Linear-Encoders
CANopen Profile
2 Sensors, Position and Speed

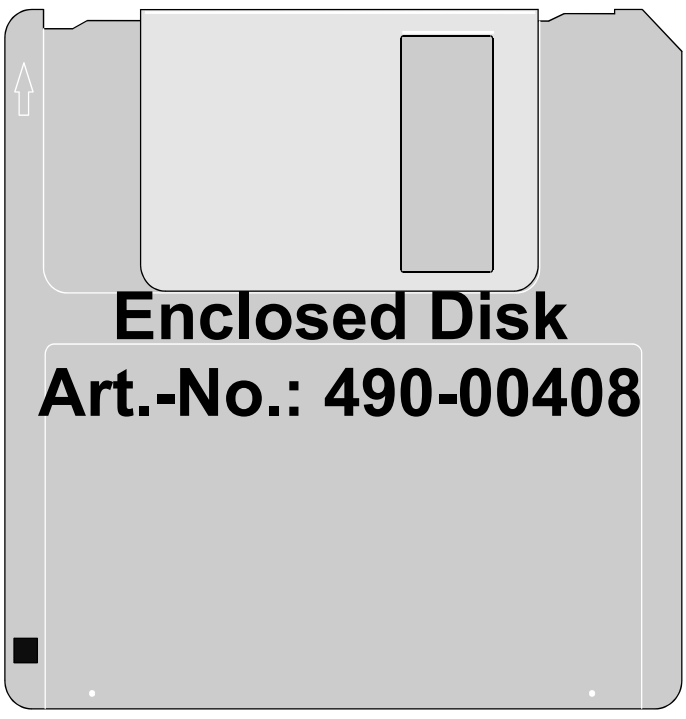
Technical Information

Please keep for further use !

Edition date/Rev. date: 19.07.2002
Document no./Rev. no.: TR - ELA - TI - GB - 0039 - 00
Software version: CiA DSP-406 V2.0
File name: TR-ELA-TI-GB-0039.DOC
Author: MÜJ / BER

TR - Electronic GmbH
Eglishalde 6
D-78647 Trossingen

Telephone + 49 (0) 7425 / 228-0
Telefax + 49 (0) 7425 / 228-33



Imprint

TR-Electronic GmbH
D-78647 Trossingen
Eglishalde 6
Tel.: (0049) 07425/228-0
Fax: (0049) 07425/228-33

Guarantee

In our ongoing efforts to improve our products, TR-Electronic reserve the right to alter the information contained in this document without prior notice.

Printing

This manual was produced on a personal computer using MS-Word for Windows. The text was printed in *Arial* type.

Fonts

Italics and **bold** type are used for the title of a document or to emphasize text passages.

Copyright Information ©

MS-WORD and Windows are registered trademarks of Microsoft Corporation.

Revision History

i

Note:
The cover of this document shows the current revision status and the corresponding date. Since each individual page has its own revision status and date in the footer, there may be different revision statuses within the document.

Document created: 19.07.2002

Revision	Date

Table of contents

1 References

1	:	ISO 11898: Road Vehicles Interchange of Digital Information - Controller Area Network (CAN) for high-speed Communication, November 1993
2	:	Robert Bosch GmbH, CAN Specification 2.0 Part A and B, September 1991
3	:	CiA DS-201 V1.1, CAN in the OSI Reference Model, February 1996
4	:	CiA DS-202-1 V1.1, CMS Service Specification, February 1996
5	:	CiA DS-202-2 V1.1, CMS Protocol Specification, February 1996
6	:	CiA DS-202-3 V1.1, CMS Encoding Rules, February 1996
7	:	CiA DS-203-1 V1.1, NMT Service Specification, February 1996
8	:	CiA DS-203-2 V1.1, NMT Protocol Specification, February 1996
9	:	CiA DS-204-1 V1.1, DBT Service Specification, February 1996
10	:	CiA DS-204-2 V1.1, DBT Protocol Specification, February 1996
11	:	CiA DS-205-1 V1.1, LMT Service Specification, February 1996
12	:	CiA DS-205-2 V1.1, LMT Protocol Specification, February 1996
13	:	CiA DS-206 V1.1, Recommended Layer Naming Conventions, February 1996
14	:	CiA DS-207 V1.1, Application Layer Naming Conventions, February 1996
15	:	CiA DS-301 V3.0, CANopen Communication Profile based on CAL, October 1996
16	:	CiA DS-406 V2.0, CANopen Profile for Encoder, May 1998

2 Definitions and Abbreviations

CAL

CAN Application Layer. The application layer for CAN-based networks as specified by CiA in Draft Standard 201 ... 207.

CAN

Controller Area Network. Data link layer protocol for serial communication as specified in ISO 11898.

CiA

CAN in Automation international manufacturer and user organisation e.V.: non-profit association for Controller Area Network (CAN).

CMS

CAN-based Message Specification. One of the service elements of the application layer in the CAN Reference Model.

COB

Communication Object. (CAN Message) A unit of transportation in a CAN Network. Data must be sent across a Network inside a COB.

COB-ID COB-Identifier. Identifies a COB uniquely in a Network. The identifier determines the priority of that COB in the MAC sub-layer too.

DBT

Distributor. One of the service elements of the application in the CAN Reference Model. It is the responsibility of the DBT to distribute COB-ID´s to the COB´s that are used by CMS.

LMT

Layer Management. One of the service elements of the application in the CAN Reference Model. It serves to configure parameters of each layer in the CAN Reference Model.

NMT

Network Management. One of the service elements of the application in the CAN Reference Model. It performs initialisation, configuration and error handling in a CAN network.

PDO

Process Data Object. Object for data exchange between several devices.

SDO

Service Data Object. Peer to peer communication with access to the Object Dictionary of a device.

3 Introduction

The CAN-Bus-Interface is defined by the international norm ISO/DIS 11898 and specifies the two lowest layers of the ISO/DIS CAN Reference Model.

The CAN-BUS-Interface with the BUS-Driver PCA83C250T is galvanic isolated of the encoder electronic and becomes the power over internal DC/DC-converter. There is no external power supply necessary for the CAN-BUS-Driver.

The conversion of the encoder information to the CAN message format (CAN 2.0A) is done by the CAN-controller PCA82C200. The function of the CAN-controller is controlled by a watchdog .

The CANopen Communication Profile (CIA standard DS 301) is a subset of CAN Application Layer (CAL) and describes, how the services are used by devices. The CANopen Profile allows the definition of device profiles for decentralised I/O.

The encoders with CANopen-protocol support the Device Profile for Encoder (CIA Draft Standard Proposal 406, Version 2.0). The encoders support the extended functions in Class C2 .

The communication functionality and objects, which are used in the encoder profile, are described in a EDS-File (Electronic Data Sheet).
When using a CANopen Configuration Tool (e.g.:CANSETTER), the user can read the objects of the encoder (SDOs) and program the functionality.

The selection of transmission rate and node number is done by hardware (switches).

4 The communication profile

Two process data objects (PDO) are implemented in the device. The transmission modes are defined as follows:

Transmission mode = 1: Transmitting is carried out synchronously on request via remote frame or SYNC-telegram.
Transmission mode = 254: Transmitting is carried out asynchronously. The cyclic timer is stored in index 6200h.

The output position value is transmitted in binary code:

COB-ID	**Output Position Value**				**Speed / CAN state**		
11 Bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
	2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}	2^7 to 2^0	2^{15} to 2^8	0

4.1 1st transmit PDO Communication Parameter

This PDO transmit the position value of the encoder of the first sensor.

Index	**Sub-Index**	**Comment**	**Default Value**
1800h	0	number of supported entries	3
	1	COB-ID used by PDO 1	180h + Node-ID
	2	transmission type	1
	3	inhibit time	0
1A00h	0	number of mapped objects	3
	1	Position value, 1. sensor	60200120h
	2	Speed, 1. sensor	60300110h
	3	CAN state, 1. sensor	63000108h

4.2 2nd transmit PDO Communication Parameter

This PDO transmit the position value of the encoder of the second sensor.

Index	**Sub-Index**	**Comment**	**Default Value**
1801h	0	number of supported entries	3
	1	COB-ID used by PDO 2	280 + Node-ID
	2	transmission type	1
	3	inhibit time	0
1A01h	0	number of mapped objects	3
	1	Position value, 2. sensor	60200220h
	2	Speed, 2. sensor	60300210h
	3	CAN state, 2. sensor	63000208h

5 Standard objects

Following table gives an overview on the supported indices in the Communication Profile Area:

Index (h)	Object	Name	Type	Attr.
1000	VAR	device type	Unsigned32	const
1001	VAR	error register	Unsigned8	ro
1002	VAR	manufacturer status register	Unsigned32	ro
1003	ARRAY	pre-defined error field	Unsigned32	ro
1004	ARRAY	Number of PDOs supported	Unsigned32	ro
1005	VAR	COB-ID SYNC-message	Unsigned32	rw
1008	VAR	device name	Vis-String	const
1009	VAR	hardware version	Vis-String	const
100A	VAR	software version	Vis-String	const
100B	VAR	Node-ID	Unsigned32	ro
100C	VAR	guard time	Unsigned32	rw
100D	VAR	life time factor	Unsigned32	rw
100E	VAR	COB-ID guarding protocol	Unsigned32	ro
1010	VAR	store parameters	Unsigned32	rw

5.1 Detailed Specification of Communication specific objects

5.1.1 Object 1000h: Device type

Contains information about the device type. The object at index 1000h describes the type of device and its functionality. It is composed of a 16 bit field which describes the device profile that is used (Device Profile Number 406 = 196h) and a second 16 bit field which gives information on the type of encoder.

Structure of parameter

Unsigned32

Device Type			
Device Profile Number		Encoder Type	
Byte 0	Byte 1	Byte 2	Byte 3
196h		2^7 to 2^0	2^{15} to 2^8

Encoder type

Code	Definition
01	Single-Turn absolute rotary encoder
02	Multi-Turn absolute rotary encoder
08	Absolute linear encoder
09	Absolute linear encoder with cyclic coding

5.1.2 Object 1001h: Error Register

This object is an error register for the device. If an alarm bit is set (object 6503), bit 5 is set in the error register.

Unsigned8

Bit	Meaning
0	generic error
1	0
2	0
3	0
4	0
5	device profile specific
6	0
7	0

5.1.3 Object 1002h: Manufacturer Status Register

This object is not used by the encoder, by read access the value is always „0“.

5.1.4 Object 1003h: Pre-defined Error Field

This object holds the error that have occurred on the encoder and have been signalled via the Emergency object.

Index	Sub-Index	Comment	Type
1003H	0	number of errors	Unsigned8
	1	standard error field	Unsigned32

Sub-index 0: The entry at sub-index 0 contains the number of errors that have occurred and recorded in sub-index 1.

Sub-index 1: The error are composed of a 16bit error code and a 16bit additional error information.

Unsigned32

Standard Error Field			
Byte 0	Byte 1	Byte 2	Byte 3
Error code		Additional Information	

5.1.5 Object 1004h: Number of PDOs supported

This object contains information about the maximum number of PDOs supported by the encoder.

Index	Sub-Index	Comment	Type
1004H	0	number of PDOs supported	Unsigned32
	1	number of synchronous PDOs	Unsigned32
	2	number of asynchronous PDOs	Unsigned32

Sub-index 0 describes the overall number of PDOs supported (synchronous and asynchronous).
Sub-index 1 describes the number of synchronous PDOs supported by the encoder.
Sub-index 2 describes the number of asynchronous PDOs supported by the encoder.

Number of PDOs			
Byte 0	Byte 1	Byte 2	Byte 3
Transmit PDOs		Receive PDOs	

Sub-index 0: Transmit PDOs = 2, Receive PDOs = 0
Sub-index 1: Transmit PDOs = 1, Receive PDOs = 0
Sub-index 2: Transmit PDOs = 1, Receive PDOs = 0

5.1.6 Object 1005h: COB-ID SYNC message

This object defines the COB-ID of the Synchronisation Object (SYNC). Further, it defines whether the device consumes the SYNC or whether the device generates the SYNC.

Unsigned32
MSB LSB

31	30	29	28-11	10-0
1	0	0	0	00 1000 0000

bit 31 = 1 , Device consumes SYNC message
bit 30 = 0 , Device does not generate SYNC message
bit 29 = 0 , 11-bit ID (CAN 2.0A)
bit 28 -11 = 0
bit 10 - 0 = 11-bit SYNC-COB-IDENTIFIER, default Value = 080H

If a SYNC-telegram with the identifier, defined in this object (080H), and data length = 0 has been received by the device, the position value of the encoder is transmitted by the 2nd Transmit PDO (object 1802).

5.1.7 Object 1008h: Manufacturer Device Name

contains the manufacturer device name (visible string).

5.1.8 Object 1009h: Manufacturer Hardware Version

contains the manufacturer hardware version (visible string).

5.1.9 Object 100Ah: Manufacturer Software Version

contains the manufacturer software version (visible string).
See also object 6507.

5.1.10 Object 100Bh: Node-ID

This object contains the Node-ID.

The value is selected by 6 hardware switches and cannot be changed using SDO services.

Unsigned32

Node_ID			
Byte 0	Byte 1	Byte 2	Byte 3
Node-ID	reserved	reserved	reserved

The value range is 1 - 64.

The Node-ID is the selected hardware number by switches + 1. That means:

all 6 switches off = 0 , Node-ID = 1
switch Bit 5 = on = 32, Node-ID = 33

5.1.11 Object 100CH: Guard Time

The objects at index 100CH and 100DH include the guard time in milli-seconds and the life time factor. The life time factor multiplied with the guard time gives the live time for the Node Guarding Protocol.

Unsigned16

Guard Time	
Byte 0	Byte 1
2^7 to 2^0	2^{15} to 2^8

5.1.12 Object 100DH: Life Time Factor

The life time factor multiplied with the guard time gives the life time for the node guarding protocol.

Unsigned8

Life Time Factor
Byte 0
2^7 to 2^0

5.1.13 Object 100EH: Node Guarding Identifier

The identifier is used for the node guarding and the life guarding procedure.

Unsigned32

MSB				LSB
31	30	29	28-11	10-0
reserved		0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-bit Identifier

Bit 10 - 0 = 11-bit identifier, value = 700H + Node-ID

5.1.14 Object 1010h: Store Parameters

This object supports the saving of parameters in non volatile memory (EEPROM).

Index	Sub-Index	Comment	Type
1010H	0	largest supported Sub-Index	Unsigned8
	1	save all parameters	Unsigned32

Sub-Index0: The entry at sub-index 0 contains the largest Sub-Index that is supported.
Value = 1.

Sub-Index1: By read access the device provides information about ist saving capability.

Unsigned32

MSB			LSB
bits	31-2	1	0
value	= 0	0	1

By read access the device provides information about its saving capability.

Bit 0 = 1, the device saves parameters only on command. That means, if parameters have been changed by the user and no „Store Parameter Command“ had been executed, at the next power on , the parameters will have there old values.

By write access the device stores the parameters to the non volatile memory.
In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the object. The signature is „save“.

Unsigned32

Signature MSB			LSB
e	v	a	s
65h	76h	61h	73h

On reception of the correct signature, the device stores the parameters. If the storing failed, the device responds with abort domain transfer, error class 6 , error code 6 (hardware fault). See also object 6503h.

If a wrong signature is written, the device refuses to store and responds with abort domain transfer, error class 8, error code 0.

6 Standardized encoder profile area

Each encoder shares the dictionary entries from 6000h to 65FFh. These entries are common to encoders.
NOTE that all indices shown in the "Index" column are hexadecimal.

The overview of all common entries are shown below:

Index	Object	Name	Data Length	Attr.
Parameters				
6000	VAR	Operating Parameters	Unsigned16	rw
6001	VAR	Measuring units per revolution	Unsigned32	rw
6002	VAR	Total measuring range in measuring units	Unsigned32	rw
6003	VAR	Preset value	Unsigned32	rw
6004	VAR	Position value	Unsigned32	ro
6005	REC	Linear encoder measuring step settings	Unsigned32	ro
6010	VAR	Preset value multi-sensor	Unsigned32	rw
6020	VAR	Position value multi-sensor	Unsigned32	r
6200	VAR	Cyclic timer	Unsigned16	rw
Diagnostics				
6500	VAR	Operating status	Unsigned16	r
6501	VAR	Single-Turn resolution (rotary), Measuring step (linear)	Unsigned32	r
6502	VAR	Number of distinguishable revolutions	Unsigned16	r
6503	VAR	Alarms	Unsigned16	r
6504	VAR	Supported alarms	Unsigned16	r
6505	VAR	Warnings	Unsigned16	r
6506	VAR	Supported Warnings	Unsigned16	r
6507	VAR	Profile and software version	Unsigned32	r
6508	VAR	Operating time	Unsigned32	r
6509	VAR	Offset value	Signed32	r
650A	VAR	Manufacturer offset value	Signed32	r
650B	VAR	Serial number	Unsigned32	r

On the following pages each single object is explained in detail.

6.1 Encoder parameters

6.1.1 Object 6000h - Operating Parameters

The Operating Parameters contain the functions for code sequence, commissioning diagnostic control and scaling function control.

Structure of parameter
Unsigned16

Bit	Function	Bit = 0	Bit = 1
0	Code Sequence	CW	CCW
1	Commissioning Diagnostic Control	Disabled	Enabled
2	Scaling function control	Disabled	Enabled
3 - 11	Reserved for further use		
12-15	Manufacturer specific functions		

Code sequence:

The code sequence defines whether increasing or decreasing position values are output when the encoder shaft rotates clockwise or counter clockwise as seen on the shaft. For linear encoders it means increasing or decreasing position values to the end.

0 = increasing, 1 = decreasing

Commissioning diagnostic control:

Not supported.

Scaling function control:

With the scaling function the encoder numerical value is converted in software to change the physical resolution of the encoder.
The parameters "Measuring units per revolution" and "Total measuring range in measuring units" are the scaling parameters. The scaling function bit is set in the operating parameters. If the scaling function bit is cleared, the scaling function is disabled and the two parameters return to their standard values.

Before writing the "Measuring units per revolution" and "Total measuring range in measuring units" to the encoder, the scaling function bit has to be set.
After writing the parameters, the Store Parameter command has to be executed to save the new parameter values.

6.1.2 Object 6001h - Measuring units per revolution

The parameter "Measuring units per revolution" sets the number of distinguishable steps per revolution.

Structure of parameter

Unsigned32

Measuring units per revolution			
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

Rotary Encoder

Standard value: 4096 = 1000H (depending on capacity, marked on the rating plate).

Linear Encoder

For linear encoder the value is always „1“.

6.1.3 Object 6002h - Total measuring range in measuring units

The parameter "Total measuring range in measuring units" sets the number of distinguishable steps over the total measuring range.

Structure of parameter

Unsigned32

Total measuring range in measuring units			
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

Rotary Encoder

Standard value: 16 777 216 = 1 00 00 00h
(depending on capacity, marked on the rating plate).

Measuring length in units = Measuring units per revolution x Number of revolutions

Linear Encoder

Standard value: The given measuring length on the rating plate multiplied with 100 corresponding to the resolution of 0,01mm

$$\text{Measuring length in increments} = \frac{\text{Measuring length}}{\text{Resolution in mm}}$$

6.1.4 Object 6003h - Preset value

The Preset Function can be used to adjust the encoder to any position value within a range of 0 to measuring length in increments -1.
The output position value of the first sensor is set to the preset value when writing to this object. The position of the second sensor is set thereby regarding the first sensor. Independent presets see object 6010h.

Structure of parameter
Unsigned32

Preset value			
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

6.1.5 Object 6004h - Position value

The object 6004h "Position value" defines the output position value for the communication object 1800h. It is output the position value of the first sensor.

Structure of parameter
Unsigned32

Position value			
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

6.1.6 Object 6005h - Linear encoder measuring step settings

The parameter "Linear encoder measuring step settings" defines the measuring step settings for the position- and speed-value for linear encoders.

Index	Sub-Index	Comment	Type
6005H	0	number of entries	Unsigned8
	1	measuring step	Unsigned32
	2	speed	Unsigned32

Unsigned32

Measuring Step / Speed			
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

Measuring step: Value in 0.001 µm
E.g.: 0.01 mm = 00 00 27 10 h = 10 000 dec.

Speed: Value in 0.01 mm / s
E.g.: 0.1 mm / s = 0Ah = 10 dec.

6.1.7 Object 6010h - Preset value multi-sensor

The parameter "Preset value multi-sensor" is similar to object 6003h. In sub-index 00h, the number of supported sensors is defined.
The output position values in the sub-indices of object 6020h are set to the sub-indices of the parameter "Preset value" in object 6010h, accordingly.
Each sensor can be set to his individual position independent of the other sensor.

Index	Sub-Index	Comment	Type
6010H	0	number of available sensors	Unsigned8
	1	Preset value sensor 1	Unsigned32
	2	Preset value sensor 2	Unsigned32

Structure of parameter
Unsigned32

Preset value multi-sensor			
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

6.1.8 Object 6020h - Position value multi-sensor

Similar to object 6004h this object defines the output position values for the communication objects 1800h and 1801h.

Index	Sub-Index	Comment	Type
6020H	0	number of available sensors	Unsigned8
	1	Position value sensor 1	Unsigned32
	2	Position value sensor 2	Unsigned32

Structure of parameter
Unsigned32

Position value multi-sensor			
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

6.1.9 Object 6200h - Cyclic timer

Defines the parameter "Cyclic timer". A Cyclic transmission of the position value is set, when the cyclic timer is programmed > 0. Values between 1 ms and 65535 ms can be selected.

E.g.: 1 ms = 1 h
256 ms = 100 h

When the encoder is started with the NODE START Command and the value of the cyclic timer is > 0, the 1st transmit PDO (object 1800h) transmit the encoder position.

6.2 Encoder diagnostics

All encoder diagnostics are read from securely stored parameters.

6.2.1 Object 6500h - Operating status

This object contains the operating status of the encoder. It gives information on encoder internal programmed parameters.

Structure of parameter
Unsigned16

Bit	Function	Bit = 0	Bit = 1
0	Code Sequence	CW	CCW
1	Commissioning Diagnostic Control	Not supp.	Supp.
2	Scaling function control	Disabled	Enabled
3 - 11	Reserved for further use		
12- 15	Manufacturer specific functions		

6.2.2 Object 6501h - Single-Turn resolution (rotary), Measuring step (linear)

The Single-Turn resolution in object 6501h has different contents depending on the encoder type.

6.2.2.1 Rotary encoder

For rotary encoders object 6501h gives the number of measuring steps per revolution that are output for the absolute single-turn position value. The maximum single-turn resolution is 2^{32}.

Structure of parameter
Unsigned32

Single-Turn resolution			
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

Standard value: 4096 = 1000H increments per revolution (depending on capacity- marked on rating plate), if no scaling.

When scaling function enabled (object 6000H), the value is the programmed number of increments per revolution.

6.2.2.2 Linear encoders

For linear encoders object 6501h indicates the measuring step that is output by the encoder. The measuring step is given in nm (0.001µm).
E.g.: 1µm = 00 00 03 E8 h

Structure of parameter
Unsigned32

Measuring step			
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

Standard value is 2710 h = 10 000 = 0,01mm

6.2.3 Object 6502h - Number of distinguishable revolutions

This object contains the number of distinguishable revolutions that the encoder can output.

Rotary Encoder

For a Multi-Turn encoder the number of distinguishable revolutions and the Single-Turn resolution gives the measuring range according to the formula below. The maximum number of distinguishable revolutions is 65536 (16 bits).

Measuring range = Number of distinguishable revolutions x Single-Turn resolution

Standard value: 4096 = 1000H revolutions (depending on capacity - marked on rating plate), if no scaling.

When scaling function is enabled (object 6000H), the value is the programmed number of revolutions.

Linear Encoder

For linear encoder the value is always "1".

6.2.4 Object 6503h - Alarms

Additionally to the emergency message, object 6503h provides further alarm messages. An alarm is set if a malfunction in the encoder could lead to incorrect position value. If an alarm occurs, the according bit is set to logical high until the alarm is cleared and the encoder is able to provide an accurate position value.

Structure of parameter

Unsigned16

Bit	Function	Bit = 0	Bit = 1
0	Position error	No	Yes
1	Commissioning diagnostics	OK	Error
2	Reserved for further use		
3	Reserved for further use		
4	Reserved for further use		
5	Reserved for further use		
6	Reserved for further use		
7	Reserved for further use		
8	Reserved for further use		
9	Reserved for further use		
10	Reserved for further use		
11	Reserved for further use		
12	EE-PROM Error	OK	Error
13	Parameter Error	OK	Error
14	Manufacturer specific functions		
15	Manufacturer specific functions		

Position Error:

The bit is set, if a rotary encoder detects a malfunction of the system or if a linear encoder has no magnet detected.

Commissioning Diagnostics:

This is not supported

EE-PROM Error:

The encoder detects a wrong checksum in the EEProm area or a writing to the EEProm had not been successful.

Parameter Error:

The value of a transmitted parameter is out of range. The check will be done after receiving the Store Parameter command.

6.2.5 Object 6504h - Supported alarms

Object 6504h contains the information on supported alarms by the encoder.

Structure of parameter
Unsigned16

Bit	Function	Bit = 0	Bit = 1
0	Position error	No	Yes
1	Commissioning diagnostics	No	Yes
2	Reserved for further use		
3	Reserved for further use		
4	Reserved for further use		
5	Reserved for further use		
6	Reserved for further use		
7	Reserved for further use		
8	Reserved for further use		
9	Reserved for further use		
10	Reserved for further use		
11	Reserved for further use		
12	EE-PROM Error	No	Yes
13	Parameter Error	No	Yes
14	Manufacturer specific functions		
15	Manufacturer specific functions		

6.2.6 Object 6505h - Warnings

This object is not supported.
By read access the value is always „0“ .

6.2.7 Object 6506h - Supported warnings

This object is not supported.
By read access the value is always „0“ .

6.2.8 Object 6507h - Profile and software version

This object contains in the 1st 16 bits the profile version which is implemented in the encoder. It is combined to a revision number and an index.

E.g.: Profile version: 1.40
Binary code: 0000 0001 0100 0000
Hexadecimal: 1 40

The 2nd 16 bits contain the software version which is implemented in the encoder. Only the last 4 number are available.

E.g.: Software version: 5022.01
Binary code: 0020 0020 0000 0001
Hexadecimal: 22 01

Structure of parameter
Unsigned32

Profile version		Software version	
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^7 to 2^0	2^{15} to 2^8

6.2.9 Object 6508h - Operating time

This object is not supported.

The operating time function is not used the operating time value is set to the maximum value (FF FF FF FF h).

6.2.10 Object 6509h - Offset value

This object contains the offset value calculated by the preset function. The offset value is stored and can be read from the encoder.

6.2.11 Object 650Ah - Manufacturer offset value

This object is not supported.

By read access the offset value is „0“.

6.2.12 Object 650Bh - Serial number

This object is not supported.

The parameter serial number is not used the value is set to maximum value
FF FF FF FF h.

7 Emergency Message

Emergency messages are triggered by the occurrence of a device internal malfunction and are transmitted from the concerned application device to the other devices with highest priority.

Emergency Message								
Byte	0	1	2	3	4	5	6	7
content	Emergency Error Code		Error register (object 1001H)	0	0	0	0	0

COB-Identifier = 080H + Node-ID

If the encoder detects an internal error, a emergency message will be transmitted with the error code of object 1003H (pre-defined error field) and the error register object 1001H. Additionally to the emergency object the according bit in the Alarm object 6503H is set.

If the error disappears, the encoder transmits an emergency message with error code "0" (reset error / no error) and error register "0".

8 Start up of Encoder on CAN-BUS

Before connecting the encoder to the bus, select the baud rate and the Node-ID with the hardware switches. The switches are only read at power-on sequence.

After power on and finishing the initialisation, the encoder goes to the Pre-Operational state and waits for command. If the encoder detects an internal error, a emergency message with the error code will be transmitted.

During this state parameterisation via SDO (e.g. using a configuration tool) is possible.

8.1 Transmit encoder position value:

Before the encoder position can be transferred the encoder has to be started with the Node Start command.

Node Start Protocol

COB-Identifier = 0

Byte 0	Byte 1
1	Node-ID

Node Start command with the Node-ID of the encoder (slave) starts only this device.

Node Start command with **Node-ID = 0** starts all slaves connected to the bus.

After the Node Start command the encoder transmit the position value one time with the COB-ID of object 1800h.

Now the encoder position value can be transmitted in different ways:

Asynchronous Transmission

The transmission PDOs with which the transmission mode is set to 254, transmit the position value of the sensors. The cyclic time is defined by the value of the cyclic timer (object 6200H). This transmission starts automatically after the Node Start command and the value of the cyclic timer is > 0.

Cyclic Transmission

The transmission PDOs with which the transmission mode is set to 1, transmit the position value of the sensors on request (remote / sync).

The encoder receives a remote frame with the COB-ID of the PDO.

The encoder receives a sync telegram with the COB-ID (default value 080h) defined in object 1005h. All slaves with this SYNC-COB-ID and PDO transmission mode = 1 will transmit the position value.
To stop the transmission of the encoder position the encoder has to be stopped with the Node Stop command.

Node Stop Protocol

COB-Identifier = 0

Byte 0	Byte 1
2	Node-ID

Node Stop command with the Node-ID of the encoder (slave) stop only this device.

Node Stop command with **Node-ID = 0** stop all slaves connected to the bus.

8.2 Read/Write Service Data Object

The transfer of the Service Data Object (SDO) is done by the CMS Multiplexed Domain protocol (CIA DS202/2) .

8.2.1 Read SDO:

(Initiate Domain Upload)

Request Protocol format:

COB-Identifier = 600H + Node-ID

Read SDOs								
Byte	0	1	2	3	4	5	6	7
content	Code	Index		Sub-index	Data 0	Data 1	Data 2	Data 3
	40h	low	high	byte	0	0	0	0

The Read SDO telegram has to be send to the slave.

The slave answers with the following telegram:

Response Protocol format:

COB-Identifier = 580H + Node-ID

Read SDOs								
Byte	0	1	2	3	4	5	6	7
content	Code	Index		Sub-index	Data 0	Data 1	Data 2	Data 3
	4xh	low	high	byte	data	data	data	data

Format Byte 0:

MSB LSB

7	6	5	4	3	2	1	0
0	1	0	0	n		1	1

n = number of data bytes (bytes 4-7) that does not contain data.

If only 1 data byte (Data 0) contains data the value of byte 0 is "4FH".

If byte 0 = 80h the transfer has been aborted.

8.2.2 Write SDO:

(Initiate Domain Download)

Request Protocol format:

COB-Identifier = 600H + Node-ID

Write SDOs								
Byte	0	1	2	3	4	5	6	7
content	Code	Index		Sub-index	Data 0	Data 1	Data 2	Data 3
	2xh	low	high	byte	0	0	0	0

Format Byte 0:

MSB LSB

7	6	5	4	3	2	1	0
0	0	1	0	n		1	1

n = number of data bytes (bytes 4-7) that does not contain data.

If only 1 data byte (Data 0) contains data the value of byte 0 is "2FH".

The Write SDO telegram has to be send to the slave.

The slave answers with the following telegram:

Response Protocol format:

COB-Identifier = 580H + Node-ID

Read SDOs								
Byte	0	1	2	3	4	5	6	7
content	Code	Index		Sub-index	Data 0	Data 1	Data 2	Data 3
	60h	low	high	byte	0	0	0	0

If byte 0 = 80h the transfer has been aborted.