User manual
Preliminary

SMAL2
ExC-D9F-/S25...

Description

This manual describes products of the SMAL2 series that works in conjunction with MTAL2 magnetic tape. The purpose of this system is to measure linear displacements on lift system and on automation systems. The device is composed by a sensor with an integrated conversion electronic that moving along the magnetic tape, generates a signal equivalent to an absolute encoder. The device can be provided with SSI, CANopen or RS485 interface or AB0 (incremental output).

CANopen SSI

DS406 - Device profile for encoder

Chapters

1 - Safety rules

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

- Install the device as close as possible to the electronic control unit.
- Always use shielded and twisted cables if possible.
- Connect shield to the ground of system.
- 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.
- static discharge can damage internal sensitive electronic components. Before handling and installing, discharge electrical charge from your body and tools which may come in touch with the device.

Connect according to the chapter 5: "Electrical connections".

2 - Identification

The device can be identified by the label's data (ordering code, production date). This information are also listed in the delivery document.

3 - Installation

To set absolute mode turn ON the switch (near LED, see chapter 11.1), to avoid unintentional turn OFF, the switch is hidden inside the housing and have to be moved using a little screwdriver.
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 - Mounting instructions

4.1 Device

Fix the device on the "L" support with self-locking nuts and than fix the support to a plane surface checking the parallelism between device and magnetic tape.

4.2 Magnetic tape

The magnetic tape must be insert on the device hole and fixed at the ends of the lift vane. The magnetic tape must be free to slide into device.

ATTENTION:

The active side of magnetic tape (black side), however installed, has to face the active part of magnetic sensor on electronic board.

5 - Electrical connections

5.1 DSub 9-pin (Output interfaces)

Color	Pin	RS485 output	SSI output
Yellow	1	n.c.	Clock+
Blue	2	B	Clock-
Black	3	0Vdc GND	0Vdc GND
Grey	4	Reset	Reset
	5	Shield	Shield
White	6	Preset	Preset
Green	7	A	Data+
Orange	8	n.c.	Data-
Red	9	+Vdc *	+Vdc *

Color	Pin	CAN output	AB0 output
Yellow	1	n.c.	A
Blue	2	CAN-L	/A
Black	3	0Vdc GND	0Vdc GND
Grey	4	Reset	0
	5	Shield	Shield
White	6	GND	B
Green	7	CAN-H	/B
Orange	8	n.c.	/0
Red	9	+Vdc *	+Vdc *

n.c. = Not connected

*: +10Vdc +30Vdc

NOTE:

Without external power supply, the internal back-up battery assure the correct working and position reading, however, the position value can not be transmitted and LEDs are not managed.

The position value is lost if the magnetic tape slides out from the device hole.

6 - Functions

The device is active after 2 seconds from power on.

6.1 Digital inputs

6.1.1 Reset

(for SSI, RS485 and CANopen outputs)

This function resets the device.

ATTENTION:

use Reset function only if the device, connected to external power supply and on moving, doesn't transmit it's position and the yellow LED is always OFF.

Perform Reset as follows:

- connect Pin 4 to power supply for at least 1ms;
- move device to zero position (datum);
- perform preset:
  - RS485 and SSI output: see chapter 6.1.2,
  - CAN output: see object 6003 (chap.9.8.3).

All steps of the procedure must be carried out carefully to avoid an incomplete Reset which may cause incorrect data transmission.

6.1.2 Preset (only for SSI and RS485 output)

This function sets to zero the position value.

- Connect Pin 6 to power supply for at least 1ms.

6.2 Switch

The device is provided with a switch used to disconnect the battery and preserve the charge during the storage.

6.3 LEDs functions

LED	Function
Yellow	OFF = device stopped ON = device on motion (LED intensity proportional to speed)
Green	OFF = power supply off ON = power supply on
Red	OFF = High battery level ON = Low battery level or not connected

7 - RS485 interface

7.1 Technical data

Function	Data
Baud rate	see chapter 7.2.3
Data bits	8
Parity bit	No
Stop bit	1
Flow control	No

7.2 Communication protocol

The communication protocol consists of messages with a fix length of 14 bytes. In every transmission a Command is sent by the PC, while the device answers with Acknowledge (except cyclic transmission).

7.2.1 Protocol structure

Byte	Field	Function
0	SOF	Start of Frame
1	ADD	Device address
2,3,4,5	CMD	Command
6	ACK	Acknowledge
7,8,9,10	DATA	Process data
11,12	CHK	Checksum
13	EOF	End of Frame

SOF Start of Frame
Start of message.
SOF = " | " (ASCII) = 7C (hex)

ADD Device address
Byte used to specify the device address. The value of ADD is hexadecimal.
eg. address 12: ADD = 0C (hex)

CMD Command
Byte used to specify the command (sent or received). CMD byte is ASCII coded (see. chap. 7.2.2 for possible values).

ACK Acknowledge
Acknowledge confirms correct transmission of data.
PC → SMAL2: ACK ="Null" (00 Hex),
SMAL2 → PC: ACK =":" (3A Hex)
Other values mean incorrect transmission.

DATA Actual position and Process data
4 bytes used to transmit the actual position or parameter data/values to be set.
DATA byte content must be hexadecimal.
Bytes 7-10 are "signed integer 32 bit" with the following structure:

byte 7			...	byte 10		
MSBit			...			LSBit
2^31	...	2^24	...	2^7	...	2^0

CHK Checksum
2 bytes used to control the correct transmission of message. Value of CHK is the sum of bytes 0-10.
Bytes 11-12 are "unsigned integer 16 bit" with the following structure:

byte 11			byte 12		
MSBit					LSBit
2^15	...	2^8	2^7	...	2^0

Checksum overflow is ignored.

EOF End of Frame
End of message.
EOF = " ♦ " (ASCII) = 04 (hex)

7.2.2 Commands

Transmitted commands have the following meaning:
"**T**" (transmit): means a command from PC to device to read a parameter value.
The device replies with the same CMD, ACK=":" and required value in the DATA field.
"**R**" (receive): means a transmission from PC to device of a DATA value.
The device acquires the value and confirms sending the same CMD, ACK=":" and same DATA values.

Possible values of CMD are the following:

Function	CMD	DATA byte
Actual position	TPOS	byte 7-10= signed 32 bit resolution = 1/10 mm
Device address	TADR	set device address in byte 1 byte 7-10= new address
	RADR	
Baud rate	TBAU	see chapter 7.2.3
	RBAU	
Battery level	TVBA	byte 7-9= 00 byte 10 = 00: low batt. 01: high batt.
Start cyclic mode	STAR	byte 7-10= signed 32 bit cyclic time on milliseconds
Stop cyclic mode	STOP	DATA byte negligible
Used by SMAL2 for cyclic transmissions	"Null"	byte 7-10= signed 32 bit resolution = 1/10 mm

NOTE:

- While sending a transmission command "T" from PC to device the contents of DATA is negligible.
- While sending a CMD "R" command from the PC to SMAL2 the parameter value to transmit has to be set in the DATA field.
- To avoid to assign the same address to more devices, the function Device Address (TADR, RADR) can be used with a single device connected to the PC.

7.2.3 Baud rate

Refer to the table for baud rate settings.

bytes 7-9 = 00 (Hex)

byte 10 (Hex)	Baud rate
00	(default) 9600 bit/s
01	19200 bit/s
02	38400 bit/s
03	57600 bit/s

The device changes its baud rate after to have sent the Acknowledge message.

ATTENTION:

For each Baud rate it is necessary to respect an inferior limit of the cyclic time:

Baud rate	min. cyclic time
9600 bit/s	40 ms
19200 bit/s	20 ms
38400 bit/s	20 ms
57600 bit/s	10 ms

Examples:

Set Device address

(Old address = 0, new address = 20 = 14h)

PC → SMAL2 CMD = RADR

	SOF	ADD	CMD	ACK	DATA	CHK	EOF
Hex	7C	00	52414452	00	00014	01B9	04

SMAL2 → PC CMD = RADR

	SOF	ADD	CMD	ACK	DATA	CHK	EOF
Hex	7C	00	52414452	3A	00014	0207	04

Start cyclic transmission

(cycle time = 100ms = 64h)

PC → SMAL2 (address 0) CMD = STAR

	SOF	ADD	CMD	ACK	DATA	CHK	EOF
Hex	7C	00	53544152	00	00064	021A	04

SMAL2 → PC CMD = STAR

	SOF	ADD	CMD	ACK	DATA	CHK	EOF
Hex	7C	00	53544152	3A	00064	0254	04

Cyclic transmission of actual position

SMAL2 → PC

	SOF	ADD	CMD	ACK	DATA	CHK	EOF
Hex	7C	00	00000000	3A	003E8	01A1	04

Position = 00 00 03 E8 h = 100.0mm

Stop cyclic transmission

PC → SMAL2 (address 0) CMD = STOP

	SOF	ADD	CMD	ACK	DATA	CHK	EOF
Hex	7C	00	53544F50	00	0000	01C2	04

SMAL2 → PC CMD = STOP

	SOF	ADD	CMD	ACK	DATA	CHK	EOF
Hex	7C	00	53544F50	3A	0000	01FC	04

8 - SSI interface

8.1 LSB right aligned protocol

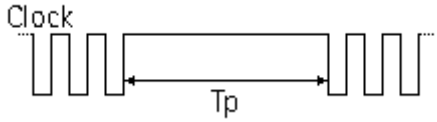
The type of transmission protocol is "right aligned" with a length of 25 bit.
The transmission starts with MSB (most significant bit) and ends with LSB (less significant bit).
Unused bits are set to 0 (zero).
Output code can be Binary or Grey code (see order code).
The position value is transmitted in 1/10 mm.

8.2 Recommended transmission rates

The SSI interface has a frequency of data transmission from 100 kHz to 1 MHz.
The CLOCK signal and DATA signal follow the "EIA standard RS-422".
The transmission rate (baud rate) depends on the length of cables.

Cable length	Baud rate
< 50 m	< 400 kHz
< 100 m	< 300 kHz
< 200 m	< 200 kHz
< 400 m	< 100 kHz

The time interval between two Clock sequence transmissions must be at least 30µs (Tp > 30µs).



9 - CANopen interface (DS 406)

Stem units 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.

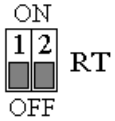
LEDs indicator

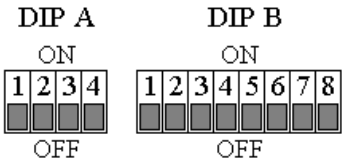
No LEDs are present to indicate CANopen interface status. LEDs functions are specify on chapter 6.3.

File EDS

CANopen devices are supplied with EDS file SMAL2_DS406_V1.eds (see enclosed support or visit www.lika.biz).
Install EDS file on CANopen master device.

9.1 Switch





9.1.1 Bus termination: RT

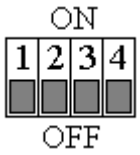
A resistor is provided in the electronic circuit, which must be used as a line termination on the last device. To activate it slide the RT switch.

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

9.1.2 Baud rate: DIP A

The bit rate can be set both hardware and software mode.
If DIP A bit 4 = "OFF" the bit rate is defined by the object 3000h of the Object Dictionary, it can be modified by SDO messages.
If DIP A bit 4 = "ON" the bit rate is defined by DIP A.

DIP A:



With device switched off, set the bit rate in binary value: ON=1, OFF=0

bit	1 LSB	2	3 MSB	4
	2^0	2^1	2^2	ON/OFF

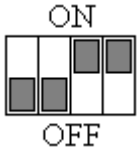
Baud rate value table:

Decimal value	Binary value	Baud rate
0	000	20 Kbit/s
1	001	50 Kbit/s
2	010	100 Kbit/s
3	011	125 Kbit/s
4	100	250 Kbit/s
5	101	500 Kbit/s
6	110	800 Kbit/s
7	111	1000 Kbit/s

Example:

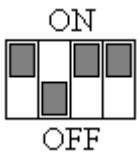
Set 250Kbit/s:
$4_{10} = 100_2$ (binary value, see previous table)

bit	1	2	3	4
	2^0	2^1	2^2	2^3
	OFF	OFF	ON	ON



Set 500Kbit/s:
$5_{10} = 101_2$ (binary value, see previous table)

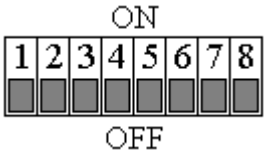
bit	1	2	3	4
	2^0	2^1	2^2	2^3
	ON	OFF	ON	ON



9.1.3 Node number: DIP B

The node number can be set both hardware and software mode.
Permissible addresses lie between 1 and 127.
If all bits of DIP B are "OFF" the node number is defined by the object 3001h of the Object Dictionary, it can be modified by SDO messages.
If at least bit of DIP B is "ON" the node number is defined by DIP B.

DIP B:

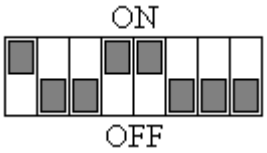


With device switched off, set the node number in binary value: ON=1, OFF=0

bit	1 LSB	2	3	4	5	6	7 MSB	8 not used
	2^0	2^1	2^2	2^3	2^4	2^5	2^6	

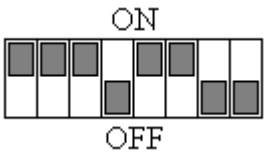
Example:
Set node number = 25:
$25_{10} = 001\ 1001_2$ (binary value)

bit	1	2	3	4	5	6	7	8
	2^0	2^1	2^2	2^3	2^4	2^5	2^6	
	ON	OFF	OFF	ON	ON	OFF	OFF	OFF



Set node number = 55:
$55_{10} = 011\ 0111_2$ (binary value)

bit	1	2	3	4	5	6	7	8
	2^0	2^1	2^2	2^3	2^4	2^5	2^6	
	ON	ON	ON	OFF	ON	ON	OFF	OFF

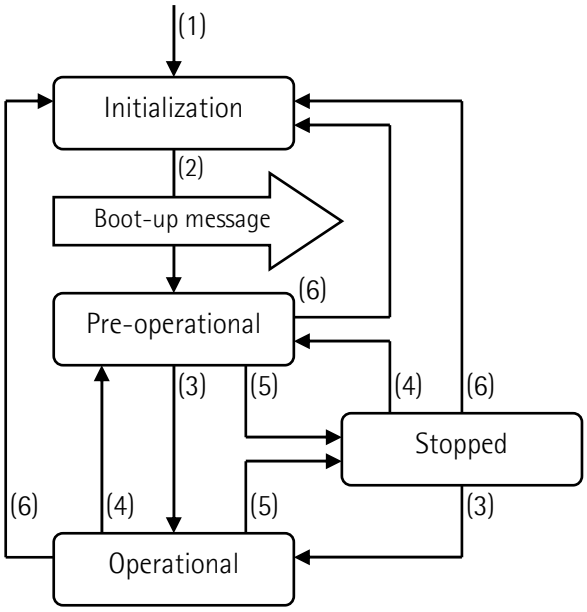


ATTENTION:

If baud rate and node number are set in software mode, at first start up, the master device have to synchronized to slave device 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.

9.2 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."

9.2.1 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.

9.2.2 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.

9.2.3 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.

9.2.4 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.

9.3 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

9.3.1 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.

9.4 NMT objects

NMT structure:

COB-ID (11 bit)	
Func.code	NodeID
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

9.5 Boot-up objects

Boot-up message structure:

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

9.6 PDO objects

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

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

4 CAN Data Bytes			
Byte 0	Byte 1	Byte 2	Byte 3
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 device 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 device will send back their process value. Each device will reply on NODE-ID order. If a device 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.

9.7 SDO objects

SDOs messages are used to know or modified device 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)	
FC	Node-ID

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

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

9.7.1 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

9.8 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

9.8.1 Standard objects (DS 301)

1000-00 Device type [Unsigned32, ro]
Default=00080196h: absolute linear encoder, DS406

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

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" 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 → Slave

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

Slave → 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 → Slave

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

Slave → Master (confirmation)

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

Master → Slave (reset node)

COB.ID	Cmd	Slave ID
000	81	ID

Slave → 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.

9.8.2 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 → Slave

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

Slave → Master (confirmation)

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

Master → Slave (reset node)

COB.ID	Cmd	Slave ID
000	81	ID

Change to new value the master baud rate
Slave → 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 → Slave

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

Slave → Master (confirmation)

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

Master → Slave (reset node)

COB.ID	Cmd	Slave ID
000	81	old ID

Slave → 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.

9.8.3 Device profile objects (DS 406)

6000-00 Operating parameters [Unsigned16, rw]

Bit	Function	bit = 0	bit = 1
0...1	not used		
2	Scaling function	Disable	Enable
3	Measuring direction	Standard	Inverted
4...11	not used		
12	Limit switch min	Disable	Enable
13	Limit switch max	Disable	Enable
14...15	not used		

Default = 0000h

- Measuring direction defines which device direction increase or decrease the position value.
- Scaling function: if disable the device uses the hardware resolution (see object 6501h), if enable it uses the resolution sets in objects 6002 e 6005-01.

6001-00 Total measuring range [Unsig32, rw]
See object 6002h.

6002-00 Total measuring range [Unsigned32, rw]
This object sets the number of distinguishable steps over the total measuring range.
To enable this function set to "1" the bit 2 of object 6000h.

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 device. The position value is set to the desired process value by the parameter preset.
The preset value must not exceed the total measuring range 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.

6005-01 Position step setting [Unsigned32, rw]
This object defines the measuring step setting, it shall be given in 0.001µm. To enable this function set to "1" the bit 2 of object 6000h.
Eg.: 1mm= 1 000 000nm = 000F 4240hex
Attention: object 6005 ≥ object 6501.

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...1	not used		
2	Scaling function	Disable	Enable
3	Measuring direction	Standard	Inverted
4...11	not used		
12	Limit switch min*	posit. > obj_2104	posit. < obj_2104
13	Limit switch max*	posit. < obj_2105	posit. > obj_2105
14	Battery level status	High	Low
15	not used		

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

6501-00 Measuring step [Unsigned32, ro]
This object defines the measuring step that is output by the device. The measuring step is given in nanometer [nm].
To use different value see object 6005-01.

6502-00 Number of revolution [Unsigned16, ro]
Default = 0001h (not used)

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]
Default = 0000 0000h (not used)

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.

9.9 Setup

Below some examples of parameters setting with, in evidence, data exchange between Master and Slave. A generic value "ID" is used to indicate the slave 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 Position step setting

(eg. 1mm = 1000000nm = 000F 4240h)

Master → Slave (Set request)

COB.ID	Cmd	Index		Sub	Process data			
600+ID	23	05	60	01	40	42	0F	00

Slave → Master (Set confirmation)

COB.ID	Cmd	Index		Sub	Process data			
580+ID	60	05	60	01	00	00	00	00

Set Operating parameter

(Scaling function: enable, measuring direction: standard, limit switch: disable)

Master → Slave (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)

Slave → 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 → Slave (Set request)

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

Slave → 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 → Slave (Req request)

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

Slave → 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 → Slave (Set request)

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

Slave → 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 → Slave (Set request)

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

Slave → 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 → Slave (Req request)

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

Slave → 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 &t= 0x7F;

Set new COB-ID used by PDO1:

Master → Slave (Set request)

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

Slave → 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.

9.10 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

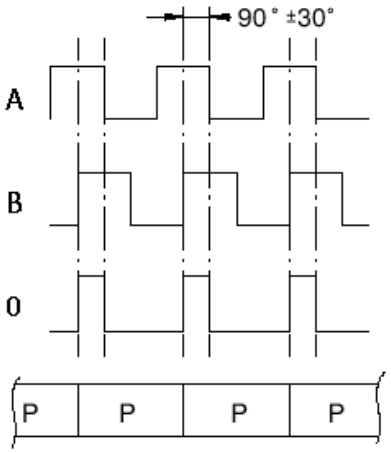
Defined error codes:
1000h = Node guarding error
5530h = Flash memory error

9.11 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.

10 - AB0 interface (incremental output)

Output signals
A, B and 0 signals (standard counting direction):



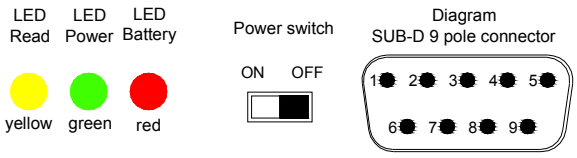
As the sensor is moved along the magnetic tape, it detects the displacement and produces an output signal equivalent to that of an incremental encoder or a linear scale. The signal output is proportional to the measuring speed and to the displacement of the sensor.
Resolution after quadrature (4 edge reading) is 0.1mm.
P = 5mm is the pole pitch of magnetic tape.
Index (0 signal) is every 5 mm.

NOTE:
All sensors have inverted signals:
A = A signal
/A = inverted A signal (or complementary signal).
All sensors have A, /A, B, /B, 0, /0 output signals. We recommend always connecting the inverted signals if the receiving device will accept them. Otherwise each output should be insulated separately.
Involuntary contact between unused signal may cause permanent damage to the sensor.

11 - Specifications

Refer to the catalogue for the remaining technical features of the product.

11.1 Output panel specifications



lika

LIKA Electronic
Via S. Lorenzo, 25 - 36010 Carrè (VI) - Italy

Tel. +39 0445 806600
Fax +39 0445 806699

Italy: eMail info@lika.it - www.lika.it
World: eMail info@lika.biz - www.lika.biz