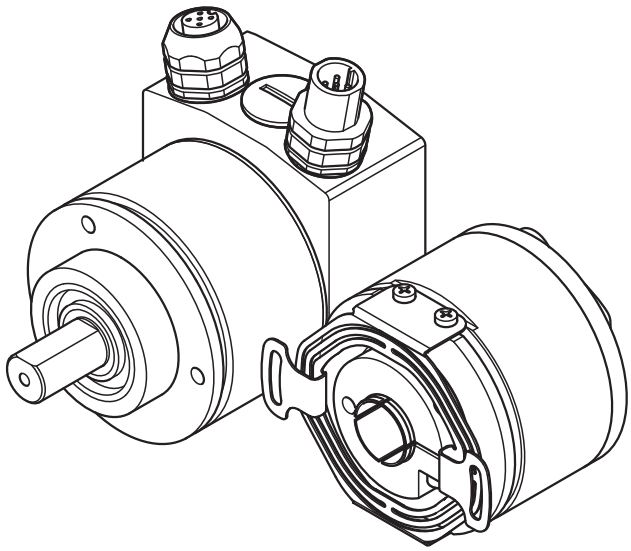
LENORD +BAUER
... automates motion.

Reference



D-02R-2xCO (2.0)

Device manufacturer and publisher:

Lenord, Bauer & Co. GmbH
Dohlenstraße 32
46145 Oberhausen ● Deutschland
Phone: +49 208 9963–0 ● Fax: +49 208 676292
Internet: www.lenord.de ● E-Mail: info@lenord.de

Doc. no. D-02R-2xCO (2.0)

Table of contents

1 General

1.1 About this document

This description deals with the CANopen fieldbus connection of the following absolute rotary encoders:

- GEL 235 CO x x x x x x x x x
- GEL 2035 CO x x x x x x x x x x
- GEL 2352 CO x x x x x x x x

It is intended for technicians who are already familiar with the functionality of the absolute encoders and who have basic knowledge of fieldbus connections – in particular relating to CAN bus and CANopen. For further information refer to the concerning standards of the *CAN in Automation* (CiA) organisation (www.canopen.org).

Information on functions, handling and specifications of the absolute encoders is provided in the accompanying documents and data sheets in the download area of www. lenord.de.

Numerical data:

If not otherwise indicated, decimal values are shown as digits without an affix (e.g. 1408). Binary values are indicated by a “b” (e.g. 1101b) and hexadecimal values by a “h” (e.g. 680h) that is appended to the digits.

Abbreviations:

The type specification **2x** covers the absolute encoders listed above.

The term **CO x …** is part of the type/order code of the encoder and is no longer mentioned in the following.

The term **encoder** is used as a synonym for absolute (rotary) encoder.

ST and **MT** mean single-turn and multi-turn.

1.2 Description

The GEL 2x encoders implements the CANopen slave concept according to DS-301 (V4.02) communication profile and support several of the objects defined in the DS-406 (V3.2) device profile for absolute encoders. The complete specification of the concerning profiles can be seen at the *CAN in Automation e.V.*:

- Protocol layer DS-301: *CANopen Application Layer and Communication Profile, CiA Draft Standard 301, February 2002*
- Device profile DS-406: *CANopen Device profile for encoders, CiA Draft Standard 406, Version: 3.2, 18 December 2006*

The supported communication and device features as well as the functions of the respective encoder are defined in the EDS file of the encoder.

The GEL 235 encoder is connected to the bus via a configurable interface module (cable or connectors). With the other encoders, two M12 connectors are implemented in the standard housing.

2 Connection and setup elements

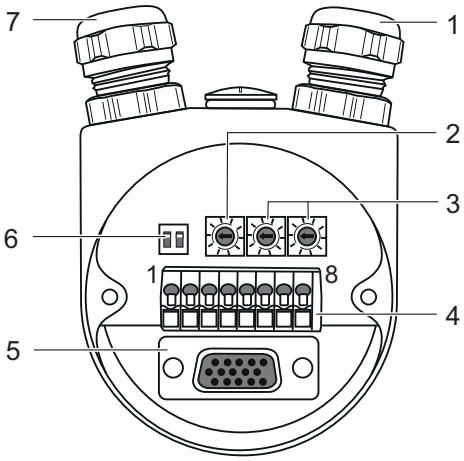
2.1 GEL 235 (separate bus interface)

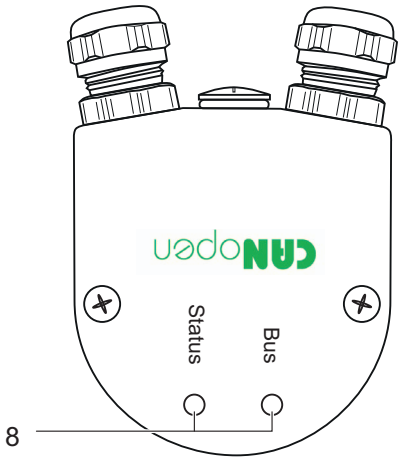
All configuration elements are inside the bus interface. For this reason, the necessary configuration settings must be made before the absolute encoder is operated on the CAN bus.

The bus interface can be removed after unscrewing the two screws on the rear of the encoder housing.

Changes made to the rotary switch are evaluated during start-up of the encoder (after reset or power-on).

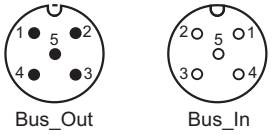
Overview





1 Bus_In (cable screw or M12 connector)
2 Baudrate configuration
3 User address configuration
4 Terminal strip for cable connection
5 Encoder interface
6 Bus terminating resistance on/off
7 Bus_Out (cable screw or M12 connector)
8 LED indicators

Bus connector [1], [7]

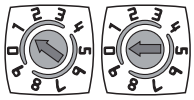


1 Screen
2 $+U_S$
3 GND
4 CAN_H
5 CAN_L

Baudrate [2]

0	Autobaud	4	100 kBit/s	8	800 kBit/s
1	–	5	125 kBit/s	9	1 MBit/s
2	–	6	250 kBit/s		
3	50 kBit/s	**7**	**500 kBit/s**		

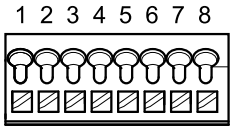
User address [3]



x1 *x10*

The switch setting **00** (factory setting) corresponds to the node ID 1 and permits the assignment of another node ID via CAN bus (LSS service).

Cable connection [4]



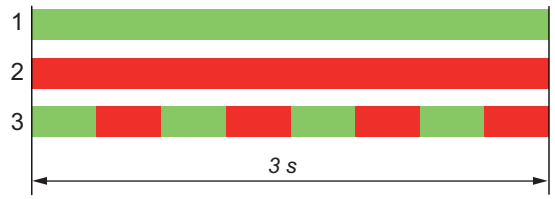
1	CAN_H	CAN_{Out} +
2	CAN_L	CAN_{Out} –
3	GND	GND_{Out}
4	+VS	Supply $V_{S,\ Out}$ (10–30 VDC)
5	CAN_H	CAN_{In} +
6	CAN_L	CAN_{In} –
7	GND	GND_{In}
8	+VS	Supply $V_{S,\ In}$ (10–30 VDC)

Bus terminaton [6]

If the absolute encoder is the last device on the CAN bus, the integrated termination resistances (CAN_H, CAN_L) must be connected if no external resistors are used. In this case, set **both** miniature switches to the ON position (set to off at the factory).

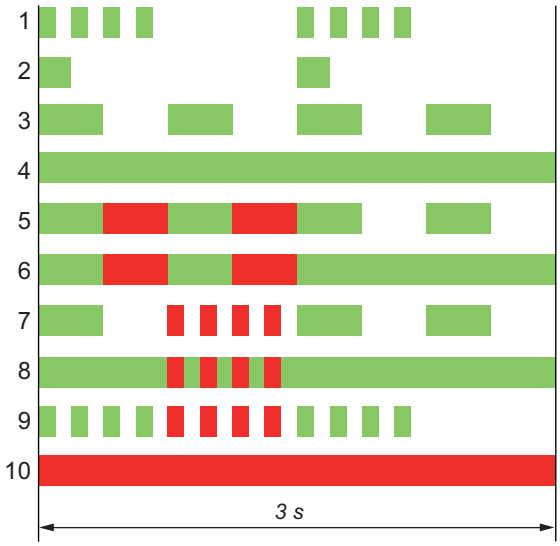
LED indicators [8] (1)

- Status



1 Voltage ok
2 Internal device error
3 Automatic Baudrate recognition

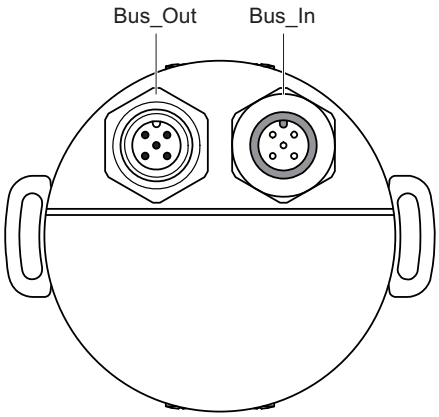
- Bus

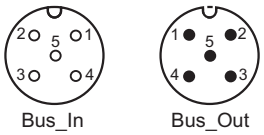


1 *Init*
2 *Stopped*
3 *Pre-operational*
4 *Operational*
5 *Pre-operational*, bus warning
6 *Operational*, bus warning
7 *Pre-operational*, bus passive
8 *Operational*, bus passive
9 *Init*, bus passive
10 *Bus off*

2.2 GEL 2035, GEL 2352

Bus connector M12





1 CAN_GND
2 $+V_S$
3 GND
4 CAN_H
5 CAN_L

(1) For monochrome representation: ■ ≙ green, ■ ≙ red

3 Object directory

The object directory contains all supported CANopen characteristics of the encoders. The data is located power failure-proof in the device's flash memory and are copied into the RAM after power on or reset. If data is changed in the object directory this is done only temporarily in the RAM. For permanent saving, data must be transferred to the flash memory via object 1010h, thus overwriting the original data.

Access to the object directory is made by means of SDO services.

The object directory is divided into three categories:

- Communication parameters according to CANopen standard DS-301
- Device parameters according to CANopen standard DS-406
- Vendor-specific parameters

Entries in the object directory are addressed using a 16-bits index. Each index entry can be further subdivided using a subindex.

Information on the object directory given below:

- Acc. (access mode): ro = read only, rw = read/write
- (Data) Type: U*xx* = Unsigned*xx* (xx = 8/16/32 → 1/2/4 bytes), S*xx* = Signed*xx* (xx = 16/32 → 2/4 bytes), STR = ASCII string
- Sub = subindex (data format: U8)

3.1 Communication parameters according to DS-301

<table>
<tr><th>Index</th><th>Name</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>1000h</td><td>Device type</td><td>ro</td><td>U32</td><td>Value: 00h xxh 01h 96h, with xx =
01: absolute encoder, single-turn
02: absolute encoder, multi-turn
03: absolute encoder, single-turn with electronic revolution counter</td></tr>
<tr><td>1001h</td><td>Error register</td><td>ro</td><td>U8</td><td>Bit 0: 1 = general error (alarm message by the encoder)
Bit 1–7: not used</td></tr>
<tr><td>1003h</td><td>Pre-defined error field</td><td>ro</td><td>U32</td><td>
<table>
<tr><th>Sub</th><th>Content</th></tr>
<tr><td>00h</td><td>number ≤ 20 (acc.: rw)</td></tr>
<tr><td>01h</td><td>last error</td></tr>
<tr><td>02h</td><td>last but one error</td></tr>
<tr><td>⋮</td><td></td></tr>
<tr><td>14h</td><td>first of the last 20 errors</td></tr>
</table>
Erase error memory: 00h → subindex 0
</td></tr>
</table>

<table>
<tr><th>Index</th><th>Name</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>1005h</td><td>COB-ID SYNC</td><td></td><td></td><td>
<table>
<tr><th>Bit</th><th>Content</th></tr>
<tr><td>0–10</td><td>identifier, standard ID: 80h</td></tr>
<tr><td>11–29</td><td>reserved for 29-bits identifier (0)</td></tr>
<tr><td>30</td><td>1 = device does not generate Sync messages</td></tr>
<tr><td>31</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>1008h</td><td>Manufacturer status register</td><td>ro</td><td>STR</td><td>e.g. “Lenord+Bauer GEL2352 CAN” as ASCII string</td></tr>
<tr><td>1009h</td><td>Hardware Version</td><td>ro</td><td>STR</td><td>e.g. “V4.00”</td></tr>
<tr><td>100Ah</td><td>Software Version</td><td>ro</td><td>STR</td><td>e.g. “V1.06”</td></tr>
<tr><td>100Ch</td><td>Guard time</td><td>rw</td><td>U16</td><td rowspan="2">This function is obsolete; CiA recommends using the Heartbeat function (producer / consumer), → object 1017h.</td></tr>
<tr><td>100Dh</td><td>Life time factor</td><td>rw</td><td>U16</td></tr>
<tr><td>1010h</td><td>Store parameters</td><td>rw</td><td>U32</td><td>Transfer of the parameter values from RAM to flash memory
● Write
Write code word “save” in reverse order (65766173h) into the respective subindex.
● Read
The value 1 is output.
<table>
<tr><th>Sub</th><th>Content</th></tr>
<tr><td>00h</td><td>number of entries = 4 (acc.: ro)</td></tr>
<tr><td>01h</td><td>all parameters</td></tr>
<tr><td>02h</td><td>communication parameters (DS-301) only</td></tr>
<tr><td>03h</td><td>device parameters (DS-406) only</td></tr>
<tr><td>04h</td><td>vendor-specific parameters only</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Index</th><th>Name</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>1011h</td><td>Restore default parameters</td><td>rw</td><td>U32</td><td>Device parameters will be reset to the factory setting and not to the values saved via object 1010h
Write
Write code word “load” in reverse order (64616F6Ch) into the respective subindex.
Lesen
The value 1 is output always.
<table><tr><th>Sub</th><th>Content</th></tr><tr><td>00h</td><td>number of entries = 4 (acc.: ro)</td></tr><tr><td>01h</td><td>all parameters</td></tr><tr><td>02h</td><td>communication parameters (DS-301) only</td></tr><tr><td>03h</td><td>device parameters (DS-406) only</td></tr><tr><td>04h</td><td>vendor-specific parameters only</td></tr></table></td></tr>
<tr><td>1014h</td><td>COB-ID EMCY</td><td>rw</td><td>U32</td><td><table><tr><th>Bit</th><th>Content</th></tr><tr><td>0–10</td><td>identifier, standard ID: 80h + node ID</td></tr><tr><td>11–29</td><td>reserved for 29-bits identifier (0)</td></tr><tr><td>30, 31</td><td>0</td></tr></table>Emergency error code as for object 1003h, emergency error register as for object 1001h.
After an error has been eliminated a single emergency message is sent with error code 0000h, then the error codes of all errors still existing will follow.</td></tr>
<tr><td>1017h</td><td>Producer Heartbeat time (ms)</td><td>rw</td><td>U16</td><td>a value ≠ 0 deactivates Node Guarding
Recommended by the CiA, this function should be used instead of Node Guarding.</td></tr>
</table>

<table>
<tr><th>Index</th><th>Name</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>1018h</td><td>Object identification</td><td>ro</td><td>U32</td><td>
<table>
<tr><th>Sub</th><th>Content</th></tr>
<tr><td>00h</td><td>number of entries = 4</td></tr>
<tr><td>01h</td><td>vendor ID: 1C5h</td></tr>
<tr><td>02h</td><td>product code: e.g. 235C0h</td></tr>
<tr><td>03h</td><td>revision no.: e.g. 01100100h</td></tr>
<tr><td>04h</td><td>serial no.: xxxxxxxxh</td></tr>
</table>
</td></tr>
<tr><td>1800h
1801h
1802h</td><td>1st/2nd/3rd transmit PDO parameter configuration</td><td>rw</td><td>U32</td><td>
<table>
<tr><th>Sub</th><th>Content</th></tr>
<tr><td>00h</td><td>number of entries = 5 (acc.: ro)</td></tr>
<tr><td>01h</td><td>COB-ID used by PDO
(standard: 180h + node ID)</td></tr>
<tr><td>02h</td><td>transmission mode for PDO
(standard: 1, cyclic)</td></tr>
<tr><td>03h</td><td>min. waiting time for PDO (in ms)</td></tr>
<tr><td>04h</td><td>not used</td></tr>
<tr><td>05h</td><td>event timer for PDO (in ms); after the time has expired the PDO is sent automatically</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Index</th><th>Name</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>1A00h
1A01h
1A02h</td><td>1st/2nd/3rd transmit PDO mapping</td><td>rw</td><td>U32</td><td>Depending on type, the encoders use 2 or 3 transmit PDOs:
● GEL 235: PDO1–3; length 8 bytes
● GEL 2352 MT: PDO1+2; length 4 bytes
● GEL 2352 ST and GEL 2035: PDO1+2; length 2 bytes
<table><tr><th>Sub</th><th>Content</th></tr><tr><td>00h</td><td>number of entries = 2</td></tr><tr><td>01h</td><td>1st user object: current position
● 32 bits (60040020h) with GEL 235 and GEL 2352 MT
● 16 bits (60040010h) with GEL 2035 and GEL 2352 ST</td></tr><tr><td>02h</td><td>GEL 235 only: 2nd user object, default depending on object
● 1A00h: data output 1, 32 bits (20350020h)
● 1A01h: data output 2, 32 bits (20360020h)
● 1A02h: data output 3, 32 bits (20370020h)</td></tr></table></td></tr>
</table>

3.2 Device parameters according to DS-406

<table>
<tr><th>Index</th><th>Name</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>6000h</td><td>Operating parameters</td><td>rw</td><td>U16</td><td>
<table>
<tr><th>Bit</th><th>Content</th></tr>
<tr><td>0</td><td>code sequence (sense of rotation)
0/1 = increasing with **clockwise**/counter clockwise rotation (**cw**/ccw)</td></tr>
<tr><td>1</td><td>diagnosis request
0/1 = **disabled**/enabled, → object 6503h</td></tr>
<tr><td>2</td><td>scaling function
0/1 = **disabled**/enabled, → objects 6001h and 6002h</td></tr>
</table>
</td></tr>
<tr><td>6001h</td><td>Measuring units per revolution (resolution)</td><td>rw</td><td>U32</td><td>The desired resolution for the single-turn part of the absolute encoder can be set with this object. The encoder internally calculates the corresponding scaling factor.
Value range: 0 to max. physical resolution per revolution (e.g. 2000h with 13 bits ST)
The current position value *Pos* results from: Pos = code value × 6001h value / 6501h value
Any change deletes a preset value set before (object 6003h).</td></tr>
<tr><td>6002h</td><td>Total measuring range in measuring units</td><td>rw</td><td>U32</td><td>Value range: 0 to max. total physical resolution (value from object 6501h × number of possible revolutions, e.g. 1000000h with 12 bits ST and MT each)
When using an unfavourable value minor rounding errors may occur for output due to the system.
Any change deletes a preset value set before (object 6003h).</td></tr>
</table>

<table>
<tr><th>Index</th><th>Name</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>6003h</td><td>Preset value</td><td>rw</td><td>U32</td><td>Matching the zero position of the encoder with the machine's zero point

Value range: 0 to set total resolution; FF FF FF FFh deletes preset

Internally, the preset value is converted to a corresponding offset value and added to the position value (offset = preset – position → object 6509h).

The preset value will be deleted if the code sequence is changed or the scaling is activated/changed.</td></tr>
<tr><td>6004h</td><td>Position value</td><td>ro</td><td>U32</td><td>Current position value of the absolute encoder after correction with scaling factor, preset and offset (mapped to PDO1, subindex 1).

With GEL 2035 and GEL 2352 ST, only the lower 16 bits can be used.</td></tr>
<tr><td>6008h</td><td>High precision position value (64 bits)</td><td>ro</td><td>U64</td><td>GEL 235 only (→ objects 2030h–2032h)</td></tr>
<tr><td>6030h

6040h</td><td>Speed value (16 bits)

Acceleration value (16 bits)</td><td>ro</td><td>S16</td><td>not GEL 235
<table><tr><th>Sub</th><th>Content</th></tr><tr><td>00h</td><td>number of entries = 2</td></tr><tr><td>01h</td><td>current value</td></tr><tr><td>02h</td><td>moving average over the number of measured values defined in object 2101h</td></tr></table></td></tr>
</table>

<table>
<tr><th>Index</th><th>Name</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>6200h</td><td>Cyclic timer (ms)</td><td>rw</td><td>U16</td><td>The cycle time for transmitting the PDOs is defined here. The timer controlled output is activated when a cycle time >0 is input.

Value range: 0 to FFFFh (65535) ms

If the transmission type 254 is used (asynchronously event-controlled) the cycle time must be longer than the bus transmission time so that the PDOs can be issued 'without interference'.

Baud rate 10 kbits/s: cycle time ≥ 14 ms
Baud rate 20 kbits/s: cycle time ≥ 10 ms
Baud rate 50 kbits/s: cycle time ≥ 4 ms

With cycle time = 0 (i.e. PDO transmission at value change) the baud rate must be ≥ 125 kbits/s.</td></tr>
<tr><td>6400h</td><td>Area state register</td><td>ro</td><td>U8</td><td>GEL 235 only

The current output status of the work areas can be obtained with this object. The reference value is configured and activated using the object 2001h. The status registers then indicate the position of the reference value in the work area.</td></tr>
<tr><td>6401h

6402h</td><td>Work area low limit

Work area high limit</td><td>rw</td><td>S32</td><td>GEL 235 only

The start/end values (low/high limits) of the work areas are stored in this objects. If a work area (X) is undershot, the corresponding bit in the status register is set (object 6400h, subindex X).

<table>
<tr><th>Sub</th><th>Content</th></tr>
<tr><td>00h</td><td>number of entries = 32 (20h)</td></tr>
<tr><td>01h</td><td>start/end of work area 1</td></tr>
<tr><td>02h – 20h</td><td>start/end of work areas 2 to 32</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Index</th><th>Name</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>6500h</td><td>Operating status</td><td>ro</td><td>U16</td><td>The configuration made with object 6000h can be queried with this object:<table><tr><th>Bit</th><th>Content</th></tr><tr><td>0</td><td>code sequence (0/1 = increasing with clockwise/counter clockwise rotation, cw/ccw)</td></tr><tr><td>1</td><td>diagnosis request (0/1 = disabled/enabled)</td></tr><tr><td>2</td><td>scaling function (0/1 = disabled/enabled)</td></tr></table></td></tr>
<tr><td>6501h</td><td>Single-turn resolution and measuring steps</td><td>ro</td><td>U32</td><td>The ST resolution (i.e. the number of position values) can be queried with this object.</td></tr>
<tr><td>6502h</td><td>Number of distinguishable revolutions</td><td>ro</td><td>U16</td><td>This object can be used to query the number of revolutions that the MT encoder can record.</td></tr>
<tr><td>6503h</td><td>Alarms</td><td>ro</td><td>U16</td><td>In addition to the errors reported by emergency messages, further messages can also be queried here (1 = error occured):<table><tr><th>Bit</th><th>Content</th></tr><tr><td>0</td><td>position error</td></tr><tr><td>1</td><td>hardware diagnosis</td></tr></table></td></tr>
<tr><td>6504h</td><td>Supported alarms</td><td>ro</td><td>U16</td><td>see object 6503h</td></tr>
<tr><td>6505h</td><td>Warnings</td><td>ro</td><td>U16</td><td>Warnings indicate that tolerance limits of internal encoder parameters have been exceeded. Unlike emergency messages, the output value associated with a warning may actually be valid. The bit belonging to the warning remains set for as long as the warning message applies.<table><tr><th>Bit</th><th>Content</th></tr><tr><td>2</td><td>CPU watchdog status (1 = watchdog has triggered a reset)</td></tr><tr><td>3</td><td>operating time (1 = exceeded; optional)</td></tr></table></td></tr>
<tr><td>6506h</td><td>Supported warnings</td><td>ro</td><td>U16</td><td>see object 6505h</td></tr>
</table>

<table>
<tr><th>Index</th><th>Name</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>6507h</td><td>Profile and software version</td><td>ro</td><td>U32</td><td>This object can be used to query the current version numbers of the profile (DS-406) and firmware of the encoder:
<table><tr><th>Byte</th><th>Content</th></tr><tr><td>0</td><td>profile revision (e.g. 20)</td></tr><tr><td>1</td><td>profile version (e.g. 3)</td></tr><tr><td>2</td><td>software revision (e.g. 21)</td></tr><tr><td>3</td><td>software version (e.g. 1)</td></tr></table>
Example: 01210320h = software version 1.21 and profile version 3.2</td></tr>
<tr><td>6508h</td><td>Operating time</td><td>ro</td><td>U32</td><td>not GEL 235 (transmits FFFFFFFFh)</td></tr>
<tr><td>6509h</td><td>Offset value</td><td>ro</td><td>U32</td><td>see object 6003h</td></tr>
<tr><td>650Ah</td><td>Module identification</td><td>ro</td><td>S32</td><td><table><tr><th>Sub</th><th>Content</th></tr><tr><td>00h</td><td>number of entries = 3</td></tr><tr><td>01h</td><td>offset</td></tr><tr><td>02h</td><td>minimum position</td></tr><tr><td>03h</td><td>maximum position</td></tr></table></td></tr>
<tr><td>650Bh</td><td>Serial number</td><td>ro</td><td>U32</td><td>see object 1018h, subindex 4</td></tr>
</table>

3.3 Vendor-specific objects

<table>
<tr><th>Index</th><th>Designation</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>2000h</td><td>Activation of the work area output</td><td>rw</td><td>U16</td><td>GEL 235 only

This object can be used to define for each of the 32 programmable work area whether and how the state of the current position is to be indicated in reference to the defined work area. Each work area state occupies a particular bit position in one of the data outputs 1 to 3.

The definition that a data output transfers work area states at all and that they are to be transferred in the PDO is made using objects 2030h to 2032h.

The start and end values of the work area are defined with objects 6401h and 6402h.

<table>
<tr><th>Sub</th><th>Content</th></tr>
<tr><td>00h</td><td>number of entries = 32 (20h)</td></tr>
<tr><td>01h</td><td>function of the output bit 0 (work area 1) for the data output:
0: work area output inactive
1: high level in the work area
2: high level outside the work area
3: high level before the start of the work area
4: high level after the end of the work area</td></tr>
<tr><td>02h – 20h</td><td>function of the output bits 1 – 31 (work area 2 – 32) for the data output: like subindex 1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Index</th><th>Designation</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>2001h</td><td>Reference value for the work area output</td><td>rw</td><td>U16</td><td>GEL 235 only
<table>
<tr><th>Sub</th><th>Content</th></tr>
<tr><td>00h</td><td>number of entries = 32 (20h)</td></tr>
<tr><td>01h</td><td>reference value for work area 1
0: current position (computed with multiplier, preset value and zero offset)
1: current relative position (absolute position computed with multiplier)
2: current absolute position (absolute encoder position from ST and MT)
3: rotational speed
4: relative speed
5: relative acceleration</td></tr>
<tr><td>02h – 20h</td><td>reference value for work areas 2 – 32: like work area 1</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Index</th><th>Designation</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>2002h</td><td>Status query of work area</td><td>rw</td><td>U16</td><td>GEL 235 only
<table>
<tr><th>Sub</th><th>Content</th></tr>
<tr><td>00h</td><td>number of entries = 32 (20h)</td></tr>
<tr><td>01h</td><td>state of work area 1:
0000h: inactive
0001h: start > end
0002h: start < counting range
0004h: start > counting range
0008h: end < counting range
0010h end > counting range
0020h outside the definitions
0040h start = end
0080h valid
0100h: reference value valid
0200h: area function valid
0400h: area activated
4000h: area written to RAM
8000h: area written to flash memory</td></tr>
<tr><td>02h – 20h</td><td>state of work areas 2 – 32: like work area 1</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Index</th><th>Designation</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>2003h</td><td>Output of work areas (work area register)</td><td>ro</td><td>U32</td><td>GEL 235 only
The maximum of 32 work areas can be evaluated with this object (see also object 2000h).
The work area register can be output via PDO by setting the required data output type (→ objects 2000h and 2030h – 2032h).
<table>
<tr><th>Sub</th><th>Content</th></tr>
<tr><td>00h</td><td>number of entries = 8</td></tr>
<tr><td>01h</td><td>evaluation of the work areas 1 – 4
Bit 0: position in work area 1
Bit 1: position outside work area 1
Bit 2: position before work area 1
Bit 3: position after work area 1
Bit 4–7: work area 2 (like bits 0–3)
Bit 8–11: work area 3 (like bits 0–3)
Bit 12–15: work area 4 (like bits 0–3)</td></tr>
<tr><td>02h</td><td>evaluation of the work areas 5 – 8
Bit 0–15: like subindex 1</td></tr>
<tr><td>03h – 08h</td><td>evaluation of the work areas 29 – 32
Bit 0–15: like subindex 1</td></tr>
</table></td></tr>
<tr><td>2005h</td><td>Current temperature</td><td>ro</td><td>S16</td><td>GEL 235 only (not supported at the moment)</td></tr>
<tr><td>2006h</td><td>Current speed value 16 bits</td><td>ro</td><td>S16</td><td rowspan="2">GEL 235 only
The difference in value is determined from a table of position values at the programmed interval as a measure of the rotational speed of the encoder shaft (gate time, → object 2025h). This value is output in the case of a read access to the object as a signed 16- or 32-bit value (positive sign = rotation in clockwise direction with frontal shaft view).
If using object 2006h the gate time must not be set too high in order that the resulting speed values do not become so high that 16 bits are no longer sufficient for correct indication.</td></tr>
<tr><td>2007h</td><td>Current speed value 32 bits</td><td>ro</td><td>S32</td></tr>
</table>

Index	Designation	Acc.	Type	Meaning
2008h	Current acceleration value 16 bits	ro	S16	GEL 235 only The difference in value is determined from a table of speed values as a measure of the acceleration of the encoder shaft, for which the same gate time as in speed measurements is used (→ object 2025h). The value determined in this way is output in the case of a read access to the object as a signed 16- or 32-bit value (positive sign = rotation in clockwise direction with frontal shaft view). If using object 2008h the gate time must not be set too high in order that the resulting acceleration values do not become so high that 16 bits are no longer sufficient for correct indication.
2009h	Current acceleration value 32 bits	ro	S32	
200Ah	Current rotational speed	ro	S32	GEL 235 only The current value of the recorded number of revolutions can be obtained with this object. The value is output as a signed 32-bit value (positive value = clockwise direction of rotation with frontal shaft view) in reference to the gate time defined in object 2025h.

<table>
<tr><th>Index</th><th>Designation</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>200Bh</td><td>Current rotational direction</td><td>ro</td><td>U16</td><td>GEL 235 only
The evaluation of the rotation status can be obtained with this object. As a result, the direction of rotation can be obtained as a function of the speed or revolutions.
<table>
<tr><th>Bit</th><th>Content</th></tr>
<tr><td>0</td><td>Forward direction of rotation derived from speed calculation</td></tr>
<tr><td>1</td><td>Backward direction of rotation derived from speed calculation</td></tr>
<tr><td>2</td><td>Standstill derived from speed calculation</td></tr>
<tr><td>4</td><td>Forward direction of rotation derived from revolution calculation</td></tr>
<tr><td>5</td><td>Backward direction of rotation derived from revolution calculation</td></tr>
<tr><td>6</td><td>Standstill derived from revolution calculation</td></tr>
</table></td></tr>
<tr><td>200Ch</td><td>Current gear mode (electronic gear)</td><td>ro</td><td>U16</td><td>GEL 235 only
Electronic gear signifies that the counting range and an internal multiplier are automatically adapted for ST and MT parts whenever a value is changed via objects 6001h or 6002h. In this case, the object shows the following results:
<table>
<tr><th>Bit</th><th>Content</th></tr>
<tr><td>0</td><td>no multiplier is active</td></tr>
<tr><td>1</td><td>ST multiplier is active</td></tr>
<tr><td>2</td><td>MT multiplier is active</td></tr>
<tr><td>3</td><td>ST and MT multipliers are active</td></tr>
</table></td></tr>
<tr><td>200Dh</td><td>Current MT counter overflow</td><td>ro</td><td>S32</td><td>GEL 235 only
A MT counter overflow can be queried with this object. The value is not saved power failure-proof.</td></tr>
</table>

<table>
<tr><th>Index</th><th>Designation</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>2010h

2011h</td><td>Current state of work areas 1–16

Current state of work areas 17–32</td><td>ro</td><td>U16</td><td>GEL 235 only

The status of the first 16 activated work area signals can be obtained with this object:
Bit 0: work area signal 1/17
Bit 1: work area signal 2/18
⋮
Bit 15: work area signal 16/32</td></tr>
<tr><td>2015h</td><td>Current encoder configuration</td><td></td><td></td><td>GEL 235 only

The configuration of the absolute encoder can be queried and modified with this object; see the explanatory notes further below.</td></tr>
<tr><td>2020h</td><td>Fault recorder</td><td>ro</td><td>U8</td><td>The last 20 stored faults can be obtained with this object. The fault storage is reset by writing subindex 0. The faults are not saved power failure-proof.
<table>
<tr><th>Sub</th><th>Content</th></tr>
<tr><td>00h</td><td>number of entries = 21 (15h)</td></tr>
<tr><td>01h</td><td>number of occurred faults</td></tr>
<tr><td>02h</td><td>fault no. 1:

0: no fault message existing
1: supply voltage too low
2: watchdog has triggered
3: internal encoder fault
4: data transmission faulty
5: no data on fieldbus</td></tr>
<tr><td>03h – 15h</td><td>faults no. 2 – 20: like no. 1</td></tr>
</table></td></tr>
<tr><td>2021h</td><td>Warning recorder</td><td>ro</td><td>U8</td><td>No function at the moment</td></tr>
</table>

<table>
<tr><th>Index</th><th>Designation</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>2025h</td><td>Gate time</td><td>rw</td><td>U16</td><td>GEL 235 only
This object can be used to set the gate time for the speed and acceleration measurement as a concrete value, and to configure the gate time for the revolution measurement from a table.
<table><tr><th>Sub</th><th>Content</th></tr><tr><td>00h</td><td>number of entries = 2</td></tr><tr><td>01h</td><td>gate time for speed measurement: 1 to 1000 ms, 0 = function inactive</td></tr><tr><td>02h</td><td>gate time for speed measurement from table:
0: function inactive
1: 1 ms
2: 5 ms
3: 10 ms
4: 50 ms
5: 100 ms
6: 250 ms
7: 500 ms
8: 1000 ms
9: 6000 ms (calculated)</td></tr></table></td></tr>
<tr><td>2030h
2031h
2032h</td><td>Function of data outputs 1, 2, 3</td><td>rw</td><td>U8</td><td>GEL 235 only, see explanatory notes further below</td></tr>
<tr><td>2035h
2036h
2037h</td><td>Data outputs 1, 2, 3</td><td>ro</td><td>U32</td><td>GEL 235 only
These objects can be used to obtain the current data assigned to the corresponding data output via objects 2030h to 2032h. Reference is taken to the objects in the PDO mapping (objects 1A00h to 1A02h).</td></tr>
</table>

<table>
<tr><th>Index</th><th>Designation</th><th>Acc.</th><th>Type</th><th>Meaning</th></tr>
<tr><td>2060h</td><td>Business card</td><td>ro</td><td>STR</td><td>GEL 235 only
<table><tr><th>Sub</th><th>Content</th></tr><tr><td>00h</td><td>number of entries = 6</td></tr><tr><td>01h</td><td>Lenord, Bauer & Co. GmbH</td></tr><tr><td>02h</td><td>Dohlenstrasse 32</td></tr><tr><td>03h</td><td>46145 Oberhausen</td></tr><tr><td>04h</td><td>Germany</td></tr><tr><td>05h</td><td>sensors@lenord.de</td></tr><tr><td>06h</td><td>www.lenord.de</td></tr></table></td></tr>
<tr><td>2101h</td><td>Measuring parameters</td><td>rw</td><td>U16</td><td>GEL 2352 only
<table><tr><th>Sub</th><th>Content</th></tr><tr><td>00h</td><td>number of entries = 4 (Data: ro)</td></tr><tr><td>01h</td><td>speed unit
1: increments per second
2: increments per minute
3: revolutions per second
4: revolutions per minute</td></tr><tr><td>02h</td><td>number of readings for speed average (50 to 500)</td></tr><tr><td>03h</td><td>number of readings for acceleration average (50 to 500)</td></tr><tr><td>04h</td><td>gate time for speed and acceleration measurement (0 to 1000 ms)</td></tr></table></td></tr>
<tr><td>2130h

2140h</td><td>Speed value 32 bits

Acceleration value 32 bits</td><td>ro</td><td>S32</td><td>GEL 2352 only
<table><tr><th>Sub</th><th>Content</th></tr><tr><td>00h</td><td>number of entries = 2</td></tr><tr><td>01h</td><td>current value</td></tr><tr><td>02h</td><td>moving average over the number of readings defined via object 2101h</td></tr></table></td></tr>
</table>

Current encoder configuration (object 2015h)

Sub	Function/content	Acc.	Type
00h	number of entries = 14 (0Eh)	ro	U8
01h	number of measuring steps per revolution	ro	U32
02h	number of MT revolutions	ro	U32
03h	calculated preset value	ro	U32
04h	current ST operating range	rw	U32
05h	current MT operating range	rw	U32
06h	preset value	rw	U32
07h	zero shift, see explanation further below	rw	S32
08h	count direction of the encoder: cw (0), ccw (1)	rw	U32
09h	MT count overflow (→ object 200Dh)	rw	S32
0Ah	reset preset value (1)	rw	U32
0Bh	enable actual value multiplier function (1), see explanation further below	rw	U32
0Ch	actual value multiplier numerator	rw	U32
0Dh	actual value multiplier denominator	rw	U32
0Eh	validation of actual value multiplier and activation if it is correct (1)	rw	U32

Function of data output 1, 2, 3 (objects 2030h, 2031h, 2032h)

PDO structure

COB-ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Type 0	Position value 32 bits				—	—	—	—
Type 1	Position value 32 bits				Data output 32 bits			
Type 2	Position value 32 bits				Data output 16 bits		Data output 16 bits	
Type 3	Position value 64 bits							

Function assignment

Sub	Function/content
00h	number of entries = 4
01h	selection of PDO type: 0: type 0 = sole output of the position as a 32-bit value * 1: type 1 = output of the position and one 32-bit data value (default) ** 2: type 2 = output of the position and two 16-bit data values ** 3: type 3 = sole output of the position as a 64-bit value ** * GEL 2035 and GEL 2352 ST: 16 bits ** GEL 235 only

Sub	Function/content
02h	function of the 32-bit data value (bytes 5–8): 0: no output, function inactive 1: absolute position value (hardware reading) 2: relative position value (with multiplier, but without preset and offset) 4: speed value 5: acceleration value 6: rotational speed value 7: MT count overflow 8: work area signals 1–32 9: ST position 10: MT count
03h, 04h	function of the 16-bit data values (bytes 5+6 and 7+8): 0: no output, function inactive 1: absolute position value (hardware reading), low word 2: absolute position value (hardware reading), high word 3: relative position value (with multiplier, but without preset and offset), low word 4: relative position value (with multiplier, but without preset and offset), high word 7: speed value, low word 8: speed value, high word 9: acceleration value, low word 10: acceleration value, high word 11: rotational speed value, low word 12: rotational speed value, high word 13: MT count overflow, low word 14: MT count overflow, high word 15: work area signals 1–16 16: work area signals 17–32 17: ST position 18: MT count

4 SDO communication

The service data objects (SDO) represent the communication channel for the transmission of device parameters (e.g., programming of the encoder resolution). Since these parameters are transferred acyclically (e.g., once only during start-up of the network), the SDOs have a low priority (high COB identifier).

Structure of the SDO telegram

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
SDO identifier	Data length	Command	Index L	Index H	Subindex	Data 0	Data 1	Data 2	Data 3

The SDO identifier is defined as follows:

Client (control) → Server (absolute encoder): **600**h + node ID

Server (absolute encoder) → Client (control): **580**h + node ID

The data length (DLC) is always 8: 1 command byte + 2 index bytes (object) + 1 subindex byte + 4 data bytes

The command defines whether data is to be written (download) or read (upload), and how many bytes of payload there are:

Command	Description	Payload	Function
22h	SDO(rx), download request	undefined	Send parameters to the absolute encoder
23h		4 bytes	
2Bh		2 bytes	
2Fh		1 byte	
60h	SDO(tx), download response	—	Confirm acceptance of parameters to the control
40h	SDO(rx), upload request	—	Request parameters from the absolute encoder
42h	SDO(tx), upload response	undefined	Send parameters to the control
43h		4 bytes	
4Bh		2 bytes	
4Fh		1 byte	
80h	SDO(tx), abort domain transfer (because of a fault)	4 bytes	Encoder reports error code to the control

In the case of an error, an error message with the 80h command (SDO abort message) replaces the normal response. The index and subindex belong to the previously specified object. The output error code is contained in the bytes 5 to 8 as follows:

Structure of the error telegram

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
580h + node ID	8	80h	Index L	Index H	Subin-dex	ErrByte 0	ErrByte 1	ErrByte 2	ErrByte 3

Possible error messages

ErrByte 3	ErrByte 2	ErrByte 1	ErrByte 0	Error
05h	04h	00h	01h	command byte not supported
06h	01h	00h	00h	illegal access to an object
06h	01h	00h	01h	read access to a write-only object
06h	01h	00h	02h	write access to a read-only object
06h	02h	00h	00h	object not supported
06h	09h	00h	11h	subindex not supported
06h	09h	00h	30h	parameter value beyond the limits
06h	09h	00h	31h	parameter value too large
06h	09h	00h	32h	parameter value too small
08h	00h	00h	00h	general error
08h	00h	00h	20h	false saving signature (“save”)
08h	00h	00h	21h	parameters cannot be saved