:**
    - Individual sources of error are listed under *Operating status error sources* and can be read out via the EnDat 2.2 additional data (note: only possible for error messages, not for warnings)

Not all encoder variants support all defined *operating status error sources* and *warnings.* Thus this information should be read out of the encoder EEPROM as follows:

- Supported *Warnings*:
  Read out from word 36 in memory area *"Parameters of the encoder manufacturer"* (see also EnDat Specification D297403, section 3.5.21)
- Supported *Operating status error sources*:
  Read out from word 35 in memory area *"Parameters of the encoder manufacturer for EnDat 2.2"* (see also EnDat Specification D297403, section 3.9.26)

Table 2 shows an overview of the additional *operating status error sources* and *warnings* supported by rotary encoders with battery-buffered revolution counter.

Table 2: Overview of the additional error sources and warnings of rotary encoders with battery-buffered revolution counter and their assignment.

	Word 0 in memory area *"Operating status"* (error messages)[1]	Word 1 in memory area *"Operating status"* (warnings)	Group message F1 (error message 1) or group message WRN (warnings)	*Operating status error sources*
S System	0x04 [1]		F1	0x0100
M Pos 1	0x04 [1]		F1	0x0400
M System	0x04 [1]		F1	0x1000
M Power interruption	0x04 [1]		F1	0x2000
Overflow/Underflow [2]	0x04 [1]		F1	0x4000
Battery charge		0x08	WRN	-

Abbreviations: M = Multiturn; S = Singleturn

1) It must be considered that bit 2 [*position error*] in word 0 of the memory area *"Operating status"* has several *operating status error sources* assigned to it.

2) The *Overflow/Underflow* error message is deactivated for the standard variants of the encoders.

In order to differentiate between individual error sources and to reliably detect the *M Power interruption* and *Overflow/Underflow* error sources that are of maximum importance for the proper functioning of the encoders, the *operating status error sources* must be read out.

The *operating status error sources* can be selected as *additional data 2* and read out by the EnDat command *"Encoder send position values with additional data"* [mode command *"111000"*] (see EnDat Specification D297403, section 2.3.5, 2.4.2.2).

Table 3 shows conditions for the occurrence of the *M Power interruption* error message and the *Battery charge* warning.

Table 3: Conditions for an *M Power interruption* error message and *Battery charge* warning

Main voltage U_P [1]	**Battery voltage U_{BAT}** [1]	**Status of communication between encoder and subsequent electronics**	**Status of *M Power interruption* error message**	**Status of the *Battery charge* warning**
Available (Within the specified range)	**Available** (Between the maximum permissible voltage and the limit value for the *Battery charge* warning[1])	**Possible** (Encoder in normal operating mode)	**0** (Not set)	**0** (Not set)
Available (Within the specified range)	**Insufficient** (Below the limit value for the *Battery charge* warning [1])	**Possible** (Encoder in normal operating mode)	**0** (Not set)	**1** (Set)
Not available (below the specified range)	**Available** (Between the maximum permissible voltage and the limit value for the *Battery charge* warning [1])	**Not possible** (Encoder in battery-buffered operating mode)	**0** (Not set)	**0** (Not set)
Not available (below the specified range)	**Insufficient** (Between the limit value for the *Battery charge* warning [1] and the limit value for the *M Power failure* error message [1])	**Not possible** (Encoder in battery-buffered operating mode)	**0** (Not set)	**0 | 1** [2) 3)] (Unknown)
Not available (below the specified range)	**Not available** (Below the limit value for the *M Power failure* error message [1])	**Not possible** (Encoder completely without voltage)	**1** [2] (Set)	**0 | 1** [2) 3)] (Unknown)

1) The values are stated in the product-specific documentation.
2) Readout of error messages and warnings from encoder only possible after power supply and communication between encoder and subsequent electronics have been restored.
3) The status of *Battery charge* is unknown. Whether *Battery charge* has been set by the encoder depends on how the completely powerless state was reached.

S System error message

An *S System* error message signals that a singleturn-related internal error during the initialization phase of the encoder has occurred.

⚠ **The transferred position value is not reliable under these conditions.**

Corrective measure: The power-on cycle should be repeated. If this also fails, a hardware problem of the encoder is probable.

M Pos 1 error message

An *M Pos 1* error message signals that the multiturn scanning or processing of the multiturn position has failed during encoder operation.

⚠ **The multiturn part of the transferred position value is not reliable under these conditions.**

Corrective measure: The consistency of the revolution counter value must be checked. If this is not possible, the axis must be referenced (see section 6, *"Referencing of rotary encoders with battery-buffered revolution counter"*).

M System error message

An *M System* error message signals that a multiturn-related internal error during the initialization phase of the encoder has occurred.

⚠ **The multiturn part of the transferred position value is not reliable under these conditions.**

Corrective measure: The power-on cycle should be repeated. If this also fails, a hardware problem of the encoder is probable. The consistency of the revolution counter value must also be checked. If this is not possible, the axis must be referenced (see section 6, *"Referencing of rotary encoders with battery-buffered revolution counter"*).

M Power interruption error message

The *M Power interruption* error message signals that both the voltage of the buffer battery (U_{BAT}) and the main supply voltage (U_P) have fallen below the defined limit values (see product-specific documentation).

The subsequent electronics can only read out this error message when the main voltage supply has been restored.

⚠ **The multiturn part of the transferred position value is not defined and therefore faulty. The possibly simultaneously occurring *M Pos 1*, *M System* and *Overflow/Underflow* error messages are to be ignored.**

Corrective measure: The axis must be referenced again (see section 6, *"Referencing of rotary encoders with battery-buffered revolution counter"*).

Overflow/Underflow error message (if supported)

The *Overflow/Underflow* error message signals that the specified (multiturn) counting range of the encoder has been exceeded (see Figure 2). The encoder remains fully functional. The *Overflow/Underflow* error message can only be cleared when the multiturn value is again in the specified counting range.

Corrective measure: Bring the axis into the specified counting range of the encoder again, and clear error messages according to the EnDat specification.

Note: To return to the defined counting range more quickly, the process of resetting the revolution counter (*Multiturn Counter Reset,* see Figure 4) can be used (if the resulting loss of the previous reference point of the revolution counter is not problematic).

Battery charge warning

A *Battery charge* warning signals that the external buffer battery is not connected or that the voltage of the buffer battery (U_{BAT}) has fallen below the defined limit value for battery replacement (see the product-specific documentation). As long as no *M Power interruption* error message occurs, the encoder remains fully functional (including correct position values).

Corrective measure: The external buffer battery must be replaced or connected (see section 7, *"Exchanging the external buffer battery").*

Note: The *Battery charge* warning should be cleared after exchanging the buffer battery according to the EnDat specification (see EnDat specification D297403, chapter 3.4.2).

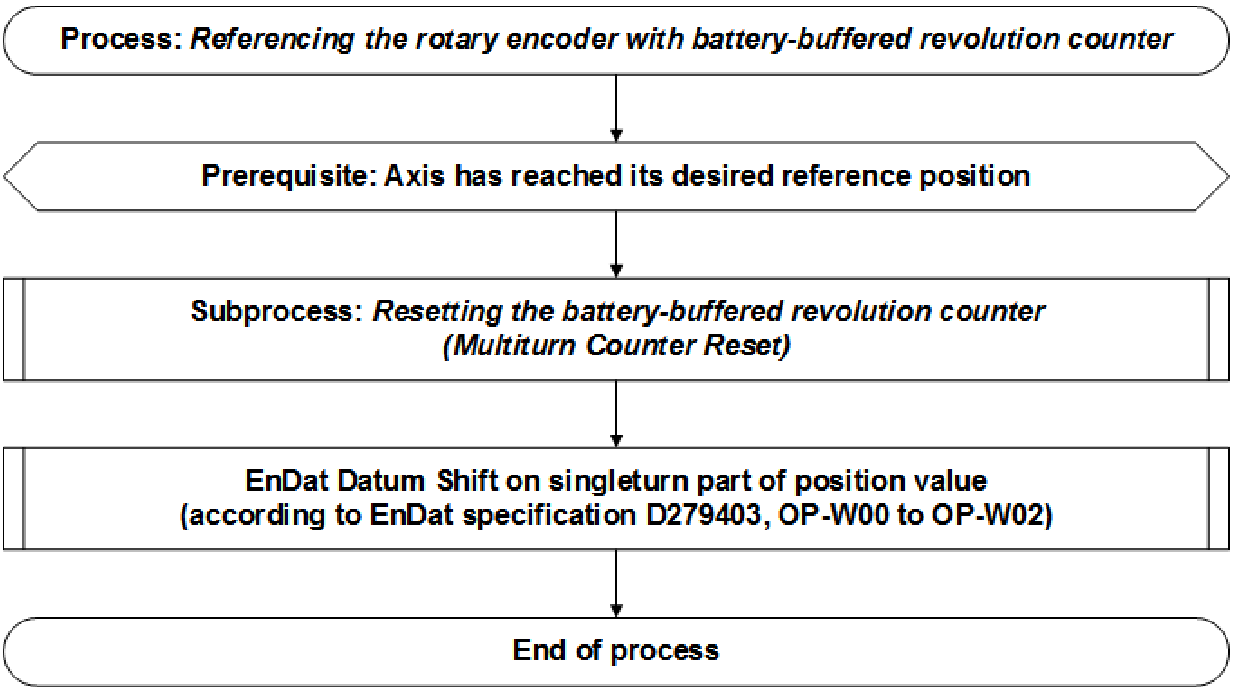
⚠ **A cyclic across-the-board deletion of warnings** (e.g. in closed-loop operation) **can result in damage to the encoder** (pay attention to the limited number of permissible write accesses to the integrated EEPROM; see also Chapter 2 *"Sequences and Data Structures"*, *"Clearing errors or warnings"* section).

13.6 Referencing of rotary encoders with battery-buffered revolution counter

The following process should be used for referencing battery-buffered rotary encoders:

- First, reset the revolution counter.
- Then, apply the EnDat datum shift to compensate for the remaining singleturn position deviations (to achieve a correct motor commutation).

⚠ **It is NOT RECOMMENDED to also apply the EnDat datum shift to the multiturn part of the position value,** as the overflow limits of the position value range are not moved with this (because these limits are connected to the internal revolution counter value of the ASIC).



Abbreviation:
OP-Wxx: Memory range 'Operating Parameters', word xx

Figure 3: Sequence of referencing process for encoders with battery-buffered revolution counters

For the datum-shift sequence, please refer to Chapter 10 *"EnDat Datum Shift"* and to the EnDat Specification D297403, section 3.6.2.

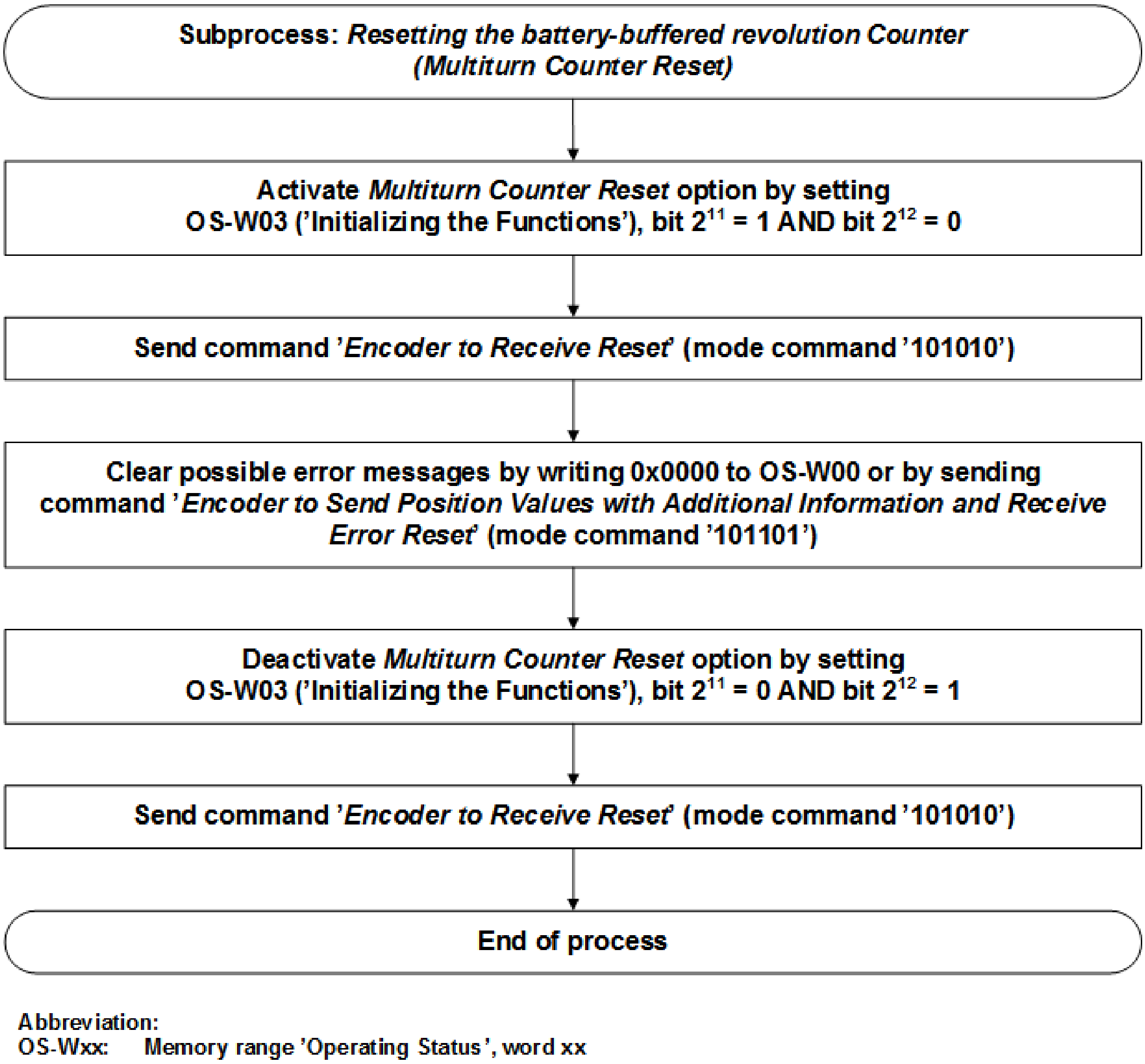


Figure 4: Sequence for *resetting the battery-buffered revolution counter* (*M Counter Reset*)

13.7 Exchanging the external buffer battery

To replace the external buffer battery, the encoder has to be in *normal operating mode* (U_P applied). If this is not the case, the revolution counter information is lost (signalized by the *M Power interruption* error message) and the axis has to be referenced again.
The *Battery charge* warning remains after the battery has been exchanged. It must be cleared by the operator after exchanging the buffer battery according to the EnDat specification (see EnDat specification D297403, chapter 3.4.2).

⚠ **A cyclic across-the-board deletion of warnings** (e.g. in closed-loop operation) **can result in damage to the encoder** (pay attention to the limited number of permissible write accesses to the integrated EEPROM; see also Chapter 2 *"Sequences and Data Structures"*, *"Clearing errors or warnings"* section).

The need to replace the battery must be indicated in plain text to the operator of the system.

14 Implementation Examples

The following describes two examples for implementing the EnDat interface. The first example shows a flexible solution, which permits the connection of very different EnDat encoders to a control. The second example shows the principle for minimum implementation. For the EIB 74x there is also a demo program that demonstrates the implementation of most EnDat features.
In both examples, use of the EnDat memory accesses in closed-loop operation are omitted as these accesses should only be used if they are specifically required via the application (also see Chapter 1). Thus from the possible EnDat commands only a limited number are used:

Example for flexible implementation:

		Mode bit					
	Mode command	**M2**	**M1**	**M0**	**(M2)**	**(M1)**	**(M0)**
Type 2.1 (as of EnDat version 2.1)	**Encoder send position values**	**0**	**0**	**0**	1	1	1
	Selection of memory area	**0**	**0**	**1**	1	1	0
	Encoder receive parameter	**0**	**1**	**1**	1	0	0
	Encoder send parameter	**1**	**0**	**0**	0	1	1
	Encoder receive reset	**1**	**0**	**1**	0	1	0
	~~**Encoder send test values**~~	~~**0**~~	~~**1**~~	~~**0**~~	~~1~~	~~0~~	~~1~~
	~~**Encoder receive test command**~~	~~**1**~~	~~**1**~~	~~**0**~~	~~0~~	~~0~~	~~1~~
Type 2.2 (as of EnDat version 2.2)	**Encoder send position values with Additional data**	**1**	**1**	**1**	0	0	0
	Encoder send position values and selection of the memory area	**0**	**0**	**1**	0	0	1
	~~**Encoder send position values and receive parameter**~~	~~**0**~~	~~**1**~~	~~**1**~~	~~0~~	~~1~~	~~1~~
	~~**Encoder send position values and send parameter**~~	~~**1**~~	~~**0**~~	~~**0**~~	~~1~~	~~0~~	~~0~~
	~~**Encoder send position values and receive test command**~~	~~**1**~~	~~**1**~~	~~**0**~~	~~1~~	~~1~~	~~0~~
	~~**Encoder send position values and receive error reset**~~	~~**1**~~	~~**0**~~	~~**1**~~	~~1~~	~~0~~	~~1~~
	~~**Encoder receive communication command**~~	~~**0**~~	~~**1**~~	~~**0**~~	~~0~~	~~1~~	~~0~~

1) Power-on:
- See the EnDat Specification, also FAQ section or EnDat seminar

2) Basic initialization:
- See “Recommended Procedure After Switch-On”
- Typical EnDat clock frequency: 200 ... 300 kHz

3) Read out EnDat 2.1 parameters of the encoder manufacturer:
- EnDat 2.1 mode commands are typically used for reading out the parameters.
- Typical EnDat clock frequency: 200 ... 300 kHz; due to the EEPROM access time of max. 12 ms the clock frequency is not of major importance
- To read out all EnDat 2.1 parameters, the following MRS codes and word addresses must be used:
  - MRS 0xA1; word 0x09 ... 0x0F
  - MRS 0xA3; word 0x00 ... 0x0F
  - MRS 0xA5; word 0x00 ... 0x0E

4) Read out operating status and operating parameters:

- For MRS codes and addresses, see the EnDat Specification.

5) Determine the basic communication parameters:

- Are EnDat 2.2 mode commands supported?
  See the EnDat Specification: EnDat 2.1 parameters, word 37
- Are 1 V_{PP} incremental signals supported or used?
  This can be determined, for example, by simultaneously checking the status (error bit) of an EnDat position request and the input level of the incremental signals.
  Please note:
  Before the EnDat position request, the EnDat error messages must be cleared.
  - For example, if the level of the incremental signals is greater than 0.25 V_{PP}, valid incremental signals are present and can be used for feedback control.
  - If the EnDat position request returns a valid position (i.e. error bit is not set) and the test of the incremental signals results in a value less than 0.25 V_{PP}, for example, then the encoder is operated in purely serial mode.
  - If both requests return invalid values, it is most likely that there is a malfunction in the encoder.

- In addition, word 40 in the EnDat 2.1 parameter range can be checked. The EnDat order designation is stored in this word.

This results in the following typical communication parameters:

Word 40	Are incremental signals supported?	Are EnDat 2.2 mode commands supported?	Typical EnDat clock frequency	Delay compensation
EnDat01	Yes	Depends on encoder	< 300 kHz	No
EnDat02	Yes	Yes	< 300 kHz	No
EnDat21	No	Depends on encoder	< 2 MHz	Yes
EnDat22	No	Yes	typ. 8 MHz	Yes

Also see “Maximum permissible clock frequency” in Chapter 1

6) Delay compensation:

- Required, depending on communication parameters; see above.
- Delay compensation is required if high transmission frequencies and long cable lengths are to be used. The EnDat master is responsible for delay compensation.
- For details, see the EnDat Specification as well as the specifications for the EnDat master used.

7) Read out EnDat 2.2 parameters:

- For the following steps, an EnDat clock frequency of 1 MHz is recommended.
- The MRS code 0xBD needs to be sent once to the encoder in order to be able to access the EnDat 2.2 parameters. The corresponding EnDat 2.2 mode command is required for this; the EnDat 2.1 mode command leads to an error message in this case. The EnDat clock frequency must be set so that it suits the setting for recovery time t_M; see the EnDat Specification.
- Besides EnDat 2.2 mode commands, EnDat 2.1 mode commands can be used to read out the parameters. Only the MRS code must be selected with the corresponding EnDat 2.2 mode command:
  - MRS code 0xBD; word 0x09 ... 0x3E

8) Set the recovery time t_M:

- The recovery time t_M can be set to two values: 10...30 µs or 1.25...3.75 µs.
  The factory default setting of 10...30 µs should only be reduced to 1.25…3.75 µs if this is necessary for maintaining the cycle time of the EnDat requests.
- The recovery time t_M can be changed in the operating status, word 3 “Function initialization”.
- Changing the setting only affects EnDat 2.2 mode commands, not EnDat 2.1 mode commands.
- Notes on / Recommendations for changing the recovery time t_M:
  Request to find out whether recovery time t_M can be reduced: EnDat 2.2 parameter, word 2
  - Read out the setting of t_M
  - Check whether the value for t_M matches the desired value
  - Change the setting if required and permissible
  It is not advisable to program a value without prior request, because the EEPROM is specified for 100 000 write accesses and unnecessary programming may cause problems, depending on the application.
- If the value for t_M is changed to 1.25…3.75 µs, the EnDat clock frequency must not be smaller than the minimum EnDat clock frequency of 1 MHz.

Note: Also see "What must be considered for the reduced recovery time?" in the FAQ section.

9) Configuration of online diagnostics:

- The diagnostics status can be read out from EnDat 2.2 parameter, word 5.
- All supported valuation numbers should be activated; see the EnDat Specification, operating parameters, word 3. However, you should first check whether all supported valuation numbers have already been activated.
- It is not advisable to program the value without a prior request, because the EEPROM is specified for 100 000 write accesses. Unnecessary programming could therefore cause problems, depending on the application.
- For details see the "EnDat diagnosis" application note.

10) Set the EnDat clock frequency for closed-loop operation:

- The maximum permissible EnDat clock frequency can be read out from EnDat 2.2 parameter, word 25.
- The EnDat clock frequency should be adapted to the application; see "Readout" or "Cycle Time".

Example for minimum implementation:

1) Set the EnDat clock frequency to 300 kHz

2) Reset the encoder
Send "Encoder receive reset" mode command to the encoder

3) Clear the error messages, if required (also see "Sequences and Data Structures“)
The following mode commands are necessary for this (all values given in hexadecimals)
- Selection of the memory area (MRS code 0xB9)
- Encoder receive parameter (address 0x00 and parameter 0x0000)
- Encoder receive reset

4) Clear the warnings, if required (also see "Sequences and Data Structures")
The following mode commands are necessary for this (all values given in hexadecimals)
- Selection of the memory area (MRS code 0xB9)
- Encoder receive parameter (address 0x01 and parameter 0x0000)
- Encoder receive reset

5) Check whether the expected encoder is actually connected. The test is based on the ID number of the encoder, in this example for the LIC 4000.
The following mode commands are necessary for this (all values given in hexadecimals)
- Selection of the memory area (MRS code 0xA3)
- Encoder send parameter (address 0x08); buffer the result
- Encoder send parameter (address 0x09); buffer the result
- Encoder send parameter (address 0x0A); buffer the result

It must now be checked whether the ID number matches the expectation. In this example (LIC with ID 651871-01) the memory content should be as follows:
- Address 0x08 ➔ Memory content 0x3031
- Address 0x09 ➔ Memory content 0xF25F
- Address 0x0A ➔ Memory content 0x0009

These contents correspond to the ID number above (see EnDat specification: EnDat 2.2 parameters of the encoder manufacturer, words 24 to 26)

6) The LIC 4000 permits EnDat clock frequencies up to 16 MHz; therefore:
- Activate propagation-delay compensation
- Set the desired EnDat transmission frequency

7) Cyclical reading out of the position can be started
Remark: Items 2) and 3) are an adaptation of “Procedure After Switch-On” (see Chapter 2) to the LIC 4000.

The example below shows a basic communication setup with a specific selection of encoders. The example is only available in English.

Basic communication with pure serial absolute linear, singleturn and gear-based multiturn encoders from HEIDENHAIN

Application:

- Only pure serial encoders with order designation: EnDat22 or EnDat21
- Only absolute encoders
    - Absolute linear
    - Absolute singleturn
    - Absolute multiturn, gear based

Related documents:

- EnDat specification
- EnDat Application note

The following setup is a subset of the data-flow shown in the “EnDat application note”, chapter 2.
For a basic communication only some certain parameters need to be read from the encoders memory (see also EnDat specification, chapter 3.5.1):

- Needed for EnDat master configuration:
    - Word 13: number of bits for position → EnDat Master configuration
    - Word 40: order designation
        - “3232”→ “EnDat22”: 8 MHz clock frequency
        - “any other” → 2 MHz
- Needed for position calculation:
    - Word 14: encoder model
    - Word 17: multiturn information
    - Word 20+21: measuring step / steps per revolution

This setup can be used for a basic communication with the encoder. Further features, like reading diagnosis information, … and other encoder types: incremental, battery buffered, .. can be added. Further information as well as other examples for an implementation can be found in the “related documents”, especially EnDat Application note: “how to get started” and “implementation examples”.

Remark
The EnDat µC demonstration code (see chapter “NOTES ON THE IMPLEMENTATION OF ENDAT MODE COMMANDS) using Arduino Mega ADK board also uses this basic communication.

The following table shows the parameter contents for some example encoders:

Parameter	LC 415	EQN 1337	ECI 1119
Word 13 (E21W13)	0x8024 ➔ nr. of bits: “36”	0x8025 ➔ nr. of bits: “37”	0x8013 ➔ nr. of bits: “19”
Word 14 (E21W14)	0x4001 ➔ “absolute linear”	0xE001 ➔ “absolute multiturn, gear based”	0xC001 ➔ “absolute singleturn”
Word 17 (E21(W17)	0x0000 ➔ not relevant for linear encoders	0x1000 ➔ upper 12 bit are multiturn information	0x0000 ➔ not relvant for singleturn encoder
Word 20 + 21 (E21W20,21)	0x0000000A ➔ measuring step: 10 nm	0x02000000 ➔ lower 25 bit are singleturn resolution	0x00080000 ➔ 19 bit singleturn resolution
Word 40 (E21W40)	0x3232 ➔ EnDat22 encoder; 8 MHz	0x3232 ➔ EnDat22 encoder; 8 MHz	0x3232 ➔ EnDat22 encoder; 8 MHz
position value	formula 5	formula 8	formula 7

Position value calculation, according to chapter “position data formats”

Formula 5:
Absolute position [m] = transferred position value * content of E21W20,21 * 1E-9

Formula 7:
Absolute angle value [°] = 360° * (transferred position value / content of E21W20,21)

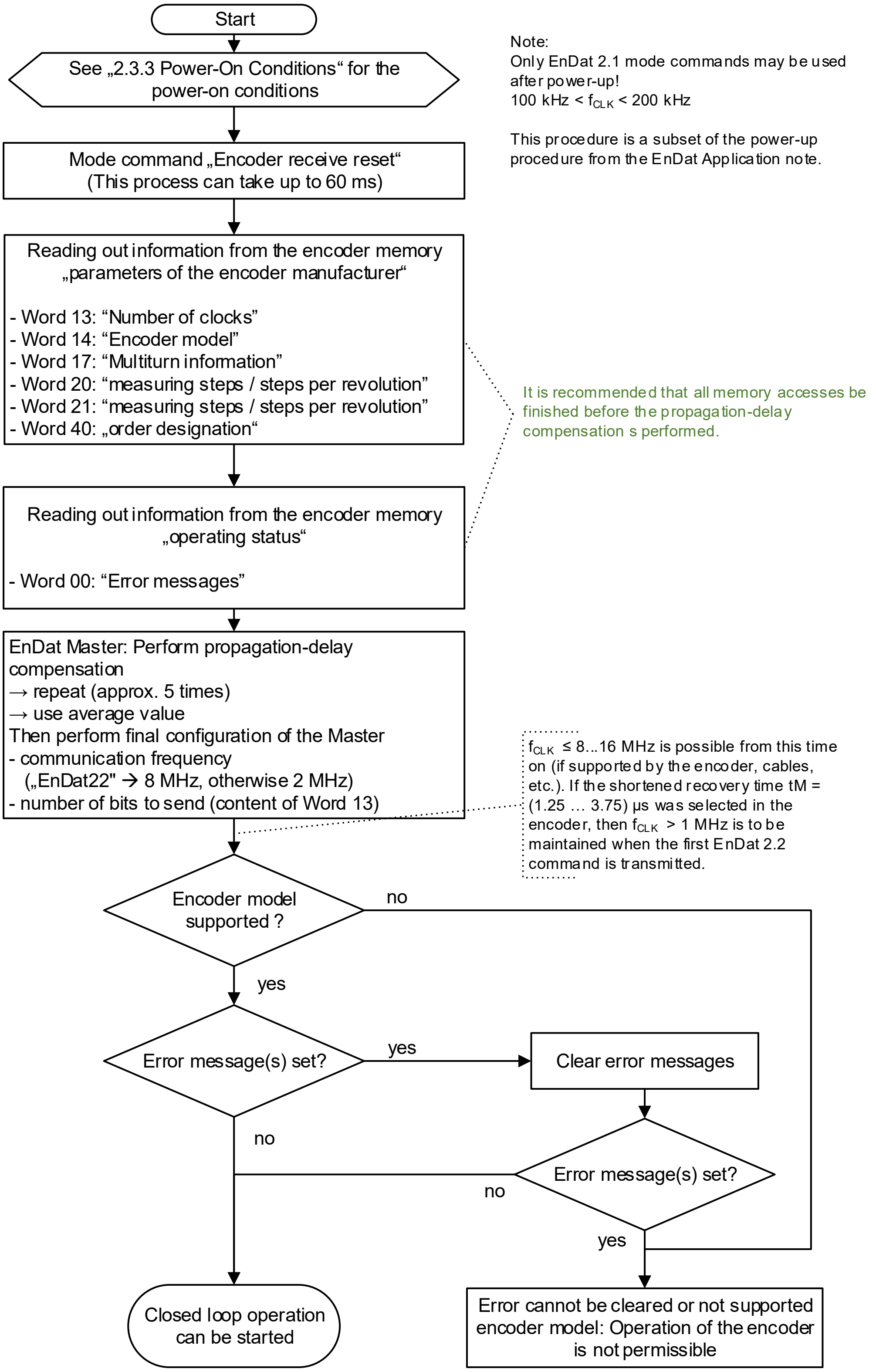
Formula 8:
The structure of the position value is according to the EnDat specification and the “Transfer format for position values” chapter.
The multiturn part must initially be extracted from the transferred position value. E21W17 or E22W34 is used for this. In practice only values corresponding to 2^x are used. Thus 2^M must be determined according to the depiction in the EnDat specification. M can be determined from parameter E21W17 or E22W34.

Multiturn part of the position: Depiction of 2^(S+M) to 2^(S+1);
Truncation of the singleturn part from the transferred position value and “push to right”

Singleturn part of the position: Depiction of 2^S to 2^1
“Zeroing” of the higher-value bits

Absolute angle value [°] = 360° * (content of E21W20,21 / singleturn part of the position)

Sample procedure: EnDat22 and EnDat21 pure serial absolute encoders:
- absolute singleturn encoders
- absolute gear-based multiturn encoders
- absolute linear encoders



The following sequence shows a report from the “EnDat Demotool Software” using a LC 415.
This sequence shows which mode commands need to be sent before closed loop operation can be started:

```
"Encoder to receive reset" transmitted!
mode command 52: "101 010"
Reset done!
(Rmrk.: Reset --> BOTH additional information 1 and 2 are now DESELECTED!)
```

Reading out parameter from "parameters of encoder manufacturer, MRS 0xA1, 0xA3, 0xA5 (see EnDat specification, chapter 3.5.1)"

```
"Selection of memory area" (A1 h Parameters of the encoder manufacturer) transmitted!
mode command 16: "001 110"
Memory area $A1 selected.
```

Word 0x0D "number of clocks" → configuration EnDat Master

```
"Encoder to send parameters" transmitted!
mode command 43: "100 011"
Parameter $8024 was read at address $0D.
```

Word 0x0E "encoder model"

```
"Encoder to send parameters" transmitted!
mode command 43: "100 011"
Parameter $4001 was read at address $0E.

"Selection of memory area" (A3 h Parameters of the encoder manufacturer) transmitted!
mode command 16: "001 110"
Memory area $A3 selected.
```

Word 0x01 "multiturn information"

```
"Encoder to send parameters" transmitted!
mode command 43: "100 011"
Parameter $0000 was read at address $01.
```

Word 0x04 "measuring steps / steps per revolution, lower 16 bit"

```
"Encoder to send parameters" transmitted!
mode command 43: "100 011"
Parameter $000A was read at address $04.
```

Word 0x04 "measuring steps / steps per revolution, upper 16 bit"

```
"Encoder to send parameters" transmitted!
mode command 43: "100 011"
Parameter $0000 was read at address $05.

"Selection of memory area" (A5 h Parameters of the encoder manufacturer) transmitted!
mode command 16: "001 110"
Memory area $A5 selected.
```

Word 0x08 "order designation"

```
"Encoder to send parameters" transmitted!
mode command 43: "100 011"
Parameter $3232 was read at address $08.
```

Read content of error register

```
"Selection of memory area" (B9 h Operating condition) transmitted!
mode command 16: "001 110"
Memory area $B9 selected.

"Encoder to send parameters" transmitted!
mode command 43: "100 011"
Parameter $0000 was read at address $00.
```

Only in case if errors are set → delete errors

```
"Selection of memory area" (B9 h Operating condition) transmitted!
mode command 16: "001 110"
Memory area $B9 selected.

"Encoder to receive parameters" transmitted!
mode command 34: "011 100"
Parameter $0000 was written at address $00.

"Encoder to receive reset" transmitted!
mode command 52: "101 010"
Reset done!
(Rmrk.: Reset --> BOTH additional information 1 and 2 are now DESELECTED!)
```

For testing the processes above, development boards or demo boards from various manufacturers or the EIB 74x can be used.
For more information about FPGA development boards, see the “News” section on the website.

For more information, see the Web site at “EnDat Master.”

Reference design for the connection of EnDat 2.2 encoders based on components from Texas Instruments (TIDA-00172): http://www.ti.com/tool/TIDA-00172

Implementation of EnDat 2.2 based on processors from Texas Instruments:
http://www.ti.com/lsds/ti/apps/automation/industrial_communication/overview.page

15 Notes on the Implementation of EnDat Mode Commands

General information:
The EnDat specifications provide a description of the timing diagrams of the EnDat mode commands. This section also provides information on the following topics:

- Basic structure of EnDat communication
- Compensation of cable propagation time
- Transmission supplement for EnDat 2.2 mode commands
- EnDat 2.1 and 2.2 position commands
- Polling of start bit
- EnDat master

Also, the functioning of the sequential control of the EnDat master basic is described in a flowchart, and the mode commands are depicted as structural blocks.

The EnDat master can be integrated by means of a micro controller (µC) or an FPGA (Field Programmable Gate Array) or ASIC. The solutions with µC are used when the desired clock frequencies are relatively low (i.e. there is no compensation of the propagation time). Integration in an FPGA or ASIC is chosen primarily for high transmission frequencies with purely serial data transfer.

For the integration in a µC, HEIDENHAIN can provide an example code for the implementation of the EnDat 2.1 mode commands:

- Integration based on an Arduino Mega ADK board (see below)
- Integration based on an 80C165, for more information, please refer to your contact person at HEIDENHAIN.

Basic structure of EnDat communication
The same data structures with fixed bit widths exist for the various mode commands. These data structures are then assembled based on the respective mode command. The data direction must also be considered for this.
A distinction is made between the following data structures:

- Mode command
- Command supplement
- Additional data
- Position values
- Acknowledgment of command supplement

Mode command:

M2	M1	M0	(M2)	(M1)	(M0)

Command supplement:

Clock ①	0	...	7	8	...	23
Data	**Address**			**Data**		
	MSB	...	LSB	MSB	...	LSB

① Clock pulse period with respect to the command supplement

Additional data:

<table>
<tr><td>Clock ①</td><td>0</td><td colspan="4">...</td><td>7</td><td>8</td><td>...</td><td>15</td><td>16</td><td>...</td><td>23</td></tr>
<tr><td rowspan="3">Data</td><td colspan="6">Status & information ②
Byte 0</td><td colspan="3">Address or data
Byte 1</td><td colspan="3">Data
Byte 2</td></tr>
<tr><td>WRN</td><td>RM</td><td>Busy</td><td>I4</td><td>...</td><td>I0</td><td>MSB</td><td>...</td><td>LSB</td><td>MSB</td><td>...</td><td>LSB</td></tr>
<tr><td colspan="3">Status</td><td colspan="3">Information</td><td>A7/D7</td><td></td><td>A0/D0</td><td>D7</td><td></td><td>D0</td></tr>
</table>

① Clock pulse period with respect to the additional data
② I0 .. I4 contain the number of the transmitted additional data

Transfer format for position values

Clock ①	0	1	2 ②	3 .. (3+position width)	(4 .. 9)+position width
Data	Start	F1	F2	Position (LSB first)	CRC

① Clock pulse period with respect to the position information
② Is only transmitted for the EnDat 2.2 position command

The number of clock pulses required for transferring the position value (without mode, start, alarm or CRC bits) is variable and depends on the encoder. The number of clock pulses must be read out from the encoder manufacturer's memory area. The subsequent electronics must support a **data width of the position value** of 48 bits. The structure of the position information depends on the encoder model. See the EnDat Specification.

Acknowledgment of command supplement:

<table>
<tr><td>Clock ①</td><td>0</td><td>...</td><td>7</td><td>8</td><td>...</td><td>23</td></tr>
<tr><td rowspan="2">Data</td><td colspan="3">Address</td><td colspan="3">Data</td></tr>
<tr><td>MSB</td><td>...</td><td>LSB</td><td>MSB</td><td>...</td><td>LSB</td></tr>
</table>

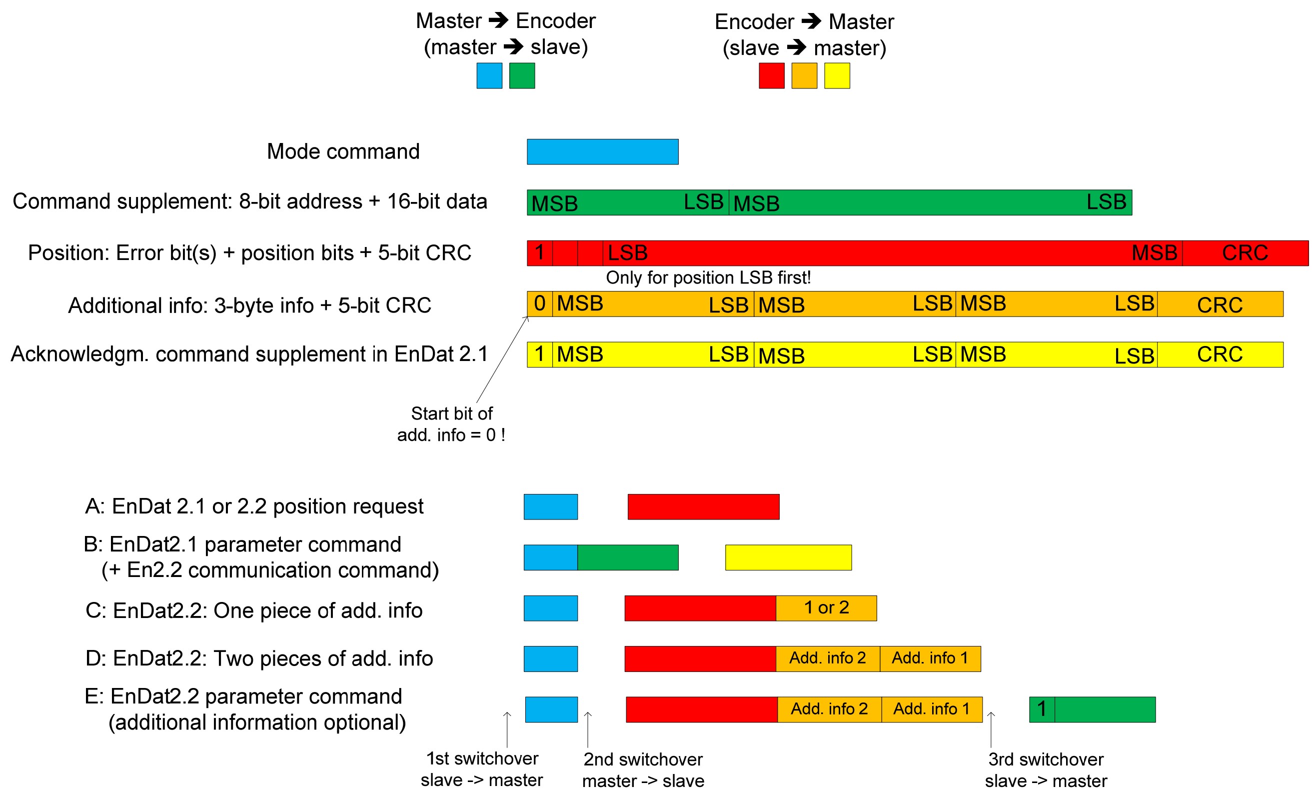
① Clock pulse period with respect to the acknowledgement of command supplement

As the result of these data structures there are five different basic structures for the EnDat mode commands (A .. E).

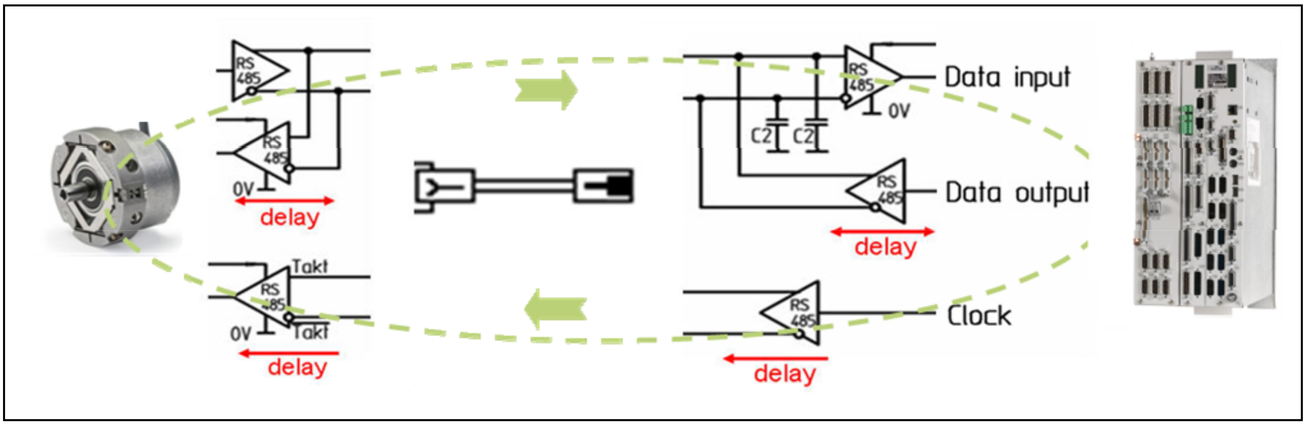
Mode command	**Structure**
Encoder send position values	A
Encoder send position values with additional data ①	A, C, E
Selection of memory area	B
Encoder send position values with additional data and selection of memory area or of the additional data	E
Encoder send parameter	B
Encoder send position values with additional data and send parameter	E
Encoder receive parameter	B
Encoder send position values with additional data and receive parameter	E
Encoder receive reset	B
Encoder send position values with additional data and receive error reset	E
Encoder receive test command	B
Encoder send position values with additional data and receive test command	E
Encoder send test values	B
Encoder receive communication command	B

1)The structure depends on the additional data selected.

This results in the following structure for the mode commands:



Compensation of cable propagation time:
The high transmission frequency and the cable length up to max. 100 m result in propagation time differences between the clock and the data.



The signal propagation time is calculated from 4 x Delay_Transceiver + 2 x cable length x cable propagation time.

However, the EnDat master needs to establish the relationship between the clock and the associated data. The cable propagation time must therefore be compensated. Compensation is mainly required for the following actions:

- Clocking-in data (position, additional data, ..) in the receive register
- Maintaining the timing for the transmission supplement for the EnDat 2.2 mode commands "Encoder send position value with additional data and ..."

The compensation of cable propagation time must therefore serve two purposes:

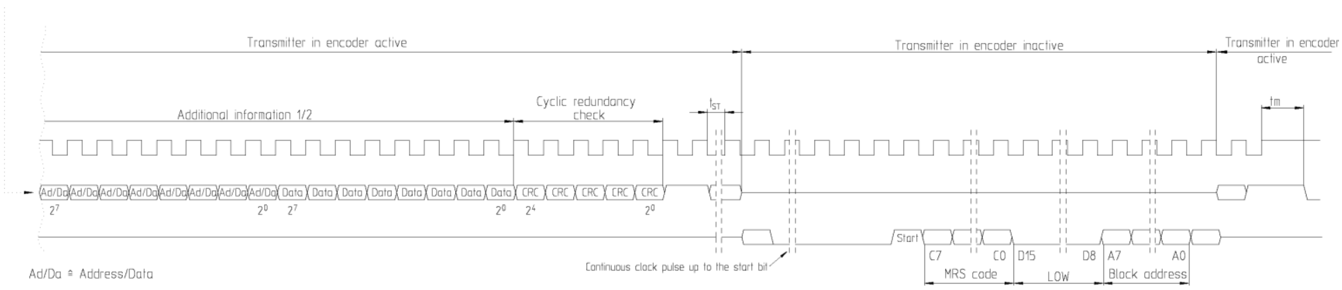
a. Moving the internal data-transfer edge of the receive register of the EnDat master to the middle of the associated data bit. Especially for large cable lengths, the data is to be accepted in the middle of the data bit in order to attain the highest possible noise immunity and quality of the data transfer.
b. In order to ensure a proper timing of the transmission supplement, it must be known how many clock pulses are in the data line without the master having received the associated data bits.

Two basic methods can primarily be used for compensating the cable propagation time:

a. Measuring the propagation time and deriving the information described above.
b. Measurement of propagation time for deriving the number of clock pulses on the line and detection of the rising edge of the start bit (including the subsequent movement of the internal data transfer edge)

Note on b):
With detection of the rising edge of the start bit it must be considered that the low level applied by the encoder before sending the start bit is delayed due to the cable propagation time. This means that a waiting time must be added before the detection of the start bit. The waiting time is either 2 µs (worst case for 100 m cable) as a default, or it is derived from the measurement of the cable propagation time (including the corresponding safety margin).

Transmission supplement for EnDat 2.2 mode commands:
This transmission supplement is necessary for the EnDat mode commands "Encoder send position value with additional data …." and makes it possible to select the MRS code, the addresses and parameters in a closed control loop.



A change in the direction of communication is critical for the timing. If such a change takes place, the delay time between the clock and the data must be taken into account; this means that several clock cycles may be in the data line without the associated information having reached the master. It must therefore be known how many clock pulses are already in the data line.

EnDat 2.1 and 2.2 position commands:
When transmitting an EnDat 2.1 position command, you must ensure that the encoder is not set unintentionally to "continuous clock" mode due to the transmission of too many clock pulses (interruption of recovery time $_M$).
The "continuous clock" mode is not available for EnDat 2.2 position commands. It is therefore no problem to send too many clock pulses to the encoder. The encoder reacts in such a way that recovery time t_M is restarted with every clock pulse.

Polling of start bit:
During polling of the start bit, the clock must be sent continuously to the encoder, also see "FAQ" at www.endat.de.

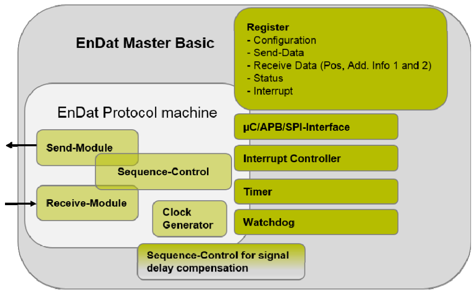
EnDat master:

The EnDat master is responsible for communication with EnDat encoders from HEIDENHAIN. It allows simple transmission of position data and additional data to the higher-level application. The EnDat master can be integrated by means of a micro controller (µC) or an FPGA (Field Programmable Gate Array) or ASIC.

The µC solutions are used if the intended clock frequencies are relatively low. For µC integration, HEIDENHAIN offers an example code for the implementation of the EnDat 2.1 mode commands; for more information, please refer to your contact person at HEIDENHAIN.

Integration in an FPGA or ASIC is chosen primarily for high transmission frequencies with purely serial data transfer. Different variants are available for integration in an FPGA or ASIC; the following graphic shows the block diagram for the EnDat master variants:

- EnDat master basic
- EnDat master reduced (only EnDat protocol machine)
- EnDat master light (only EnDat protocol machine)



For more information, see www.endat.de

Especially the "reduced" and "light" EnDat master variants show the implementation of the EnDat mode commands. In this case, functions—such as compensation of cable propagation time—are stored in the VHDL source code.

Excerpt from the documentation for the EnDat master basic:

The send register contains data to be transmitted to the EnDat encoder:

- Mode command
- MRS code/address/port address (depends on the mode command)
- Parameters/instructions (depends on the mode command)

Address [hex]	3	2	1	**0**
	Byte 4 Bit [29:24]	Byte 3 Bit [23:16]	Byte 2 Bit [15:8]	Byte 1 Bit [7 :0]
	Mode bits M [5 :0]	MRS code / Address / Port address A [7 :0]	Parameters / Instructions D [15:8]	Parameters / Instructions D [7:0]

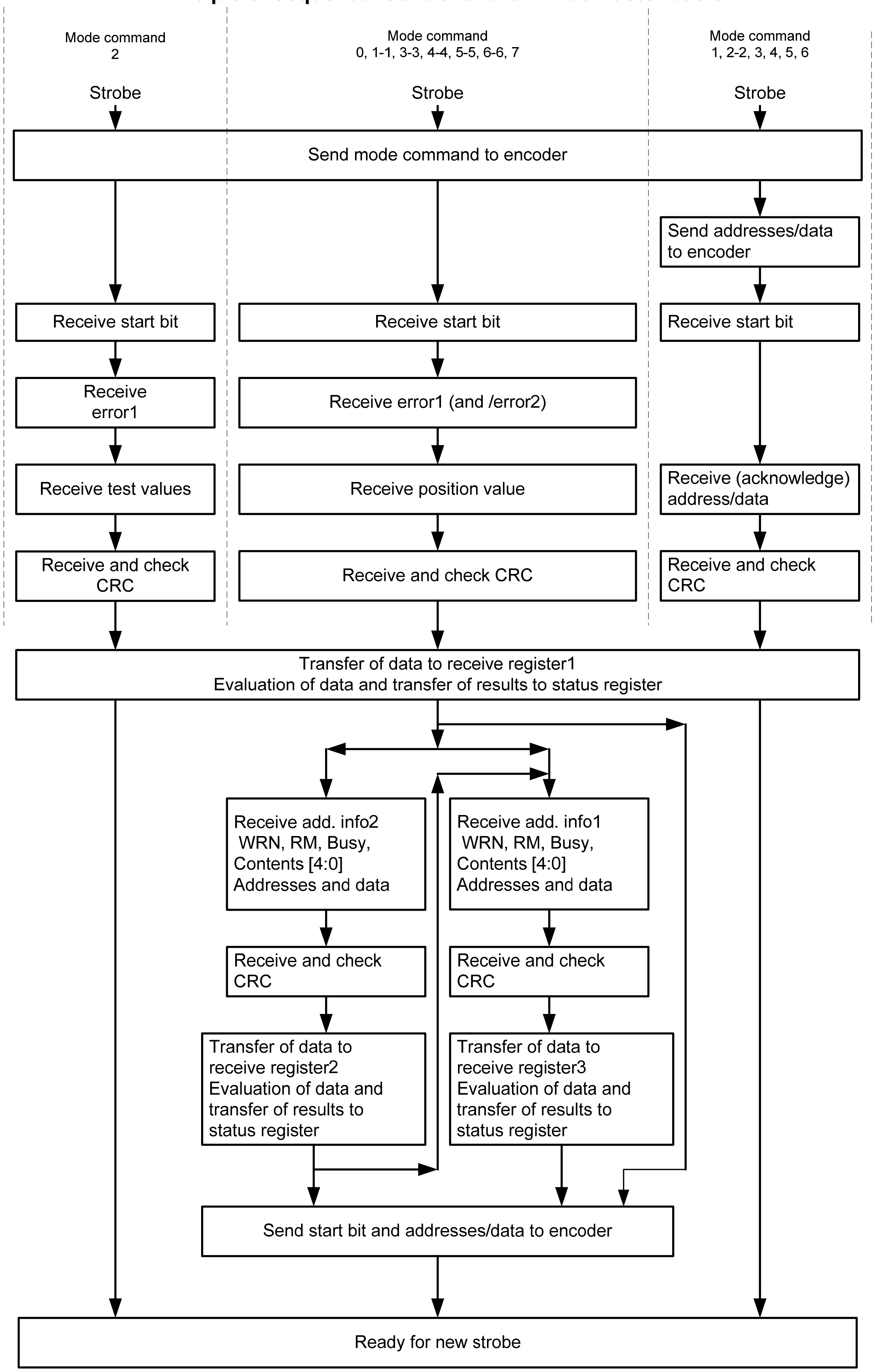
Overview of the supported mode commands with the data to be entered into the send register.

Mode command	Mode command description	Byte 4 Bit [29:24]	Byte 3 Bit [23:16]	Bytes 1 & 2 Bit [15:0]
0	Encoder send position values	07	---	---
7	Encoder send position values with additional data	38	---	---
1	Selection of memory area	0E	C(7:0)	xxx
3	Encoder receive parameter	1C	A(7:0)	Parameter
4	Encoder send parameter	23	A(7:0)	xxx
5	Encoder receive reset	2A	A(7:0)	xxx
2	Encoder send test values	15	---	
2-2	Encoder receive communication command	12	A(7:0)	Instruction
6	Encoder receive test command	31	P(7:0)	xxx
1-1	Encoder send position values with additional data and selection of the memory area	09	C(7:0)	xxx
3-3	Encoder send position values with additional data and receive parameter	1B	A(7:0)	Parameter
4-4	Encoder send position values with additional data and send parameter	24	A(7:0)	xxx
5-5	Encoder send position values with additional data and receive error reset	2D	A(7:0)	xxx
6-6	Encoder send position values with additional data and receive test command	36	P(7:0)	xxx

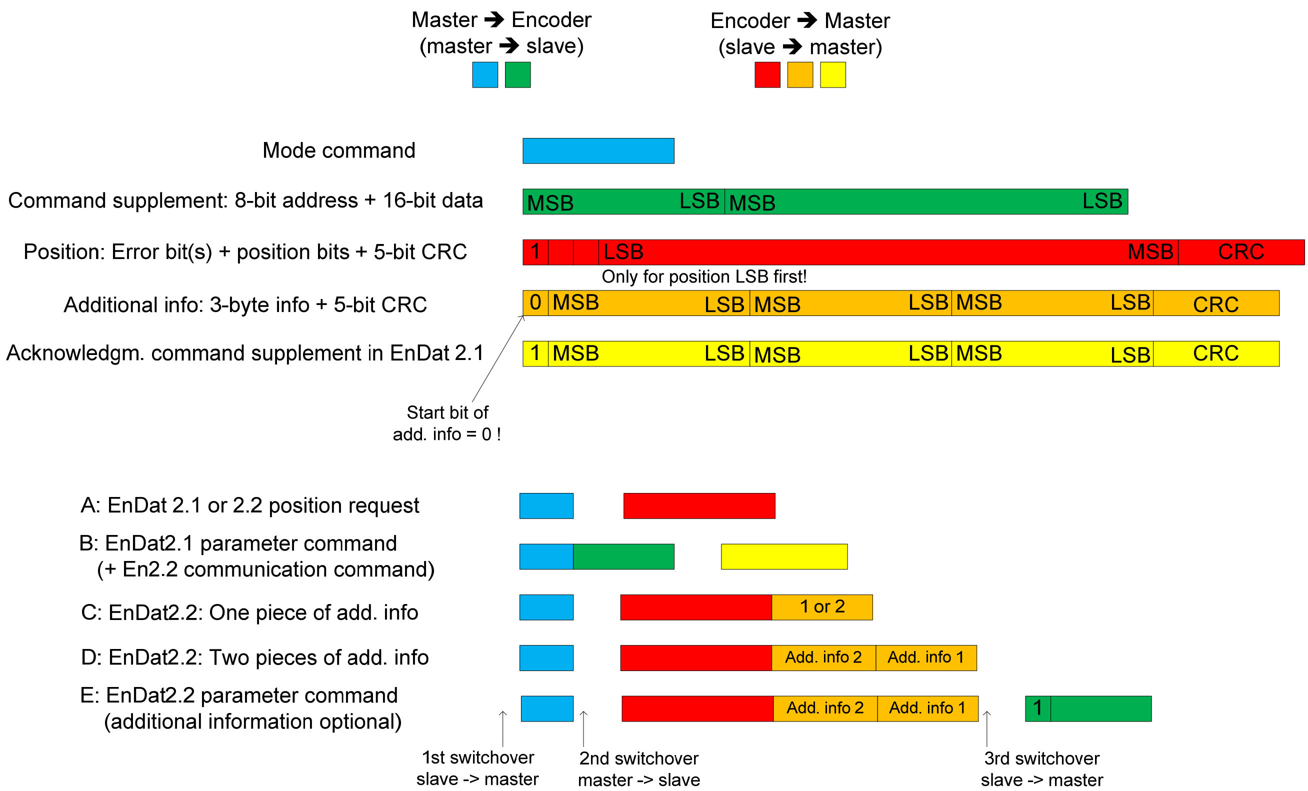
MRS code: C(7:0), addresses: A(7:0), port addresses: P(7:0), xxx: free

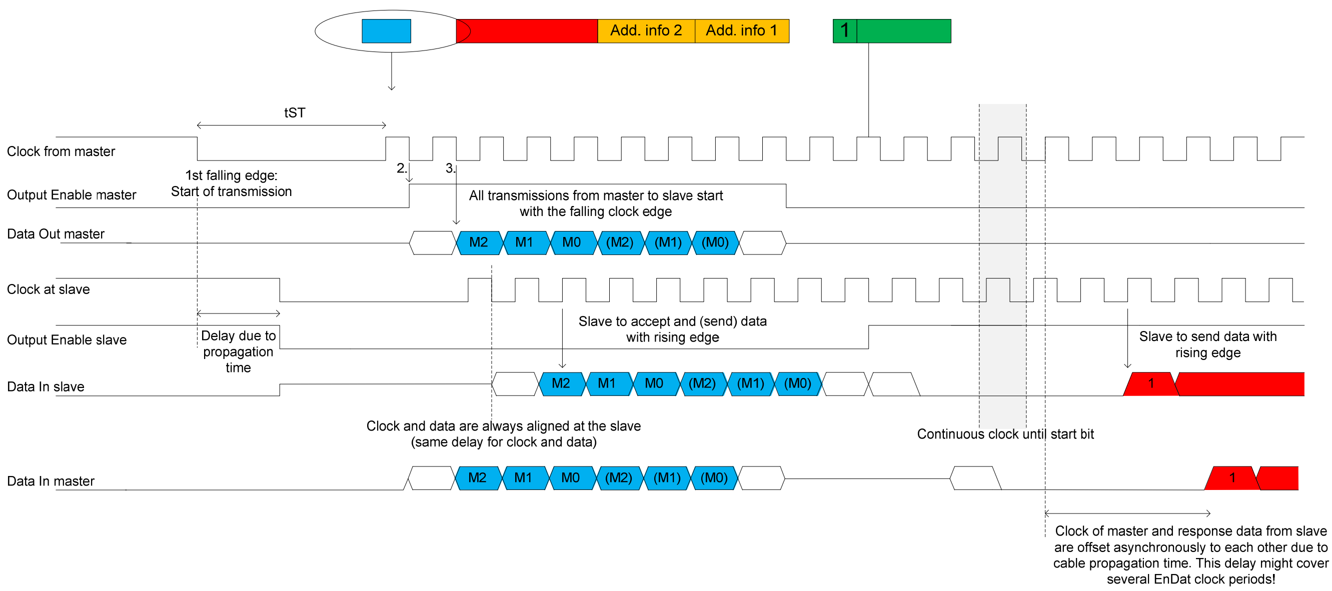
The following page describes the principle of sequential control for execution of the mode commands

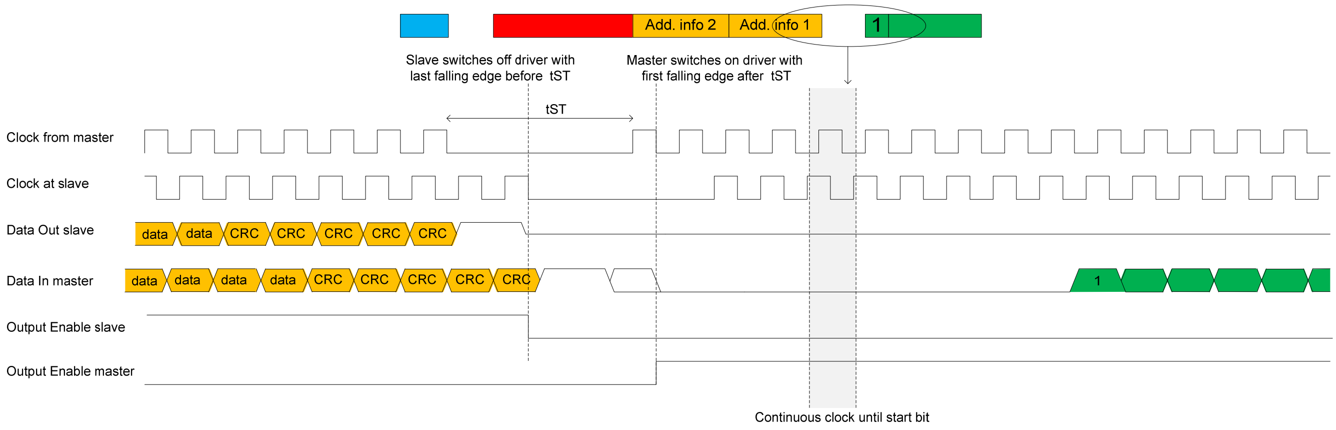
Principle of sequential control of the EnDat master basic

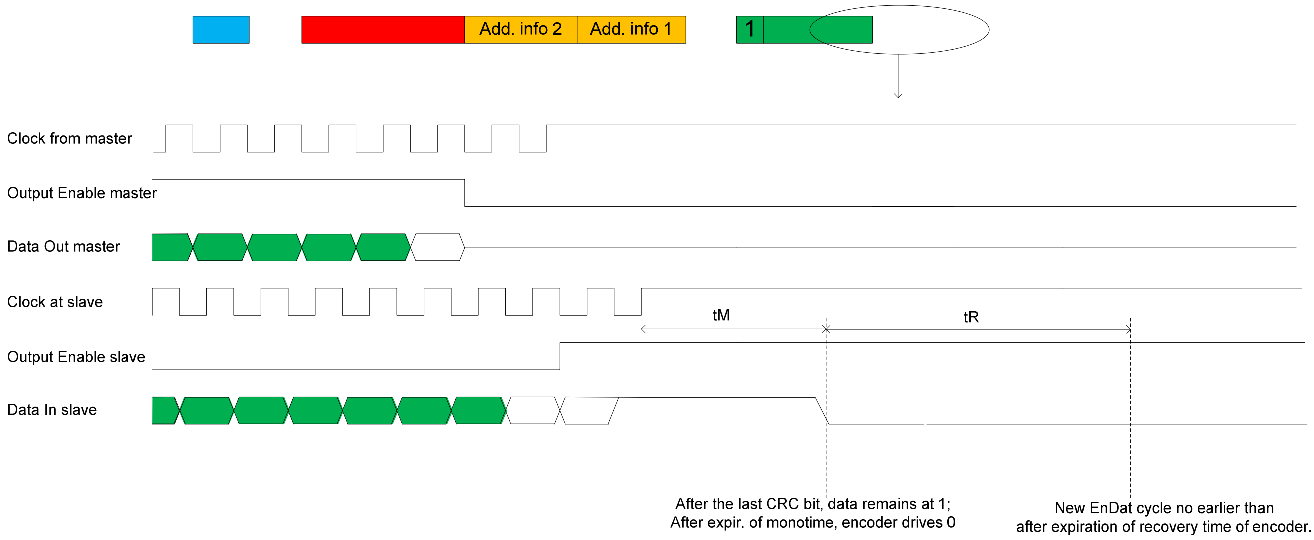


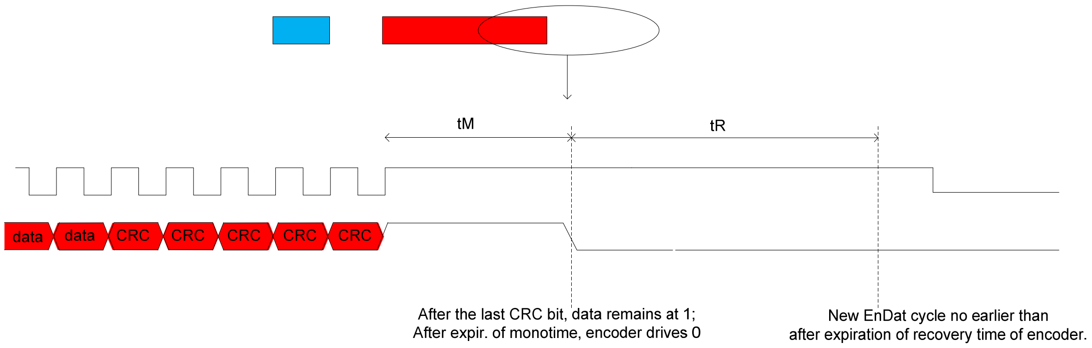
Depiction of mode commands as structural blocks:



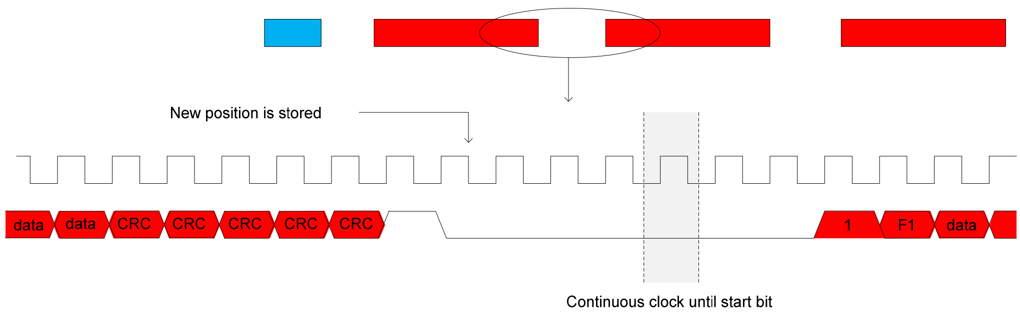








Continuous clock (only with EnDat21 position command)

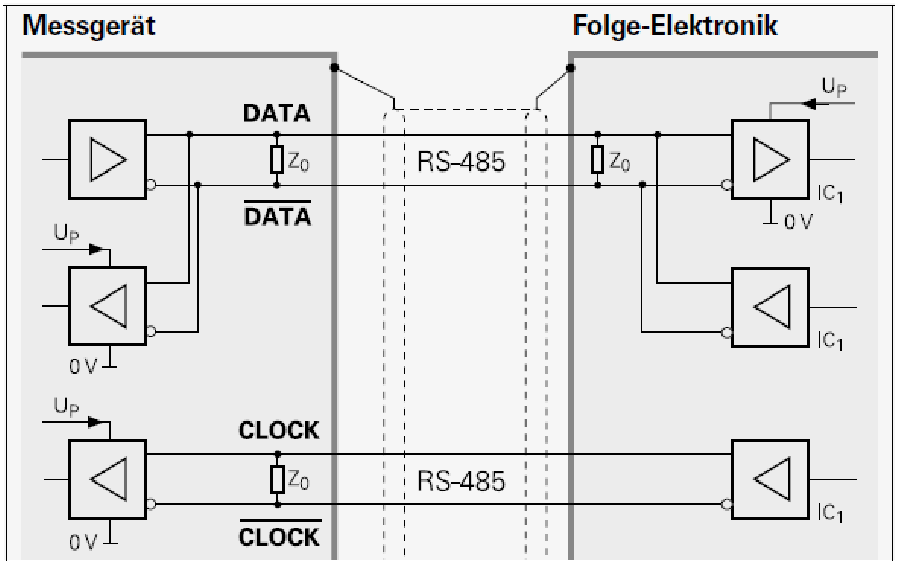


Implementation of EnDat communication with µC

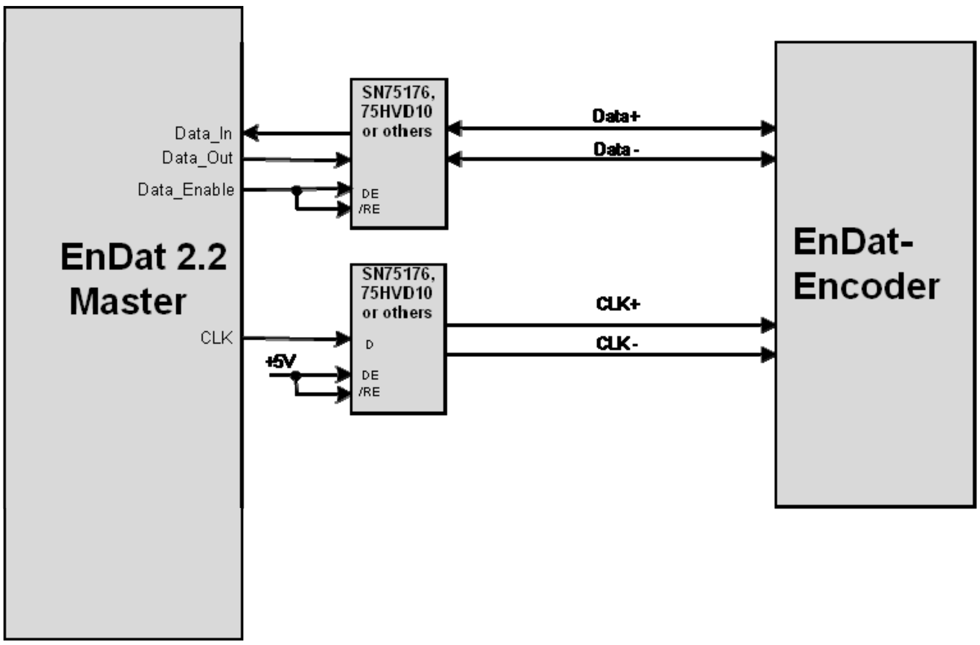
Constraints:

- A relatively low clock frequency is used. For cable lengths of up to 150 m, the propagation time thus does not have to be considered.
  The following constraints should be complied with:
  - High phase of clock pulse: 2 µs < t < 10 µs
  - Low phase of clock pulse: 2 µs < t < 1 ms
- Clock line to “high” for > 30 µs after every transmission

Physical layout:



Control lines to the µC:



Example code of EnDat communication with a µC

Mode commands:

- Only the EnDat 2.1 commands without
  - o Encoder send test values
  - o Encoder receive test command

  will be implemented.

Supported encoder types:

The example code supports the following types of encoders:

- Absolute singleturn encoders
- Absolute multiturn encoders with gear-based multiturn
- Absolute linear encoders

Background: The other encoder types require the EnDat 2.2 functionality, which is not part of the example code.

Regarding the sequence of the EnDat communication, the example code is based on the implementation example “Basic communication with purely serial absolute linear single and multiturn encoders” in the EnDat Application Note.

Hardware platform:

An Arduino Mega ADK board was used for the implementation.

Basic communication:

- Step 1: Send mode command or mode command + address + data
- Step 2: Poll to start bit
- Step 3: Load encoder data
- Load 4: Load CRC
- Step 5: Conclude communication: Set CLK line to “high” (for > 30 µs)

Thus, the entire communication can be set up with one main routine.
This routine is called directly for the position transfer. For the other mode commands, the address and data have to be preprocessed (sorting the data in the order of sending). Furthermore, the error type II is determined after the transmission.
To optimize the speed, it may be useful to further subdivide the individual parts. This depends very much on the processor being used.

Reading out parameters:

The following parameters are read out of the encoder:

- EnDat 2.1 parameter, word 13:
  Number of clock pulses for position value transfer
- EnDat 2.1 parameter, word 14:
  Type of encoder
- EnDat 2.1 parameter, words 20 + 21:
  Measuring step or measuring steps per revolution
- EnDat 2.1 parameter, word 17:
  Number of distinguishable revolutions

After the basic initialization, the encoder is checked for any logged errors. If an error is pending, a single attempt will be made to delete it.

Initialization:
The following initialization is performed after switch-on:

- Waiting time: 2 s
  Background: Booting time of the EnDat encoder up to 1.3 s
- CLK line: Level from high → low → high + waiting time > 30 µs
  This is a hardware reset for the encoder.
- Switch off the data driver in the control

Cyclic redundancy check:
A CRC is to be conducted, see EnDat Specification, Chapter 4.4.2.
In the example code, the CRC calculation is carried out upon loading of each bit.
However, the calculation can also take place after concluding the EnDat communication. In this case, it must be ensured, however, that the input data (Data_In) pass through the CRC algorithm in the correct order (i.e. in the same direction as during the data import):

- Position: LSB first
- Address + data: MSB first

Note:
There are alternative possibilities regarding the implementation of the CRC algorithm. Consult your contact person at HEIDENHAIN if required.

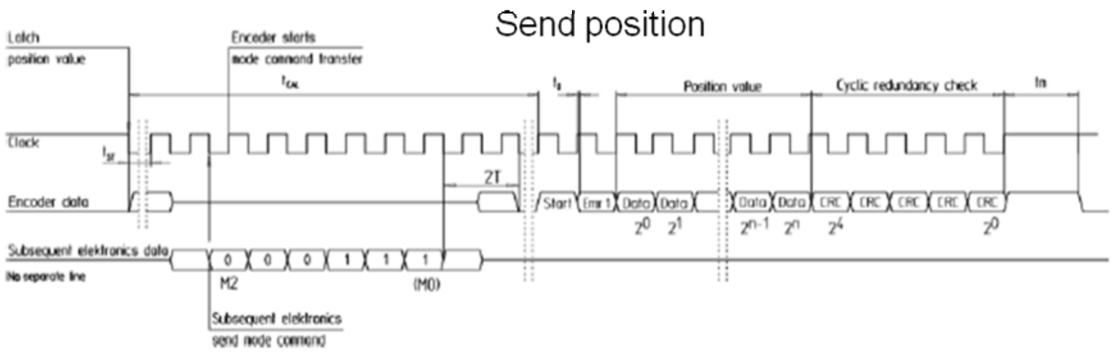
Type I and II error handling:
Error handling in accordance with the EnDat Specification, Chapter 4.2, is to be performed.

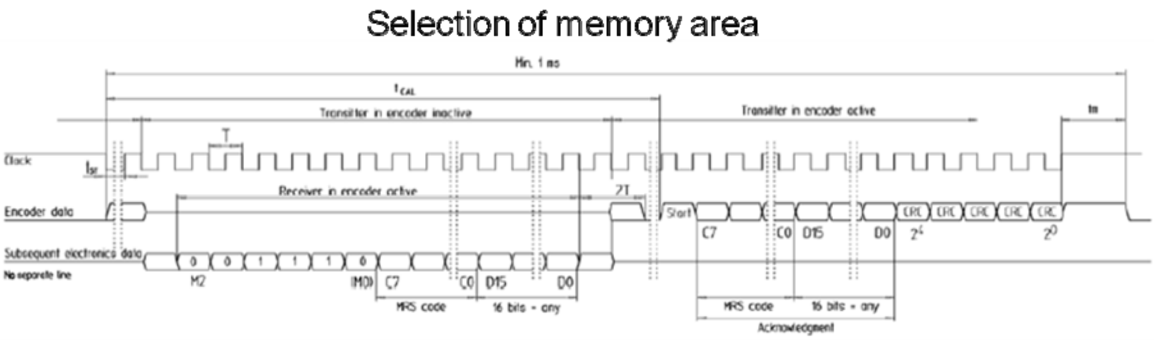
- Error type I:
  - The state of the data line depends on whether pull-up or pull-down resistors are being used on the data line and on the preferred condition of the transceiver at "high-impedance" state.
  - Error type I is not evaluated explicitly for the example code. If a type I error occurs, a CRC error is displayed.
- Error type II:
  - Evaluation of the transmitted and returned address
- Error type III:
  - No evaluation in the example code because the EnDat 2.2 functionality is not supported

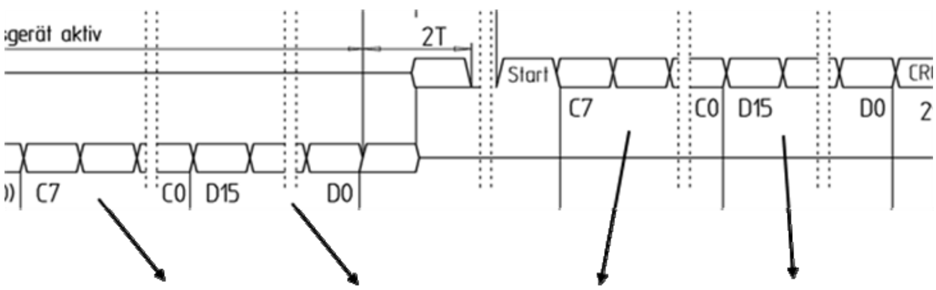
Advanced error handling:
An advanced error handling, e.g. watchdog for EnDat communication, is not performed in the example code. The expansion is to be made depending on the application.

Details on the EnDat communication:

Basic structure of the supported mode commands:







Mode command	C7 .. C0 (or A7 .. A0)	D15 .. D0	C7 .. C0 (or A7 .. A0)	D15 .. D0
Selection of memory area	MRS	Any	MRS (acknowledgment)	Any
Encoder receive parameter	Address	Parameter	Address (acknowledgment)	Parameter (acknowledgment)
Encoder send parameter	Address	Any	Address (acknowledgment)	Parameter
Encoder receive reset	Any	Any	Any	Any

Shift register for Data_In and Data_Out

In the simplest case, separate shift registers (= variables) are used. However, it is also possible to use a shift register (= variable) for both data directions.

The following configuration is used in the example program:

- Data_Out:
  - Data width: 32 bits
  - The LSB is sent and then a right-shift is made (this means that the direction of the MRS and the address and data needs to be “reversed”), see also “Data_Out structure” table: Adjusting the orientation of data and address.
- Data_In:
  - EnDat position: max. 48 bits → 64-bit variable is used.
  - 64-bit writing and shifting operations are usually relatively time-consuming, which is why a 16-bit array with three entries is used. At the end of the EnDat transmission, a 64-bit position value is formed from it.
  - Position command: Data_In is written to the MSB and then shifted to the right.
  - Further mode commands: Data_In is written to the LSB and then shifted to the left.

Structure of Data_Out:

Mode Command	**Data_Out (right-aligned)**
Encoder to Send Position Values (EnDat 2.1)	**"111000"**
Selection of Memory Area	**„Data[0:15]"+“MRS[0:7]“+“011100“**
Encoder to Send Parameters	**„Data[0:15]"+“Adresse[0:7]“+“110001“**
Encoder to Receive Parameters	**"Data[0:15]"+“Adresse[0:7]“+“001110“**
Encoder to Receive Reset	**"Data[0:15]"+“Adresse[0:7]“+“010101“**

EnDat communication:

Step 1:
Data of Data_Out are sent to the encoder:

- Clock cycles 1 and 2: Switchover of data direction
- Output of data from Data_Out:
  - Position mode command: Six clock cycles for mode command
  - Other mode commands: 30 clock cycles for mode command + address + data
- The following two clock cycles: Switchover of the data direction
- Set CLK line to low level, waiting time > 2 µs

The waiting time of 2 µs is necessary to ensure a stable low level on the data line even in a worst-case scenario. This is important because the next step involves polling to the start bit. For this purpose, the state of the data line is read in after the “1” → “0” transition of the clock line, and the start bit is thus detected.

Step 2:
Detection of start bit

Step 3:
Importing the data transmitted by the encoder
Position mode command: Number of clock pulses: see EnDat 2.1 parameter, word 13.
Other mode commands: Address + data, i.e. 24 bits have to be read in.
The data are imported in groups of 3 x 16-bit variables.

Step 4:
Importing CRC

Step 5:
Conclusion of communication.
The CLK line must be at high level for at least 30 µs.

Example code:
Please get in touch with your contact person at HEIDENHAIN to obtain the example code.

16 Recommendations for the EnDat Implementation Test

General information:

Depending on the selected type of implementation, testing with different encoders, different cable lengths and different clock frequencies are recommended. In general, all required information about timing, etc. is provided in the EnDat specifications. This Application Note contains general information as well as detailed information on the topic of testing the implementation.
If one of the recommended EnDat masters is used, then all test points which relate to the timing of the EnDat transmission or the run-up behavior can be omitted.

For information on testing the EnDat implementation, see Appendix 1.
Note: Only available in English

Further supplementary information:

Supported encoder models and position data formats:

- Encoder models:
  - See EnDat Specification, EnDat 2.1 parameter word 14
  - See also Chapter 1 “Supported encoder models”
  - See also Chapters 11 and 13
- Position data formats:
  - See also Chapter 4

Encoder functions:

- See Chapter 1 “Encoder functions”
- See Chapters 8, 9, 10, 12

Monitoring functions:

- See Chapter 7
- See Chapter 17

For information on measurements of the power-up phase and the EnDat transfer lines Clock and Data, see Appendix 2.

For information on which encoders should be used for the test, see Appendix A3.

16.1 EnDat implementation test

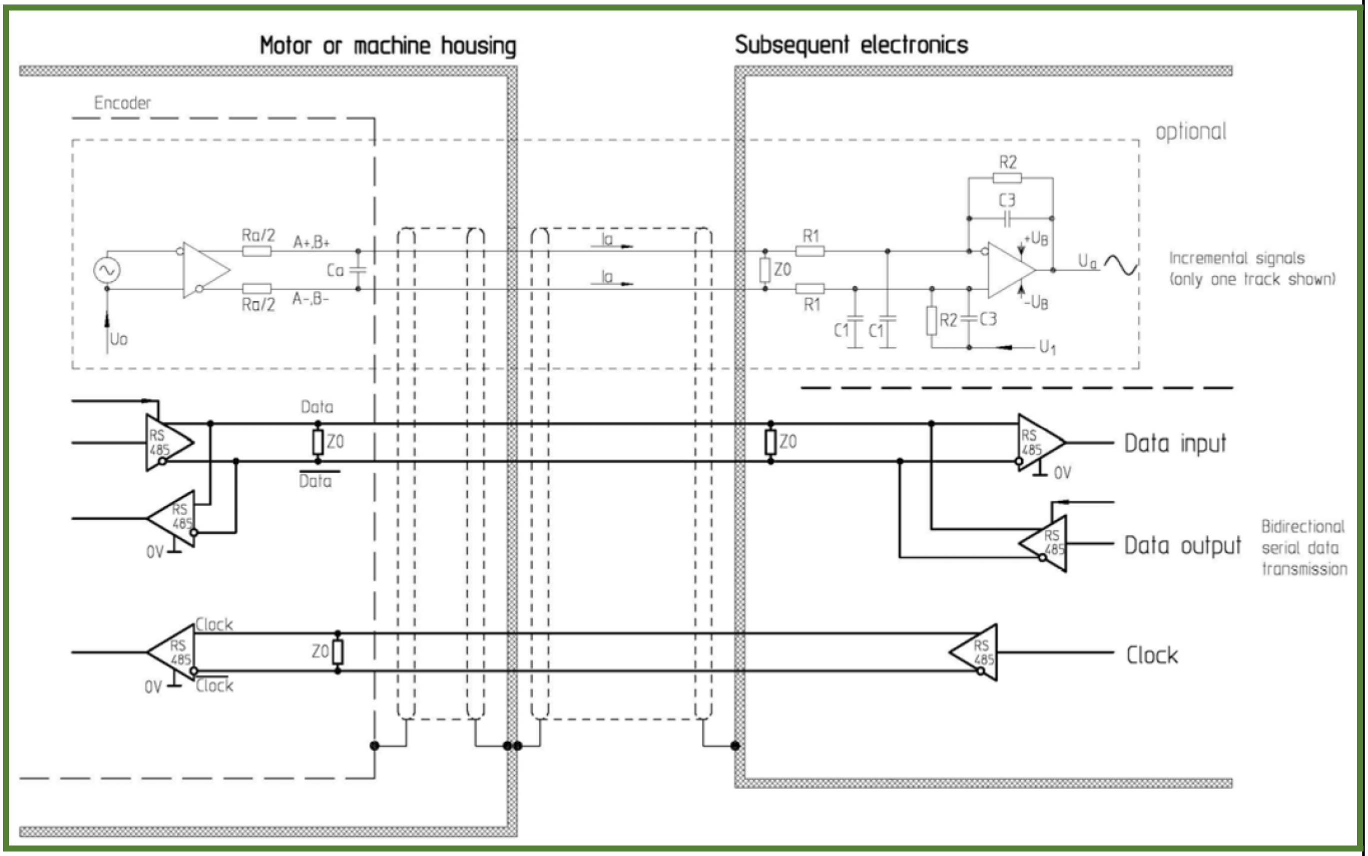
Test	Remark	Result
Physics		
Transceiver (timing, level, direction switching, …)	Transceiver must be able to handle the max. frequency; transceiver must be able to be activated / deactivated within one clock cycle; levels at the differential lines should be checked whether they are within the specifications	
Termination resistors	120 Ohm (without capacitors) is the recommendation	
1 V_{pp} receiver	Check bandwidth limitation and EMC decoupling capacitors; check if amplifier can handle the max. voltage swing according to 1 V_{PP} specification	
EMC decoupling	Check for RS-485 and 1 V_{PP} receiver; tolerance of capacitors should be closely matched (CMRR)	
Duty cycle of clock	Optimum is 50%	
High (max. 10 µs) / low (max. 30 µs, 50 ms) phases of clock	Limits are 10 µs maximum high time (lower 100 kHz limit); maximum low time is 30 µs (for LC) / 50 ms (rotary encoder)	
Power-up		
Timing (power ramp-up, t_1, t_2, t_3, t_{Sat}, …)	Please refer to the comments 1 to 7: can be found in the FAQ area or the EnDat Seminar	
50 ms until clock is stable?	Please check as per EnDat Specification (power-on conditions)	
Level (clock and data)	See above	
Clocks during power up	Up to the time where t_3 starts, no clock pulses are allowed to be sent to the encoder	
"Reset-pulse" at clock line	Please refer to the comments 1 to 7: can be found in the FAQ area or the EnDat Seminar	
Sequence after power-up		
Sequence due to specification?	See EnDat Specification Appendix A9 "Processes"	
2.1 mode commands		
Timing 2.1 Position	Check the timing according to specification: t_{ST}, first two clock pulses for changing data direction, transmission of mode command; 2nd change of direction; check if polling of Start bit starts by the time when there are stable conditions at the data line	
Timing 2.1 Parameter r/w, MRS, Reset; (are the 1 ms / 12 ms / 50 ms followed)	Minimum length between commands like MRS, parameter read or write is 1 ms according to EnDat Specification	
Check cycle time	Is the cycle time long enough so that the recovery time t_M can elapse	
Recovery time sequence correct?	Is the recovery time sent correctly by the encoder; make sure that the recovery time is not interrupted by any additional clock pulse (this puts the encoder into continuous clock mode; see Specification)	
Behavior at different clock rates	Check the above topics at different clock rates; influence on start-bit polling, recovery-time behavior, cycle-time,…	
CRC checking	Check if CRC checking works (to test, try to send one clock pulse less for position)	
Switching of data direction	Make sure that enabling and disabling is done at the right phase of the clock	
Correct number of clock cycles	Make sure that the correct number of clock pulses is sent to the encoder; sending too many clock pulses can bring the encoder to continuous clock mode (position command) or cause Error-type I reactions of the encoder	
Point of time for data takeover	The optimum point for taking data over is the rising edge of the following clock cycle (see Specification)	
Polling of Start bit	Make sure that you poll for the Start bit; clock pulses must be sent down to the encoder (see Specification)	

Request of position values		
Test with optical and inductive types of encoders	This is to make sure that the whole variety of HEIDENHAIN encoders is supported; especially Start-bit polling is tested with this, as well as slight differences of the timing of the different encoders	
Test with EnDat01, 21, 02, 22	This is to make sure that the whole variety of HEIDENHAIN encoders is supported; especially Start-bit polling is tested with this, as well as slight differences of the timing of the different encoders	
Test with LC xx3	This is to make sure that the whole variety of HEIDENHAIN encoders is supported; especially Start-bit polling is tested with this, as well as slight differences of the timing of the different encoders	
Test with EIB	This is to make sure that the whole variety of HEIDENHAIN encoders is supported; especially Start-bit polling is tested with this, as well as slight differences of the timing of the different encoders	
Position value formation, absolute + incremental	Check if the connection of absolute and incremental data works correctly for different encoders	
Parameter handling		
OEM programming (read-out of start and stop addresses) + relative programming	To program the OEM memory, start and stop addresses must be read out of the encoder, because different encoders have different sections	
Acknowledgement of 8-bit and 16-bit extension	To check if the encoder has understood the MRS address / parameter / ... correctly: the information sent down is sent back for acknowledgement; this must be checked, because if the encoder doesn’t understand the parameter the inverted data is sent back; see Error-type 2 in Appendix 1 of the EnDat Specification	
Readout of data-width for position	This is essential for sending down the right number of clock pulses to the encoder	
Parameter handling for different encoder types	Parameters need to be interpreted differently whether it is a linear or rotary encoder, or a singleturn or multiturn encoder	
Readout of EnDat 2.1 Parameters	Check if reading out the parameters works correctly	
Readout of EnDat 2.2 Parameters (MRS selection + readout in 2.1 mode)	To get access to EnDat 2.2 parameters, the MRS selection must be done with the EnDat 2.2 mode command; after that, parameter reading can be done with 2.1 or 2.2 mode commands	
Parameter readout in 2.2 mode	Please refer to the EnDat seminar; there is a theoretical section about that as well as some practical training	
Datum shift	Check if datum shift works correctly; make sure that if 1 V_{PP} is used the datum shift is done "EnDat conform" (see Specification)	
Initializing the functions	Check if, for example, shortening of the recovery time works (if supported by the encoder)	
Write-protection	Check if the process is according to the specification; be aware: once write-protection is set it cannot be reset	
Speed-dependent checking of absolute - incremental data	If absolute data is checked against incremental data, the different speed areas according to EnDat 2.1 parameters must be taken into account	
Errors / Warnings / Error Types		
Delete errors and warnings	Sequence of clearing errors and warnings must be checked; sequence can be found in Appendix A9 "Processes"	
Test of the sequence "F1 occurs"	With the occurrence of an error the control must bring the system into a "safe system status"; this needs to be checked	
How warnings are processed	If the control checks for warnings, this procedure must be tested	
Detection + Reaction Error-Type I, II, III	The 3 Error types according to Appendix A1 must be checked for whether they are detected correctly; Error type 1 → write wrong MRS address; Error type 2 → write to an EnDat 2.1 parameter address; Error type 3 → select additional data that is not supported	

Data width		
Position 40 / 48 bits?	The maximum data width is 40 bits for EnDat 2.1 and 48 bits for EnDat 2.2; this must be supported by the EnDat master, which means that the width of the receiving registers must be able to hold that number of bits	
Parameter, Additional Data, …	Additional data has a fixed length of 30 bits; a maximum of 2 pieces of additional data can be selected -> 60 bits must be reserved; Parameters are 16 bits wide and addresses are 8 bits wide	
Incremental data		
Is quadrant connection checked with absolute data?	Only if 1 V_{PP} is used: To check if the absolute data fits the incremental data (check for the code connection), at least two bits of overlap between absolute and incremental data is provided	
µC handling		
Behavior at interrupt (Clock → low)	If an interrupt forces the µC to stop the communication with the EnDat encoder the clock should be brought (or kept) to a low level, cause at a low level the encoder can be "parked" for up to 50 ms (exception LC: 30 µs), at a high level the encoder will interrupt the communication and reset itself after 10 µs (elapse of the recovery time)	
High / low timing	See also "Physics"	
Cable		
Test at different cable lengths	This is important to check if change of data direction, data takeover and polling for the Start bit are working correctly; EnDat allows up 100 m (150 m at very low clock frequencies), this should be tested, or at least the maximum cable length that is allowed for your application	
Stability of signal delay compensation	If signal delay compensation is used (EnDat 2.2) it must be checked if this is working with different cable lengths; Temperature has an influence on the signal delay of the cable, depending on the method that is used for signal delay compensation this influence needs to be checked	
2.2 mode commands		
Timing 2.2 Position	EnDat 2.2 position command provides 2 error bits and additional data is transferred as well, if selected; it should be checked if one or two pieces of additional data is processed correctly	
Selection and readout of the additional data; cyclical switching between additional data	Additional data is stored in the ASIC and can (it's not a must) therefore be changed from one EnDat request to the other; check if the process of selecting and de-selecting of additional data is working properly and if the control is sending the right number of clock pulses to read out the additional data	
Timing 2.2 Parameter r/w, MRS, Reset	EnDat 2.2 mode commands like "..select MRS..", "…send parameter.." all have the same structure, but different data content for address and data; the critical part regarding the timing is the third change in data direction from the encoder back to the control, therefore it must be known how many clock pulses are already on the line (this is a very essential part of the signal delay compensation); this needs to be verified very carefully, especially using different cable lengths	
Check cycle time	Is the cycle time long enough so that the recovery time t_M can elapse; for EnDat 2.2 mode commands the recovery time can be shortened (see "Function initialization" in the EnDat specification); check also if this function is used	
Recovery time sequence correct (short and long recovery time)?	If you have selected the shortened recovery time (only valid for EnDat 2.2 commands) check if the control processes this correctly, so for EnDat 2.2 mode commands the minimum clock frequency is then 1 MHz	
Behavior at different clock rates	Check the above topics at different clock rates; influence on start-bit polling, recovery-time behavior, cycle-time,…	

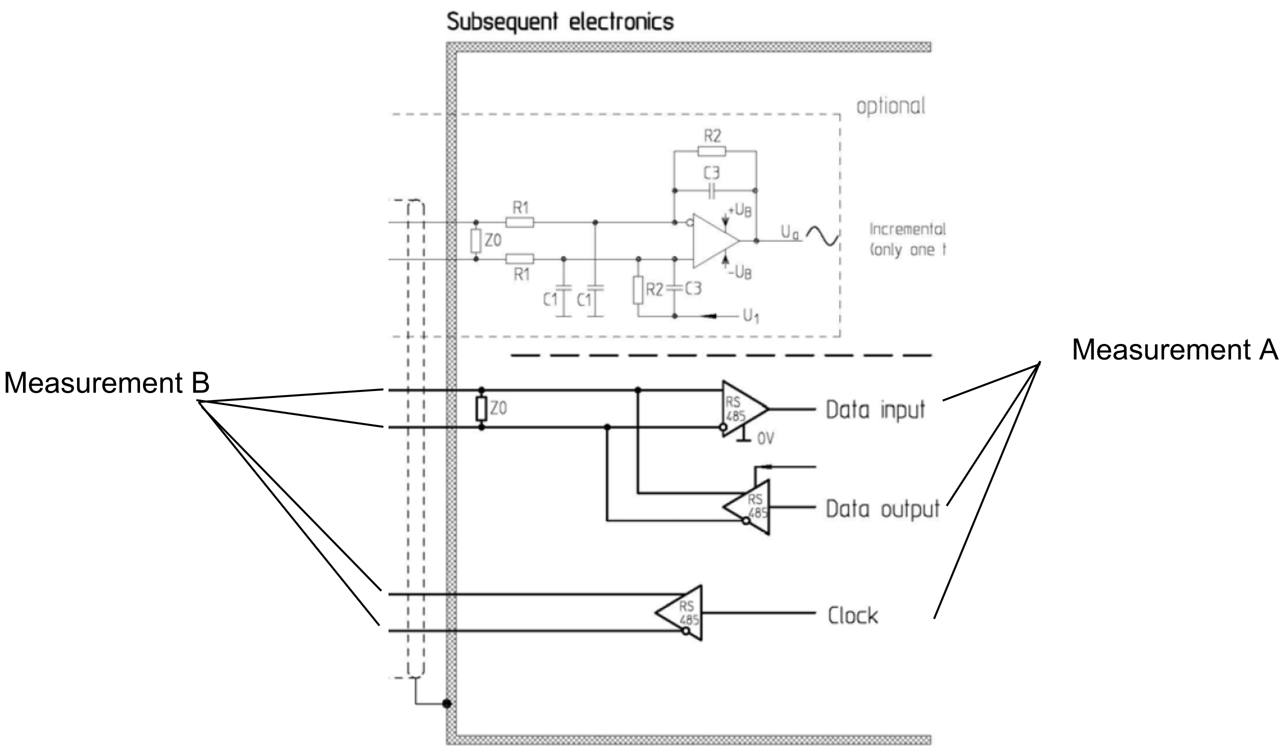
CRC checking: position and additional data	There is separate CRC checking for Position, additional data 1 and 2; please verify that all CRCs are checked	
Switching of data direction 1, 2 and 3	There can be up to 3 switches in data direction within one EnDat communication cycle; especially critical is the third switch (only for part of the EnDat 2.2 mode commands; see above); please verify, since very critical	
Correct number of clock cycles	Please check	
Check signal delay compensation (100..200 kHz) / point of time for data takeover	There are two basic methods for the signal delay compensation: measuring the signal delay and doing an internal shift of the receiving register based on this measurement or detection of the edge of the start bit and internal shift of the receive register based on the edge detection. Please check if the method you are using is stable for different clock frequencies and cable lengths.	
Polling of Start bit	Make sure that you poll for the Start bit; clock pulses must be sent down to the encoder (see Specification)	
Diagnostics data	Check if the process of reading out diagnostics data (see EnDat Seminar, chapter 9) is implemented correctly and works with different types of encoders	

16.2 Recommendation: Measurement of Clock and Data



Overview over an EnDat system consisting of encoder (left side) and subsequent electronic (right side)

Recommendation: Measuring of clock and data



Preferred is to have Measurement A and B. Measurement B should be done closest possible to the subsequent electronic.
Measurement should be done with a 4-channel oscilloscope to have all measurements on one sheet.
Resolution of Measurements:

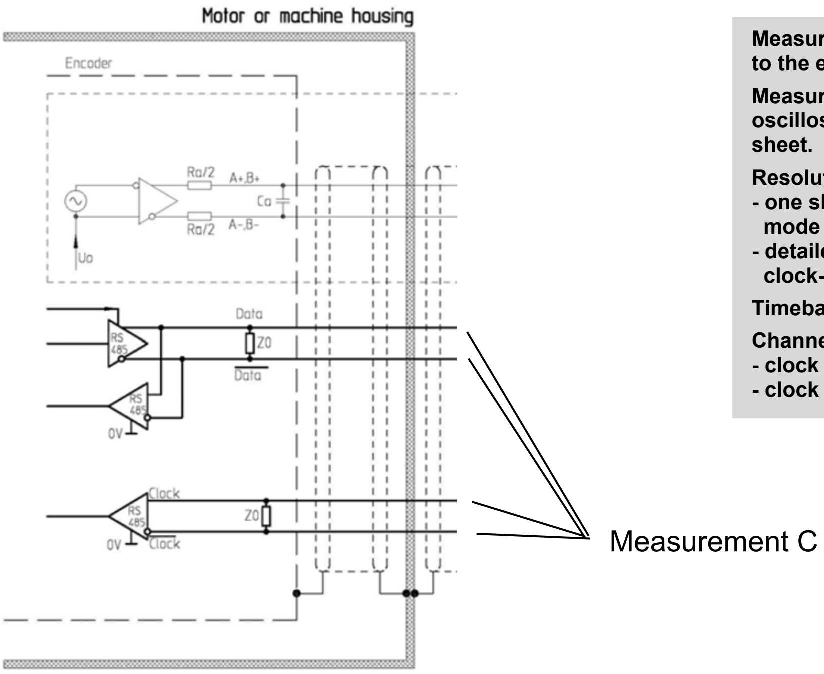
- one should show the complete cycle of the mode command, including recovery time
- detailed plots with a resolution of about one clock-cycle per division

Timebase and voltage level must be visible.

Channels should be named, e.g.:

- Clock
- clock inverted

Recommendation: Measuring of clock and data



Measurement C should be done closest possible to the encoder.

Measurement should be done with a 4-channel oscilloscope to have all measurements on one sheet.

Resolution of Measurements:
- one should show the complete cycle of the mode command, including recovery time
- detailed plots with a resolution of about one clock-cycle per division

Timebase and voltage level must be visible.

Channels should be named, e.g.:
- clock
- clock inverted

Measurement C should be done closest possible to the encoder. Measurement should be done with a 4-channel oscilloscope to have all measurements on one sheet.

Resolution of Measurements:

- one should show the complete cycle of the mode command, including recovery time
- detailed plots with a resolution of about one clock-cycle per division

Timebase and voltage level must be visible.

Channels should be named, e.g.:

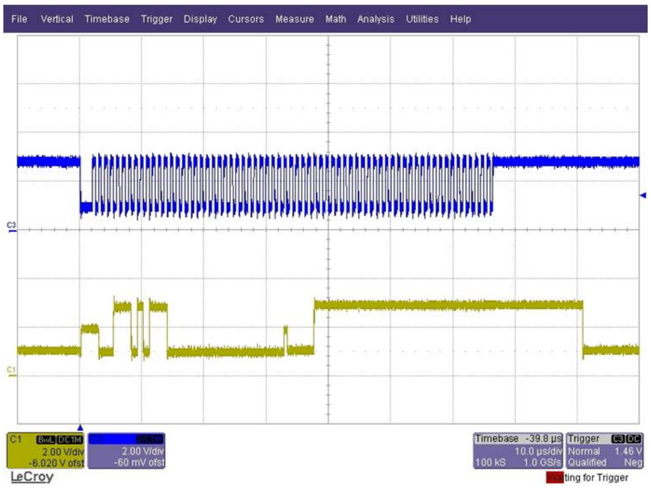
- clock
- clock inverted

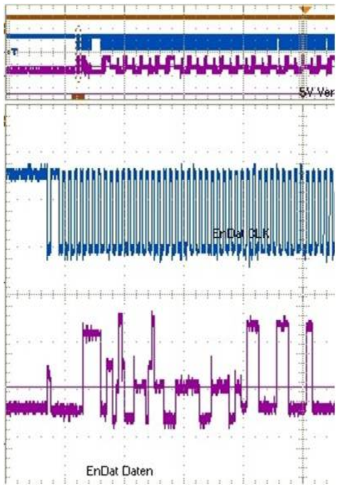
Recommendation: Measuring of clock and data

Example for Measurements:

Resolution of Measurements:

- one should show the complete cycle of the mode command, including recovery time
- additionally detailed plots with a resolution of about one clock-cycle per division

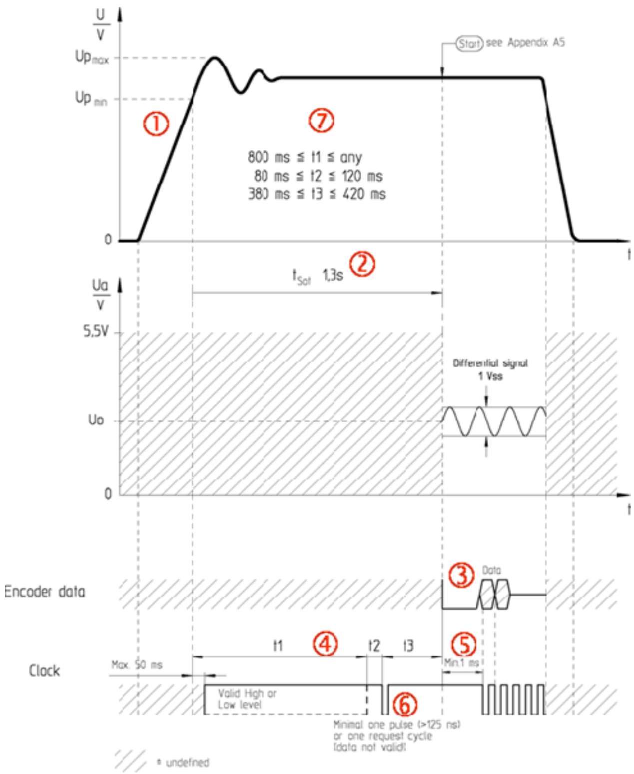




Example:
Showing Error Type I because of wrong MRS-code

Example showing trouble with Transceiver:
Level shift for high and low level (comparison to other levels) is due to the fact that both Transceiver are active.

Recommendation: Measuring of clock and data



① The rise time of the supply voltage until U_P,min is reached should be > 10 V/sec.

② The time until the 1-V_{PP} incremental signals assume valid values is max. 1.3 sec.

③ The encoder can be recognized as an EnDat or SSI encoder by the logic level on the data line, after power has been switched on (see enclosure A5).

④ Clock pulse edges during t1 or t2 can interrupt booting; this can only be corrected by switching the encoder off and then on again.

⑤ A first EnDat request (falling edge) is permissible once t3 has ended after at least 1 ms (there is no maximum time limit). After the first clock pulse, the direction of data on the date line is reversed (this is why the data line is then at "high impedance").

⑥ The encoder requires a defined reset:
Falling edge + end of recovery time;
During the Low phase this is: 0.125 < t_{LOW} < 30 µs

⑦ t1: Boot or reset time of the EnDat encoder
t2: Initialization phase of the EnDat encoder
t3: Must be maintained for downward compatibility to EnDat 2.1

Power-up and boot phases:
The power-up phase (see at right, excerpt from the EnDat seminar and EnDat FAQ) is also of importance for the encoder to function correctly. Therefore, this phase should also be measured with the oscilloscope.

Measurement C (see page 3) should be used for this, and the supply voltage U_P should be measured as well.

In case power is switched on again, the switch-off and switch-on phases should be recorded.

The recording should also include the first “valid” EnDat request. The sampling time of the oscilloscope must be adapted correspondingly. If possible, the clock and data lines should be recorded as well.

16.3 Recommendation: Encoders for testing the EnDat implementation

General information

The EnDat interface supports different encoder models that should be tested for a broad test coverage. Testing all possible encoder models is not necessary because it is not relevant within the scope of the test coverage—for example regarding incremental encoders—whether one reference mark or distance-coded reference marks are being used (this concerns the evaluation in the encoder or the EIB). Furthermore, not all of the possible encoder models are supported at present.

Encoder models according to EnDat 2.1 parameter, word 14:

Encoder model	**Remark on the test**
Incremental linear encoder With distance-coded reference marks Without distance-coded reference marks With backup battery With distance-coded reference marks Without distance-coded reference marks	Testing a combination of these four possible models is sufficient. The versions with buffer battery backup are currently not yet supported. Test recommendation: EIB x92 + LS 487C
Absolute linear encoder With cyclic coding	Test recommendation: LC 415 Currently not supported.
Touch probes	Currently not supported.
Incremental rotary encoder or incremental angle encoder With distance-coded reference marks Without distance-coded reference marks With backup battery With distance-coded reference marks Without distance-coded reference marks	Testing a combination of these four possible models is sufficient. The versions with buffer battery backup are currently not yet supported. Test recommendation: EIB x92 + ROD 486
Singleturn rotary encoder or singleturn angle encoder	Test recommendation: ROC 425
Multiturn rotary encoder with gears	Test recommendation: ROQ 437
Multiturn rotary encoder with battery buffer	Test recommendation: EBI 1135
EIB	Currently not supported because the EIB is always shipped preconfigured for the encoder.

17 EnDat Error Injector

The EnDat error injector enables manipulation of an EnDat transmission. Manipulation can occur in both directions, i.e. from master to encoder or from encoder to master. A special PWM 20 version forms the basis for the error injector.
If you require more information, please get in touch with your HEIDENHAIN contact person.

18 Commissioning Diagnostics for Encoders with Gray-Code Scanning

Contents are no longer relevant for the current generations of encoders; consult your contact person at HEIDENHAIN if required.

19 Processing of various temperature sensor types

Via the EnDat supplementary information 1, temperature values can be transmitted from internal measuring or external sensors. The temperature specification has so far been based on the use of KTY 84-130 sensors. In order to be prepared in the future for the automatic processing of further types of temperature sensors by the measuring device or the subsequent electronics, as well as for the processing of multiple temperature sensors, corresponding information was introduced in the EnDat specification.

Additional sensors:

- For the definition of external sensors for detecting the motor temperature, see Chapter 2.3.5.1 of the EnDat specification.
- Due to the limitation to 12 bits for the output value, a fixed scaling of the values was introduced, see Chapter 3.9.6 of the EnDat specification.

Adjustability of evaluation of the temperature sensor type

- Currently not supported (as of October 2017, reserved for future encoder generations).
- For the evaluation settings, see Chapter 3.6.8 of the EnDat Specification.
- To find out which evaluations the measuring instrument supports, see Chapter 3.9.36 of the EnDat specification.

Storing the information on the connected temperature sensor location types:

- Words have been introduced for a subsequent electronics to perform a compensation calculation for unsupported temperature sensor types, (see Chapter 3.6.9 of the EnDat specification).
- If these words are filled, then the follower electronics can check whether evaluation (e.g., KTY 84-130) and actually connected temperature sensor location type (e.g., PT 1000) match. If required, a compensation calculation can then be made in the subsequent electronics.
- This allows in a specific case: “Measuring device supports KTY 84-130 evaluation, but a PT 1000 sensor is connected,” the automatic compensation calculation in the subsequent electronics. For the compensation polynomial, see the brochure “Encoders for Servo Drives” in the chapter “Connectable temperature sensors.”
  An appropriate compensation calculation is integrated in the ATS software version v3.2.0.

The following tables show the recommendation for implementing the above-mentioned items by the motor manufacturer and the control manufacturer. The recommendation for the motor manufacturer refers both to existing units that do not yet support configuration of the temperature evaluation and future units that will support this. This example focuses on the evaluation of KTY 84-130 and PT 1000.

Abbreviations:

- Operating parameter, wort 9 OPW9
- Operating parameter, word 10 and word 11 OPW10+11
- EnDat 2.2 parameter, wort 50 E22W50
- NA Not applicable

Recommendation for the motor manufacturer
The recommended programming for OPW9 and OPW 10+11 depends on the sensor model installed in the motor and on the content of the E22W50.

Note:
OPW 10+11 = 2 means

- Assign the value 0x02 to OPW10
- Assign the value 0x00 to OPW11

Sensor model	E22W50	Programming OPW9	Programming OPW10+11	Encoder evaluation	Remark
KTY 84-130	Bit 15 = 0	0	1	KTY 84-130	
	Bit 15 = 1	1	1	KTY 84-130	
PT 1000	Bit 15 = 0	0	2	KTY 84-130	Control must calculate compensation
	Bit 15 = 1, bit 2 = 0, bit 1 = 1	1	2	KTY 84-130	Control must calculate compensation
	Bit 15 = 1 and bit 2 = 1	2	2	PT 1000	
PT 100	Bit 15 = 0	0	3	KTY 84-130	Control must calculate compensation
	Bit 15 = 1, bit 3 = 0, bit 1 = 1	1	3	KTY 84-130	Control must calculate compensation
	Bit 15 = 1 and bit 3 = 1	2	3	PT 100	
KTY 83-110	Bit 15 = 0	0	4	KTY 84-130	Control must calculate compensation
	Bit 15 = 1, bit 4 = 0, bit 1 = 1	1	4	KTY 84-130	Control must calculate compensation
	Bit 15 = 1 and bit 4 = 1	4	4	KTY 83-110	

Recommendation for the control manufacturer
The combination of evaluation in the encoder and the sensor model depends on whether E22W50, OPW9, and OPW10+11 is the case. If these two do not match, then the compensation calculation must be performed in the control.

E22W50	OPW9	OPW10+11	Encoder evaluation	Sensor model	Action required in control
Bit 15 = 0	NA	0 or 1	KTY 84-130	KTY 84-130	None
Bit 15 = 0	NA	2	KTY 84-130	PT 1000	Compensation correction KTY 84-130 --> PT 1000
	NA	3 ... 255	KTY 84-130	Other	Depends on sensor model
Bit 15 = 1 and bit 1 = 1	1	1 or 0	KTY 84-130	KTY 84-130	None
Bit 15 = 1 and bit 1 = 1	1	2	KTY 84-130	PT 1000	Compensation calculation or switchover to PT 100 evaluation in encoder
Bit 15 = 1 and bit 1 = 1	1	3 ... 255	KTY 84-130	Other	Depends on sensor model
Bit 15 = 1 and bit 2 = 1	2	2 or 0	PT 1000	PT 1000	None
Bit 15 = 1 and bit 2 = 1	2	3 ... 255	PT 1000	Other	Depends on sensor model
Bit 15 = 1 and bit 3 = 1	3	3 or 0	PT 100	PT 100	None
Bit 15 = 1 and bit 3 = 1	3	Not equal to 3	PT 100	Other	Depends on control
Bit 15 = 1 and bit 4 = 1	4	4 or 0	KTY 83-110	KTY 83-110	None
Bit 15 = 1 and bit 4 = 1	4	Not equal to 4	KTY 83-110	Other	Depends on control
Bit 15 = 1	0 or 5..7				Evaluation not possible

Remark:
If E22W50 bit 15 =1 is set, then word OPW10+11 is set to “0” by default.
OPW 10+11 should be set by the motor manufacturer in accordance with the installed sensor.
In the default value “0,” it is assumed that the sensor corresponds to the configured evaluation (e.g., KTY 84-130).