CANopen Encoder Gateway

Application and Installation information

1. INTRODUCTION 3

1.1 CANopen communication model 3

1.2 Profile overview 3

1.3 References 4

2. FUNCTIONAL OVERVIEW 4

2.1 Default Identifiers 4

2.2 Boot-up message 5

2.3 Encoder functionality 5

3. SCALING FUNCTION 6

3.1 Overview 6

3.2 Scaling formulas 7

3.2.1 Measuring range 7

4. PRESET VALUE 8

4.1 Overview 8

4.2 Preset formula 9

5. INSTALLATION 9

5.1 Power supply cable 9

5.2 Bus cable 10

5.3 Encoder cable 10

5.4 Address setting 10

5.5 Baudrate setting 11

5.6 EDS and DCF files 11

5.7 Parameter settings 12

5.8 Incremental signal output (optional) 12

6. MANUFACTURER SPECIFIC FUNCTIONS 13

6.1 Object 5000h - Transmission rate 13

6.2 Object 5001h - Node number 14

6.3 Object 5500h – Gateway Serial number 15

7. ENCODER TYPES 16

7.1 Singleturn Encoders 13 bit 16

7.2 Multiturn Encoders 25 bit 16

7.3 Angle Encoders 16

7.4 Linear Encoders 16

8. ENCODER SETUP EXAMPLE 17

APPENDIX A, HISTORY 18

1. Introduction

This document contains application and installation information on the Leine & Linde CANopen Encoder Gateway based on the CiA CANopen Encoder Profile DS 406. The Encoder Gateway support all functions of the profile.
The gateway is the interface between the CAN bus and the encoder. The communication between the gateway and the encoder is handled through a fast Bidirectional Synchronous-Serial Interface called Endat. In the EnDat encoder advanced diagnostics are integrated together with the position sensing parts to make it possible to check the correctness of the position value very thoroughly.

1.1 CANopen communication model

The CANopen communication profile is based on the CAN Application Layer (CAL) specification from the CiA (CAN in Automation). CANopen comprises a concept to configure and communicate real-time data using both synchronous and asynchronous messages. Four types of message (objects) are distinguished:

1. Administrative messages (Layer Management, Network Management and Identifier Distribution Messages)
2. Service Data Messages (SDO)
3. Process Data Messages (PDO)
4. Pre-defined Messages (Synchronisation-, Time-stamp-, Emergency Messages)

For further information please see the CANopen specification.

1.2 Profile overview

The Encoder Profile defines the functionality of encoders connected to CANopen. The operating functions are divided in two device classes:
- Class 1, the Mandatory class with a basic range of functions that all Encoders must support. The class 1 encoder can optionally support selected class 2 functions, these functions must however be implemented according to the profile.
- Class 2, where the Encoder must support all class 1 functions and all functions defined in class 2. The full class 2 functionality includes:

- Absolute position value transfer using either polled, cyclic or sync mode.
- Change of code sequence
- Preset value settings
- Scaling of the encoder resolution

Advanced diagnostics including:

- Encoder identification
- Operating status
- Operating time
- Alarms and warnings

All programming and diagnostic parameters are accessible through SDOs. The output position value from the encoder is presented in binary format.

1.3 References

http://www.can-cia.de

CAN Application Layer, DS 201 …207	CiA
CAL Based Communication Profile, DS 301	CiA
Device Profile for Encoders, DS 406	CiA
CAN Specification Version 2.0 A	Robert Bosch Gmbh
82527 Architectural Overview	Intel

2. Functional overview

2.1 Default Identifiers

In order to reduce configuration effort 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, SYNC and node guarding) is indicated by a module-ID of zero.
In CANopen the 11 bit identifier is build as follows:

Bit-Nr.	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:

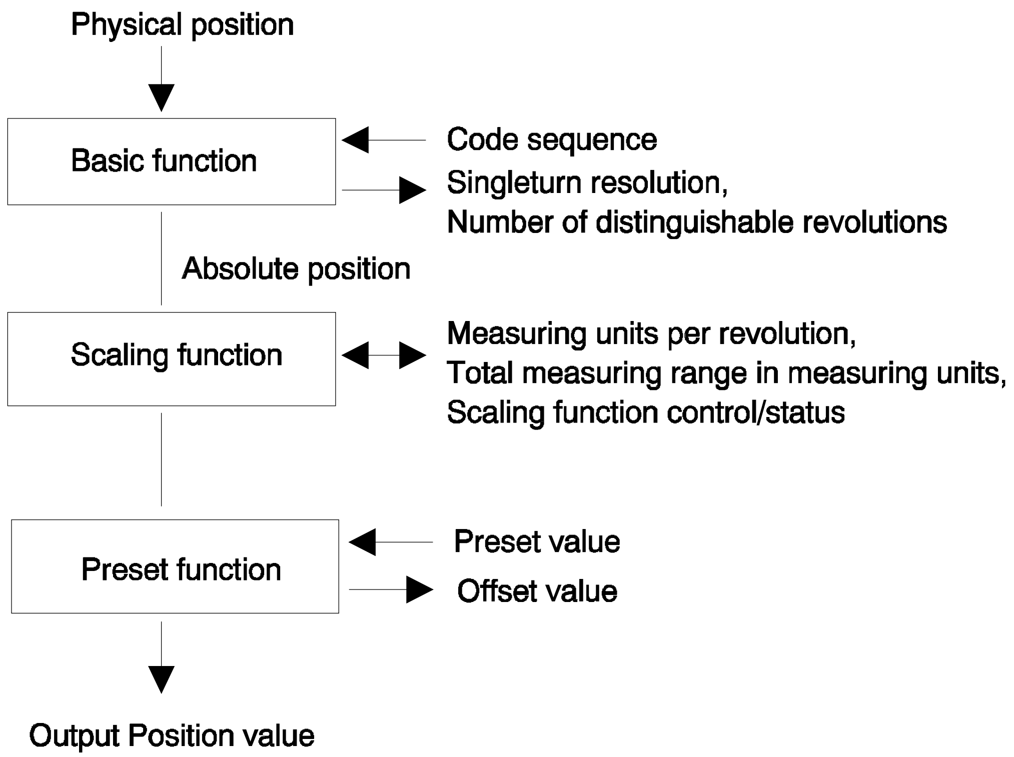
Object	Function Code (binary)	Resulting Identifier (COB-ID)	Priority group
EMERGENCY	0001	129 – 255	0, 1
PDO1 (tx)	0011	385 – 511	1, 2
PDO2 (tx)	0101	641 – 767	2, 3
SDO (tx)	1011	1409 – 1535	6
SDO (rx)	1100	1537 – 1663	6, 7
Nodeguard	1110	1793 – 1919	-

2.2 Boot-up message

The encoder sends a Boot-up 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 sending node directly from the used identifier (COB-ID) as it is a function of the node number, see 2.1.

2.3 Encoder functionality

The figure below gives an overview of the Encoder functions.



3. Scaling function

3.1 Overview

With the scaling function the Encoder internal 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 set to operation with the scaling function control bit.
NOTE! When scaling a Multiturn encoder the parameter “Measuring units per revolution” must be sent before the parameter “Total measuring range in measuring units”.
The data type for both scaling parameters is unsigned 32 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 permissible value for the ”Measuring units per revolution” is between 1 and 2^{13} (8192) and for the ”Total measuring range in measuring units” the permissible value is between 1 and 2^{25} (33 554 432).
The scaling parameters are securely stored in case of voltage breakdown and reloaded at each start-up.

Scaling parameter format:

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 - Measuring units per revolution			

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 measuring units			

3.2 Scaling formulas

The scaling function used in the CAN encoder gateway is limited to a singleturn resolution within one step. After downloading new scaling parameters the preset function should be used to set the encoder starting point to the required starting point.

NOTE! Changing the scaling function parameters should only be used at encoder standstill.

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

Formula for the multiturn scaling function:

A = measuring_units_per_revolution * singleturn_position / 8192
output_position = (revolution_number * measuring_units_per_revolution) + A

where:

singleturn_position =	the Absolute position value of the encoder singleturn disk
revolution_number =	the Absolute revolution number of the encoder multiturn disks

3.2.1 Measuring range

The measuring range is set by the parameter ”Total measuring range in measuring units”. The encoder has two different operating modes depending on the specified measuring range. When the encoder recieves new scaling parameters it checks the values for binary scaling and chooses operating mode A if binary scaling detected, see explanation below.

A. Cyclic operation (Binary scaling)

Used when operating with $\mathbf{2^x}$ number of turns (2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096 number of turns).

If the desired measuring range is equal to the specified singleturn resolution $\mathbf{* 2^x}$ (where x <= 12) the encoder operates in endless cyclic operation (0 - max - 0 - max ...). For example: If the position value increases above the maximum value (measuring range-1) by rotating the encoder beyond the maximum value the encoder continues from 0.

Example of a cyclic scaling:

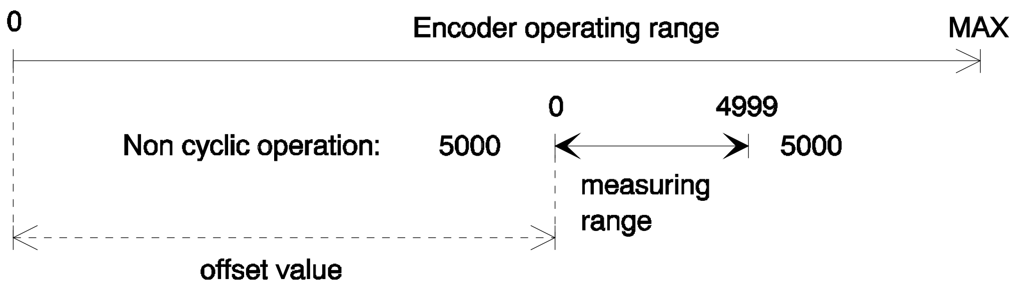
Measuring units per revolution	= 1000
Measuring range	= 32000 ($\mathbf{2^5}$ = 32 number of turns)

B. Non cyclic operation

If the measuring range is used to limit the encoder value range to a value not equal to the specified singleturn resolution $\mathbf{* 2^x}$ the output position value is limited within the operating range. If the position value increases or decreases outside the measuring range by rotating the encoder beyond the maximum value (measuring range-1) or below 0 the encoder outputs the total measuring range value, see figure below.

Example of a non cyclic scaling:

Measuring units per revolution: 100
Total measuring range: 5000 steps (50 turns)



4. Preset value

4.1 Overview

The preset function supports adaptation of the encoder zero point to the mechanical zero point of the system. The preset function sets the actual position of the encoder to the preset value. The preset function is used after the scaling function which means that the preset value is given in the current measuring units.
A preset is handled by the encoder in the following way:
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 with the calculated offset value. The offset value can be read with the diagnostic function (Object 6509h) and is securely stored in case of voltage breakdown and reloaded at each start-up.
NOTE! The preset function should only be used at encoder standstill.

Preset value format:

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			

4.2 Preset formula

An offset_value is calculated when the encoder receives the preset_value, see setup calculation below. The offset_value is then used during runtime to shift the current position to get the required output position, see runtime calculation below.
NOTE! 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.

Setup calculation:
offset_value = preset_value - current_position
NOTE! A previously set offset_value is not included in the current position.

Runtime calculation:
output_position = current_position + offset_value

5. Installation

This section handles the installation issues of the Encoder Gateway.

5.1 Power supply cable

The gateway should be supplied with 9-30VDC through the integrated screw terminal block. A shielded power cable must be used.
Installation:
1. Remove the cover of the gateway box.
2. Strip the cable ends to the appropriate length, leave app. 15mm of the shield for connection to the cable gland.
3. Insert the power cable through the cable gland.
4. Connect the cables to the +E and 0V screw terminal block. Tighten the terminal screws.
5. Tighten the cable gland and make sure the shield is connected to the gland.
6. Close the cover of the gateway box.

5.2 Bus cable

The Encoder gateway includes an isolated bus interface separating the power supply and the CAN transceiver supply. The transceiver ground is available at the screw terminals marked CAN_GND.
A shielded twisted pair wire with specifications according to the CAN standard must be used.
The gateway includes a build in T-coupling with a screw terminal block for CAN_H, CAN_L and CAN_GND. When the gateway is used as the first or last station on the BUS, the cable can be connected on either IN or OUT terminal with the terminating resistor switch (TERM.R.) set to ON.
Installation:
1. Remove the cover of the gateway box.
2. Strip the cable ends to the appropriate length, leave app. 15mm of the shield for connection to the cable gland.
3. Insert the bus cable through the cable gland.
4. Connect the cable conductors to the screw terminal block marked BUS (IN or OUT). Make sure that the same conductor is always connected to the same terminal (e.g. green conductor always connected to CAN_H, red conductor always connected to CAN_L). Tighten the terminal screws.
5. Set the terminating resistor switch (TERM.R.) to ON if this is the first or last station on the bus.
6. Tighten the cable gland and make sure the shield is connected to the gland.
7. Close the cover of the gateway box.

5.3 Encoder cable

Installation:
1. Make sure the power supply to the gateway is switched off.
2. Connect the male cable-connector to the gateway.
3. Connect the female cable-connector to the encoder.
4. Switch on the gateway power supply.

5.4 Address setting

The physical address (node number) of the gateway must be set between 1 - 127 with the address switch inside the gateway. The address is set in binary code (the value for each switch position is marked beside the switch). The gateway reads the address switch only at power-up.
The gateway also support setting the physical address from the CAN master through a separate object, for information see the manufacturer specific functions below.
NOTE! The selected address (node number) affects the default identifier allocation, see section 2 for further information.
Setting of the address:
1. Make sure the power supply to the gateway is switched off.
2. Set the address with the dipswitches.
3. Switch on the gateway power supply.

5.5 *Baudrate setting*

The transmission rate of the gateway must be set with the Baudrate switch inside the gateway. The value is set with three dip-switches (Switch 8,9,10 on SW1) as a binary code representing the selected transmission rate. The gateway reads the Baudrate switch only at power-up.
The gateway also support setting the transmission rate from the CAN master through a separate object, for further information see the manufacturer specific functions below.
Setting of the transmission rate:
1. Make sure the power supply to the gateway is switched off.
2. Set the transmission rate code with the dipswitches, see table below.
3. Switch on the gateway power supply.

Dip-switch value	Transmission rate	Switch 8	Switch 9	Switch 10
0	10 kBit/s	OFF	OFF	OFF
1	20 kBit/s	ON	OFF	OFF
2	50 kBit/s	OFF	ON	OFF
3	125 kBit/s	ON	ON	OFF
4	250 kBit/s	OFF	OFF	ON
5	500 kBit/s	ON	OFF	ON
6	800 kBit/s	OFF	ON	ON
7	1.000 kBit/s	ON	ON	ON

5.6 *EDS and DCF files*

An EDS-file (Electronic Data Sheet) is delivered together with the CANopen Encoder Gateway. The EDS file describes:

- The communication functionality and objects as defined in the CANopen communication profile DS-301.
- The 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 one device type.
A DCF-file (Device Configuration File) is generated from the EDS-file describing a specific configuration of the device including object values, selected baudrate and module-Id.
CANopen configuration tools are available to support CANopen network configuration and device configuration via the CAN bus. The information about the device is obtained from the EDS-file.
NOTE! The EDS Installation procedure depends on your configuration tool, please consult your tool supplier if you run into problems.
Installation example for PRO-CANopen:

1. Copy the EDS file into the EDS directory created by PRO-CANopen.
2. Select the type of device to be configured, a list of EDS-files is displayed, select the required EDS-file.
3. An empty DCF file is created and the configuration of the device can start.

5.7 Parameter settings

The parameters are usually set from the configuration tool when the device is in the Pre-Operational state using the objects obtained from the EDS-file. The parameters can also be changed during runtime (Operational state), please be careful as the position data is directly effected by some parameters and will change directly following such parameter message. Therefore *changing the scaling function parameters and the code sequence should only be used at encoder standstill*.

NOTE! The parameterisation procedure depends on your configuration tool, please consult your tool supplier if you run into problems.

5.8 Incremental signal output (optional)

The absolute encoder connected to the CAN Gateway provides two incremental sinusoidal outputs A and B. The signals is distributed through a connector on the Gateway. The signal amplitude is 1Vpp with terminating resistor Z_0 = 120 ohm. See enclosed information for recommended input circuitry.

Chassis connector: EUCHNER SD 4 K
Cable connector: EUCHNER BS 4 K (Leine & Linde Art.nr: 002 01 029)

Pin No.	Signal	Colour
1	A+	Green/black
2	A-	Yellow/black
3	B+	Blue/black
4	B-	Red/black

6. Manufacturer specific functions

6.1 Object 5000h - Transmission rate

Eight transmission rates are supported by the encoders. The selection of the transmission rate can be done either by Object 5000h or by hardware. By writing the value FFh to Object 5000h the encoder gets the transmission rate by hardware. Changing to Software mode is done by writing the desired transmission rate code to Object 5000h. The transmission rate will be valid after next power-up (next initialisation sequence).

OBJECT DESCRIPTION

INDEX	5000 H
Name	Transmission rate
Object Code	VAR
Data Type	Unsigned 8

VALUE DESCRIPTION

Object Class	Manufacturer specific
Access	Rw
PDO Mapping	No
Value Range	Unsigned8
Mandatory Range	No
Default Value	FFh = Hardware settings

STRUCTURE OF PARAMETER

Value	Function
0	10 kBit/s
1	20 kBit/s
2	50 kBit/s
3	125 kBit/s
4	250 kBit/s
5	500 kBit/s
6	800 kBit/s
7	1.000 kBit/s
FFh	Hardware settings

6.2 Object 5001h - Node number

The selection of the node number can be done either with Object 5001h or by hardware (e.g.: dip-switches, pins or wires). By setting the value FFh the encoder gets the node number by hardware. Changing to Software mode is done by writing the desired node number to Object 5001h. The node number can be set in the range 1 – 127.
The node number will be valid after next power-up (initialisation sequence).

OBJECT DESCRIPTION

INDEX	5001 H
Name	Node number
Object Code	VAR
Data Type	Unsigned 8

VALUE DESCRIPTION

Object Class	Manufacturer specific
Access	Rw
PDO Mapping	No
Value Range	1 – 127, FFh
Mandatory Range	No
Default Value	FFh = Hardware settings

STRUCTURE OF PARAMETER

Bit	Function
0	Node number
1	Node number
2	Node number
3	Node number
4	Node number
5	Node number
6	Node number
7	Hardware switching

6.3 Object 5500h – Gateway Serial number

Object 5500h contains the CANopen Encoder Gateway serial number. It is given as an unsigned32 binary value. If the gateway serial number is not used the value is set to maximum value FF FF FF FF h.

OBJECT DESCRIPTION

INDEX	5500 H
Name	Gateway Serial number
Object Code	VAR
Data Type	Unsigned32

VALUE DESCRIPTION

Object Class	Manufacturer specific
Access	R
PDO Mapping	No
Value Range	Unsigned32
Mandatory Range	No
Default Value	No

7. Encoder types

A number of different EnDat encoders can be connected to the CANopen gateway. This section of the document gives you information on the supported functionality.

7.1 Singleturn Encoders 13 bit

The 13 bit singleturn encoder supports the full class2 functionality.
The Scaling function accepts all values between 2 and 8192 measuring units per revolution. The total measuring range is equal to the scaled singleturn resolution.

7.2 Multiturn Encoders 25 bit

The 25 bit Multiturn encoder supports the full class2 functionality.
The Scaling function parameter ”measuring units per revolution” accepts all values between 2 and 8192. The encoder has two different operating modes (Cyclic or Non-cyclic) depending on the value of the parameter “total measuring range in measuring units”, please check the section Scaling formulas above for further information.

7.3 Angle Encoders

EnDat singleturn Angle encoders with a resolution of 20 and 23 bits is currently supported. The Angle encoders support the full class2 functionality with the scaling function limited to binary scaling. This means that the parameter “measuring units per revolution” accepts only values with is equal to $\mathbf{2^x}$ where x is a value from one to 20 (for a 20 bit encoder). Scaling a 20 bit encoder to 18 bit gives the parameter “measuring units per revolution” the value $\mathbf{2^{18}}$ = 262144.

7.4 Linear Encoders

EnDat Linear encoders is supported with the preset function but currently no other class2 parameter or the change of code sequence. However access to all diagnostic objects as the operating time monitor, alarms, warnings and the encoder serial number is supported. For Linear encoders the Object 6501h indicates the measuring step in nm that is output by the encoder.

8. Encoder setup example

This example shows a simple setup of the Encoder Gateway for cyclic transmission of the Position value. Read section 2 of this Application note before proceeding.

1. Set the physical address (Node Number) of the Encoder Gateway by Dip-switches, see section 5.2 for further information.

2. Set the baudrate of your Encoder Gateway by Dip-switches, see section 5.5 for further information.

3. Power up the Encoder Gateway.

4. The Gateway will send a Boot-up message on the default emergency identifier (ID = 128 + encoder address), the message has no data bytes.

5. The Gateway is now ready for configuration through the SDO message. To set a cyclic transmission of the position value with 10ms repetition rate we need to send an SDO request message (ID = 1536 + Gateway address) to the cyclic timer object (Object 6200h) with the data below. The Gateway will confirm with the SDO response message (ID = 1408 + Gateway 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

6. To get the Gateway in operation we need to send an NMT “start remote node” message, ID = 0, two data bytes with the following contents:

Byte 0	Byte 1
0x01	The Gateway address (Node Number)

7. The Gateway has now entered the operational state and the position message (ID = 384 + Gateway address) is transferred with a 10ms repetition rate. If an error occur the Gateway will send an emergency message (ID = 128 + Gateway address).

Appendix A, History

Revision	Date	Changes
Rev. 1.0	98-02-24	First release
Rev. 1.1	98-03-17	1. Default identifer allocation scheme included
		2. Boot-up message included
Rev. 1.2	98-05-25	EnDat Linear encoders is supported with the preset function.
Rev. 1.3	99-03-19	Encoder setup example included.