User manual

Ax58x CB
CC-CB

CANopen

DS 406 - Device profile for encoders

Chapters

1 - Safety summary

For the electrical connections, we recommend to closely follow these electrical instructions. In particular, according to the 89/336/EEC norm on electromagnetic compatibility, following precautions must be taken:

- Install the encoder as close as possible to the electronic control unit.
- Always use shielded and twisted cables if possible.
- Avoid running the signal cables near high voltage power cables (e.g. drive cables).
- Install EMC filters on sensor power supply if needed.
- Avoid mounting sensor near capacitive or inductive noise sources and switching power supplies.

Connect according to the chapter 4: "Electrical connections".

2 - Identification

The device can be identified by the label's data (ordering code, serial number). This information is listed in the delivery document. For technical features of the product, refer to the technical catalogue.

c UL us File E211871

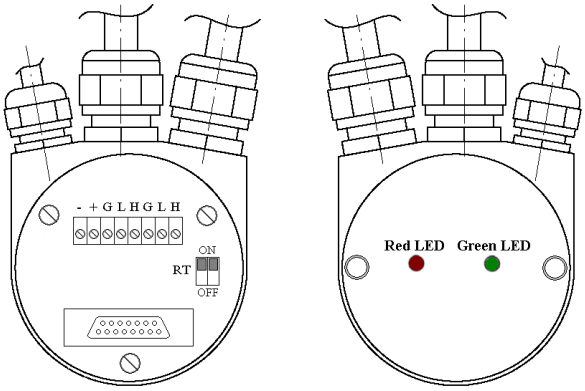
3 - Installation

Install the device according to the provided protection level. Protect the system against knocks, friction, solvents and respect the environmental characteristics of the unit.

4 - Electrical connections

This device is to be supplied by a Class 2 Circuit or Low-Voltage Limited Energy or Energy Source not exceeding 30 Vdc.

4.1 CANopen cover



Three cables connect the rotary encoder. The power supply is achieved with a two-wire connection cable through one PG9. Each one of the twisted-pair and shielded bus lines are guided in and out through two PG9 on the right side (as seen on clamps). We strongly recommend to use the appropriate shielded and twisted cables.

Clamp	Description
-	0 Vdc Supply voltage
+	+10Vdc +30Vdc Supply voltage
G	CAN GND
L	CAN Low
H	CAN High

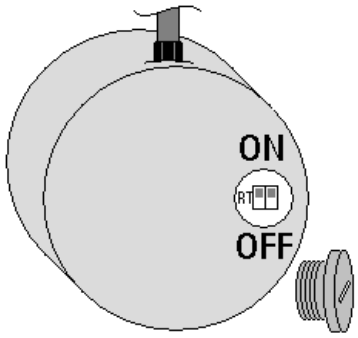
LED indicator

Two LEDs on the backside of the connection cap show the status of the CAN interface following the table below:

GREEN LED	Description
ON	The encoder is in state Operational
Single flash	The encoder is in state Stopped
Blinking	The encoder is in state Pre-Operational

RED LED	Description
ON	The CAN controller is bus off
Double flash	Node guarding error
Single flash	Warning limit reached
Blinking	Flash memory error
OFF	No error

4.2 CANopen cable output



Colour	Description
Red	+10Vdc +30Vdc Supply voltage
Black	0 Vdc Supply voltage
White	CAN_H
Blue	CAN_L

With CANopen cable output the LEDs are not provided.

4.3 Bus termination

A resistor is provided in the connection cap or under threaded cap (with cable output), which must be used as a line termination on the last device. To activate it slide the RT switch.

RT	Description
ON	if encoder is last device of CANbus line
OFF	if encoder is not last device of CANbus line

4.4 Node number and baud rate

Attention:

Hardware setting of the node number and the bit rate are not expected, only software setting are possible.

To set the node number, refer to the object 3001h of Object Dictionary.

To set the bit rate, refer to the object 3000h of Object Dictionary.

At first start up, the master device have to synchronized to slave device (encoder) with slave baud rate, when a communication is established, set baud rate and node number wishes (objects 3000h and 3001h). Send a reset node and store the parameters.

To avoid conflict with other devices, this operation must be done with only one slave connected to the CAN network.

5 - CANopen interface

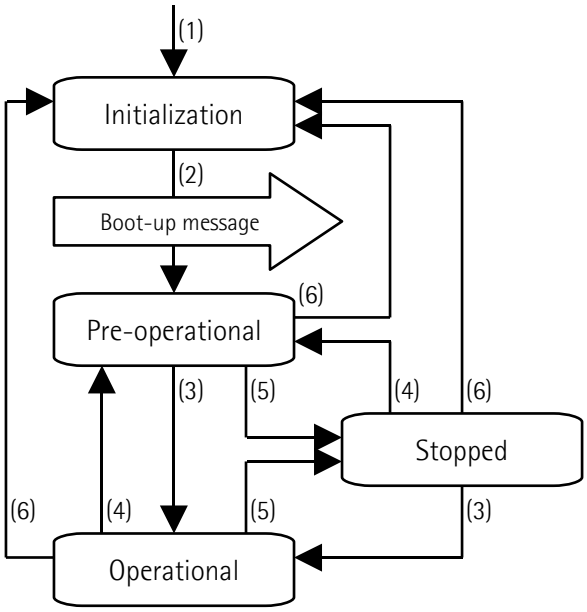
Lika encoders are always slave devices and they respect the "Device profile for encoders", Class 2.
For every omitted specify, refer to the documents "CiA Draft Standard 301" and "CiA Draft Standard 406" available on www.can-cia.org.

5.1 File EDS

CANopen encoders are supplied with EDS file **AxCB_DS406_V1.eds** (see enclosed support or **www.lika.biz > products > fieldbus > CANopen**). Install EDS file on CANopen master device.

5.2 State machine

The CANopen device provide a state working, the device may be switched in different state sending a specific NMT message.
The state diagram is show below:



(1)	Power on
(2)	State initialization finished, the boot-up message is sent automatically
(3)	NMT message: "Start remote node"
(4)	NMT message: "Enter pre-operational"
(5)	NMT message: "Stop remote node"
(6)	NMT message: "Reset node" or "Reset comm."

5.2.1 Initialization

This is the first state the CANopen device enters after power-on or hardware reset. After finishing the basic CANopen device initialization the device read the parameters stored in EPROM, than the device send a boot-up message and enters autonomously into the "Pre-operational" state.

5.2.2 Pre-operational

In this state communication via SDOs is possible. PDOs do not exist, so PDO communication is not allowed. Configuration of PDOs and parameters may be performed by a configuration application.
The device may be switched into the Operational state directly by sending a "Start remote node" message.

5.2.3 Operational

In this state all communication objects are active. The constructor uses the parameters as described in the object dictionary and may sent process data using PDO. Object dictionary access via SDO is possible.
The device may be switched into the Pre-operational state directly by sending a "Enter pre-operational" message.

5.2.4 Stopped

In this state the device is forced to stop the communication altogether (except node guarding, if active). PDO and SDO communications are not allowed.
The device may be switched into the Operational state or Pre-operational state directly by sending the specific NMT message.

5.3 Communication objects

There are 4 type of communication messages:

- Network management NMT: the NMT master controls the NMT state of the NMT slaves.
- Process Data Objects PDO: used to transfer the real-time data.
- Service Data Object SDO: used to provide direct access to entries of a CANopen devices object dictionary.
- Special Function Object: Sync: provides the basic network synchronization mechanism. After this service the consumers may sent real-time data. Emergency: object transmitted only once per error event. Nodeguard: used to know the slave status.

Relation between device states and communication objects:

	Initial.	Pre-oper.	Operat.	Stopped
NMT		X	X	X
PDO			X	
SDO		X	X	
Sync			X	
Emerg		X	X	
Boot-up	X			
Nodeg.		X	X	X

5.3.1 Pre-defined connection set

Master → Slave broadcast		
COB (Object) Kind	Function code (binary)	COB-ID (hex)
NMT	0000	000
SYNC	0001	080

pear-to-pear object		
EMERGENCY	0001	081 - 0FF
PDO 1 (tx)	0011	181 - 1FF
PDO 2 (tx)	0101	281 - 2FF
PDO 3 (tx)	0111	381 - 3FF
SDO (tx)	1011	581 - 5FF
SDO (rx)	1100	601 - 67F
Nodeguard	1110	701 - 77F

"COB kind" (tx or rx) is seen from the slave device point of view.
Boot-up messages use the nodeguard COB-ID.

5.4 NMT objects

NMT structure:

COB-ID (11 bit)	
Func.code	Node ID
0000	0

2 CAN Data Bytes	
Command	Slave ID
NMT Func.	Slave ID

if Slave ID = 00h, the NMT message is directed to all network node.

NMT Function:

Command	NMT Function	Status node
01 hex	Start remote node	Operational
02 hex	Stop remote node	Stopped
80 hex	Enter pre-operational	Pre-operational
81 hex	Reset node	Pre-operational
82 hex	Reset communication	Pre-operational

5.5 Boot-up objects

Boot-up message structure:

COB-ID(hex)	1 CAN Data Bytes
700+Node ID	00

5.6 PDO objects

PDO(tx) messages are always composed by 4 CAN Data Bytes and they are used from the encoder to transmit the position value.
PDO structure:

IDENTIFIER		4 CAN Data Bytes			
COB-ID(hex)		Byte 0	Byte 1	Byte 2	Byte 3
F.C.	Node-ID	2^7 - 2^0	2^{15} - 2^8	2^{23}- 2^{16}	2^{31}- 2^{24}
		Low	...	...	High

3 kind of PDO are defined:

PDO1 Cyclic mode: asynchronous transmission.
The absolute rotary encoder transmits cyclic, without being called by the host, the current process value. The cycle time can be programmed in milliseconds for values between 1 ms and 65536 ms (see cyclic time: object 6200h).
To enable (disable) the cyclic mode, set to "0" ("1") the most significant bit of COB-ID used by PDO1 (object 1800h, sub1).

PDO2 and **PDO3** Sync mode: synchronous transm.
Synchronous means that the PDO is transmitted after the Sync. The Sync is a high-priority COB transmitted by Master to all devices after which the encoders will send back their process value. Each device will reply on NODE-ID order. If an encoder has not to reply to all Sync command, it is possible to set it for replying only after n Sync commands.
For PDO2 the value of n can be set on object 1801h sub 2. For PDO3 the value of n can be set on object 1802h sub 2.
To enable (disable) the Sync mode, set to "0" ("1") the most significant bit of COB-ID used by PDO (object 1801h / 1802h, sub1).

NOTE:
More than one transmission mode can be active at the same time.

5.7 SDO objects

SDOs messages are used to know or modified encoder parameters, these parameters are enclosed in the "Object dictionary". Max 4 bytes are used for CAN data, other 4 bytes are used for Command, Index and Sub-index fields. SDOs are always follow by confirmation: when Master send a SDO to Slave, it always reply (with warning in case of problem).

SDO structure:

IDENTIFIER	
COB-ID(hex)	
F.C.	Node-ID

4 CAN Data Bytes			
0	1	2	3
Command	Index		Sub index
1 byte	LSB	MSB	1 byte

From 1 to 4 CAN Data Bytes			
4	5	6	7
Process data			
LSByte	...	...	MSByte

5.7.1 Command

The command byte contents the kind of telegram which is sent across the CAN network.
There are three kinds of telegrams:

- Set: to send configuration parameters to a device;
- Req: used by Master to read data from a device;
- Warnings: used by slave to send to Master error messages (es. index does not exist, illegal parameter, ...).

Command	COB	COB type	Data length
22h	Set	M → S request	not spec.
23h	Set	M → S request	4 byte
2Bh	Set	M → S request	2 byte
2Fh	Set	M → S request	1 byte
60h	Set	S → M confirmation	
40h	Req	M → S request	0 byte
42h	Req	S → M reply	not spec.
43h	Req	S → M reply	4 byte
4Bh	Req	S → M reply	2 byte
4Fh	Req	S → M reply	1 byte
41h	Req	S → M reply segmented SDO	
80h	Warning	S → M reply	4 byte

5.8 Object dictionary

Each implemented object is listed as follows:
Index-subindex Object name [data types, attribute]

- Index and subintex are written in hexadecimal notation.
- Attribute: ro = read only access
  rw = read and write access
- Unsigned16 data type:

Process data byte	
byte 4	byte 5
LSByte	MSByte

- Unsigned32 data type:

Process data byte			
byte 4	byte 5	byte 6	byte 7
LSByte	...	...	MSByte

5.8.1 Standard objects (DS 301)

1000-00 Device type [Unsigned32, ro]
Default = 0001 0196h =single turn encoder,DS 406
0002 0196h =multi turn encoder,DS 406

1001-00 Error register [Unsigned8, ro]
If a bit is set to "1" the specified error has occurred.
Default = 00h

1003 Pre-defined error field
This object holds the errors that have occurred on the device.
-00 Number of actual errors [Unsigned8, rw]
Writing 00h deletes the entire error history.
-01 Last error occurred [Unsigned32, ro]
-02...04 Older error occurred [Unsigned32, ro]

1005-00 COB_ID SYNC message [Unsigned32, rw]
Default = 0000 0080h

1008-00 Manufacturer device name [String, ro]
Default = "Lika"

1009-00 Hardware version [String, ro]

100A-00 Software version [String, ro]

100C-00 Guard time [Unsigned16, ro]
Default = 03E8h (milliseconds)

100D-00 Life time factor [Unsigned8, ro]
Default = 05h

"Node life time" = "Obj_100C" * "Obj_100D"
"Node guarding" is enable if "Node life time" ≠ 0.
If the slave is not guarded within its lifetime, it informs its local application about that event with "Life Guarding Event", red LED indicates the node guarding error and the object 1001h and 1003h are up to date. To delete the error send a "Reset node".

1010-01 Store parameters [Unsigned32, rw]
This object supports the saving of all parameters in non-volatile memory. The signature that shall be written is "save":
Master → Encoder

COB-ID	Cmd	Index		Sub	Data bytes			
600+ID	23	10	10	01	73	61	76	65

Encoder → Master (confirmation)

COB-ID	Cmd	Index		Sub	Data bytes			
580+ID	60	10	10	01	00	00	00	00

1011-01 Restore default parameters [Unsig32, rw]
With this object the default values of all parameters are restored. The signature that shall be written is "load". The default values are set valid after the device is reset:
Master → Encoder

COB-ID	Cmd	Index		Sub	Data bytes			
600+ID	23	11	10	01	6C	6F	61	64

Encoder → Master (confirmation)

COB-ID	Cmd	Index		Sub	Data bytes			
580+ID	60	11	10	01	00	00	00	00

Master → Encoder (reset node)

COB-ID	Cmd	Slave ID
000	81	ID

Encoder → Master (Boot-up)

COB-ID	Cmd
700+ID	00

To keep the default value execute the "Store parameters" function (see object 1010h).

1014-00 COB-ID EMCY [Unsigned32, rw]
This object defines the COB-ID for the EMCY write service.
Default = 80h+NodeID

1015-00 Inhibit time EMCY [Unsigned16, rw]
The value is given in multiples of 100 µs.
Default = 00h (disable)

1018 Identity object
-01 Vendor number [Unsigned32, ro]
-02 Product number [Unsigned32, ro]
-03 Revision number [Unsigned32, ro]

1800 Transmit PDO1 parameters
This PDO transmits asynchronously the position value of the encoder. To set the cyclic timer see the 6200h object.
-01 COB-ID of the PDO1 [Unsigned32, rw]
Default = 4000 0180h+NodeID (no RTR, COB-ID)
-02 Transmission type [Unsigned8, rw]
Default = FEh (asynchronous transmission)

1801 Transmit PDO2 parameters
This PDO transmits synchronously the position value of the encoder.
-01 COB-ID of the PDO2 [Unsigned32, rw]
Default = 4000 0280h+NodeID (no RTR, COB-ID)
-02 Transmission type [Unsigned8, rw]
Default = 01h(synchronous transmission every Sync)
For replying only after n Sync commands the value of n can be set on object 1801h sub 2.

1802 Transmit PDO3 parameters
This PDO transmits synchronously the position value of the encoder.
-01 COB-ID of the PDO3 [Unsigned32, rw]
Default = C000 0380h+NodeID (disable, no RTR)
-02 Transmission type [Unsigned8, rw]
Default = 01h(synchronous transmission every Sync)
For replying only after n Sync commands the value of n can be set on object 1802h sub 2.

NOTE:
To enable (disable) the transmission of PDO1, PDO2 and PDO3, set to "0" ("1") the most significant bit of COB-ID used by PDO (object 180xh, sub1).

1A00-01 PDO1 mapping parameter [Unsig32, rw]
This object follows device profile DS406 and contains the mapped position value of the encoder.
Default = 6004 0020h

1A01-01 PDO2 mapping parameter [Unsig32, rw]
See object 1A00h, sub1.

1A02-01 PDO3 mapping parameter [Unsig32, rw]
See object 1A00h, sub1.

5.8.2 Manufacturer specific objects

2104-00 Limit switch min [Unsigned32, rw]
If the position value is less than object 2104h, the bit 12 of object 6500h is set to "1". To enable this function set to "1" the bit 12 of object 6000h.
Default = 0000 0010h

2105-00 Limit switch max [Unsigned32, rw]
If the position value is higher than object 2105h, the bit 13 of object 6500h is set to "1". To enable this function set to "1" the bit 13 of object 6000h.
Default = 003F FFF0h

3000-00 baud rate [Unsigned8, rw]
This object defines the baud rate of the device following the table below:

Data byte	Baud rate
00h	20 Kbit/s
01h	50 Kbit/s
02h	100 Kbit/s
03h	125 Kbit/s
04h	250 Kbit/s
05h	(default) 500 Kbit/s
06h	800 Kbit/s
07h	1000 Kbit/s

The correct procedure to change the baud rate is: set object 3000h, send a "reset node" (or "reset communication"), store parameter.

Master → Encoder

COB-ID	Cmd	Index		Sub	Data byte
600+ID	2F	00	30	00	see table

Encoder → Master (confirmation)

COB-ID	Cmd	Index		Sub	Data byte
580+ID	60	00	30	00	00

Master → Encoder (reset node)

COB-ID	Cmd	Slave ID
000	81	ID

Change to new value the master baud rate
Encoder → Master (Boot-up with new baud rate)

COB-ID	Cmd
700+ID	00

Store parameters (see object 1010h), otherwise at next power up will be load the old baud rate value.

3001-00 Node-ID [Unsigned8, rw]
This object defines the node identifier of the device. The correct procedure to change the Node-ID is: set object 3001h, send a "reset node", store parameter.
Default = 01h

Master → Encoder

COB-ID	Cmd	Index		Sub	Data byte
600+ID	2F	01	30	00	new Node-ID

Encoder → Master (confirmation)

COB-ID	Cmd	Index		Sub	Data byte
580+ID	60	01	30	00	00

Master → Encoder (reset node)

COB-ID	Cmd	Slave ID
000	81	old ID

Encoder → Master (Boot-up with new Node-ID)

COB-ID	Cmd
700+ID	00

Store parameters (see object 1010h), otherwise at next power up will be load the old Node-ID value.

5.8.3 Device profile objects (DS 406)

6000-00 Operating parameters [Unsigned16, rw]

Bit	Function	bit = 0	bit = 1
0	Code sequence	Clockwise	CCW
1	not used		
2	Scaling function	disable	enable
3...11	not used		
12	Limit switch min	disable	enable
13	Limit switch max	disable	enable
14...15	not used		

Default = 0000h

- The code sequence defines whether increasing or decreasing position values are output when the encoder shaft rotates clockwise or counterclockwise as seen on the shaft.
- Scaling function: if disable the device uses the physical resolution (see objects 6501h and 6502h), if enable it uses the resolution set on objects 6001h and 6002h with the following relationship:

$$posTx = \frac{obj\_6001}{obj\_6501} \cdot RealPos \leq obj\_6002$$

6001-00 Resolution per revolution [Unsig32, rw]
This object sets the number of distinguishable steps per revolution. Enable if bit 2 of 6000h = "1".

To avoid counter error, check that $\frac{obj\_6501}{obj\_6001}$ is an integer value.

6002-00 Total measuring range [Unsigned32, rw]
This object sets the number of distinguishable steps over the total measuring range.
Enable if bit 2 of 6000h = "1".

Example:
Multiturn encoder with "total physical resolution"= 16777216, obj_6501 = 4096 and 2048 steps per revolution (obj_6001) are required:
obj_6002 = 16777216 * 2048 / 4096 = 8388608

6003-00 Preset value [Unsigned32, rw]
The preset value is the desired position value, which should be reached at a certain physical position of the encoder shaft. The position value is set to the desired process value by the parameter preset.
The preset value must not exceed the total physical resolution to avoid run-time errors.

6004-00 Position value [Unsigned32, ro]
The object defines the output position value for the communication objects 1800h, 1801h and 1802h.

6200-00 Cyclic time [Unsigned16, rw]
Cyclic timer is used, in asynchronous communication, to adjust the timing from a PDO1 (object 1800h) transmission to the next.
Default = 0064h (100ms)

6500-00 Operating status [Unsigned16, ro]

Bit	Function	bit = 0	bit = 1
0	Code sequence	Clockwise	CCW
1	not used		
2	Scaling function	Disable	Enable
3...11	not used		
12	Limit switch min*	posit. > obj_2104	posit. < obj_2104
13	Limit switch max*	posit. < obj_2105	posit. > obj_2105
14...15	not used		

*: to use this function sets to "1" bits 12 and 13 of 6000h.

6501-00 Physical resolution per revolution [Unsigned32, ro]
This object defines the physical number of distinguishable steps per revolution. To use different value see object 6001h

6502-00 Physical number of revolution [Unsigned16, ro]
This object defines the physical number of distinguishable revolution.
"total physical resolution"= "obj_6501" * "obj_6501"

To use different value see objects 6001h and 6002h.

6504-00 Supported alarms [Unsigned16, ro]
Default = 0000h (no supported alarms)

6506-00 Supported warnings [Unsigned16, ro]
Default = 0000h (no supported warnings)

6507-00 Profile and software version [Unsig32, ro]
Default = 0301 0101h: software version = 1.1
profile for encoder version = 3.1

6508-00 Operating time [Unsigned32, ro]
Default = FFFF FFFFh (not used)

6509-00 Offset value [Integer32, ro]
This object contains the offset value, it is calculated by the preset function and shifts the position value with the calculated value.

650A-01 Manufacturer offset value [Integer32, ro]
This object contains the manufacturer offset value. This value gives information on the shift of the zero point in the number of positions from the physical zero point of the encoder disk.

650B-00 Serial number [Unsigned32, ro]
Default = FFFF FFFFh (not used)

NOTE:
To keep the parameters changed execute the "Store parameters" function (see object 1010h).
In case of "reset node" command, "reset communication" command or power off, if parameters are not stored they will be lost.

6 - Setup

Below some examples of parameters setting with, in evidence, data exchange between Master and Device. A generic value "ID" is used to indicate the encoder address.
Following values are written in hexadecimal notation.

Set Operational, Pre-operational status

NMT message Master → Slave

	COB-ID	Cmd	Node
Operational:	000	01	ID
Pre-operational:	000	80	ID

Set resolution per revolution (2^{16}=0001 0000h)

Master → Encoder (Set request)

COB-ID	Cmd	Index		Sub	Process data			
600+ID	23	01	60	00	00	00	01	00

Encoder → Master (Set confirmation)

COB-ID	Cmd	Index		Sub	Process data			
580+ID	60	01	60	00	00	00	00	00

Set total resolution (2^{28}=1000 0000h)

Master → Encoder (Set request)

COB-ID	Cmd	Index		Sub	Process data			
600+ID	23	02	60	00	00	00	00	10

Encoder → Master (Set confirmation)

COB-ID	Cmd	Index		Sub	Process data			
580+ID	60	02	60	00	00	00	00	00

Set Operating parameter (Code sequence: CW, scaling function: enable, limit switch: disable)

Master → Encoder (Set request)

COB-ID	Cmd	Index		Sub	Process data			
600+ID	2B	00	60	00	04	00	-	-

Encoder → Master (Set confirmation)

COB-ID	Cmd	Index		Sub	Process data			
580+ID	60	00	60	00	00	00	-	-

Set Preset value (preset = 1000 = 03E8h)

Master → Encoder (Set request)

COB-ID	Cmd	Index		Sub	Process data			
600+ID	23	03	60	00	E8	03	00	00

Encoder → Master (Set confirmation)

COB-ID	Cmd	Index		Sub	Process data			
580+ID	60	03	60	00	00	00	00	00

Set Sync counter (n = 5 = 05h)

Master → Encoder (Set request)

COB-ID	Cmd	Index		Sub	Process data			
600+ID	2F	01	18	02	05	-	-	-

Encoder → Master (Set confirmation)

COB-ID	Cmd	Index		Sub	Process data			
580+ID	60	01	18	02	00	-	-	-

Disable Sync mode

Read COB-ID used by PDO2:

Master → Encoder (Req request)

COB-ID	Cmd	Index		Sub	Process data			
600+ID	40	01	18	01	-	-	-	-

Encoder → Master (Req reply)

COB-ID	Cmd	Index		Sub	Process data			
580+ID	43	01	18	01	B0	B1	B2	B3

COB-ID used by PDO2 =
((B3<<24) | (B2<<16) | (B1<<8) | B0)
set to 1 the most significant bit:

```
B3 |= 0x80;
```

Set new COB-ID used by PDO2:

Master → Encoder (Set request)

COB-ID	Cmd	Index		Sub	Process data			
600+ID	23	01	18	01	B0	B1	B2	B3

Encoder → Master (Set confirmation)

COB-ID	Cmd	Index		Sub	Process data			
580+ID	60	01	18	01	00	00	00	00

Enable Cyclic mode

Set cyclic time (100ms = 64h)

Master → Encoder (Set request)

COB-ID	Cmd	Index		Sub	Process data			
600+ID	2B	00	62	00	64	00	-	-

Encoder → Master (Set confirmation)

COB-ID	Cmd	Index		Sub	Process data			
580+ID	60	00	62	00	00	00	-	-

Read COB-ID used by PDO1:

Master → Encoder (Req request)

COB-ID	Cmd	Index		Sub	Process data			
600+ID	40	00	18	01	-	-	-	-

Encoder → Master (Req reply)

COB-ID	Cmd	Index		Sub	Process data			
580+ID	43	00	18	01	B0	B1	B2	B3

COB-ID used by PDO1 =
((B3<<24) | (B2<<16) | (B1<<8) | B0)
set to 0 the most significant bit:

```
B3 &= 0x7F;
```

Set new COB-ID used by PDO1:
Master → Encoder (Set request)

COB-ID	Cmd	Index		Sub	Process data			
600+ID	23	00	18	01	B0	B1	B2	B3

Encoder → Master (Set confirmation)

COB-ID	Cmd	Index		Sub	Process data			
580+ID	60	00	18	01	00	00	00	00

To keep the parameters changed execute the "Store parameters" function (see object 1010h).
In case of "reset node" command, "reset communication" command or power off, if parameters are not stored they will be lost.

6.1 Warning objects

In order to know the meaning of warning message make reference to the document "CiA Draft Standard 301" on chapter "SDO abort codes" available on www.can-cia.org.

6.2 Emergency objects

Emergency objects are triggered by the occurrence of the device internal error situation.

EMCY structure:

IDENTIFIER	CAN Data Byte			
COB-ID(hex)	0	1	2	3...7
see object 1014h	Error code		Error register	Specific code
	LSB	MSB	1001	00...00

Defined error codes:
1000h = Node guarding error
5530h = Flash memory error

Rev.	Man.Vers.	Description
0	1.0	1st issue
		Manual update
3	1.3	Manual update
4	2.0	SW and HW CANopen interface update, manual update
5	2.1	Add cable output (chapter 4)