ifm electronic

UK

Device manual Encoder with CANopen interface

efector400®

RM7xxx
RN7xxx

704434/01 08 / 2010

Contents

UK

Licences and trademarks

1 Preliminary note

1.1 Symbols used

►	Instruction
>	Reaction, result
[…]	Designation of pushbuttons, buttons or indications
→	Cross-reference
❗	Important note Non-compliance can result in malfunctions or interference.
ℹ	Information Supplementary note

1.2 Warning signs used

ATTENTION

Warning of damage to property.

2 Safety instructions

This manual is part of the device. It contains information and illustrations about the correct handling of the device and must be read before installation or use.

Observe the operating instructions.

Non-observance of the instructions, operation which is not in accordance with use as prescribed below, wrong installation or handling can affect the safety of people and machinery.

The installation and connection must comply with the applicable national and international standards. Responsibility lies with the person installing the unit.

Only the signals indicated in the technical data or on the device label may be supplied to the connections or wires.

3 General information

3.1 Absolute encoders

For an absolute encoder each angular position is assigned a coded position value. This is generated by a coded disc with several parallel fine code segments which are detected individually. For singleturn encoders, i.e. an encoder producing absolute positions within one revolution, the absolute position information is repeated with every revolution. A multiturn encoder can also distinguish between revolutions via a gearbox containing magnets which are individually detected by Hall elements. The number of the individual revolutions is determined by the resolution of the multiturn detection and is repeated after the total resolution has been reached.

3.2 CANopen technology

The CANopen communication profile is based on the CAN Application Layer (CAL) specification from CiA (CAN in Automation). CANopen is considered as a robust fieldbus with highly flexible configuration options. It is used in many various applications which are based on different application profiles. CANopen comprises a concept to configure and communicate real-time data using synchronous and asynchronous messages. Four message types (objects) are distinguished.

1. Administration messages (layer management, network management and identifier distribution)
2. Service data messages (SDO)
3. Process data messages (PDO)
4. Predefined messages (synchronisation, time stamp, emergency)

You can find more information in the CANopen specification.

3.2.1 Certification of CANopen products

To achieve interoperability and a suitable device functionality CANopen products are approved by external notified bodies. A copy of the certificate is attached to this manual.

3.3 References

http://www.can-cia.org

CAN Application Layer, DS 201…207	CiA
CAL based communication profile, DS 301	CiA
Device profiles for encoders, DS 406	CiA
CAN specification version 2.0 A	Robert Bosch GmbH
CANary CAN controller	Atmel

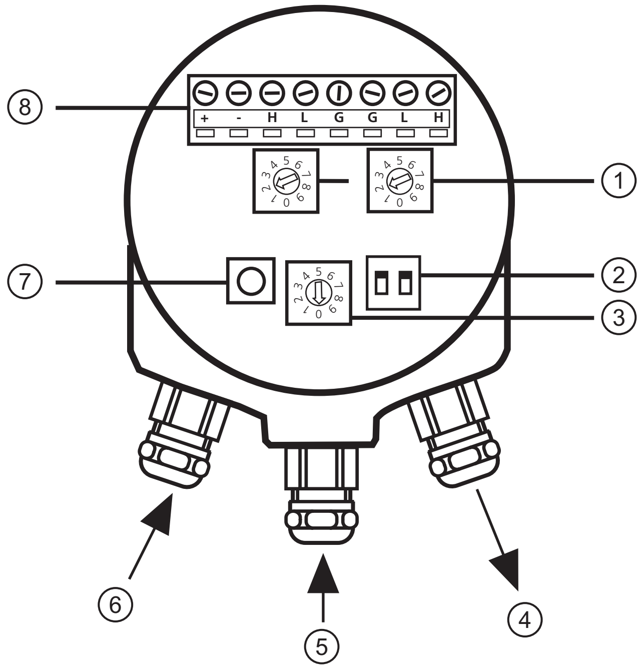
3.3.1 Abbreviations

CAN	Controller Area Network
CiA	CAN in Automation
CAL	CAN Application Layer
EDS	Electronic Data Sheet
DCF	Device Configuration File
SDO	Service Data Object
PDO	Process Data Object
TPDO	Transmit PDO
COB-ID	Communication Object Identifier
NMT	Network Management
IRT	Isochronous Real Time

4 Installation of the encoder

4.1 Settings of the encoder

The node address of the encoder, the baud rate and the bus termination must be configured during the set-up of the device. This is done by removing the cover of the housing on the back.



1: switches node address
2: bus termination on/off
3: baud rate switch
4: bus output
5: bus input

6: supply +U_B
7: zero set button
8: screw terminals for bus and power supply connection

4.2 Node address

The node address of the device can be set via two decimal rotary switches in the device. The step increment x10 and x1 is specified next to the switches. The permitted address range is between 3 and 98. The address range 0 to 2 is reserved for the master. Address 0 is used for broadcasting, i.e. master broadcasting to multiple slaves. Note that each address used in a CANopen network must be unique and must not be used by other devices.

The device address is read and adopted when the encoder power supply is switched on (or the NMT command Reset_Communication or Reset_Node). Both actions are required to adopt changes to the address settings.

4.3 Bus termination

In a CANopen network all devices are connected in a bus structure. Up to 32 devices (masters and/or slaves) can be connected in one segment. If more devices are needed, repeaters must be used to amplify the signals between the segments. An active termination must be added at the beginning and end of each bus segment to ensure an error-free operation. These terminations are integrated into the device and can be activated via DIP switches.

The active termination is only activated if the encoder is switched on. If the device is switched off, the CAN_H and CAN_L lines are internally terminated by a 121 Ω resistor.

Bit 1	Bit 2	Effect
on	on	121 ohm resistor between CAN_H and CAN_L
on	off	no valid setting
off	on	no valid setting
off	off	no resistor between CAN_H and CAN_L

4.4 Baud rate switch

The communication baud rate can be set using the rotary switch inside the encoder. The baud rate is set according to the following table.

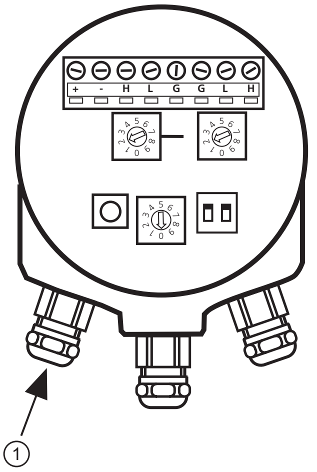
Settings baud rate switch

Baud rate [kbits]	Baud rate switch
10	0
20	1
50	2
125	3
250	4
500	5
800	6
1000	7
400	8

4.5 Electrical connection of the encoder

NOTE

The unit must be connected by a qualified electrician.
Disconnect power before connecting the unit.



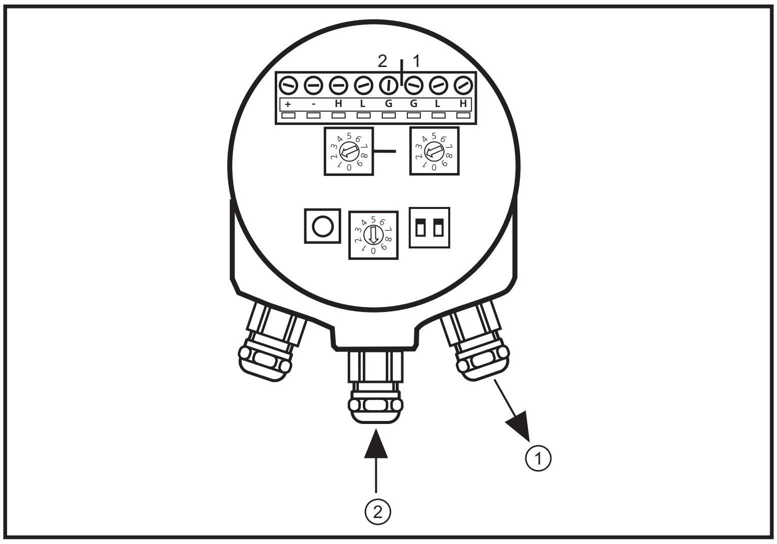
1: Voltage supply U_B

Function	Terminal
$+U_B$ (9...36 V DC)	+
0 V	-

The cable glands of the encoder must always be equipped with a shielded power supply cable with a wire cross-section between 0.34 mm^2 and 1.5 mm^2. The permissible outer cable diameter is ø 8...ø 10 mm. Two screw terminals with the required power supply terminals marked (+) and (-) are located inside the cover.

The (+) terminal is used for the connection to the +U_B line (9...36 V DC). The (-) terminal is used for the connection to the 0 V line.

4.6 BUS lines



1: bus output
2: bus input

Function	Terminal
CAN shield	cable gland
CAN GND	G
CAN_H	H
CAN_L	L

The cable glands of the encoder must be equipped with a twisted pair cable according to EN50170. The guidelines recommend a wire cross-section > 0.34 mm$^{2.}$. The permissible outer cable diameter is ø 6 ... to ø 8 mm. Six screw terminals with the required bus line terminals marked H, L and G are in the terminal chamber.

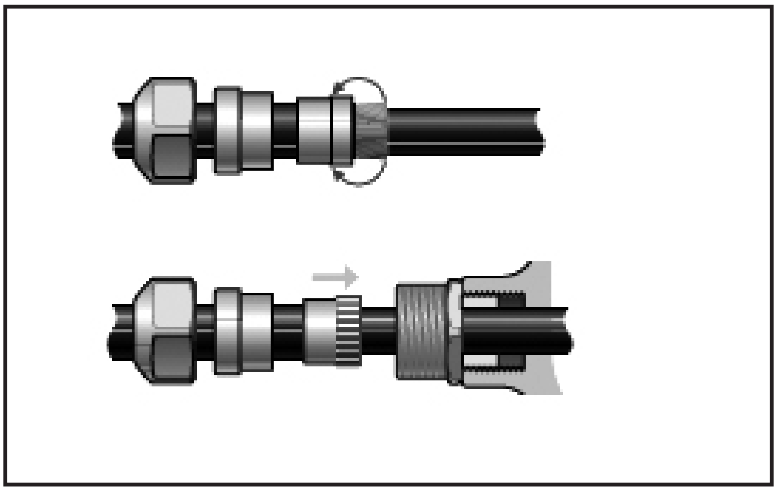
► Connect the (H) terminal to the CAN_H cable.

► Connect the (L) terminal to the CAN_L cable.

► Connect the (G) terminal to the CAN_G cable.

The G, H and L terminals are internally connected to each other so that the bus cables can be connected to any pair.

4.7 Shield



To achieve the highest possible noise immunity and electromagnetic compatibility the bus and power supply cables must always be shielded. The shield must be connected to ground on both ends of the cable. In certain cases a compensation current can flow across the shield. Therefore a compensation wire for the potential is recommended.

4.8 EDS file

An EDS file can be downloaded from our website:

→ www.ifm.com

Contents of the EDS file:

- communication functions and objects as defined in the CANopen communication profile DS-301
- device-specific objects as defined in the encoder profile DS-406
- manufacturer-specific objects

The EDS file serves as a template for different configurations of a device type. A DCF file is generated by the EDS file and describes a specific configuration of the device including object values, selected baud rate and module ID.

CANopen configuration tools are available to support the CANopen network configuration and the device configuration via the CAN bus. The information about the device is in the EDS file.

ℹ The EDS installation process depends on your configuration tool. In case of problems please contact your controller supplier.

4.9 Parameter setting

If the device is in the pre-operational status, the parameters are set by the configuration tool using the objects in the EDS file. During runtime the parameters can also be changed (operating status).

ℹ The position data is directly affected by some parameters and changes directly after such a parameter message.

► Only change scaling function parameters and the code sequence if the shaft is stationary.

ℹ The parameter setting process depends on your configuration tool. In case of problems please contact your controller supplier.

4.10 LED display

Two LEDs are on the cover of the encoder to indicate the encoder status. The module LED indicates the status of the module itself. The status LED indicates the module status on the bus.

The LEDs can be permanently on or out, blink or flash.

Blinking: LED 200 ms on, 200 ms out

Flashing: LED 200 ms on, 1000 ms out (single flashing)

LED 200 ms on, 200 ms out, 200 ms on, 1000 ms out (double flashing)

4.10.1 Module LED

The module LED is a two-colour LED with the following functionality:

LED	Display
out	no voltage supply
green	voltage supply ok
red	position error, the encoder cannot provide a correct position value
3 x blinking green, out	zero set button pressed and position set to zero after blinking three times the LED automatically returns to the previous status
red blinking	wrong switch settings

4.10.2 Status LED

The status LED is a two-colour LED with two functions; a green LED (run status) and a red LED (error status).

LED	Display
green flashing	encoder in the NMT status operational
green blinking	encoder in the NMT status stopped
green lighting	encoder in the NMT status pre-operational
red out	no error
red blinking	overflow on the error counter
red double blinking	guard event or heartbeat event occurred
red lights	the encoder is in the bus off status

If the communication of the encoder is error free in the operating status, the module LED shows green and the status LED flashes green.

5 Profile overview

The encoder profile defines the functions of the encoders connected to CANopen. The operating functions are divided in two device classes:

Class 1

The mandatory class with basic functions which all encoders must support. As an option, class 1 encoders can support selected functions of class 2. However, these functions must be implemented according to the profile.

Class 2

The encoder must support all functions of class 1 as well as all functions defined in class 2.

Functions of class 2

- Transmission of the absolute position value with polling, cyclic or SYNC mode.
- Speed and acceleration output values
- Change of the code sequence
- Presettings
- Scaling of the encoder resolution

Advanced diagnosis

- Encoder identification
- Operating status
- Operating time

- Alarms and warnings

All programming and diagnostic parameters can be accessed via SDOs. The output position value of the encoder is represented in binary format.

6 Functionality of the encoder

6.1 Basic functionality of the encoder

The figure below gives an overview of the basic functions of the encoder and how they are implemented in the encoder.

Physical position	
Basic function	← code sequence
	→ singleturn resolution → number of distinguishable revolutions
Absolute position	
Scaling function	↔ steps per revolution ↔ total measuring range in steps ↔ scaling function control/status
Preset function	← preset value
	→ offset value
Output position value	

6.2 Default identifiers

In order to facilitate configuration a default identifier allocation scheme is defined for CANopen devices. This ID allocation scheme consists of a functional part which determines the object priority and a module ID part which is equal to the node number (1 to 127). Broadcasting of non confirmed services (NMT and SYNC) is indicated by a module ID of zero.

For CANopen the 11-bit identifier is set up as follows:

bit no. 0...6 node number

bit no. 7...10 function code

Bit no.	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	function code				node number						

The following broadcast objects with default identifiers are available in the encoder:

Object	Function code (binary)	Resulting identifier (COB-ID)	Priority group
NMT	0000	0	0
SYNC	0001	128	0

The following peer-to-peer objects with default identifiers are available in the encoder:

Object	Function code (binary)	Resulting identifier (COB-ID)	Priority group
EMERGENCY	0001	129...255	0, 1
PDO1 (tx)	0011	385...511	1, 2
PDO2 (tx)	0101	641...767	2, 3
SDO (tx)	1011	1409...1535	6
SDO (rx)	1100	1537...1663	6, 7
Node guard	1110	1793...1919	-

6.3 Boot message

The encoder transmits a boot message after power on and initialisation. This message uses the default emergency identifier and has no data bytes. With this message the user can retrieve the transmitting node directly from the used identifier (COB-ID) as this is a function of the node number → chapter 6.2.

6.4 Operating parameters

Object 6000h, operating parameters, controls the functions for code sequence and scaling.

Bit	Function	Bit = 0	Bit = 1	Class 1	Class 2
0	code sequence	CW	CCW	M	M
2	scaling function control	switched off	enabled	O	M
4...11	reserved for further use				
12...15	manufacturer-specific parameters	no indication	no indication	O	O

The code sequence defines whether increasing or decreasing position values are provided when the encoder shaft rotates clockwise or counterclockwise (seen on the shaft). The control of the scaling function is used for enabling/disabling the steps per revolution in the object (6001h) and for the complete measuring range in steps in the object (6001h) (→ chapter 6.5).

If the scaling function bit is set, the scaling parameters affect the output position value. If the scaling function bit is set to 0, the scaling function is disabled.

6.5 Scaling function

6.5.1 Overview

With the scaling function the internal numerical value of the encoder is converted in the software to change the physical resolution of the encoder. The parameters "steps per revolution" (object 0x6001h) and "total measuring range in steps" (object 0x6002h) are the scaling parameters which operate with the scaling function control bit.

ℹ For scaling a multiturn encoder the parameter "steps per revolution" must be transmitted before the parameter "total measuring range in steps".

The data type for both scaling parameters is 32 (without sign) with a value range from 1 to 2^{32} (limited by the encoder resolution). For a 25-bit encoder with a singleturn resolution of 13 bits the permitted value for "steps per revolution" is between 1 and 2^{13} (8192). For the "total measuring range in steps" the permitted value is between 1 and 2^{25} (33554432). The scaling parameters are stored in a non-volatile memory and reloaded at each start-up.

Format of the singleturn scaling parameters

Byte	**3**	**2**	**1**	**0**
Bit	31 - 24	23 - 16	15 - 8	7 - 0
Data	2^{31} - 2^{24}	2^{23} - 2^{16}	2^{15} - 2^{8}	2^{7} - 2^{0}
object 6001h - steps per revolution				

Format of the multiturn scaling parameters

Byte	**3**	**2**	**1**	**0**
Bit	31 - 24	23 - 16	15 - 8	7 - 0
Data	2^{31} - 2^{24}	2^{23} - 2^{16}	2^{15} - 2^{8}	2^{7} - 2^{0}
object 6002h - total measuring range in steps				

6.5.2 Scaling formulas

The scaling function used in the CANopen encoder is limited to a singleturn resolution within a step. After the download of new scaling parameters the preset function should be used to set the start point of the encoder.

Only change the scaling function parameters if the shaft is stationary.

In the following formula a 25-bit multiturn encoder with a singleturn resolution of 13 bits is used as an example.

Formula for the multiturn scaling function:

A = (singleturn_position x steps_per_revolution) / 8192

output_position = (revolution_number x steps_per_revolution) + A

Where: singleturn_position = absolute singleturn position value
revolution_number = absolute multiturn number

6.6 Preset value

6.6.1 Overview

The preset function (object 0x6003h) supports the adaptation of the encoder to the mechanical zero point or to a preset value. The preset function is used after the scaling function. The preset value is then provided as a measured value.

A preset value is determined by the encoder as follows:

The encoder reads the current position value and calculates an offset value from the preset value and the read position value. The position value is shifted by the calculated offset value. The offset value can be read with the diagnostic function (object 6509h), is stored in a non-volatile memory and reloaded at each start-up.

Only use the preset function if the shaft is stationary.

Format of the preset value

Byte	**3**	**2**	**1**	**0**
Bit	31 - 24	23 - 16	15 - 8	7 - 0
Data	2^{31} - 2^{24}	2^{23} - 2^{16}	2^{15} - 2^{8}	2^{7} - 2^{0}
object 6003h - preset value				

6.6.2 Calculation of the preset value

An offset value is calculated when the encoder receives the preset value, see the set-up calculation below. The offset value is then used during runtime to shift the current position to the required output position, see the runtime calculation below.

In the formulas below the current position is the absolute position of the encoder disk after the scaling function. The calculations are made with signed values.

Set-up calculation: *offset_value = preset_value - current_value*

A previously set offset value is not included in the current position.

Runtime calculation: *output_position = current_position + offset_value*

6.7 Zero setting

Two methods can be used for zero setting.

Setting via software

If the preset object is used and the preset value is set to zero (00 00 00 00h), the encoder is set to zero.

Setting via pushbutton

If the zero set button is pressed for at least 1 second, the position of the encoder is set to zero (00 00 00 00h).

Display module LED
Green, out, green, out, green, out to confirm that the position value is set to zero.

6.8 Speed and acceleration

The encoder supports the output of the speed object (0x6030) and of the acceleration object (0x6040). In order to maintain the accuracy irrespective of the rotational speed of the encoder different steps can be set. The speed object is limited to a signed 16-bit value. It is necessary to optimise the assumed rotational speed of the shaft with regard to the selected resolution to avoid an overflow of data.

The object 0x5003 (speed type) is a manufacturer-specific object which sets the update time and resolution (steps / second or rpm) of the speed object (0x6030) and the acceleration object (0x6040). The speed type object is described in chapter 7.1.

6.9 PDO mapping

Dynamic PDO mapping enables changes of the objects transmitted in a PDO. The RM and RN type encoders can map three different objects in the PDOs.

Name	Object	Subindex	Length
Position	0x6004		4 bytes
Speed	0x6030	1	2 bytes
Acceleration	0x6040	1	2 bytes

The encoder has two transmit PDOs. PDO1 (cyclically transmitted by the cyclic timer) and PDO2 (transmitted on receipt of a SYNC message). As default both PDOs are mapped to transmit only position data. Both PDOs can be changed separately to transmit a combination and sequence of the object above.

The structure of the entries of the object "transmit PDO mapping parameter" in subindices 1...3 is as follows:

Byte MSB	Byte MSB - 1	Byte LSB + 1	Byte LSB
object		subindex	object length (number of bits)

6.9.1 PDO configuration

To change the PDO mapping the encoder must be in the NMT mode pre-operational. The PDO must be set to "not valid". This is done by deleting bit 31 (MSB) in subindex 1 "COB-ID used by PDO" in the object "transmit PDO communication parameters".

The PDO "transmit PDO mapping parameter" must be deactivated. To do so, set the subindex 0 to 0.

To reconfigure the PDO mapping transmit the data of the corresponding object, the subindex and length of the first object to "transmit PDO mapping parameter" in subindex 1. Then proceed in the same way for the optional second and third objects and transmit to "transmit PDO mapping parameter" in subindices 2 and 3.

The "transmit PDO mapping parameter" subindex 0 must be set to the number of the objects mapped in the PDO (1-3).

The reconfigured PDO mapping must be set to "valid" by the setting bit 31 (MSB) in subindex 1 "COB-ID used by PDO" in the object "transmit PDO communication parameters". After setting the encoder in the NMT mode operational the reconfigured PDO mapping is enabled.

The PDO mapping can be stored in the non-volatile EEPROM using the object 0x1010 "store parameter field" (subindex 1 "all parameters" or subindex 2 "communication parameters").

6.9.2 PDO configuration example

The following chapter shows how to map PDO1 with position and speed (the address of the encoder is 0x0F, all data in hexadecimal format).

Step	ID	Data	Note
1	0	80 0F	Set the encoder in the NMT mode pre-operational
2	60F	23 00 18 01 8F 01 00 80	Set PDO1 to not valid and COB-ID to 0x18F
3	60F	2F 00 1A 00 00 00 00 00	Set the subindex 0 to 0 to transmit "transmit PDO mapping parameter" (mapping deactivated)
4	60F	23 00 1A 01 20 00 04 60	Map position (object 0x6004) to the first position in the PDO
5	60F	23 00 1A 02 10 01 30 60	Map speed (object 0x6030) to the second position in the PDO
6	60F	2F 00 1A 00 02 00 00 00	Set the subindex 0 "transmit PDO mapping parameter" to 2 (number of objects mapped in the PDO)
7	60F	23 00 18 01 8F 01 00 00	Set PDO1 to valid and COB-ID to 0x18F
8	0	01 0F	Set the encoder if it is in the NMT mode operational

The mapping of PDO1 is now finished. The PDO1 message can for example be as follows:

ID	Data
18F	4E C9 B2 00 53 01

"4E C9 B2 00" is position data and "53 01" is the speed value.

To save the PDO mapping in the EEPROM transmit:

ID	Data	Note
60F	23 10 10 02 73 61 76 65	Save all communication parameters by transmitting the ASCII code for "SAVE" to object 0x1010, subindex 2

6.10 Heartbeat

The RM and RN type encoders can act as heartbeat producer. The time between two heartbeats is configured in the object "producer heartbeat time" (0x1017) and is in the millisecond range (1...65535). If the "producer heartbeat time" (0x1017) is zero (0), heartbeat is deactivated.

The object "producer heartbeat time" (0x1017) is stored in the non-volatile EEPROM and reloaded at start-up.

6.11 IRT mode

In order to enhance the real-time characteristics the encoder can operate in the IRT (Isochronous Real Time) mode. In the normal operating mode the position value is sampled cyclically every 0.5 ms. If "read position at SYNC" is disabled, the PDO2 (transmit data at SYNC) uses the last sampled position of the encoder. This adds a non real-time characteristics to the output position value. In the IRT mode "read position at SYNC" is set. The position value is only sampled if the SYNC message is received.

If the bit "read position at SYNC" is set in the operating parameter object (0x6000), the following changes are made:

- The speed object (0x6030) and the acceleration object (0x6040) are disabled because the cyclic position sampling is mandatory for calculating these values.
- PDO1 (transmit data cyclically) is disabled, object 0x1800, subindex 1, bit 31 is set and stored in the EEPROM.
- The object "transmit PDO1" (0x1800) is only read if the bit "read position at SYNC" is set.
- PDO2 (transmit data at SYNC) is only set to transmit position data and the new PDO2 mapping is stored in the EEPROM.
- The object "PDO2 tx mapping" (0x1A01) is only read if the bit "read position at SYNC" is set.

If the speed and acceleration values are needed during operation in the IRT mode, it is recommended to calculate these values in the master application and to use the master clock (SYNC message) as a reference.

Bit	Parameter
0	code sequence

Bit	Parameter
1	no indication
2	scaling function control
3...14	no indication
15	read position at SYNC

Operating parameters (object 0x6000)

6.11.1 Diagnosis of the encoder

The diagnosis of the encoder can be read by the objects 65xxh. The operating status, alarm and warning diagnosis is described in the following chapters. For a complete overview of the supported diagnosis please refer to the EDS file.

6.11.2 Operating status

In the object 6500h the operating status can be read. The function for each bit corresponds to the operating parameters, see chapter 6.4.

In the operating status the scaling function control (bit 2) is set depending on the setting in the operating parameters. In addition, the actual scaling values used in the encoder can be read as diagnosis, object 6501h (singleturn resolution) and object 6502h (multiturn resolution).

6.12 Alarms and warnings

If an internal alarm is detected by the encoder, it automatically passes into the pre-operational status. A COB-ID EMCY message (object 0x1014h) is transmitted by the encoder. This message transmits the alarm type occurred. To return to the operating status an NMT command must be transmitted. The encoder supports the following alarms.

Bit	Supported alarms
0	position error
1-11	
12	E2prom error
13...15	

Alarms (object 0x6506/0x6505)

Bit	Supported warnings
0	
1	lighting control
2	watchdog

Bit	Supported warnings
3...15	

Warnings (object 0x6504/0x6503)

7 Manufacturer-specific objects

7.1 Object 0x5003, speed type

The object 0x5003 sets the update time and resolution (steps / second or rpm) of the speed information. This object affects the speed object (0x6030) and the acceleration object (0x6040).

Speed type	Setting
0	200 ms update time, steps / s
1	10 ms update time, steps / 10 ms
2	100 ms update time, steps / 100 ms
3	200 ms update time, rpm

The speed object is limited to a signed 16-bit value. In order to avoid data overflow and optimise the accuracy it is recommended to calculate the optimum setting of the speed type. Also transfer the scaling to the encoder. The amount of data is limited and overflow is avoided as the calculation of the speed value is based on the scaled singleturn value.

If the rotation of the shaft is faster than 1000 rpm and the speed type is 0 (steps/s), a data overflow occurs. In this case a higher resolution is required, i.e. steps/100 ms.

The accuracy of the speed measurement depends on the selected resolution. The figures in this table should be considered as reference values.

Speed type	Shaft rotation
0	> 100 rpm
1	> 1000 rpm
2	> 1000 rpm
3	> 100 rpm

The table shows from which shaft speed the accuracy of the measured value deviates less than 1%. In general, the accuracy improves irrespective of the selected speed type, the higher the shaft rotation is.

7.2 Object 0x5A03, serial number 2

The object 0x5A03 is a manufacturer-specific object where the serial number can be read.

8 Example encoder configuration

This example shows a simple set-up of the encoder for the cyclic transmission of the position value.

1. Set the physical address (node number) of the encoder using the address switches. You can find more information in chapter 4.
2. Ensure that the baud rate of your CANopen network and the baud rate of the encoder are the same. You can find more information regarding the baud rate setting of the encoder in chapter 4.4.
3. Switch on the encoder.
4. The encoder transmits a boot message in the default emergency identifier (ID = 128 + encoder address). The message has no data bytes.
5. The next step is to configure the encoder via the SDO message. To set a cyclic transmission of the position value with a repetition rate of 10 ms it is necessary to transmit an SDO request message (ID = 1536 + encoder address) to the cyclic timer object (object 6200h) with the data below. The encoder confirms with the SDO response message (ID = 1408 + encoder address).

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x22	0x00	0x62	0x00	0x0A	0x00	0x00	0x00

SDO request message

6. For the set-up of the encoder you have to transmit an NMT "start remote node" message which consists of ID = 0 and two data bytes with the following contents:

Byte 0	Byte 1
0x01	encoder address (node number)

NMT "start remote node" message

7. The encoder is now in the operating status and the position message (ID = 384 + encoder address) is transferred with a repetition rate of 10 ms. If an error occurs, the encoder transmits an emergency message (ID = 128 + encoder address).