User manual

Absolute Position Indicator

with CANopen - interface
and Firmware-Softwareversion 01 and 05

AP04

Table of contents

Detailed description of objects 24

1. General Informaton

This user manual is valid for the absolute position indicator AP04 with firmware version H1.02 or higher and is intended to provide the necessary information for handling this device.

The ***Installation Instructions AP04*** contains important information on warranty, safety, mechanical assembly, electrical connection as well as commissioning of the AP04. Please read that information carefully as well.

The present ***User Manual*** will provide a detailed description of the functionality of the CANopen interface.
The communication options via RS485 are described in the ***Supplement to the Installation Instructions AP04 RS485.***

Definitions

This symbol precedes passages in the text that should be read particularly carefully to ensure flawless use of the device and to exclude dangers.

This symbol provides important information for proper handling of the display. Disregard of these hints may result in failures of functioning of the display or its environment.

This symbol indicates instructions for actions.

CAL	CAN Application Layer. Application layer (layer 7) in the CAN communication model
CAN	Controller Area Network
CiA	CAN in Automation. International Association of Users and Producers of CAN products.
COB	Communication Object. Transport unit in the CAN network (CAN message). Data is sent within a COB via the network.
COB-ID	COB-Identifier. Unambiguous identification of a CAN message. The identifier determines the priority of the COB on the network.
ID	Identifier, see COB ID
LSB	Least significant bit/byte
MSB	Most significant bit/byte
NMT	Network Management. Service element of CAL, responsible for initialization, configuration and error handling on the network.
PDO	Process Data Object. Object for exchanging process data.
RTR	Remote Transmission Request; data request telegram
SDO	Service Data Object; communication object that enables the master to access the object directory of a node.

SYNC Synchronization telegram. Bus stations respond to the SYNC command by sending their process values.

Figures if not explicitly stated otherwise, decimal values are given as figures without an extension (e. g., 1234), binary values are marked after the figure with a **b** (e. g. 19011b), hexadecimal values with an **h** (e. g., 280h).

Intended use

The AP04 position indicator is a high-precision measuring instrument. It serves exclusively for the acquisition and output of position values, for processing and providing measured values as electrical output signals for an upstream control as well as for the indication of target values and positioning aids. The AP04 must be used exclusively for these applications.

- Conversion or alteration of the device not approved by SIKO is forbidden for safety reasons.
- Refrain from any operation that may compromise safety with the device.

The AP04 product family

At present, the product family of absolute indicators consists of the following 2 types:

- **AP04 with RS485 interface**
- **AP04 with CAN interface (CANopen)**

Both versions are available in hollow-shaft design and standard dimensions with 20 mm shaft diameter. Although designed with a bus interface, the position indicator is very compact.

The AP04 functions with the following communication protocols:

- **CAN (CANopen interface)**
- **SN3 (RS485 with SIKONETZ3 protocol)**
- **SN4 (RS485 with SIKONETZ4 protocol)**
- **SSP (RS485 with Service Standard protocol)**

2. Brief description

In general

Absolute position indicator with a 20 mm hollow shaft suitable for direct shaft mounting. Set point and actual value are displayed via the 2-line LC display. In case of non-conformance of actual value and set point a direction indication (arrow) is shown. The direction of the arrow indicates into which direction the shaft should be rotated in order to arrive at the set point. The user can set the threshold of deviation where the arrows will be shown. Addionally, various visualization tasks can be realized by means of two coloured LEDs (green and red).

The device parameters can be adjusted by means of 3 keys. The set point can be changed and individual device parameters adjusted via the integrated CAN bus interface.

Scanning is magnetically incremental. In the currentless state, scanning and saving of changes of the position value are battery-supported. The battery can be replaced. The battery symbol on the display will blink when the battery voltage falls below a critical level. If the battery voltage falls below a minimum level, the symbol will glow permanently.
Display and interface are active with external power supply only.

Display

2 lines of each 5 7-segment readings

4 special characters: "<", ">", incremental measure symbol ⟼⟼; Battery symbol ▭

Displayable number range:
-99999 to 99999

If this number range is exceeded, the "FULL" message will be displayed. However, the value is available for transmission via interface.

If the value to be displayed falls below -19999, the negative arithmetic sign will not be indicated permanently. Indication of the highest-value figure will alternate cyclically with the arithmetic sign.

Keyboard

The AP04 has ✱ -, ⬆ - and ⬅ keys serving for adjustment of device parameters.

⬆ - Key

By pressing the ⬆ key, the incremental measurement function is switched on or off. During this action, the incremental measurement symbol ⟼⟼ is shown or hidden on the display. For this purpose, the incremental measurement function must be enabled.

During configuration, the current value can be changed by means of the ⬆ key.

✱ - Key

If the ✱ key is pressed for more than 5 s, then the current position value is zeroed after releasing the key. For doing this, zeroing must be enabled.

Position value = 0 + calibration value + offset value.

"rESEt" blinks in the lower line of the display during the 5 s delay period. After expiry of the delay period and as long as the button is still being pressed, "rESEt" will be displayed permanently until the button is released.

During configuration the ✱ key serves for acknowledging the current value and switching over to the next parameter.

⬅ - Key

By pressing the ⬅ key, the set bus address ("1" in the example) and baud rate (250 kbit/s) will be displayed.

E. g.:

Id 1
250

When actived during more than 15 s, AP04 will switch to configuration mode.

Display will then show the first configuration menu point.

Battery buffering

The battery makes possible the detection of currentless displacement. Battery life is approx. 5 to 8 years depending on the duration of battery operation (including storage) and frequency of currentless adjustments. Battery voltage is checked at intervals of approx. 5 min. If battery voltage drops below a specified value, the battery symbol will blink on the display. The battery is nearly empty. If the battery voltage continues to drop, will be displayed permanently. The battery should be replaced as soon as the battery symbol appears on the display. The battery can be replaced by the SIKO distribution partners or at the SIKO main factory. For battery replacement it is mandatory to follow the instructions of the ***Installation Instructions***.

3. Start-up

Prior to starting up the display the following work should be performed:

- Correct assembly (see AP04 Installation instructions)
- Correct connection of the supply and bus lines (see AP04 Installation instructions)
- Setting of the Node ID (must be present in the system only once)
- Setting of the CAN baud rate valid for the system
- Setting of additional configuration parameters if necessary

Switching on the supply voltage

The AP04 will be initialized after switching on the supply voltage.

During initialization, a display and LED test is carried out and the configuration parameters are loaded from the non-volatile memory to the main memory of the controller.

With the display still unconfigured all parameters are set to their default values. See to it that the bus will be connected only after correct adjustment of baud rate and ID. The AP04 functions with the data last parameterized.

After completing the initialization procedure, the AP04 with CAN interface sends a specific NMT command, the **Boot-Up Message,** which informs the system about the availability of the display. The AP04 is now in the **Pre-Operational** Mode. In this state, the display can be parameterized via SDO commands in accordance with the requirements of the application. This applies to configuration parameters as well as to the way it makes available to the system its position values (asynchronous or synchronous data transmission).

If no boot-up message can be sent because the baud rate was set wrongly, the AP04 will be reinitialized completely (warm start) and will try again to send the message. This becomes visible by a repeated display test. If no bus is connected, the AP04 will also try permanently to send the boot-up message but will only be reinitialized after a defined number of failed attempts.

Display

In the first line, the current position value is always shown.

The second line indicates the operational mode (see also section "Network Management Services (NMT)". In the operational mode the valid target value is displayed. If no valid target value has been received, the display will indicate "- - - - -" (see section "Transmission of process data"). The display of the target value can be deactivated via object 5F0B.

Configuration

Parameter list

For a detailed description of the parameters refer to section "Detailed description of objects". The following parameters are relevant for the display and measurement function of the AP04.

Parameter	Value range	Default	Meaning / Note	Object
Id	1 … 127	1	bus address	5F0Ah
baud	125, 250, 500, 1000	250	baud rate in kbit/s	-
OFFST	-9999 … +9999	0	offfset value	2001h
F0SET	0, 1	1	zeroing enable	2003h
FKETT	0, 1	1	incremental measurement enable	2004h
ExHbtimer	0 … 65535	300	expected heartbeat cycle time	5F09h
Hidetarget	0, 1	0	hiding of target value	5F0Bh
InPos	-9999 … +9999	5	deviation window from target to actual value	5F10h
DEZ	0, 0.0, 0.00, 0.000, 0.0000	0.0	display of decimals	5F11h
DISPLDIR	0, 180	0	display orientation	5F12h
GrEEn	0, 1	0	green LED is lighted when target window reached	5F12h
rEd	0, 1	0	red LED is lighted when position outside the target window	5F12h
FLASh	0, 1	0	LED blinks when switched on	5F12h
DIV	1, 10, 100, 1000	1	display divisor	5F13h
Loop	-9999 … +9999	0	loop reversal point (in display unit)	5F14h
LPDIR	DIR, +, -,	DIR	positioning direction for loop	5F15h
ExHbSource	0, 1	1	trigger source of ext. heartbeat	5F18h
LPHYS	0 … 99999	0	hysteresis for detection of reversal of sense of rotation	5F1Ah
DIR	I, E	E	sense of rotation cw or ccw	6000h
APU	0 … 59999	720	indication per revolution	6001h
CAL	-9999 … +9999	0	calibration value	6003h
Code	0 … 99999 00100 11100	0	for SIKO-internal test purposes / diagnosis trimming travel load factory settings	- - 1011h

Configuration via keyboard

In the configuration mode, Id and baud rate can be set via keyboard. The functions of the "Code" parameter can be called up as well. All other parameters are configured via interface.

To this purpose the parameter is indicated in the 1st line of the display and the associated value in the 2nd line.

Using the ⇧ key, the current value (e. g. "baud" "250" -> "500") can be changed, in case of multi-digit numbers ("Code") at the blinking position.

The ← key serves for advancing to the next digit in case of multi-digit numbers.

By pressing the * key, the set value will be acknowledged and saved non-volatilely. If no key is pressed, the configuration mode will be exited after approx. 30 s without saving the value last displayed, i. e., the original value remains unchanged.

Configuration via CAN interface

Except for the baud rate all parameters can be configured via the CAN interface. For a detailed description of all parameter objects refer to section "6 Directory of objects".

Sending the position value

Before the display can send its position value, the AP04 must be switched to the **Operational Mode** via the **Node Start** NMT command.

COB-ID	Command byte	Node number
0h	1h	0h ... 1Fh (0 ... 31)

If the Node ID of the display is indicated as the node number, then only this display will start. If the value 0 is transmitted for the node number, then all devices connected to the bus will start.

Now the display can transmit its position value as specified via PDO1 or PDO2, respectively.

Synchronous transmission

The device supports this type of transmission with its factory setting.

If the AP04 receives a SYNC telegram in the operational mode, the display will respond with the position value plus the status byte. For a more detailed description of transmission refer to section "Transmission of process data".

To make possible synchronous transmission of the position value, the PDO2 must be enabled (bit31 of the COB ID PDO2 to 0 = default). Furthermore, a value between 1 and 240 (= F0h) must be written in object 1801h, sub-index 2 (1 = default).

Another possibility of transmitting the PDO2 consists in the response to a RTR. To this purpose, the value 253 (=FDh) must be written to object 1801h, sub-index 2.

Asynchronous (cyclic) transmission

PDO1 is responsible for this type of transmission. The position value (plus status byte) is cyclically sent in accordance with the time parameterized in object 1800h, sub-index 5.

Stopping Transmission of the Position Value

General:

To stop data transmission from the display, the display can be switched back to the **Stopped Mode** or to the **Pre-Operational Mode**.

Stop Mode command

COB-ID	Command byte	Node number
0h	2h	0h ... 1Fh (0 ... 31)

Pre-Operational Mode command:

COB-ID	Command byte	Node number
0h	80h	0h ... 1Fh (0 ... 31)

All devices connected to the bus are addressed via node address 0.

Synchronous transmission:

Naturally, there is no synchronous transmission if no SYNC telegram is received or no RTR is obtained.

Asynchronous transmission:

Asynchronous, i. e. timer-controlled transmission is suppressed if the timer value is set to 0 (see object 1800h).

4. General information on the CAN bus

Originally, the CAN bus (CAN: Controller Area Network) was developed by Bosch and Intel for fast and low-cost data transmission in the car industry. Today it is also used in industrial automation. The CAN bus is a field bus, which enables communication of devices, actuators and sensors of different manufacturers. The standards are defined by the Association CAN in Automation (CiA).

CAN bus features

- Bus medium is a shielded twisted pair cable.
- The CAN bus is a multi-master bus, i. e., several CAN stations can request the bus at the same time. The message with the highest priority (determined by the identifier) prevails.
- Data rate up to 1bit/s permissible (with 40 m network range).
- Closed network on both sides.
- Theoretically, up to 127 stations possible on one bus; however, practically only up to 32 stations due to the driver.
- Message-oriented communication: The message is marked with a message identification (identifier). By means of the identifier, all bus stations check whether the message is relevant for each of them.
- All bus station receive each message at the same time. Therefore, synchronization is possible.
- The identifier determines the priority of the message. The lower the value of the identifier, the higher is the priority of the message. This enables fast transmission of important messages via the bus.
- High transmission safety thanks to various error identification mechanisms, which complement each other.

- Localization of faulty or disabled bus stations. The CAN protocol includes function monitoring of bus stations. The functionality of the latter will be limited or disconnected from the network if they are faulty.

CANopen

The CANopen profile was developed on the basis of the layer 7 specification CAL (CAN Application Layer) under the direction of the Steinbeis Transferzentrum für Automatisierung (Transfer centre for automation). Compared to CAL, only the functions appropriate for this use are included in CANopen. Thus, CANopen is a subset of CAL optimized for the application and enables a simplified system design as well as the use of simplified devices. CANopen has been optimized for fast data exchange in real-time systems.

The organization CAN in Automation (CiA) is responsible for the applicable standards of the respective profiles

The position indicator AP04 with CANopen interface fulfils the conditions specified in the "CANopen Application Layer and Communication Profile" (CiA Draft Standard 301, version 4.02) and in the "CANopen Device profile for encoders" (CiA Draft Standard 406, version 3.1) (CAN 2.0A).

CANopen enables:

- easy access to all device and communication parameters,
- synchronisation of several devices,
- automatic configuration of networks
- cyclic and event-triggered data traffic

CANopen consists of four communication objects (COB) with different features:

- Process Data Objects (PDOs) for real-time data.
- Service Data Objects (SDOs) for parameter and program transfer,
- Network Management (NMT),
- Predefined objects (for synchronization, emergency message)

The description of the device functionality via an object directory is the central element of the CANopen standard. The object directory is subdivided into an area containing general information on the device (device identification, manufacturer's name, etc.) and communication parameters, and an area describing the specific device functionality.

An entry ("object") of the object directory is identified via a 16bit index and an 8b bit sub-index. By means of these entries, the "application objects" of a device (e. g. position value with encoders) are made accessible in a standardized form via the network.

The functionality and features of a CANopen device can be described in the ASCII format as a standardized "Electronic Data Sheet" (**EDS**).

The EDS file (CANopen configuration file) of the AP04 can be downloaded from the homepage of SIKO GmbH (www.siko-global.com/p/ap04).

The encoder device profile (CiA Draft Standard 406)

This profile describes a manufacturer-independent and binding specification of the interface for rotary encoders. The profile defines which CANopen functions are used and how they are to be used. This standard enables the creation of an open and manufacturer-independent bus system.

The device profile is divided into two object classes:

- Standard class C1 describes all basic functions, which the encoder must contain,
- The extended class C2 contains a wide range of additional functions that must either be supported by these encoders (mandatory) or are optional. Thus, devices of the C2 class contain all the C1 and C2 mandatory functions as well as – manufacturer-dependent – additional, optional functions.

Additionally, an addressing range is defined in the profile for assignment of special proprietary functions.

The AP04 supports class C2.

5. Data transfer according to the CANopen communication model

The communication model underlying CANopen provides two types of communication mechanisms:

- Unconfirmed transfer of data having a length of max. 8 bytes **(Process Data Objects, PDOs).** These data is transferred with high priority (low COB identifier). PDOs are broadcast messages and provide their data to all addressees on the bus at the same time.
- Confirmed transfer also of longer data sets (parameters) between two stations with direct access to the entries of the addressee's object directory **(Service Data Objects, SDOs).** As a rule, these parameters are transferred acyclically (e. g. only once when the system is started) and have, therefore, low priority (= high COB identifier).

The priority of the message objects is determined via the COB identifier.

CANopen message structure

For easier management of the identifiers, CANopen uses the "Pre-Defined Connection Set". Here, all identifiers are defined in the object directory with standard values. However, the customer has the possibility of changing these identifiers via SDO access to meet his requirements. However, only 11 bit identifiers are supported (CAN 2.0A).

The 11bit identifier (COB identifier) consists of a 4bit function code and a 7bit node number.

Bit no.	10	9	8	7	6	5	4	3	2	1	0
Type	Function code				Node number (node ID)						
Assignment	x	x	x	x	0	0	x	x	x	x	x

Hint: Thus, a maximum of 127 different node numbers can be set (node number 0 is illegal!).

Hint: Only a maximum of 31 bus stations is permitted!

The function code informs about the type of message and its priority. The higher the value of the identifier, the lower the priority.

Function code

The following function codes have been defined in the "Pre-defined Connection Set" (only the function codes used by the AP04 are represented):

Object	Function code	Resulting COB ID	assigned communication parameter for index
NMT	0000b	0	-
SYNC	0001b	128 (80h)	1005h
EMERGENCY	0001b	128 (80h) + Node-ID	1014h
PDO1 (tx)[1]	0011b	384 (180h) + Node-ID	1800h
PDO1 (rx)[1]	0100b	512 (200h) + Node-ID	1400h
PDO2 (tx)[1]	0101b	640 (280h) + Node-ID	1801h
PDO2 (rx)[1]	0110b	768 (300h) + Node-ID	1401h
SDO (tx)[1]	1011b	1408 (580h) + Node-ID	1200h
SDO (rx)[1]	1100b	1536 (600h) + Node-ID	1200h
HEARTBEAT	1110b	1792 (700h) + Node-ID	1017h

[1] (tx) and (rx) seen from the position indicator

Node number (node ID)

The 7bit node number is set on the AP04 via configuration and displayed upon pressing the ← key during operation.

Node number 0 is reserved and must not be changed by any node. Therefore, resulting node numbers are in the range of 1 … 127. Any freshly set node number is only taken over after the next reset/power-on of the encoder.

Ex works, the position indicator is delivered with node number 1.

Transmission of process data

The four PDO services, PDO1 (tx), PDO1 (rx), PDO2 (tx) and PDO2 (rx), are available.

From the AP04 to the bus master (position value + status byte)

PDO transfer from the display to the bus master (TPDO) can be initiated as a result of various events:

- asynchronous, controlled by an internal device timer
- synchronous as a response to a SYNC telegram
- as a response to a RTR telegram

Both PDOs provide the current position as well as a status byte of the display and are determined via objects 1800h, 1801h, 1A00h, 1A01h, 2800h, 2801h and 6200h.

With the AP04, TPDO1 is assigned to asynchronous and TPDO2 to synchronous process data transfer. As a standard, TPDO2 is enabled after each power-on of the encoder and must be disabled on request via SDO.

Request of the position value via RTR telegram is also only possible via TPDO2.

The TPDO message has the following structure:

COB-ID	Process data in binary code				
11bits	Byte 0 (LSB)	Byte 1	Byte 2	Byte 3	Byte 4 (MSB)
TPDO1: 180h+Node-ID TPDO2: 280h+Node-ID	Position value in two's complement representation see Object 6004h				Status byte see object 5F19h

The bits of the status byte have the following meaning:

Bit	7	6	5	4	3	2	1	0
0	-	battery not flat	"<" off	">" off	Increm. Meas. = 0	Batt. ok	ACT < TARG	Not IN-POS
1		flat battery	"<" on	">" on	Increm. Meas. = set	Batt.warn	ACT > TARG	IN-POS

Synchronous data transfer (factory setting)

The AP04 is delivered with this type of transmission preset and the AP04 responds to a SYNC telegram received by sending the TPDO message.
In the softwareversion 01 and 05 the object 5F09h (ext. heartbeat timer) is additionally active. This means that with factory setting the AP04 expects a SYNC telegram at an interval of max. 300 ms. Otherwise the operational mode will be exited.
The validity of the transmitted target value is acknowledged with the additional control byte. The data received is interpreted as a valid target value only if the control byte has a value > 0.

To be able to send process data synchronously, a value between 1 and 240 (= F0h) must be written in object 1801h, sub-index 2.

In synchronous operation, the PDO2 is requested by a master via the SYNC telegram (SYNC-COB ID = 80h).

If the PDO2 is to be requested via an RTR telegram, then the value 253 (= FDh) must be written in object 1801h, sub-index 2.

Asynchronous data transfer

If a PDO is to be sent cyclically, then the cycle time must be entered into object 1800h, sub-index 5, in milliseconds. The PDO1 will not be sent if the value 0ms is written. The function is disabled. The minimum value to be set is 1 (= 1 ms).

From bus master to AP04 (target value)

By means of PDO transmission from the bus master to the AP04 (RPDO), the target value can be transferred to the display. The following PDOs are accepted.

COB-ID	Process data in binary code				
11bits	Byte 0 (LSB)	Byte 1	Byte 2	Byte 3	Byte 4 (MSB)
RPDO1: 200h+Node-ID RPDO2: 300h+Node-ID	Position value in two's complement representation see Object 6004h				Control byte see object 5F0Ch

Transfer of the SDO data (parameterization)

The object directory of the position indicator can be accessed via an SDO message. All device parameters are stored in this object directory under standardized addresses (indexes) and can be written to and read by means of SDOs. SDOs are exchanged between two stations using the request/response method.

Two SDO services are available:

- SDO (tx) (AP04 ➡ master): **580h** + Node ID
- SDO (rx) (master ➡ AP04): **600h** + Node ID

The SDO identifiers cannot be changed!

SDO messages are set up as follows:

COB ID	Command	Index		Sub-index	Service data (parameters)			
SDO + Node-ID	Byte 0 (read / write)	Byte 1 (LSB)	Byte 2 (MSB)	Byte 3	Byte 4 (LSB)	Byte 5	Byte 6	Byte 7 (MSB)

For the meaning of index, sub-index and data please refer to chapter, "6 Directory of objects".

The command byte specifies the length of the service data (parameters). In the case of the AP04 the following command bytes are valid:

Command byte	Type	Function
23h	SDO (rx), Initiate Download Request	Send parameter to AP04 (data length = 4bytes)
60h	SDO (tx), Initiate Download Response	Acknowledgement of data acquisition to master
40h	SDO (rx), Initiate Upload Request	Request parameter from AP04
42h	SDO (tx), Initiate Upload Response	Parameter to master (data length = 4bytes)
80h	SDO (tx), Abort Domain Transfer	AP04 reports error code to master

- An error message (command 80h) replaces the normal response in case of a fault.
- The error message includes communication protocol errors as well as object directory access errors (e. g. write attempt on read-only object, wrong index, etc.).

The error codes are described in the CANopen profile (DS 301) or in the encoder profile (DSP 406), respectively. The table below shows the error codes used in the AP04:

Error code	Description
06010000h	Wrong access to an object.
06010001h	Read access to Write-Only
06010002h	Write access to Read-Only.
06020000h	Object doesn't exist in the object directory.
06070010h	Wrong data type, incorrect data length.
06090011h	Sub-index does not exist
06090030h	Wrong value range of selected parameter.
06090036h	Maximum value smaller than minimum value.
08000020h	Parameters cannot be transferred to application or stored.
08000022h	Parameters cannot be transferred to application or stored due to the current device status.

SDO examples:

Request of value by a master with a slave ➡ Operating Status (object 6500h):

COB-ID	Command	Index L	Index H	Sub-index	Data 0	Data 1	Data 2	Data 3
600h + Node ID	40h	00h	65h	00h	x	x	x	x

Response to the request by the slave

COB-ID	Command	Index L	Index H	Sub-index	Data 0	Data 1	Data 2	Data 3
580h + Node ID	42h	00h	65h	00h	a	b	c	d

Writing a value from master to a slave ➡ object 1800, sub-index 5 (Event Timer):

COB-ID	Command	Index L	Index H	Sub-index	Data 0	Data 1	Data 2	Data 3
600h + Node ID	23h	00h	18h	05h	E8h	03h	00h	00h

Response from slave to writing the value:

COB-ID	Command	Index L	Index H	Sub-index	Data 0	Data 1	Data 2	Data 3
600h + Node ID	60h	00h	18h	05h	00h	00h	00h	00h

Emergency Service

Internal device errors or bus problems trigger an emergency message. The corresponding telegram is set up as follows:

COB ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80h + Node ID	Error Code		Error Register	Alarms (object 6503h)		Warnings (object 6505h		00h

If value "11h" is in the error register, the meaning of bytes 3 – 6 in the emergency telegram will be changed. The value "11h" indicates errors that have occurred during transfer of data on the CAN bus (see description „Error codes"). The encoder has changed to the "Error Passive" state.

With diminished interference on the CAN bus the encoder returns automatically to the normal state designated "Error active". Otherwise, if interference continues to increase, the encoder will change to the "bus off" state with subsequent restart characterized by a "boot-up message" and an additional "emergency message" (byte3 = TEC, byte4 = REC). TEC and REC are Transmit or Receive Error Counters. The bus status of the encoder depends on their statuses.

Emergency message in the case of bus errors:

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80h + Node ID	Error Code		Error Register	Transmit Error Counter	Receive Error Counter	00h	00h	00h

As with the SDO error messages, pre-defined error messages are assigned to the EMERGENCY object as well. A subset of these error codes described in the CAN Application Layer DS301 is used by the AP04. They are described in the table below:

Byte 0 ... Byte 1: Error Code

Error Codes	Description
0000h	no error
8120h	encoder is in the Error Passive Mode
8140h	recovered from Bus Off

Byte 2: Error Register

Bit no.	Description
0	set bit indicates general error condition; bit is set with every error occurring.
4	communication error; is set when a CANbus communication error occurs (acknowledgement, form, C, and stuff error).
8	Manufacturer-specific instrumental error (battery dead)

Byte 3 ... Byte 4: Alarms

Bit no.	Description
0	Position value invalid if bit set (= 1)
14	Battery warning (critical charge condition)
15	Battery alarm (battery dead)

Byte 5 ... Byte 6: Warnings

Bit no.	Description
4	battery status critical

Byte 7: not used

Network Management Services (NMT)

The network management can be subdivided into two groups:

- NMT service for device control; serves for initializing, starting and stopping of the encoder,
- NMT service connection monitoring ("heartbeat").

Description of the NMT commands

The commands are transferred as unconfirmed objects (broadcast messages) and are set up as follows:

COB-ID	Byte 1	Byte 2
0h	Command byte	Node number (node ID)

The COB ID for NMT commands is always zero (highest priority). The node ID is transferred in byte 2 of the NMT command.

The node number corresponds with the node ID of the desired station. With node number = 0, all bus stations are addressed.

Command byte

Command byte	Description	State transition (see State diagram, fig. 1)
01h	Start_Remote_Node; change from state "Pre-Operational" or "Stopped" to "Operational"	1
02h	Stop_Remote_Node; change to state "Stopped"	2
80h	Enter_PRE-OPERATIONAL_State; change to state "Pre-Operational"	3
81h	Re-initialization of CAN connection	4
82h	Reset AP04 (warm start)	5

NMT status

After initializing, the encoder is in the "Pre-Operational" state. SDO parameters can be read and written in this state. To request PDOs, the encoder must first be switched to the "Operational" state.

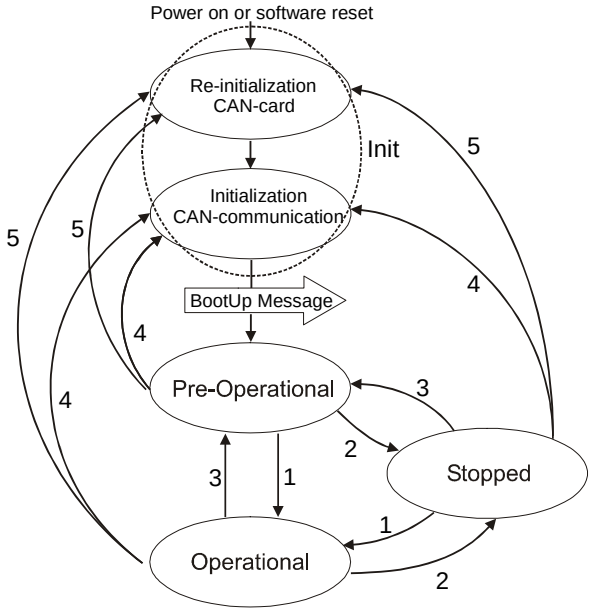


Fig. 1: CAN status diagram

The individual NMT states

Init:

After initialization, the encoder logs in at the CAN bus with a boot-up message. Afterwards, the encoder changes automatically to the "Pre-operational" state.

The COB ID of the boot-up message is made up of 700h and the node ID.

COB-ID	Byte 0
700h + Node ID	00h

Pre-Operational Mode:

SDOs can be read and written in the Pre-Operational Mode.

Operational Mode:

In the operational mode, the encoder sends the PDOs requested. Additionally, SDOs can be read and written.

Stopped Mode:

Only NMT communication is possible in the Stopped Mode. No SDO parameters can be read or written.

State change

The following applies to all commands listed below: If node number 0h is sent, the command will apply to all nodes connected.

Start Remote Node (1)

With the "Start_Remote_Node" command, the encoder is set to the "Operational Mode" state.

COB-ID	Command byte	Node number
0h	1h	0h ... 1Fh (0 ... 31)

Stop Remote Node (2)

With the "Stop_Remote_Node" command, the encoder is set to the "Stopped" state.

COB-ID	Command byte	Node number
0h	2h	0h ... 1Fh (0 ... 31)

Enter_PRE-OPERATIONAL-Mode (3)

Change to the "Pre-Operational" state.

COB-ID	Command byte	Node number
0h	80h	0h ... 1Fh (0 ... 31)

Re-initialization of CAN parameters (4)

COB-ID	Command byte	Node number
0h	81h	0h ... 1Fh (0 ... 31)

Re-initialization of the CAN card (5)

COB-ID	Command byte	Node number
0h	82h	0h ... 1Fh (0 ... 31)

Heartbeat

Two optional monitoring mechanisms are intended for ensuring proper functioning of the CANopen network nodes: Each network node can be monitored by a higher-order master via the so-called "Node Guard" or, alternatively, announce its ability to communicate by cyclic sending of a so-called "heartbeat" message.

The "heartbeat" method is intended for the AP04.

One or several network subscribers can receive this message and, thus, monitor the assigned subscriber.

In object 1017h, "Producer Heartbeat Time", the time of the heartbeat interval can be deposited. The value 0 disables heartbeat.

The heartbeat message consists of the COB ID and an additional byte. In this byte, the current NMT state is deposited.

COB-ID	Byte 0
700h + Node ID	NMT state

NMT states:

- 0: Boot-Up
- 4: Stopped
- 5: Operational
- 127: Pre-Operational

External heartbeat

In addition to the function described above, the NMT state of the AP04 can be controlled via external heartbeat. To this purpose a value has been entered in object 5F09. This value corresponds to an interval in ms. If no external heartbeat is received by the AP04 within this interval, the display will change to the pre-operational state. In object 5F18 it is stored which telegram is to be interpreted as an external heartbeat.

6. Directory of objects

In the object directory of a CANopen device, all features and parameters of this device are deposited.

Specific parameters of the objects directory are deposited in a power-failure-safe memory of the encoder and are copied into the main memory during power-on or re-initialization.

Access to the directory of objects is via the SDO services described in section "Transfer of the SDO data (parameterization)".

The object directory is subdivided into three separate areas:

- standard objects applicable to all CANopen instruments, 1h ... 1FFFh, (CiA DS 301)
- manufacturer-specific objects, 2000h ... 5FFFh
- device-specific objects, 6000h ... BFFFh, (CiA DS 406)

The address (index) pointing to each entry in the object directory is also standardized in the profiles except for the manufacturer-specific area. This fact ensures that all instruments always provide the functions described in the profile (standard and optional functions) under the same index. This is a precondition of an open system and of exchangeability of the instruments.

The entries of the object directory are addressed by a 16-bit index. Each index can be further subdivided by a sub-index.

Overview of objects

Index	Name	Description	*see page*
1000h	Device Type	indicates the device profile and the encoder type	*24*
1001h	Error Register	indicates error states of the encoder	*25*
1002h	Manufacturer Status Register	indicates the contents of the CAN bus-specific "TransmitErrorCounter" or "ReceiveErrorCounter", respectively	*25*
1003h	Pre-Defined Error Field	the object stores the 8 error states that have occurred last	*25*
1005h	COB ID SYNC message	Setting of the COB ID of the SYNC object.	*26*
1008h	Manufacturer Device Name	Short designation of the device type	*26*
1009h	Manufacturer Hardware Version	hardware version of the encoder	*27*
100Ah	Manufacturer Software Version	software version of the encoder	*27*
1010h	Store Parameters	the object indicates non-volatile storage of parameters by the encoder with no user input.	*27*
1011h	Restore Parameters	the object indicates that the encoder automatically loads parameters from the non-volatile memory.	*29*
1014h	COB-ID Emergency Object	COB ID of the Emergency object	*29*
1017h	Producer Heartbeat Time	setting of the cycle time of the heartbeat timer	*30*
1018h	Identity Object	contains the manufacturer number and device variant assigned by CiA	*30*
1200h	Server SDO Parameter	SDO parameter	*31*
1400h	Receive PDO1 Communication Parameter	Receive PDO for asynchronous operating mode	*31*
1401h	Receive PDO2 Communication Parameter	Receive PDO for synchronous operating mode	*32*
1600h	Receive PDO1 Mapping Parameter		*32*
1601h	Receive PDO2 Mapping Parameter		*33*
1800h	Transmit PDO1 Communication Parameter	Transmit PDO for the asynchronous operation mode (timer-controlled)	*34*

Index	Name	Description	see page
1801h	Transmit PDO2 Communication Parameter	Transmit PDO for the synchronous operation mode, including output of the position value via RTR	35
1A00h	Transmit PDO1 Mapping Parameter		35
1A01h	Transmit PDO2 Mapping Parameter		36
2001h	Manufacturer Offset	manufacturer-specific offset value (is added to the position value encoder-internally)	37
2002h	Zero encoder	set position value to value 0 (condition: pre-set value 0)	37
2003h	Enable for zeroing	indicates whether zeroing via key actuation is enabled	38
2004h	Enable for incremental measurement	indicates whether setting the position value as an incremental measurement via key actuation is enabled	38
5F09h	External heartbeat timer	expected heartbeat cycle time	38
5F0Ah	Node ID	Node ID, ! Change only active after re-initialization	39
5F0Ch	Control byte	Controls the validity of the target value and the LEDs	39
5F10h	Target window	max. deviation from target value, if actual value within the window: target value achieved	39
5F11h	Decimal places	Number of decimal places	40
5F12h	Display orientation and LED	0° or 180° Function of the LEDs	40
5F13h	Display divisor		41
5F14h	Loop width	Width of the loop	41
5F15h	Loop direction	Direction from which travel to target value must be started	41
5F16h	Read target value	Read current target value; write access only via PDO	42
5F18h	Trigger source of ext. heartbeat	Signal which triggers external heartbeat	43
5F19h	Device status	is attached to position value in PDO	43
5F1Ah	Hysteresis reversal of sense of rotation	Parameter for loop positioning	43
6000h	Operating Parameters	setting of sense of rotation and scaling function	44
6001h	Measuring units per Revolution	parameterization of the resolution in steps / revolution of the encoder	44
6002h	Total measuring range in measuring units	parameterization of the total measuring range of the encoder	45
6003h	Preset Value	parameterization of a pre-set (calibration) value	45
6004h	Position Value	position value (offset with pre-set and manufacturer offset value)	45

Index	Name	Description	see page
6200h	Cycle Timer PDO1	value in ms, identical with object 1800h, sub-index 5	*46*
6500h	Operating Status	indicates the sense of rotation and scaling function currently set	*46*
6501h	Resolution	indicates the maximum possible resolution in steps / revolution	*46*
6502h	Number of distin-guishable Revolutions	indicates the maximum possible number of revolu-tions	*47*
6503h	Alarms	indication of error states	*47*
6504h	Supported Alarms	indicates which alarm messages are supported	*47*
6505h	Warnings	indication of warnings	*48*
6506h	Supported Warnings	indicates which warnings are supported	*48*
6507h	Profile and Software Version	indicates the version number of the device profile used and the version number of the encoder's firmware	*48*
6508h	Operating Time	outputs the value FFFFFFFFh (function is not sup-ported)	*49*
6509h	Offset Value	corresponds with the encoder's zero point value	*49*
650Ah	Module Identification	device-specific parameters (Manufacturer offset, Manufacturer min position value, Manufacturer max position value) can be represented via sub-indexes	*49*
650Bh	Serial Number	outputs the value FFFFFFFFh (function is not sup-ported)	*50*

Detailed description of objects

Object 1000h (Device Type)

Sub-index	**00h**			
Description	Information on device type and device profile			
Access	ro			
Data type	UNSIGNED 32			
EEPROM	no			
Default	00030196h			
Data content	Device profile number		Encoder type	
	Byte 0	Byte 1	Byte 2	Byte 3
	96h	01h	03h	00h

0196h (= 406): CANopen Device Profile for Encoders, version 3.01

0003h: Single-turn angle encoder, absolute, with battery-buffered electronic revolu-tion counter

Object 1001h (Error Register)

Sub-index	**00h**	
Description	Device errors occurring are indicated here	
Access	ro	
Data type	UNSIGNED 8	
EEPROM	no	
Default	no	
Data content	Bit	Meaning
	0	set bit indicates the occurrence of any error condition
	4	set bit indicates communication error on the CAN bus (Acknowl-edgement-, Form-, CRC- and Stuffbit)
	1-3, 5-7	not used

Object 1002h (Manufacturer Status Register)

Sub-index	**00h**			
Description	The counts of the registers "Transmit Error Counter" and "Receive Error Counter" can be read via this object. The contents of these registers provide information on the transmit faults present at the mounting site of the encoder.			
Access	ro			
Data type	UNSIGNED 32			
EEPROM	no			
Default	0			
Data content	Byte 0	Byte 1	Byte 2	Byte 3
	Receive Error Counter	Transmit Error Counter		

For details on the above-mentioned counters refer to the relevant CAN bus publications.

Object 1003h (Pre-defined Error Field)

- the object stores the 8 error states that have occurred last.
- the entry under sub-index 0 indicates the number of errors saved.
- each newly error state added is stored under sub-index 1. Previous error messages "slip" in their position by one digit.
- the whole error list is deleted by writing the value 0 at sub-index 0.
- The entries in the error list have the format described in "Emergency Service".

Sub-index	**00h**
Description	number of the error messages stored
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 8
EEPROM	no
Default	0
Value range	0 – 8

Sub-index	**01h .. 08h**
Description	error messages that occurred
Access	ro
Data type	UNSIGNED 32
EEPROM	no
Default	0

Object 1005h (COB ID SYNC message)

Sub-index	**00h**	
Description	Defines the COB ID of the synchronization object (SYNC)	
Access	rw (writable in the "Pre-Operational" state only)	
Data type	UNSIGNED 32	
EEPROM	yes	
Default	80h	
Data content	Bit 31:	not defined
	Bit 30	0: encoder generates no SYNC message 1: encodes generates SYNC messages
	Bit 29	0: 11bits identifier (CAN 2.0A) 1: 29bits identifier (CAN 2.0B) ! not supported !
	Bit 28 ... 11	0: if bit 29 = 0
	Bit 10 ... 0	X: bits 10 – 0 of the SYNC-COB-ID

Object 1008h (Manufacturer Device Name)

Sub-index	**00h**			
Description	short encoder designation in ASCII			
Access	const			
Data type	Visible_String			
EEPROM	no			
Default	AP04			
Data content	Byte 0	Byte 1	Byte 2	Byte 3
	41h ('A')	50h ('P')	30h ('0')	34h ('4')

Object 1009h (Manufacturer Hardware Version)

Sub-index	**00h**			
Description	hardware version in ASCII			
Access	const			
Data type	Visible_String			
EEPROM	no			
Default	"V100"			
Data content	Byte 0	Byte 1	Byte 2	Byte 3
	56h ('V')	31h ('1')	30h ('0')	30h ('0')

Object 100Ah (Manufacturer Software Version)

Sub-index	**00h**			
Description	hardware version in ASCII			
Access	const			
Data type	Visible_String			
EEPROM	no			
Default	"H1.00"			
Data content	Byte 0	Byte 1	Byte 2	Byte 3
	48h ('H')	31h ('1')	30h ('0')	30h ('0')

Object 1010h (Store Parameters)

This object serves only for information that the encoder automatically stores specific parameters in the EEPROM. The "Store Parameter" command is not required for parameter storage!

Sub-index	**00h**
Description	describes the number of entries present in sub-index 1.
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	1h

Sub-index	**01h**
Description	describes the behavior of the encoder, how parameters are stored in the EEPROM.
Access	ro
Data type	UNSIGNED 32
EEPROM	no
Default	2h

Data content	Bit 31-2	0
	Bit 1:	0: encoder does not store parameters automatically 1: encoder stores parameters automatically following write access to relevant object
	Bit 0:	0: encoder does not store parameter by command 1: encoder stores parameter by command

The following table represents the parameters, which are stored non-volatilely in the EEPROM:

Object	Sub-index	Description	Default value
1005h	0h	SYNC ID	80h
1014h	0h	EMCY ID	80h + Node ID
1017h	0h	Producer Heartbeat Time	0h
1400h	1h	RPDO1 ID	40000200h + Node ID
1401h	1h	RPDO2 ID	40000300h + Node ID
1800h	1h	TPDO1 ID	40000180h + Node ID
1800h	5h	PDO1 Event Timer	0h
1801h	1h	PDO2 ID	80000280h + Node ID
1801h	2h	PDO2 Transmission Type	1h
2001h	0h	Manufacturer Offset	0h
2003h	0h	Enable for zeroing	1h
2004h	0h	Enable for incremental measurement	1h
5F09h	0h	External heartbeat timer	300
5F0Ah	0h	Node ID	1h
5F0Ch	0h	Hysteresis reversal of sense of rotation	0h
5F10h	0h	Target window	5h
5F11h	0h	Decimal places	0h
5F12h	0h	Display orientation and LED	0h
5F13h	0h	Display divisor	0h
5F14h	0h	Loop reversal point	0h
5F15h	0h	Loop direction	0h
5F18h	0h	Trigger source of external heartbeat	1h
6000h	0h	Operating Status	0h
6001h	0h	Resolution	720
6002h	0h	Total measurement range	+/-5242320
6003h	0h	Preset value	0h
6200h	0h	PDO1 Event Timer	see object 1800-5

Object 1011h (Load Default Parameters)

This object serves for setting the encoder to its default values (see *5.2.9*). To be protected against unintended loading of the default values, the string "load" must be written in sub-index 1h.

COB ID	Command	Index Low	Index High	Sub-index	Data 0 (LSB)	Data 1	Data 2	Data 3 (MSB)
600h + Node ID	23h	11h	10h	01h	'l' (6Ch)	'o' (6Fh)	'a' (61h)	'd' (64h)

The write access to the respective sub-indexes results in the values represented below:

Sub-index	**00h**
Description	indicates the largest supported sub-index
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	1h

Sub-index	**01h**	
Description	all default values are loaded	
Access	rw (writable in the "Pre-Operational" and "Operational" states)	
Data type	UNSIGNED 32	
EEPROM	no	
Default	0h	
Data content	Bit 31-1	0
	Bit 0:	0: Encoder does not permit loading of default parameters. 1: Encoder permits loading of default parameters.

Object 1014h (COB ID Emergency Object)

Sub-index	**00h**	
Description	Defines the COB ID of the Emergency object (EMCY)	
Access	rw (writable in the "Pre-Operational" state only)	
Data type	UNSIGNED 32	
EEPROM	yes	
Default	80h + Node ID	
Data content	Bit 31:	0: EMCY object exists / is valid 1: EMCY object does not exists / is invalid
	Bit 30:	always 0
	Bit 29:	0: 11bits identifier (CAN 2.0A) 1: 29bits identifier (CAN 2.0B) ! not supported !
	Bit 28 ... 11	0: if bit 29 = 0
	Bit 10 ... 0	X: bits 10 – 0 of the EMCY-COB ID

Object 1017h (Producer Heartbeat Time)

Sub-index	**00h**
Description	defines the cycle time of the heartbeat monitoring service
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 16
EEPROM	yes
Default	0h
Value range	10 ... 65535 (Ah ... FFFFh); the numerical value corresponds to a multiple of 1 ms. Value 0 disables the service values in the range of 1 ... 9 trigger an error message!

Object 1018h (Identity Object)

Sub-index	**00h**
Description	number of entries
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	2h

Sub-index	**01h**
Description	The manufacturer identification number (vendor ID) for the company SIKO GmbH allocated by the CiA (see www.can-cia.org)
Access	ro
Data type	UNSIGNED 32
EEPROM	no
Default	195h

Sub-index	**02h**			
Description	indicates the display version ASCII-coded.			
Access	ro			
Data type	UNSIGNED 32			
EEPROM	no			
Default	"CAN"			
Data content	Byte 0	Byte 1	Byte 2	Byte 3
	43h ('C')	41h ('A')	4Eh ('N')	00h

Object 1200h (Server SDO parameter)

Sub-index	00h
Description	largest sub-index supported
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	2h

Sub-index	01h
Description	COB ID Client -> Server (rx)
Access	ro
Data type	UNSIGNED 32
EEPROM	yes
Default	00000600h + Node ID

Sub-index	02h
Description	COB ID Server -> Client (tx)
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	00000580h + Node ID

Object 1400h (Receive PDO1 parameter, asynchronous operational mode)

Sub-index	00h
Description	largest sub-index supported
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	2h

Sub-index	01h	
Description	COB ID of RPDO1	
Access	rw	
Data type	UNSIGNED 32	
EEPROM	yes	
Default	40000200h + Node ID	Bit30 = 1: RTR for this PDO not released, bit is always set

Sub-index	**02h**
Description	Transmission Type
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	FFh (255) cannot be changed, update with PDO receipt

Objekt 1401h (Receive PDO2 parameter, synchronous operational mode)

Sub-index	**00h**
Description	largest sub-index supported
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	2h

Sub-index	**01h**
Description	COB ID of RPDO2
Access	rw
Data type	UNSIGNED 32
EEPROM	yes
Default	40000300h + Node ID Bit30 = 1: RTR for this PDO not released, bit is always set

Sub-index	**02h**
Description	Transmission Type
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	FFh (255) cannot be changed, update with PDO receipt

Objekt 1600h (Receive PDO1 Mapping parameter)

Sub-index	**00h**
Description	number of objects mapped
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	1h

Sub-index	01h
Description	Describes the 1st portion of the PDO1 message (data bytes 0 to 3)
Access	ro
Data type	UNSIGNED 32
EEPROM	no
Default	5F160020h (object 5F16h, 32bit)

Sub-index	02h
Description	Describes the 2nd portion of the PDO1 message (data byte 4)
Access	ro
Data type	UNSIGNED 8
EEPROM	yes
Default	5F120008h (Objekt 5F12h, 8bit)

Object 1601h (Receive PDO2 Mapping parameter)

Sub-index	00h
Description	number of objects mapped
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	1h

Sub-index	01h
Description	Describes the 1st portion of the PDO2 message (data bytes 0 to 3)
Access	ro
Data type	UNSIGNED 32
EEPROM	no
Default	5F160020h (object 5F16h, 32bit)

Sub-index	02h
Description	Describes the 2nd portion of the PDO2 message (data byte 4)
Access	ro
Data type	UNSIGNED 8
EEPROM	yes
Default	5F120008h (Objekt 5F12h, 8bit)

Object 1800h (Transmit PDO1 parameter, asynchronous operation mode)

Sub-index	**00h**
Description	largest sub-index supported
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	5h

Sub-index	**01h**
Description	COB ID of the PDO1
Access	rw (writable in the "Pre-Operational" state only)
Data type	UNSIGNED 32
EEPROM	yes
Default	40000180h + Node ID Bit30 = 1: RTR for this PDO not released, bit is always set

Sub-index	**02h**
Description	Transmission Type
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	FEh (254) PDO has asynchronous characteristics (PDOs are sent depending on the "Event Timer"). This value cannot be changed!

Sub-index	**03h (is not used, access attempt generates error message)**

Sub-index	**04h (is not used, access attempt generates error message)**

Sub-index	**05h**
Description	Event Timer
Access	rw (writable in the "Pre-Operational" state only)
Data type	UNSIGNED 16
EEPROM	yes
Value range	0 ... 65535 (1h ... FFFFh); the numerical value corresponds to a multiple of 1 ms. The service is disabled by writing the value 0. The content of this object is identical with object 6200h.

Object 1801h (Transmit PDO2 parameter, synchronous operation mode)

Sub-index	**00h**
Description	largest sub-index supported
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	5h

Sub-index	**01h**
Description	COB ID of the PDO2
Access	rw (writable in the "Pre-Operational" state only)
Data type	UNSIGNED 32
EEPROM	yes
Default	00000280h + Node ID

Sub-index	**02h**
Description	Transmission Type
Access	rw (writable in the "Pre-Operational" state only)
Data type	UNSIGNED 8
EEPROM	yes
Default	1h PDO has synchronous characteristics
Value range	1h … F0h (240) the PDO will be sent following every SYNC command FDh (253): encoder responds to RTR request.

Sub-index	**03h (is not used, access attempt generates error message)**

Sub-index	**04h (is not used, access attempt generates error message)**

Sub-index	**05h (is not used, access attempt generates error message)**

Object 1A00h (Transmit PDO1 Mapping parameter)

Sub-index	**00h**
Description	number of objects mapped
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	2h

Sub-index	01h
Description	Describes the 1st portion of the PDO1 message (data bytes 0 to 3)
Access	ro
Data type	UNSIGNED 32
EEPROM	no
Default	60040020h (Object 6004h, 32bit) position value

Sub-index	02h
Description	Describes the 2nd portion of the PDO1 message (data byte 4)
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	5F190008h (Object 5F19h, 8bit) AP04 status

Object 1A01h (Transmit PDO2 Mapping parameter)

Sub-index	00h
Description	number of objects mapped
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	2h

Sub-index	01h
Description	Describes the 1st portion of the PDO2 message (data bytes 0 to 3)
Access	ro
Data type	UNSIGNED 32
EEPROM	no
Default	60040020h (Object 6004h, 32bit) position value

Sub-index	02h
Description	Describes the 2nd portion of the PDO2 message (data byte 4)
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	5F190008h (Object 5F19h, 8bit) AP04 status

Object 2001h (Manufacturer offset)

Sub-index	**00h**
Description	The offset enables the shifting of a scaled value range. The offset value is added to the position value in the encoder. Positive as well as negative values are permitted. Position value = measured value + calibration value + offset value
Access	rw
Data type	SIGNED 32
EEPROM	yes
Default	0h
Value range	The minimum or maximum values to be entered depend on the values entered in object 650Ah, sub-index 2 or sub-index 3, respectively. The latter depend on the parameterized value of the total of measuring steps: lower_limit = - ½ total of measuring steps = 5242880, upper_limit = ½ total of measuring steps – 1 = 5242779 -5242880 < offset < 5242779

Object 2002h (Zeroing of encoder value)

Sub-index	**00h**
Description	The object enables "zeroing" of the measured value. Position value = measured value + calibration value + offset value
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 8
EEPROM	no
Default	no
Value range	0 ... 1; writing the value 1 on sub-index 0 sets the position value to 0. After renewed zeroing the figure 1 will be output in case of read access.

Example:

Measured value = 214 calibration value = 400, offset value = 0,
results in position value = 614 (see object 6004h)

COB ID	Com-mand	Index Low	Index High	Sub-index	Data 0 (LSB)	Data 1	Data 2	Data 3 (MSB)
600h + Node ID	23h	02h	20h	00h	01h	00h	00h	00h

Measured value = 0
results in position value = 400

This function does not depend on zeroing enable via keyboard (object 2003h)

Object 2003h (Enable zeroing of encoder value via keyboard)

Sub-index	**00h**
Description	Zeroing via * key is enabled with this object.
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 8
EEPROM	yes
Default	no
Value range	0: Zeroing via * key disabled 1: Zeroing via * key enabled

Object 2004h (Incremental measurement enable)

Sub-index	**00h**
Description	Switching on incremental measurement function via ↑ key is enabled with this object.
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 8
EEPROM	yes
Default	no
Value range	0: Incremental measurement function via ↑ key disabled 1: Incremental measurement function via ↑ key enabled

Object 5F09h (External Heartbeat Timer)

Sub-index	**00h**
Description	If a value > 0 is entered here, the AP04 will expect an event to occur in this interval (see object 5F18h). If no such event occurs, the AP04 will change to the "Pre-Operational" state.
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 8
EEPROM	yes
Default	300 (12Ch)
Value range	0 ... 65535 (0h ... FFFFh); the numerical value corresponds to a multiple of 1 ms. The function is disabled by writing the value 0.

Object 5F0Ah (Node ID)

Sub-index	**00h**
Description	Node ID of AP04
Access	rw (writable in the "Preoperational" and "Operational" states) The ID set here will become valid only after re-initialization of communication or power-up.
Data type	UNSIGNED 8
EEPROM	yes
Default	0h
Value range	1 ... 127 (1h ... 7Fh)

Object 5F0Ch (control byte)

Sub-index	**00h**	
Description	control byte	
Access	wo (writable in the "Pre-Operational" and "Operational" states) write access only via PDO	
Data type	UNSIGNED 8	
EEPROM	no	
Default	0h	
Value range	0 ... 48 (0h ... 30h)	
Coding	bit 0	0: target value invalid 1: target value valid
	bit 1	0: LEDs constant 1: LEDs blinking when ON
	bit 2 bit 3	reserved reserved
	bit 4	0: LED green OFF 1: LED green ON independent of target window (only valid when object 5F12h bit 0 = bit 1 = 0) (will be reflected from object 5F12h to bit 4)
	bit 5	0: LED red OFF 1: LED red ON independent of target window (only valid when object 5F12h bit 0 = bit 1 = 0) (will be reflected from object 5F12h to bit 5)
	bit 6 bit 7	not used not used

Object 5F10h (Target window)

Sub-index	**00h**
Description	max. valid deviation from target value, if actual value is within the window: target value achieved
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 32
EEPROM	yes
Default	5h
Value range	0 ... 4.294.967.296 (0h … FFFFFFFFh)

Object 5F11h (Decimal places)

Sub-index	**00h**
Description	Number of decimal places
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 8
EEPROM	yes
Default	0h
Value range	0 ... 4 (0h … 4h)

Object 5F12h (Display orientation and LED)

Sub-index	**00h**			
Description	Display orientation 0° or 180° additional functionality of LED			
Access	rw (writable in the "Pre-Operational" and "Operational" states)			
Data type	UNSIGNED 32			
EEPROM	Basic functions yes			
Default	0h			
Value range	0; 14516 (38B4h)			
Data content	Reserved for future use		LED	Display
	Byte 3	Byte 2	Byte 1	Byte 0
	00h	00h	0 … 56	0 or 180 (B4h)

Coding	Byte 0: Display	0h: 0° B4h: 180°
	Byte 1: LED	bit 0 = 0: LED green OFF bit 0 = 1: LED green ON when position in target window bit 1 = 0: LED red OFF bit 1 = 1: LED red ON when position outside target window bit 3 = 1: LEDs blink when ON bit 4 = 1: LED green ON independent of target window (only valid if bit 0 = bit 1 = 0) (will be reflected out of control byte bit 4) bit 5 = 1: LED red ON independent of target window (only valid if bit 0 = bit 1 = 0) (will be reflected out of control byte bit 5) Only bits 0 … 3 are saved non-volatilely bit 6 = 0: not used bit 7 = 0: not used

Object 5F13h (Display divisor)

Sub-index	**00h**
Description	Display divisor
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 8
EEPROM	yes
Default	1h
Value range	0 … 3 (0h … 3h) 0: Indication of the position value: "10000" 1: Indication of the position value: "1000" 2: Indication of the position value: "100" 3: Indication of the position value: "10"

Object 5F14h (Loop width)

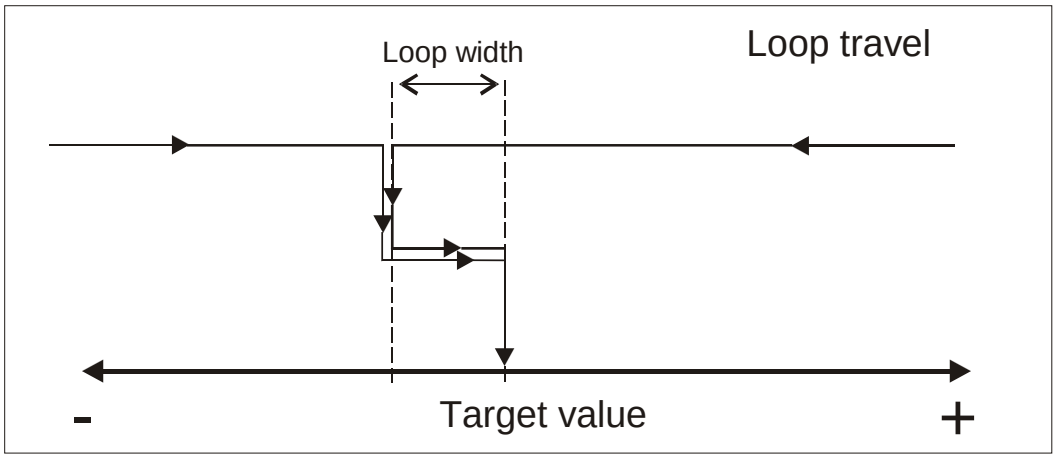
Sub-index	**00h**
Description	Loop width; the target value will be exceeded by this value in case of loop travel.
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 32
EEPROM	yes
Default	0h
Value range	0 ... 4.294.967.296 (0h … FFFFFFFFh)

Description of loop travel, see object 5F15h loop travel.

Object 5F15h (Loop direction)

Sub-index	**00h**
Description	Loop direction; travelling to the target value is always in this direction.
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 8
EEPROM	yes
Default	524944h
Value range	0h ... 524944h 0h: direct (write) 2Bh: = ASCII "+" Target value is approached from the positive direction 2Dh: = ASCII "-" Target value is approached from the negative direction 524944h: = ASCII "DIR" direct (response to read)

If the AP04 is operated on a spindle, then the spindle play can be compensated by means of loop positioning.
In this case, travelling to the target value is always from the same direction.



Example:
Loop width = 100 (Object 5F14h = 100)
The direction from which every target position shall be driven to is: *Negative* (Object 5F15h = "-")
Hysteresis of reversal of sense of rotation = 10
current position value = 1000

- Case 1 ➡ new target value = 1500

  direct travel to target position

- Case 2 ➡ new target value = 500

  The positioning aid (arrows) of AP04 requires that the target position (500) will be exceeded by the loop width (object 5F14h). Upon reaching the loop target position (400 = target position – loop width), the positioning arrows will be reversed and travel to the target position 500 is enabled.
  If, on the way between target loop position and target position, reversal of the sense of rotation > hysteresis (object 5F1A) is detected, a new loop travel will be initiated.

  In the example:
  Movement between target loop position (400) and target position (500)
  Given an actual position of = 453; movement back to position 442 => hysteresis is exceeded, a new travel to target loop position (400) is requested.

Object 5F16h (Read target value)

Sub-index	**00h**
Description	Read target value
Access	rw (writable in the "Pre-Operational" and "Operational" states) write access only via PDO
Data type	UNSIGNED 32
EEPROM	no
Default	0h
Value range	0 ... 4.294.967.296 (0h … FFFFFFFFh)

Any attempt at writing on this object via SDO will result in an error message (error code: 06010000h).

Object 5F18h (External Heartbeat source)

Sub-index	**00h**
Description	Event triggering an external heartbeat timer (object 5F09h)
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 8
EEPROM	yes
Default	0h
Value range	0 ... 1 0: Timer is triggered when receiving a PDO (target value) 1: Timer is triggered when receiving a sync

Object 5F19h (AP04 status)

Sub-index	**00h**
Description	The status byte informs about the current state of AP04.
Access	ro (writable in the "Pre-Operational" and "Operational" PDO states only)
Data type	UNSIGNED 8
EEPROM	yes
Default	0h
Value range	0 ... 2Fh

The bits of the status byte have the following meaning:

Bit	7	6	5	4	3	2	1	0
0 1	-	battery not flat flat battery	"<" off "<" on	">" off ">" on	increm. meas. = 0 increm. meas. = set	Batt. ok Batt.warn	ACT.<TARG ACT.>TARG	Not IN-POS IN-POS

Any attempt at writing on this object will result in an error message (error code: 06010000h).

Object 5F1Ah (Hysteresis of detection of reversal of sense of rotation)

Sub-index	**00h**
Description	Hysteresis, within which detection of reversal of sense of rotation does not initiate new loop travel.
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 32
EEPROM	yes
Default	0h
Value range	0 ... 4.294.967.296 (0h … FFFFFFFFh)

Explanation see Object 5F15h loop travel.

Object 6000h (Operating Parameters)

Sub-index	**00h**				
Description	This object influences the encoder's sense of rotation and the scaling function.				
Access	rw (writable in the "Pre-Operational" and "Operational" states)				
Data type	UNSIGNED 16				
EEPROM	yes				
Default	0h				
Bit definition		Bit 14 .. Bit 3:	Bit 2:	Bit 1:	Bit 0:
	Function	not used	Scaling	not used	Sense of rotation
	Bit = 0	-	disabled	-	E (CCW)
	Bit = 1	-	enabled	-	I (CW)

Explanation of the functions:

I sense of rotation: ascending position values with clockwise (CW) encoder rotation (view at the display)

E sense of rotation: ascending position values with counter-clockwise (CCW) encoder rotation (view at the display)

Scaling disabled: The encoder works with preset readout per revolution or measuring units per revolution. Any attempt at changing the readout per revolution via object 6001h will result in an error message

Scaling enabled: The readout per revolution (object 6001h) can be parameterized.

Object 6001h, (Display per revolution, APU)

Sub-index	**00h**
Description	This parameter sets the desired resolution per revolution. (max. 720 increments per revolution)
Access	rw (writable in the "Pre-Operational" and "Operational" states, if the "Scaling" bit [see object 6000h] is set)
Data type	UNSIGNED 32
EEPROM	yes
Default	720
Value range	1 … FFFFFFFFh

Example:

APU = 400, position = 0;
When the shaft is moved by one revolution, the new position will be 400

Object 6002h (Total Measuring Range [Total of measurement steps])

Sub-index	**00h**
Description	This parameter sets the total number of measuring steps. (number of countable revolutions multiplied with set APU)
Access	ro (readable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 32
EEPROM	no
Default	5242320
Value range	7281 … 4294967295 (FFFFFFFFh)

Any attempt at writing on this object will result in an error message (error code: 06010002h).

Object 6003h (Preset value)

Sub-index	**00h**
Description	The position value of the encoder is set to this preset (calibration) value when zeroing. Position value = measured value + calibration value + offset value see object 6004h
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	SIGNED 32
EEPROM	yes
Default	0h
Value range	0 ... 4.294.967.296 (0h … FFFFFFFFh)

Object 6004h (Position value)

Sub-index	**00h**
Description	This object provides the position value of the encoder offset with the scaling factors, preset and Manufacturer Offset.
Access	ro
Data type	SIGNED 32
EEPROM	no

The position value of the AP04 is calculated by using the following formula:

*Position value = (encoder value - encoder zeroing value)*RF + preset value + Manufacturer Offset*

Encoder value:	absolute value sensed by the encoder sensor system,
Encoder zeroing value:	absolute value at the time of zeroing,
RF:	Calculation (scaling) factor = readout per revolution/ 720
Preset value:	see Object 6300h
Manufacturer Offset	see Object 2100h

With the AP04, the total measuring range is subdivided into a negative and positive value range:

-1/2 total measuring range .. 0 .. +1/2 (total measuring range – 1)

Therefore, the representation of the position value is in the 2-complement format in a signed 32 bits number.

Object 6200h (Cycle Timer)

Sub-index	**00h**
Description	Defines the cycle time with which the PDO1 is output. The value is fixed-linked (identical) with the value indicated under object 1800h, sub-index 5. The timer-controlled output is activated as soon as a cycle time was parame-terized within the value range and the encoder switched over to the Opera-tional Mode.
Access	rw (writable in the "Pre-Operational" and "Operational" states)
Data type	UNSIGNED 16
EEPROM	no
Default	0h
Value range	0: Cycle timer is disabled, 1 ... 65535: cycle time in ms

Object 6500h (Operating Status)

Sub-index	**00h**
Description	The object indicates the settings programmed with object 6000h. (sense of rotation, scaling enable)
Access	ro
Data type	UNSIGNED 16
EEPROM	yes
Default	no
Bit definition	see Object 6000h

Object 6501h (SingleTurn resolution)

Sub-index	00h
Description	The object indicates the maximum possible encoder resolution.
Access	ro
Data type	UNSIGNED 32
EEPROM	no
Default	720

Object 6502h (Number of distinguishable revolutions)

Sub-index	**00h**
Description	The object indicates the maximum possible number of encoder revolutions.
Access	ro
Data type	UNSIGNED 16
EEPROM	no
Default	7281

Object 6503h (Alarms)

Sub-index	**00h**
Description	In addition to the errors reported via the emergency messages, this object provides further, encoder-specific error messages. In the case of an error, the associated bit is set to 1.
Access	ro
Data type	UNSIGNED 16
EEPROM	no
Default	0h

Bit definition:

Bit	Function	Value = 0	Value = 1
0	position error	no error	position value invalid
1 … 11	not used	-	-
12	battery warning	battery voltage OK	battery voltage near lowest tolerable value
13	battery error	battery OK or still in tolerable range	battery discharged
14 … 15	not used		

Object 6504h (Supported Alarms)

Sub-index	**00h**
Description	The object indicates which alarm messages are supported. The relevant bits are set.
Access	ro
Data type	UNSIGNED 16
EEPROM	no
Default	3001h

Bit 0: position error

Bit 12: battery warning

Bit 13: battery alarm

Object 6505h (Warnings)

Sub-index	**00h**
Description	Warnings indicate that tolerances of internal encoder parameters have been exceeded. However, unlike with alarm messages, the position value can be valid in case of a warning.
Access	ro
Data type	UNSIGNED 16
EEPROM	no
Default	0h

Bit definition:

Bit	Function	Value = 0	Value = 1
0 .. 3	not used	-	-
4	battery warning	battery voltage OK	battery voltage near lowest tolerable value
5 ... 15	not used	-	-

Object 6506h (Supported Warnings)

Sub-index	**00h**
Description	The object indicates which warnings are supported.
Access	ro
Data type	UNSIGNED 16
EEPROM	no
Default	0010h

Bit 4: Battery warning is supported

Object 6507h (Profile and Software Version)

Sub-index	**00h**
Description	The object indicates the encoder profile used (CANopen Device profile for encoders) and the version number of the firmware state.
Access	ro
Data type	UNSIGNED 32
EEPROM	no
Default	00650301h

Data content:

Firmware Version		**Profile Version**	
Byte 3 (High)	**Byte 2 (Low)**	**Byte 1 (High)**	**Byte 0 (Low)**
00h	65h	03h	01h

Object 6508h (Operating Time)

Sub-index	**00h**
Description	Operation time counter (not implemented in the AP04)
Access	ro
Data type	UNSIGNED 32
EEPROM	no
Default	FFFFFFFFh (shows that the function is not supported)

Object 6509h (Encoder Zeroing Value)

Sub-index	00h
Description	The difference between encoder value and the position value scaled and offset with preset and/or Manufacturer Offset is output via this object.
Access	ro
Data type	SIGNED 32
EEPROM	yes

$$\textit{Encoder zeroing value} = \textit{encoder value} - \frac{(\textit{position value} - \textit{preset value} - \textit{manufacturer offset}}{\textit{Scaling factor}}$$

Object 650Ah (Module Identification)

The Manufacturer Offset value (sub-index 1), the smallest (sub-index 2) and the largest (sub-index 3) position value can be read out via this object.

Sub-index	**00h**
Description	contains the number of additional sub-indexes.
Access	ro
Data type	UNSIGNED 8
EEPROM	no
Default	3h

Sub-index	**01h**
Description	manufacturer-specific offset value (is added to the position value) see object 2001h
Access	ro
Data type	SIGNED 32
EEPROM	yes
Default	**0h**

Sub-index	02h
Description	Minimum transferable position value
Access	ro
Data type	SIGNED 32
EEPROM	no
Default	-5242880

Sub-index	03h
Description	Maximum transferable position value
Access	ro
Data type	SIGNED 32
EEPROM	no
Default	5242880

Object 650Bh (Serial Number)

Sub-index	00h
Description	Provides the serial number of the encoder (not supported with the AP04).
Access	ro
Data type	UNSIGNED 32
EEPROM	no
Default	FFFFFFFFh (function is not implemented)