

		Document RXA CANopenAppl	Revision 1.1	Page 1	Total 14
Topic CANopen Encoder Application Note	Project CANopen	Sign. Thr	Date 06-03-30		

CANopen Application and Installation information

1. INTRODUCTION	2
1.1 CANopen communication model	2
1.2 Profile overview	2
1.3 References	3
2. FUNCTIONAL OVERVIEW	3
2.1 Default Identifiers	3
2.2 Boot-up message	4
2.3 Encoder functionality	4
3. SCALING FUNCTION	5
3.1 Overview	5
3.2 Scaling formulas	6
4. PRESET VALUE	6
4.1 Overview	6
4.2 Preset formula	7
5. INSTALLATION	7
5.1 Bus installation	7
5.2 Address setting	7
5.3 Baudrate setting	8
5.4 EDS and DCF files	8
5.5 Parameter settings	8
5.6 Cables and connectors	9
6. MANUFACTURER SPECIFIC FUNCTIONS	10
6.1 Object 5000h - Transmission rate	10
6.2 Object 5001h - Node number	11
6.3 Object 5002h – Address offset	12
7. ENCODER SETUP EXAMPLE	13
APPENDIX A, HISTORY	14

		Document RXA CANopenAppl	Revision 1.1	Page 2	Total 14
Topic CANopen Encoder Application Note	Project CANopen	Sign. Thr	Date 06-03-30		

1. Introduction

This document contains application and installation information on the Leine & Linde CANopen Encoders RXA 5X7 based on the CiA CANopen Encoder Profile DS 406. The Encoders support all functions of the profile.

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.

		Document RXA CANopenAppl	Revision 1.1	Page 3	Total 14
Topic CANopen Encoder Application Note	Project CANopen	Sign. Thr	Date 06-03-30		

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
CANary CAN controller	Atmel

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	-

Document	Revision	Page	Total
RXA CANopenAppl	1.1	4	14
Project		Sign.	Date
CANopen		Thr	06-03-30

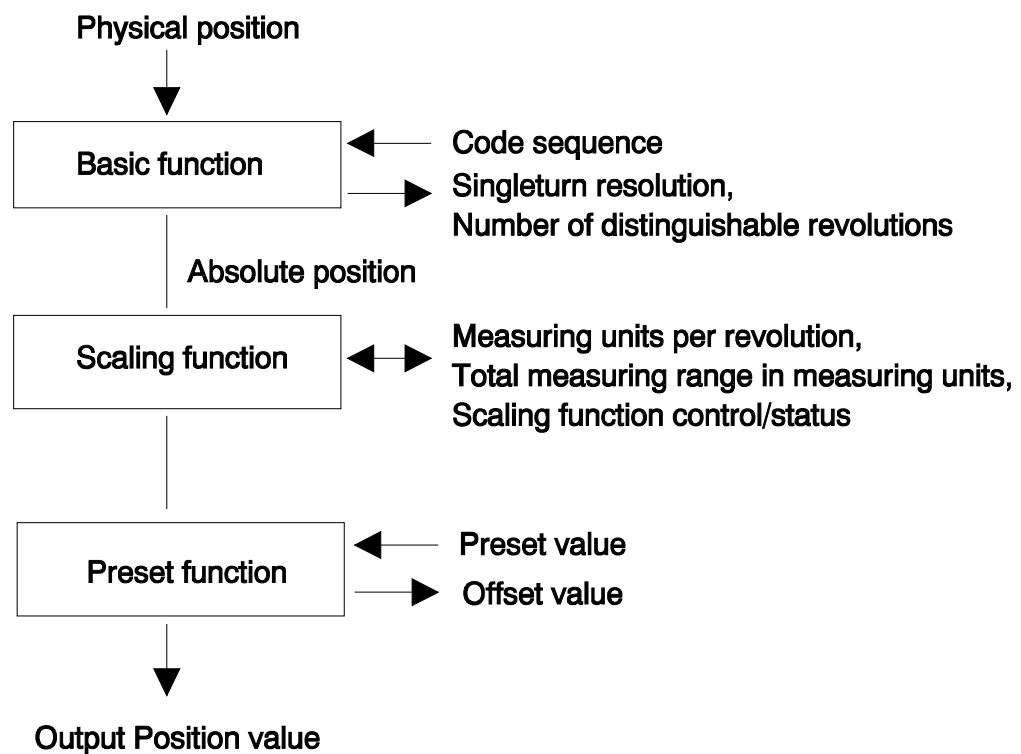
Topic	Project	Sign.	Date
CANopen Encoder Application Note	CANopen	Thr	06-03-30

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.



Document RXA CANopenAppl	Revision 1.1	Page 5	Total 14
Topic CANopen Encoder Application Note		Project CANopen	Sign. Thr
		Date 06-03-30	

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 singleturn encoder the parameter "Total measuring range in measuring units" does not need to be set as it is equal to the Measuring units per revolution.

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 13 bit encoder the permissible value for the "Measuring units per revolution" is between 1 and 2^{13} (8192).

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				

Document	Revision	Page	Total
RXA CANopenAppl	1.1	6	14
Topic	Project	Sign.	Date
CANopen Encoder Application Note	CANopen	Thr	06-03-30

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 formula a 13 bit singleturn encoder is used as an example.

Formula for the singleturn scaling function:

$$\text{Output_position} = \text{measuring_units_per_revolution} * \text{singleturn_position} / 8192$$

where:

singleturn_position = The Absolute position value of the encoder singleturn disk.

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				

		Document RXA CANopenAppl	Revision 1.1	Page 7	Total 14
Topic CANopen Encoder Application Note	Project CANopen	Sign. Thr	Date 06-03-30		

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 CANopen Encoder.

5.1 Bus installation

When the encoder is used as the first or last station on the BUS, the cable must be terminated with a terminating resistor (120 ohm) between `CAN_L` and `CAN_H`. There is no terminating resistor inside the encoder.

Installation:

1. Connect the CAN bus to the encoder terminals/cables marked `CAN_L` and `CAN_H`. Make sure that the same conductor is always connected to the same terminal/cable (e.g. green conductor always connected to `CAN_H`, yellow conductor always connected to `CAN_L`).
2. Install the terminating resistor if this is the first or last station on the bus.

5.2 Address setting

The physical address (node number) of the encoder can be set in two ways:

- Hardware address setting with address offset. The hardware addresses (1 – 15) are set through binary coded pins/wires separately grounded (connected to 0V) to get the desired address (Open = 1, Grounded = 0). The address offset (0 - 112) are set through object 5002h, for information see the manufacturer specific functions below. The encoder address are the hardware address + address offset. If all pins/wires are left open and the address offset are 0, the Encoder gets the address 15.
- Software address setting from the CANopen master (1 - 127) through Object 5001h, for information see the manufacturer specific functions below.

The encoder reads the address switch only at power-up.

NOTE! The selected address (node number) affects the default identifier allocation, see section 2 for further information.

		Document RXA CANopenAppl	Revision 1.1	Page 8	Total 14
Topic CANopen Encoder Application Note	Project CANopen	Sign. Thr	Date 06-03-30		

5.3 **Baudrate setting**

The transmission rate of the gateway must be set with the manufacturer specific object 5000h from the CANopen master, for further information see the manufacturer specific functions below.

The default baudrate is 125kbit/s.

5.4 **EDS and DCF files**

An EDS-file (Electronic Data Sheet) is delivered together with the CANopen Encoders. 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.

5.5 **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.

Document RXA CANopenAppl	Revision 1.1	Page Total 9 14
Topic CANopen Encoder Application Note	Project CANopen	Sign. Date Thr 06-03-30

5.6 Cables and connectors

Cable with hardware addressing: 12x0.25mm, shielded.

Function:	Colour:
+E Volt	Red
0 Volt	Blue
CAN_H	Green
CAN_L	Yellow
CAN_GND	Grey/Pink
Address bit 0	White
Address bit 1	Brown
Address bit 2	Grey
Address bit 3	Pink
Address 0 Volt	Red/Blue
NC	Black
NC	Violet
Case	Shield

Pin assignment for 8 Pin Male PT connector.

Pin No.	Signal	Meaning
A	Address bit 0	Device address bit 0
B	Address bit 1	Device address bit 1
C	Address bit 2	Device address bit 2
D	Address bit 3	Device address bit 3
G	CAN_H	Receive/Transmit Data
H	CAN_L	Receive/Transmit Data
F	0 V	Supply Voltage -
E	+E V	Supply Voltage +

External shield is on housing of coupling or flange socket.

Document	Revision	Page	Total
RXA CANopenAppl	1.1	10	14
Topic		Sign.	Date
CANopen Encoder Application Note		Thr	06-03-30
Project			
CANopen			

6. Manufacturer specific functions

6.1 Object 5000h - Transmission rate

Nine transmission rates are supported by the encoders. The selection of the transmission rate can be done by 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	3 = 125kbit/s

STRUCTURE OF PARAMETER

Value	Function
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
8	100 kBit/s

Document	Revision	Page	Total
RXA CANopenAppl	1.1	11	14
Topic	Project	Sign.	Date
CANopen Encoder Application Note	CANopen	Thr	06-03-30

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

Document RXA CANopenAppl	Revision 1.1	Page 12	Total 14
Topic CANopen Encoder Application Note	Project CANopen	Sign. Thr	Date 06-03-30

6.3 **Object 5002h – Address offset**

The Address offset are used together with the address pins / wires to set the encoder hardware address.

When Node number = FFh (Hardware settings):

Encoder address (1 - 127) = Hardware address (0 - 15) + Address offset (0 - 112)

OBJECT DESCRIPTION

INDEX	5002 H
Name	Address offset
Object Code	VAR
Data Type	Unsigned 8

VALUE DESCRIPTION

Object Class	Manufacturer specific
Access	Rw
PDO Mapping	No
Value Range	0 – 112
Mandatory Range	No
Default Value	0

Document	Revision	Page	Total
RXA CANopenAppl	1.1	13	14
Topic	Project	Sign.	Date
CANopen Encoder Application Note	CANopen	Thr	06-03-30

7. Encoder setup example

This example shows a simple setup of the Encoder 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 if Hardware addressing is used, see section 5.2 for further information.
2. Set the baudrate of your CAN network to the baudrate of the encoder (125 kbit if the Encoder still is using the default baudrate). See section 6.1 for further information.
3. Power up the encoder.
4. The encoder will send a Boot-up message on the default emergency identifier (ID = 128 + encoder address), the message has no data bytes.
5. The encoder 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 + encoder address) to the cyclic timer object (Object 6200h) with the data below. The encoder will confirm 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

6. To get the encoder 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 encoder address (Node Number)

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

		Document RXA CANopenAppl	Revision 1.1	Page 14	Total 14
Topic CANopen Encoder Application Note	Project CANopen		Sign. Thr	Date 06-03-30	

Appendix A, History

Revision	Date	Changes
Rev. 1.0	04-08-17	First release