SICK AG
Erwin-Sick-Str. 1
79183 Waldkirch

Commissioning instructions

ATM 60 absolute rotary encoder

with CANopen Interface

to communications interface DS 301 V 4.0

and device profile DSP 406 V 2.0

The data in this documentation describes the product. The properties, however, are not guaranteed.

SICK AG
Erwin-Sick-Str. 1
D-79183 Waldkirch

Version: Dez 2021

Contents

1. Explanation of symbols

This symbol is placed at points in the text which have to be observed in particular in order that proper use is ensured and risks are ruled out.

This symbol indicates instructions and particularly useful information. If these are not observed, errors may occur in the functioning of the encoder.

2. Abbreviations used

CAN	Controller Area Network
CiA	CAN in Automation. International association of the users and manufacturers of CAN products.
COB	Communication Object. Transport unit in the CAN network (CAN message). Data is transmitted over the network within a COB.
COB-ID	COB Identifier. Unique identification of a communication object.
DS	Draft Standard.
DSP	Draft Standard Proposal.
ID	Identifier, see COB-ID.
LSB	Least Significant Bit/Byte.
MSB	Most Significant Bit/Byte.
NMT	Network Management. Service element of CAL implementation of initialization, configuration and error handling in a CAN network.
PDO	Process Data Object. Communication object for the rapid interchange of data.
SDO	Service Data Object. Communication object for access to the object directory.
RTR	Remote Transmission Request.
SYNC	Synchronization Telegram.
CMS	CAN Message Specification

3. Safety advice

- Please ensure that the commissioning instructions are read before commissioning.
- The ATM60-Cxx encoders are measuring instruments produced in accordance with the recognized industrial regulations and meet the quality requirements of ISO 9001.
- The installation of the encoder must be performed by specialist personnel with knowledge of electrical engineering and precision mechanical engineering, taking into account the relevant safety regulations.
- An ATM60-Cxx must be used only for the purpose appropriate to its design.
- Observe the professional safety and accident prevention regulations applicable to your country.
- Switch off the voltage to all the devices/machines and plant involved in the mounting.
- For satisfactory function of the devices, care must be taken to ensure good earthing or a screen connection suitable for EMC (connecting the screen at both ends).
- Avoid striking the shaft.

4. Introduction

4.1 CAN general

The ATM60-Cxx is an absolute encoder for use in a CAN network. The CANopen protocol, based on CAN, is used for data transmission.

CAN was originally developed by Bosch for use in motor vehicles. The area of use of this bus system has expanded considerably since then. Nowadays, CAN is used in mobile systems, as a communication system within machinery or plant, in the field area of production automation, in building services engineering and in many other sectors.

CAN has a line structure, a twisted two-wire line being used as the transmission medium. The electrical levels are designed in accordance with ISO DIS 11898.

A main feature of CAN is the object-oriented data transmission. For this purpose, no bus participants (nodes) are addressed, instead each variable to be transmitted is marked by a network-wide defined identifier. In addition, in order to identify the messages, the priority of the respective variable is simultaneously defined by the identifier during the system configuration. The allocation of priorities for the bus access method used is necessary.

4.2 CAN Application Layer (CAL)

In order to facilitate the use of CAN in industrial applications, the “CAN in automation” (CiA) user organization has defined a universal application interface with communications and management services for CAN networks, the “CAN Application Layer” (CAL).

4.3 CANopen

CANopen was defined as a subset of CAL (CAN Application Layer) and, in addition, contains some extensions specifically for industrial real-time systems. A distinction is drawn between the communications profile and the device profile. In the communications profile (CiA DS 301), the type and manner of data interchange between the participants is defined. In the device profile, the functionality of the participant is defined.

4.3.1 CANopen communications profile

The communications profile of CANopen CiA DS 301 describes which properties of CAL are used in which form. Typically, two types of communication occur in a network. Firstly, there are relatively large blocks, which often contain configuration data, as they are known. Therefore, they are normally transmitted comparatively rarely, and are relatively uncritical in terms of time behaviour. Secondly, there is the process data. These are typically rather small information blocks with a high repetition rate and high requirements on uniform time behaviour.

CANopen therefore restricts the flexibility of CAL to two data objects, “Service Data Objects” (SDO) for the transmission of large data blocks, and “Process Data Objects” (PDO) for the real-time transmission of process data. With the restriction to these two data types, the number of CMS objects for each device is also reduced at the same time. This simplifies communication and helps to keep the volume of the communication profile small.

4.3.2 CANopen device profiles

In the industrial sector, a large number of standards has led to a modular system for standard tasks, from which one may choose relatively freely when constructing a system, without being referred to specific manufacturers in the process. By means of device profiles, CANopen intends to expand this situation to the sector of networked components as well. As a compromise between strict regimentation, scalability and openness for future developments, a device profile comprises three sections:

- “Mandatory Functionality” comprises all device functions which must absolutely be present in order to conform to the respective profile.

- “Optional Functionality” describes how optional functions are to be implemented. This means that, although they are not prescribed, when they are present then they are defined as here.

- “Manufacturer Specific Functionality” is, finally, the section in which each manufacturer can describe his individual functions and expansions.

For the ATM60-Cxx, the profile for encoders DSP 406 is definitive.

This profile describes a manufacturer-independent definition for encoders. It is divided into two device classes:

- Class 1
  Describes all the basic functions which the encoder must contain.

- Class 2
  Contains a large number of expanded functions which must be supported by this class of encoders (mandatory) or are optional.

5. Communications model

In CANopen, four classes of communications objects are defined:

- Process data objects (PDOs)
  CAN communications objects (< 9 byte) for the transport of process data (application objects), unconfirmed transfer of data between network participants. No protocol overhead. Content of the data field specified by “PDO mapping”.

- Service data objects (SDOs)
  Access to entries in the object directory of a device. Confirmed transfer of data of any desired length. Two CAN identifiers per SDO object (channel). Specification of the object directory entry via index and sub-index of the entry.

- Management objects
  For tasks of network management.

- Predefined objects
  For the synchronization of operations (synchronization object) and reporting device error states (emergency object).

5.1 Process data objects (PDO)

For the ATM60-Cxx, there are two transmission PDOs, PDO 1 (tx) and PDO 2 (tx) are available for data transmission. The content of the PDOs is specified by the respective “PDO Mapping” Object $1A00_{hex}$ and $1A02_{hex}$.

The type of transmission of a PDO is specified via the sub-index 3 (transmission type) of the objects 1800_{hex} and 1802_{hex} (PDO communication parameter).

The following transmission types are provided:

- Synchronous transmission
  Transmission is carried out on the basis of the synchronization object.

- Asynchronous transmission
  Transmission is carried out without reference to the synchronization object. The transmission is triggered by an internal timer or by a change in a position value.

- Transmission on request
  The transmission is carried out only after the request from another participant has been received.

The following objects in the object directory can be transmitted in a PDO, defined via the PDO mapping:

- Object 6004_{hex}; position value
- Object 6300_{hex}: status register of the CAMs
- Object 6400_{hex}; status register of the working range
- Object 6503_{hex}; alarms
- Object 6505_{hex}; warnings
- Object 2004_{hex}; speed
- Object 2005_{hex}; acceleration

5.2 Service data objects (SDOs)

The use is predominantly the device configuration via configuration tools. Access to the object directory of the encoder can be made via an SDO connection. All the device parameters of the encoder are stored in this object directory at defined addresses (index and sub-index) and can be read and written via SDOs.

The ATM60-Cxx has one SDO channel, comprising:

- A transmission SDO (tx) for messages from the encoder to the master.
- A receiving SDO (rx) for messages from the master to the encoder.

Construction of an SDO message:

SDO-ID	Command	Index		Sub-index	4-byte data			
11 bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

Command	Function	Telegram type	Meaning
22 hex	SDO, Initiate Download Request	Request	Parameters to encoder
60 hex	SDO, Initiate Download Response	Confirmation	Parameters accepted
40 hex	SDO, Initiate Upload Request	Request	Request parameters from encoder
43 hex	SDO, Initiate Upload Response	Reply	Transmit parameters to master (4 bytes)

Command	Function	Telegram type	Meaning
4B hex	SDO, Initiate Upload Response	Reply	Transmit parameters to master (2 bytes)
4F hex	SDO, Initiate Upload Response	Reply	Transmit parameters to master (1 byte)
80 hex	SDO, Abort Domain Transfer	Reply	Encoder sends error code to master

5.3 Network management objects

The following services are supported by the ATM60-Cxx.

5.3.1 Services to control the nodes

The commands are transmitted unconfirmed with the COB-ID of the NMT network object. The transmission is built up as follows.

5.3.1.1 Start Remote Node

Master						**Slave**
Request	→	COB-ID 0	Command 1	Node ID	→	Instruction

Change to operational.

5.3.1.2 Stop Remote Node

Master						**Slave**
Request	→	COB-ID 0	Command 2	Node ID	→	Instruction

Change to stopped.

5.3.1.3 Enter Pre-Operational

Master						**Slave**
Request	→	COB-ID 0	Command 128	Node ID	→	Instruction

5.3.1.4 Reset Node

Master						Slave
Request	→	COB-ID 0	Command 129	Node ID	→	Instruction

All parameters in the entire object directory are set to power-on values.

5.3.1.5 Reset Communication

Master						Slave
Request	→	COB-ID 0	Command 130	Node ID	→	Instruction

The parameters in the object directory are the communications profile are set to power-on values.

5.3.2 Error control services

- Node guarding event
- Life guarding event
- Heartbeat event.

5.3.3 Boot-up service

- Boot-up event.

5.4 Synchronization object (Sync object)

Typical applications of CANopen need a system-wide clock, with which all inputs and outputs are synchronized. As a basis for this, CANopen provides the communication cycle.

The communication cycle is defined by specific synchronization messages, which have a very high priority.

The identifier for the synchronization message is defined in object 1005_{hex}.

5.5 Emergency object (Emcy object)

An emergency message is triggered as soon as the ATM60-Cxx has detec ted an internal error. One emergency message is sent for each error which has occurred. The following error messages are supported by the emergency object:

Error message	Meaning
00xx	Error Reset or no error
10xx	General error

The meanings of the error messages are described in the communications profile of CANopen (DS 301).

6. Network management

6.1 Node monitoring

The node monitoring checks the ability of each node to communicate. In this case, a distinction is drawn between the “node guarding protocol” and the “heartbeat protocol”.

It is possible for only one form of node monitoring to be used: either node guarding or heartbeat.

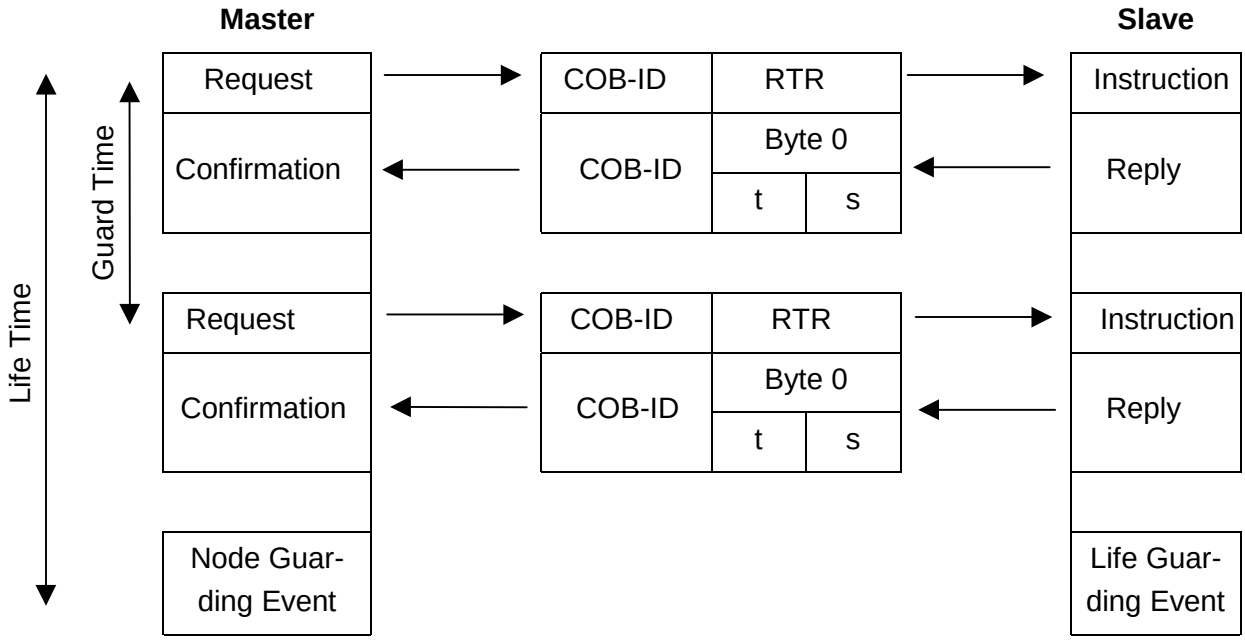
If the “producer heartbeat time (object 1017_{hex}) = 0, then the node guarding is active. If the “producer heartbeat time” (object 1017_{hex}) <> 0, then the heartbeat is active.

6.1.1 Node guarding protocol

At regular intervals (node guarding time), the NMT master sends an RTR telegram to the COB-ID of the NMT error control object. The reply from the NMT slave contains the status of the slave.

The lifetime of the slave is given by multiplying the guard time (object $100C_{hex}$) by the lifetime factor (object $100D_{hex}$).

If no RTR telegram has been received for the period of the lifetime, then the slave signals a “remote node error” via the NMT error control object.



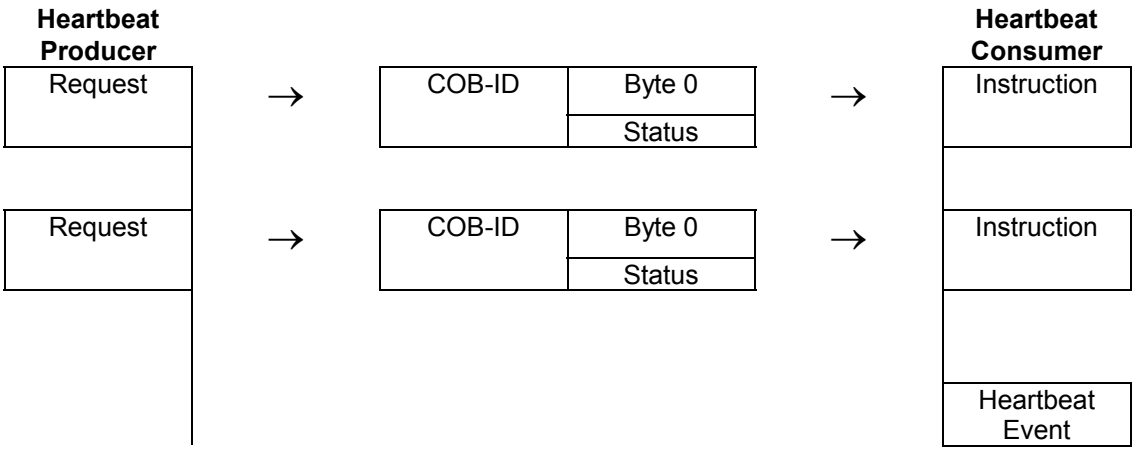
s: Status of the encoder
- 4 : Stopped
- 5 : Operational
- 127 : Pre-operational

t : Toggle bit

6.1.2 Heartbeat protocol

At regular time intervals (producer heartbeat time), the “Heartbeat Producer” sends a “Heartbeat Telegram”. This is a message with the COB-ID of the NMT error control object.

This message is received by one or more “Heartbeat Consumers”. A “Heartbeat Consumer” monitors the receipt of “Heartbeat Telegrams” within its consumer heartbeat time. If no heartbeat telegram is received within this time, then a heartbeat event is triggered.

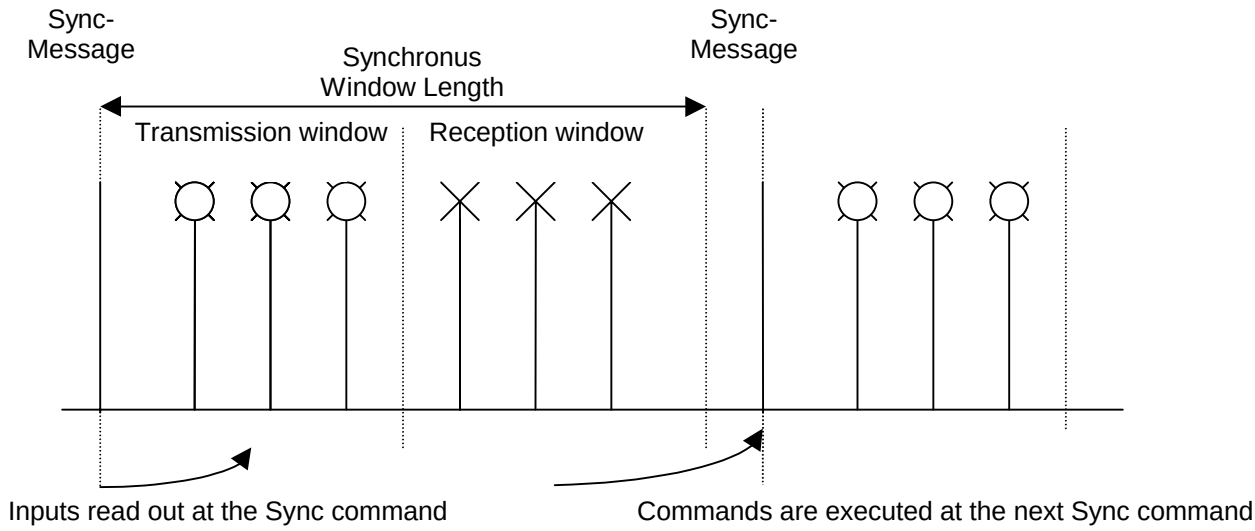


Status:

0	: Boot-up
4	: Stopped
5	: Operational
127	: Pre-operational

6.2 Synchronization

Synchronous transmission of a PDO means that the transmission has a fixed time relationship with the Sync message. The PDO is transmitted within a defined time window, based on the synchronization message.



In the transmission window, directly after the Sync telegram, nodes send their input values and output values priorities.

7. Initialization and system boot-up

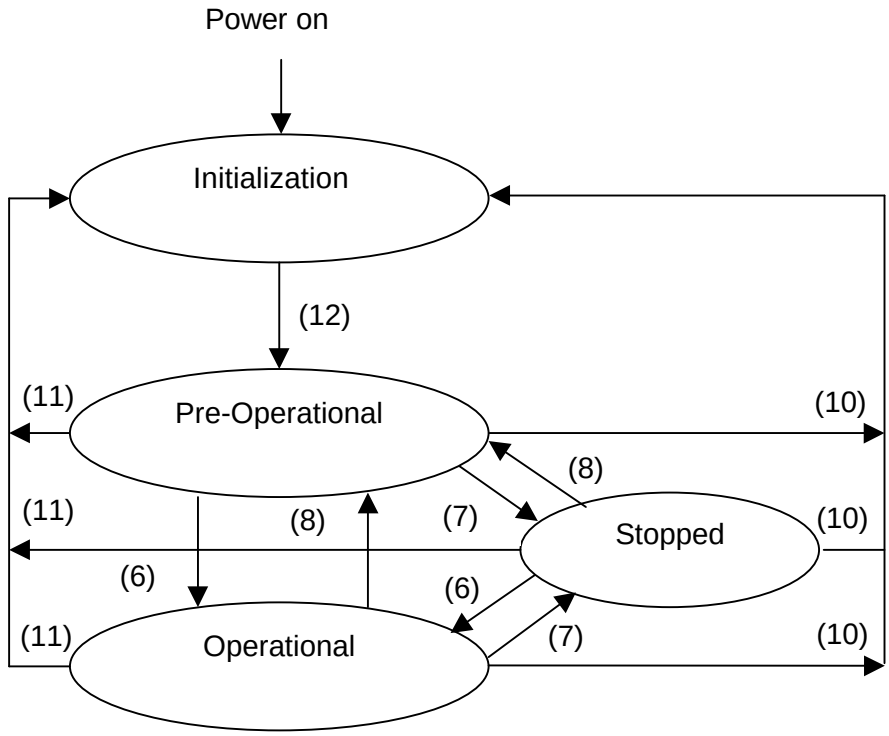
After a change from the initializing status to the pre-operational status, the encoder sends a boot-up

7.1 Initialization

The ATM60-Cxx supports the “minimum capability device” defined in CANopen. This device type permits the implementation of CANopen devices which have only few resources but also provide only little functionality.

State diagram of a CANopen device with “minimum capability”:

KEY TO FIGURE



(6) Start_Remote_Node indication
(7) Stop_Remote_Node indication
(8) Enter_Pre-Operational_State indication
(10) Reset_Node indication
(11) Reset_Communication indication
(12) Initialization concluded, automatic transition to pre-operational

Initialization:
Initial state following application of the supply voltage. After running through the initialization, the node changes automatically to the pre-operational state.

Pre-operational:
The SDO connections are active. The node can then be configured by making access to its object directory.

Operational:

Process values are transmitted via the PDOs.

Stopped:

No SDO or PDO connections are active. The node can be brought into the pre-operational or operational state via NMT commands.

7.2 Boot-up

The boot-up message signals the transition of an NMT slave from the “initializing” state to the “pre-operational” state. The boot-up message uses the COB-ID of the NMT error control object.

Master					Slave
Instruction	←	COB-ID	Status = 0	←	Request

Status = 0 signals that this is a boot-up message.

7.3 Identifier allocation

The identifier allocation is required in each ATM60-Cxx for:

- SDOs
- PDOs
- Synchronization object
- Emergency object
- Network management object.

For simple management of the identifiers, CANopen uses the “predefined connection set”. Here, the COB identifiers are allocated in accordance with the node ID (module ID) and the function code.

7.3.1 Predefined Connection Set

The 11-bit identifier is composed here of a 4-bit function code and 7-bit node ID (module ID).

Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Function code				Node ID (module ID)						

Identifier allocation is static and depends on the node ID (module ID). This permits rapid configuration of simple networks.

The node ID in the ATM60-Cxx can be set via a six-pole DIP switch. This means that, for the ATM 60-Cxx, a node ID in the range from 1 to 63 can be set via the DIP switches. Bit 6 of the node ID is therefore always 0.

Identifier allocation for broadcast objects:

Object	Function code (binary)	COB-ID
NMT Network Object	0000	0
SYNC	0001	128

Identifier allocation for communications objects:

Object	Function code (binary)	COB-ID
EMERGENCY	0001	129-255
PDO 1 (tx)	0011	385-511
PDO 1 (rx)	0100	513-639
PDO 2 (tx)	0101	641-767
PDO 2 (rx)	0110	769-895
SDO (tx)	1011	1409-1535
SDO (rx)	1100	1537-1663
NMT Error Control	1110	1793-1919

8. EDS file

The EDS file is the electronic description of the object directory of a device. The description of the object directory is required for the configuration and operation of the device.

The EDS file is read in by a configuration tool. Using the entries in the EDS file, the configuration tool provides the user with all the available objects in this device.

As a result, a device can be configured easily without any prior knowledge of its object directory.

9. Object directory of the encoder

Organization of the entire object directory in accordance with CANopen.

Index (hex)	Object
0000	Unused
0001..001F	Static data types
0020..003F	Complex data types
0040..005F	Manufacturer-specific data types
0060..007F	Device-profile-specific static data types
0080..009F	Device-profile-specific complex data types
00A0..0FFF	Reserved
1000..1FFF	Area for communications profile
2000..5FFF	Area for manufacturer-specific profile
6000..9FFF	Area for standardized device profile
A000..FFFF	Reserved

The following object directories complying with communications profile CiA DS 301 are implemented in the encoder.

- Index $1000_{hex}..1FFF_{hex}$/area for communications profile
- Index $2000_{hex}..5FFF_{hex}$/area for manufacturer-specific profile
- Index $6000_{hex}..9FFF_{hex}$/area for standardized device profile

9.1 Object directory of the communications profile

In the encoder, the following objects complying with the communications profile CiA DS 301 are implemented.

Index (hex)	Object	Name	Type	Access	M/O
1000	VAR	Device type	Unsigned 32	ro	M
1001	VAR	Error register	Unsigned 8	ro	M
1003	ARRAY	Predefined error field	Unsigned 32	ro	O
1005	VAR	COB-ID for SYNC message	Unsigned 32	rw	O

Index (hex)	Object	Name	Type	Access	M/O
1007	VAR	Window width for SYNC	Unsigned 32	rw	O
1008	VAR	Manufacturer's device name	Vis string	ro	O
1009	VAR	Manufacturer's hardware version	Vis string	ro	O
100A	VAR	Manufacturer's software version	Vis string	ro	O
100C	VAR	Guard time	Unsigned 32	rw	O
100D	VAR	Lifetime factor	Unsigned 32	rw	O
1010	VAR	Save parameters	Unsigned 32	rw	O
1011	VAR	Load default parameters	Unsigned 32	rw	O
1014	VAR	COB-ID for emergency message	Unsigned 32	rw	O
1015	VAR	Inhibit time for emergency message	Unsigned 16	rw	O
1017	VAR	Producer heartbeat time	Unsigned 16	rw	O
1018	RECORD	Identity object	Identity	ro	M
1800	RECORD	Transmit PDO 1, parameter		rw	M
1802	RECORD	Transmit PDO 2, parameter		rw	M
1A00	ARRAY	Transmit PDO 1, mapping		ro	M
1A02	ARRAY	Transmit PDO 2, mapping		ro	M

9.1.1 Description of the individual objects

9.1.1.1 Object 1000 hex: device type

Description
Contains information about the device type.

Explanation

Device type			
Device profile number		Encoder type	
Byte 0	Byte 1	Byte 2	Byte 3
196_{hex} (encoder profile $406_{decimal}$)		02_{hex} (multiturn encoder)	

9.1.1.2 Object 1001 hex: error register

Description
In this register, the encoder indicates errors. For a more detailed error analysis, object 1003_{hex} (predefined error field) must be used.

Explanation
Bit 0 => general error, 0 = no error occurred, 1 = error occurred
Bit 1..7 => are not supported

9.1.1.3 Object 1003 hex: predefined error field

Description
If the encoder detects errors in the sequence, these are entered into the predefined error field. The error field comprises a maximum of four error entries. Each new error is entered under sub-index 1, all the other errors in the error field are shifted back by one sub-index. The last entry under sub-index 4 is lost in this event.

Explanation

- Sub-index 0: number of errors that have occurred
- Sub-index 1: first entry in the effort field
- Sub-index 2: second entry in the error field
- Sub-index 3: third entry in the error field
- Sub-index 4: fourth entry in the error field.

All entries in the error field are deleted by writing a 0 to sub-index 0.

The meanings of the error messages are described in the communications profile of CANopen (DS 301).

9.1.1.4 Object 1005 hex: COB-ID for SYNC messages

Description
This object defines the COB-ID for SYNC messages.

Explanation

Bit number	Value	Meaning
31 (MSB)	1	Device is receiving SYNC messages
30	0	Device not generating any SYNC messages
29	0	11-bit ID (CAN 2.0A)

Bit number	Value	Meaning
28-11	0	Reserved for 29-bit identifier
10-0	X	Bits 10..0 of the SYNC-COB-ID

9.1.1.5 Object 1008 hex: manufacturer's device name

Description
Contains the manufacturer's device name.

Explanation
In the event of a read access to this object, the ATM60-Cxx sends the following reply:

- "AG 626 CO" in ASCII code.

9.1.1.6 Object 1009 hex: manufacturer's hardware version

Description
Contains the manufacturer's hardware version.

Explanation
In the event of a read access to this object, the ATM60-Cxx sends the following reply:

- e.g. "HW_V 1.00" in ASCII code.

9.1.1.7 Object 100A hex: manufacturer's software version

Description
Contains the manufacturer's software version.

Explanation
In the event of a read access to this object, the ATM60-Cxx sends the following reply:

- e.g. "SW_V 2.00" in ASCII code.

9.1.1.8 Object 100C hex: guard time and 100D hex: lifetime factor

Description
The objects $100C_{hex}$ and $100D_{hex}$ contain the guard time in milliseconds and the lifetime factor. The lifetime factor multiplied by the guard time gives the lifetime period for the node monitoring protocol.

Explanation
See 6.1 Node monitoring.

9.1.1.9 Object 1010 hex: save parameters

Description
Via this object, the parameters of the encoder can be saved in an Eeprom.

By writing the command “save”, the parameters are stored in an Eeprom. This prevents any inadvertent writing to this object having the effect of saving the parameters.

ASCII (MSB-LSB)	e	v	a	s
Hex (MSB-LSB)	65h	76h	61h	73h

Explanation

- Sub-index 0: highest sub-index supported

- Sub-index 1: save all parameters
  All objects listed under sub-index 2 to 4 will be saved in the Eeprom.

- Sub-index 2: save communications parameters
  The following objects are saved in the Eeprom:
  Object 1005_{hex} => COB-ID for synchronization message
  Object 1007_{hex} => time window for synchronization
  Object $100C_{hex}$ => guard time
  Object $100D_{hex}$ => lifetime factor
  Object 1014_{hex} => COB-ID for emergency message
  Object 1015_{hex} => inhibit time for emergency message
  Object 1017_{hex} => producer heartbeat time
  Object 1800_{hex} => transmit PDO 1, parameter
  Object 1802_{hex} => transmit PDO 2, parameter
  Object $1A00_{hex}$ => transmit PDO 1, mapping
  Object $1A02_{hex}$ => transmit PDO 2, mapping

- Sub-index 3: save applications parameters
  The following objects are saved in the Eeprom:
  Object 6000_{hex} => operating parameters
  Object 6001_{hex} => measuring step per revolution
  Object 6002_{hex} => total number of measuring steps
  Object 6003_{hex} => preset value
  Object 6200_{hex} => cycle timer
  Object 6301_{hex} => cams, enable register
  Object 6302_{hex} => cams, polarity register
  Object $6310..6317_{hex}$ => cams, lower limit
  Object $6320..6237_{hex}$ => cams, upper limit

Object $6330..6337_{hex}$ => cams, hysteresis
Object 6401_{hex} => working range, lower limit
Object 6402_{hex} => working range, upper limit

- Sub-index 4: save manufacturer-specific parameters
  The following objects are saved in the Eeprom:
  Object 2000_{hex} => source for node ID
  Object 2001_{hex} => node ID
  Object 2002_{hex} => baud rate
  Object 2003_{hex} => format for speed and acceleration

9.1.1.10 Object 1011 hex: load default parameters

Explanation
Via this object, the default values of the parameters of the encoder can be loaded.

By writing the command “load”, the parameters are set to their default values. This prevents inadvertent writing to this object having the effect of setting the parameters to default values.

ASCII (MSB - LSB)	d	a	o	L
Hex (MSB - LSB)	64h	61h	6Fh	6Ch

Explanation

- Sub-index 0: highest sub-index supported

- Sub-index 1: set all parameters to default values
  All objects listed under sub-index 2 to 4 are set to default values.

- Sub-index 2: set communications parameters to default values.
  The following objects are set to default values:
  Object 1005_{hex} => COB-ID for synchronization message, default = 128
  Object 1007_{hex} => time window for synchronization, default = 0
  Object $100C_{hex}$ => guard time, default = 0
  Object $100D_{hex}$ => lifetime factor, default = 0
  Object 1014_{hex} => COB-ID for emergency message, default = 128 + node ID
  Object 1015_{hex} => inhibit time for emergency message, default = 0
  Object 1017_{hex} => producer heartbeat time, default = 0
  Object 1800_{hex} => transmit PDO 1, parameter
  COB-ID, default = 384 + node ID
  Transmission type, default = 254
  Inhibit time, default = 0
  Object 1802_{hex} => transmit PDO 2, parameter

COB-ID, default = 640 + node ID
Transmission type, default = 1
Inhibit time, default = 0
Object $1A00_{hex}$ => transmit PDO 1, mapping
First object, default = 60040020_{hex}
Second object, default = 63000108_{hex}
Third object, default = 64000108_{hex}
Fourth object, default = 65030010_{hex}
Object $1A02_{hex}$ => transmit PDO 2, mapping
See object $1A00_{hex}$

- Sub-index 3: set applications parameters to default values.
The following objects are set to default values:
Object 6000_{hex} => operating parameters, default = 0
Object 6001_{hex} => measuring steps per revolution, default = 8192
Object 6002_{hex} => total number of measuring steps, default = 67108864
Object 6003_{hex} => preset value, default = 0
Object 6200_{hex} => cycle timer, default = 0
Object 6301_{hex} => cams, enable register, default = 0
Object 6302_{hex} => cams, polarity register, default = 0
Object $6310_{hex}..6317_{hex}$ => cams, lower limit, default = 0
Object $6320_{hex}..6327_{hex}$ => cams, upper limit, default = 67108864
Object $6330_{hex}..6337_{hex}$ => cams, hysteresis, default = 0
Object 6401_{hex} => working range, lower limit, default = 0
Object 6402_{hex} => working range, upper limit, default = 67108864

- Sub-index 4: set manufacturer-specific parameters to default values.
The following objects are set to default values:
Object 2000_{hex} => source for node ID, default = 0
Object 2001_{hex} => node ID, default = 1
Object 2002_{hex} => baud rate, default = 3
Object 2003_{hex} => format for speed and acceleration, default = 1

9.1.1.11 Object 1014 hex: COB-ID for emergency message

Description
This object defines the COB-ID for the emergency message.

Explanation

Bit number	Value	Meaning
31 (MSB)	0	reserved
30	0	reserved
29	0	11-bit ID (CAN 2.0A)
28-11	0	reserved for 29-bit identifier
10-0	X	bits 10..0 of the EMCY-COB-ID

9.1.1.12 Object 1015 hex: inhibit time for emergency message

Description
This object defines the minimum time in ms which must elapse between two emergency messages.

9.1.1.13 Object 1017 hex: producer heartbeat time

Description
This object defines the time in ms after which the encoder sends a heartbeat message. This is used by the master for monitoring the encoder during asynchronous PDO transmission.

Explanation
See 6.1 Node monitoring.

9.1.1.14 Object 1800 hex: transmit PDO 1, parameter

Description
This object contains the communications parameters for the first transmit PDO.

Explanation

- Sub-index 0: number of entries
- Sub-index 1: COB-ID for transmit PDO 1
- Sub-index 2: transmission type
- Sub-index 3: inhibit time. Defines the minimum time, in 100 µs steps, which must elapse between two PDO 1 messages.

COB-ID for transmit PDO

Bit number	Value	Meaning
31 (MSB)	0 1	PDO enabled PDO disabled
30	0	RTR permitted
29	0	11-bit ID (CAN 2.0A)
28-11	0	reserved for 29-bit identifier
10-0	X	bits 10..0 of the COB-ID

Transmission type

Transmission type	PDO transmission				
	cyclic	acyclic	synchronous	asynchronous	only RTR
0		X	X		
1-240	X		X		
241-251	Reserved				
252			X		
253				X	X
254				X	
255				X	

0 PDO is transmitted after a SYNC telegram if the position value has changed since the last transmitted PDO.

1-240 PDO is transmitted after 1..240 received SYNC telegrams.

252 As a result of a SYNC telegram, the content of the PDO is saved internally. However, the PDO is transmitted only upon a request from an RTR telegram.

253 The content of the PDO is updated and transmitted after an RTR telegram.

254 The content of the PDO is transmitted either under timer control (Object 6200_{hex}<>0) or after a position change (Object 6200_{hex} = 0).

9.1.1.15 Object 1802 hex: transmit PDO 2, parameter

Description
This object contains the communications parameters for the second transmit PDO.

Explanation

- Sub-index 0: number of entries
- Sub-index 1: COB-ID for transmit PDO 1
- Sub-index 2: transmission type
- Sub-index 3: inhibit time. Defines the minimum time, in 100 µs steps, which must elapse between two PDO 2 messages.

Description of the sub-indices => see object 1800_{hex}.

9.1.1.16 Object 1A00 hex: transmit PDO 1, mapping

Description
Contains the composition of the data which are transmitted via the PDO 1.
Structure of the entries of sub-index 1 to 4.

MSB		LSB
Index (16 bits)	Sub-index (8 bits)	Object length (8 bits)
Example of an entry in the status register of the cams:		
6300_{hex}	01_{hex}	08_{hex}

Explanation

- Sub-index 0: number of objects mapped in the PDO.
- Sub-index 1: first object mapped in the PDO.
  Position value (Object 6004_{hex}): 60040020_{hex}
- Sub-index 2: second object mapped in the PDO.
  Default value => status register of the cams (Object 6300_{hex}): 63000108_{hex}
- Sub-index 3: third object mapped in the PDO.
  Default value => status register of the working range (Object 6400_{hex}): 64000108_{hex}
- Sub-index 4: fourth object mapped in the PDO.
  Default value => alarm messages (Object 6503_{hex}): 65030010_{hex}

Sub-index 1 cannot be changed. This means that the position value is always contained in the PDO. Sub-index 2 to 4 can be read and written via an SDO connection.

The following objects from the object directory can be mapped in a PDO.

- Object 6004: position value
  Entry under sub-index 2..4: 60040020_{hex}
- Object 6300: status register of the cams
  Entry under sub-index 2..4: 63000108_{hex}
- Object 6400: status register of the working range
  Entry under sub-index 2..4: 64000108_{hex}
- Object 6503: alarms
  Entry under sub-index 2..4: 65030010_{hex}
- Object 6505: warnings
  Entry under sub-index 2..4: 65050010_{hex}
- Object 2004: speed
  Entry under sub-index 2..4: 20040020_{hex}
- Object 2005: acceleration
  Entry under sub-index 2..4: 20050020_{hex}

Changing the composition of a PDO must be carried out in accordance with the following scheme:

- Firstly, the sub-index 0 must be set to 0.
- Then, the necessary sub-indices 2 to 4 are written with the new objects to be mapped. In this case, a check is made as to whether the object can be transmitted in a PDO.
- Then, sub-index 0 is set to the number of sub-indices to be mapped. Here, a check is made on the overall length of the objects to be mapped to a maximum of 8 bytes. If the check is correct, the content of the PDO is changed. If not, the composition of the PDO remains the same.

9.1.1.17 Object 1A02 hex: transmit PDO 2, mapping

Description
Contains the composition of the data which are transmitted via the PDO 2.

Explanation

- Sub-index 0: number of objects mapped in the PDO.
- Sub-index 1: first object mapped in the PDO.
  Position value => 60040020_{hex}

- Sub-index 2: second object mapped in the PDO.
  Default value => status register of the cams (object 6300_{hex}): 63000108_{hex}

- Sub-index 3: third object mapped in the PDO.
  Default value => status register of the working range (object 6400_{hex}): 64000108_{hex}

- Sub-index 4: fourth object mapped in the PDO.
  Default value => alarm messages (object 6503_{hex}): 65030010_{hex}

Description of the sub-indices => see object $1A00_{hex}$.

9.2 Object directory for the device profile

Index (hex)	Object	Name	Type	Access	M/O
6000	VAR	Operating parameter	Unsigned 16	rw	M
6001	VAR	Measuring steps per revolution	Unsigned 32	rw	M
6002	VAR	Total number of measuring steps	Unsigned 32	rw	M
6003	VAR	Preset value	Unsigned 32	rw	M
6004	VAR	Position value	Unsigned 32	rw	M
6200	VAR	Cycle timer	Unsigned 16	rw	M
6300	ARRAY	Cams, status register	Unsigned 8	ro	O
6301	ARRAY	Cams, enable register	Unsigned 8	rw	O
6302	ARRAY	Cams, polarity register	Unsigned 8	rw	O
6310	ARRAY	Cam 1, lower limit	Unsigned 32	rw	O
6311	ARRAY	Cam 2, lower limit	Unsigned 32	rw	O
6312	ARRAY	Cam 3, lower limit	Unsigned 32	rw	O
6313	ARRAY	Cam 4, lower limit	Unsigned 32	rw	O
6314	ARRAY	Cam 5, lower limit	Unsigned 32	rw	O
6315	ARRAY	Cam 6, lower limit	Unsigned 32	rw	O
6316	ARRAY	Cam 7, lower limit	Unsigned 32	rw	O
6317	ARRAY	Cam 8, lower limit	Unsigned 32	rw	O
6320	ARRAY	Cam 1, upper limit	Unsigned 32	rw	O
6321	ARRAY	Cam 2, upper limit	Unsigned 32	rw	O

Index (hex)	Object	Name	Type	Access	M/O
6322	ARRAY	Cam 3, upper limit	Unsigned 32	rw	O
6323	ARRAY	Cam 4, upper limit	Unsigned 32	rw	O
6324	ARRAY	Cam 5, upper limit	Unsigned 32	rw	O
6325	ARRAY	Cam 6, upper limit	Unsigned 32	rw	O
6326	ARRAY	Cam 7, upper limit	Unsigned 32	rw	O
6327	ARRAY	Cam 8, upper limit	Unsigned 32	rw	O
6330	ARRAY	Cam 1, hysteresis	Unsigned 16	rw	O
6331	ARRAY	Cam 2, hysteresis	Unsigned 16	rw	O
6332	ARRAY	Cam 3, hysteresis	Unsigned 16	rw	O
6333	ARRAY	Cam 4, hysteresis	Unsigned 16	rw	O
6334	ARRAY	Cam 5, hysteresis	Unsigned 16	rw	O
6335	ARRAY	Cam 6, hysteresis	Unsigned 16	rw	O
6336	ARRAY	Cam 7, hysteresis	Unsigned 16	rw	O
6337	ARRAY	Cam 8, hysteresis	Unsigned 16	rw	O
6400	ARRAY	Working range, status register	Unsigned 8	ro	O
6401	ARRAY	Working range, lower limit	Unsigned 32	rw	O
6402	ARRAY	Working range, upper limit	Unsigned 32	rw	O
6500	VAR	Operating status	Unsigned 16	ro	M
6501	VAR	Single-turn resolution (physical)	Unsigned 32	ro	M
6502	VAR	Number of revolutions (physical)	Unsigned 32	ro	M
6503	VAR	Alarms	Unsigned 16	ro	M
6504	VAR	Alarms supported	Unsigned 16	ro	M
6505	VAR	Warnings	Unsigned 16	ro	M
6506	VAR	Warnings supported	Unsigned 16	ro	M
6507	VAR	Profile and software version	Unsigned 32	ro	M
6508	VAR	Operating-time counter	Signed 32	ro	M
6509	VAR	Offset value	Unsigned 32	ro	M

Index (hex)	Object	Name	Type	Access	M/O
650A	VAR	Manufacturer-specific offset value	Unsigned 32	ro	M
650B	VAR	Serial number	Unsigned 32	ro	M

9.2.1 Description of the individual objects

9.2.1.1 Object 6000 hex: operating parameters

Description
This object contains the functions for the code sequence (CW or CCW), self-diagnosis of the encoder and the scaling function.

Explanation:

Bit	Function	Bit = 0	Bit = 1
0	Code sequence	CW	CCW
1	Self-diagnosis of the encoder	disabled	enabled
2	Scaling function	disabled	enabled
3-11	Reserved		
12-15	Not used		

If the scaling function is switched on, then any existing preset and offset values are reset.

9.2.1.2 Object 6001 hex: measuring steps per revolution

Description
This object indicates the desired resolution per revolution. The value can be chosen freely between 1 and 8192.

Any change in this object has the effect of resetting the preset and offset values.

9.2.1.3 Object 6002 hex: total number of measuring steps

Description
This object indicates the desired total number of measuring steps. The value may only be 2^n * measuring steps per revolution (object 6001_{hex}). If a value outside this restriction is chosen, it is modified automatically by the encoder.

Any change in this object has the effect of resetting the preset and offset values.

9.2.1.4 Object 6003 hex: preset value

Description

By means of this object, the current position value is set to the preset value.

Explanation
The encoder calculates an offset value from the current position value and the preset value. This offset value can be read out via object 6509_{hex}. The execution of the preset function is carried out with a write access to this object. The current position value from the encoder is in this case set to the transmitted preset value. The preset value and the calculated offset value are stored in a non-volatile manner in an Eeprom.

It should be noted that the Eeprom has a write cycle of 1 million cycles. This means that this memory location can become defective if 1 million write cycles is exceeded.

The preset and offset values are deleted by means of the following actions:

- Executing the functions restore default parameters
- Changing the resolution per revolution (object 6001_{hex})
- Changing the total number of measuring steps (object 6002_{hex})
- Changing the setting of the scaling function (object 6000_{hex}, bit 1)

9.2.1.5 Object 6004 hex: position value

Description
Via this object, the current position value is read.

9.2.1.6 Object 6200 hex: cycle timer

Description
Defines the transmission time for all asynchronous PDOs. A timer-controlled output becomes active when a cycle time <> 0 is programmed.

If the value of the object = 0, then the output is controlled by a position value change.

9.2.1.7 Object 6300 hex: cams status register

Description
This object contains the status bits of the cams.

Explanation

- Sub-index 0: number of available channels
- Sub-index 1: status of channel 1

Cam status register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Cam 8	Cam 7	Cam 6	Cam 5	Cam 4	Cam 3	Cam 2	Cam 1
Status	Status	Status	Status	Status	Status	Status	Status

Relationship between the status and the polarity:

	Status = 0	Status = 1
Polarity = 0	Cam inactive	Cam active
Polarity = 1	Cam active	Cam inactive

9.2.1.8 Object 6301 hex: cams, enable register

Description
Via this object, the individual cams can be enabled for monitoring.

Explanation

- Sub-index 0: number of available channels
- Sub-index 1: enable register for channel 1

Cams, enable register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Cam 8	Cam 7	Cam 6	Cam 5	Cam 4	Cam 3	Cam 2	Cam 1
enable	enable	enable	enable	enable	enable	enable	enable

enable = 1: cam is enabled for monitoring

enable = 0: cam is not enabled for monitoring

9.2.1.9 Object 6302 hex: cams, polarity register

Description
In this object, the polarity of the status bits is defined. If the polarity bit = 1, the status bit = 0 in the case of an active cam. If the polarity bit = 0, the status bit = 1 in the case of an active cam.

Explanation

- Sub-index 0: number of available channels
- Sub-index 1: polarity of channel 1

Cams, polarity register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Cam 8	Cam 7	Cam 6	Cam 5	Cam 4	Cam 3	Cam 2	Cam 1
polarity	polarity	polarity	polarity	polarity	polarity	polarity	polarity

9.2.1.10 Object 6310 to 6317 hex: cams 1..8, lower limit

Description
Defines the lower guard limit of the individual cams.

Explanation

- Sub-index 0: number of available channels
- Sub-index 1: lower limit for channel 1

If the value for the lower limit of a cam is greater than the upper limit, then the upper limit is set to the lower limit.

9.2.1.11 Object 6320 to 6327 hex: cams 1..8, upper limit

Description
Defines the upper guard limit of the individual cams.

Explanation

- Sub-index 0: number of available channels
- Sub-index 1: upper limit for channel 1

If the value for the lower limit of a cam is greater than the upper value, then the upper and lower values are interchanged.

9.2.1.12 Object 6330 to 6337 hex: cams 1..8, hysteresis

Description
Defines the hysteresis between the cams switching on and off.

Explanation

- Sub-index 0: number of available channels
- Sub-index 1: hysteresis for channel 1

If the hysteresis is greater than the lower limit, the hysteresis is set to the value of the lower limit.

9.2.1.13 Object 6400 hex: working range, status register

Description
This object contains the status of the programmed working range of the encoder (lower limit = object 6401_{hex}, upper limit 6402_{hex}).

Explanation

- Sub-index 0: number of available channels
- Sub-index 1: status register for channel 1

Working range, status register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
reserved	reserved	reserved	reserved	reserved	range underflow	range overflow	out of range

9.2.1.14 Object 6401 hex: working range, lower limit

Description
Defines the lower guard limit of the working range.

Explanation

- Sub-index 0: number of available channels
- Sub-index 1: lower limit for channel 1

9.2.1.15 Object 6402 hex: working range, upper limit

Description
Defines the upper guard limit of the working range.

Explanation

- Sub-index 0: number of available channels
- Sub-index 1: upper limit for channel 1

9.2.1.16 Object 6500 hex: operating status

Description
This object contains the status information from the encoder.

Explanation

Bit	Function	Bit = 0	Bit = 1
0	Code sequence	CW	CCW
1	Self-diagnosis of the encoder	not supported	supported
2	Scaling function	Off	On
3-11	reserved		
12-15	Not used		

9.2.1.17 Object 6501 hex: single-turn resolution (physical)

Description
This object contains the physical number of steps per revolution. In the ATM60-Cxx, this is 8192 steps per revolution.

9.2.1.18 Object 6502 hex: number of revolutions (physical)

Description
This object contains the physical number of revolutions. In the ATM60-Cxx, this is 8192 revolutions.

9.2.1.19 Object 6503 hex: alarms

Description
This object contains further errors in addition to the emergency messages.

Explanation

Bit	Function	Bit = 0	Bit = 1
0	Position error	no	yes
1	Self-diagnosis of the encoder	OK	error
2-11	reserved		
12-15	not used		

9.2.1.20 Object 6504 hex: alarms supported

Description
This objects defines the alarm messages which are supported by the encoder.

Explanation

Bit	Function	Bit = 0	Bit = 1
0	Position error	no	yes
1	Self-diagnosis of the encoder	no	yes
2-11	reserved		
12-15	not used		

9.2.1.21 Object 6505 hex: warnings

Description
Warnings signal the fact that the encoder is outside its tolerances. As opposed to alarm messages, in the case of warnings the position value from the encoder may nevertheless be valid. As soon as the encoder moves within its tolerances again, the corresponding warning messages are withdrawn.

Explanation

Bit	Function	Bit = 0	Bit = 1
0	Excessive rotational speed	no	yes
1	Optical monitoring reserve	not reached	reached
2	CPU watchdog status	OK	reset triggered
3	Operating time	reached	not reached

Bit	Function	Bit = 0	Bit = 1
			reached
4	Battery charge	OK	too low
5	Reference point	reached	not reached
6-11	reserved		
12-15	not used		

9.2.1.22 Object 6506 hex: warnings supported

Description
This object defines the warnings which are supported by the encoder.

Explanation

Bit	Function	Bit = 0	Bit = 1
0	Speed monitoring	not supported	supported
1	Optical monitoring reserve	not supported	supported
2	CPU watchdog status	not supported	supported
3	Operating time	not supported	supported
4	Battery charge	not supported	supported
5	Reference point	not supported	supported
6-11	reserved		
12-15	not used		

Warnings on a grey background are not supported by the encoder.

9.2.1.23 Object 6507 hex: profile and software version

Description
This object contains the profile and software version of the encoder.

Explanation

Profile version		Software version	
Byte 0	Byte 1	Byte 2	Byte 3
00	02	00	02

9.2.1.24 Object 6508 hex: operating time

Description
This object contains the operating time of the encoder. This function is not supported by the ATM60-Cxx. Therefore, this object is set to FF FF FF FF_{hex}.

9.2.1.25 Object 6509 hex: offset value

Description
This object contains the offset value which is calculated by executing the preset function.

9.2.1.26 Object 650A hex: manufacturer-specific offset value

Description
This object contains the manufacturer-specific offset value. This is always 0 in the ATM60-Cxx.

9.2.1.27 Object 650B hex: serial number

Description
Contains the serial number of the encoder.

9.3 Object directory for the manufacturer-specific area

Index (hex)	Object	Name	Type	Access	M/O
2000	VAR	Source for the node ID	Unsigned 8	rw	O
2001	VAR	Node ID	Unsigned 8	rw	O
2002	VAR	Baud rate	Unsigned 8	rw	O
2003	VAR	Format for speed and acceleration	Unsigned 8	rw	O
2004	VAR	Speed	Signed 32	ro	O

Index (hex)	Object	Name	Type	Access	M/O
2005	VAR	Acceleration	Signed 32	ro	O
2006	VAR	Change of state	Unsigned 32	rw	O

9.3.1 Description of the individual objects

9.3.1.1 Object 2000 hex: source for node ID

Description
Defines the source which is used to set the node ID. A new node ID or baud rate is accepted only after a power-on of the encoder.

Explanation

- Source for node ID = 0
  The setting of the node ID and of the baud rate is carried out on the basis of the DIP switch settings.

- Source for node ID = 1
  The setting of the node ID and the baud rate is carried out on the basis of objects 2001_{hex} and 2002_{hex}.

9.3.1.2 Object 2001 hex: node ID

Description
This object defines the node ID if the setting is made not by the DIP switches but by the Eeprom. For this purpose, object 2000_{hex} (source for node ID and baud rate) must be set to 1.

9.3.1.3 Object 2002 hex: baud rate

Description
This object defines the baud rate if the setting is not carried out by the DIP switches but by the Eeprom. For this purpose, object 2000_{hex} (source for node ID and baud rate) must be set to 1.

Explanation

Baud rate = 0 : 10 kBaud
Baud rate = 1 : 20 kBaud
Baud rate = 2 : 50 kBaud
Baud rate = 3 : 125 kBaud
Baud rate = 4 : 250 kBaud

Baud rate = 5 : 500 kBaud
Baud rate = 6 : 1 MBaud

9.3.1.4 Object 2003 hex: format for speed and acceleration

Description
This object defines the format of the outputs of the speed and of the acceleration (object 2004_{hex} and object 2005_{hex}).

Explanation

- Format = 0
  Output of the speed: steps per second.
  Output of the acceleration: steps per second per second.

- Format = 1
  Output of the speed: revolutions per second.
  Output of the acceleration: revolutions per second per second.

- Format = 2
  Output of the speed: revolutions per minute.
  Output of the acceleration: revolutions per minute per second.

- Format = 3
  Output of the speed: revolutions per hour.
  Output of the acceleration: revolutions per hour per second.

9.3.1.5 Object 2004 hex: speed

Description
Via this object, the speed of the rotational movement of the encoder shaft can be read out. The output format of the speed depends on the setting of object 2003_{hex} (format for speed and acceleration).

9.3.1.6 Object 2005 hex: acceleration

Description
Via this object, the acceleration of the rotational movement of the encoder shaft can be read out. The output format of the acceleration depends on the setting of object 2003_{hex} (format for speed and acceleration).

9.3.1.7 Object 2006 hex: change of state

Description
This object indicates the size of the change of the position value at which a PDO is transmitted. For this purpose, the “transmission type” of the PDO must be set to 254 and the “cycle timer” (object 6200_{hex}) must be set to 0.

Explanation

The “transmission type” 254 means that for each change in the position value, a PDO will be transmitted. In the event of a continuous position change, this leads to a high loading on the CAN bus. In order to reduce this bus load, the difference above which a PDO will be transmitted can be set via the object “change of state”.

10. Function of the cams

The encoder has a maximum of eight cams which, on the software side, can be controlled via the CAN bus.

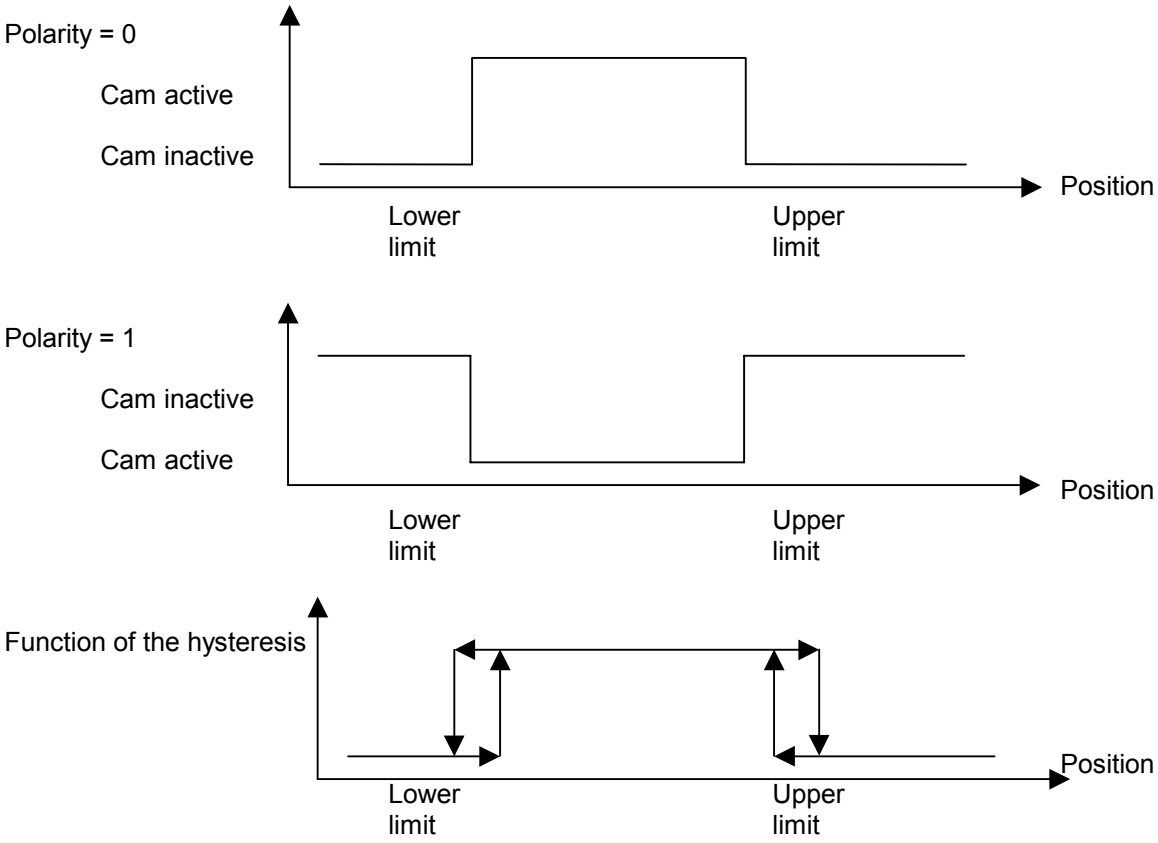
The following parameters can be set individually for each cam:

- Lower limit, upper limit and hysteresis
- Polarity, enable

The following restrictions must be complied with in the case of the cams:

- The lower limit must be smaller than the upper limit. If this is not the case, the greater value will be used as the upper limit and the smaller value as the lower limit.
- The hysteresis must be smaller than the lower limit. If this is not the case, the hysteresis is set to the value of the lower limit.

Illustration of the functionality of the cams



11. Technical description

11.1 General

The ATM60-Cxx absolute encoders have an overall resolution of 26 bits as a multi-turn. The resolution per revolution is 13 bits, the number of revolutions is likewise 13 bits. The revolutions are determined by a gear mechanism. The bus interface is located in the encoder and is an access circuit for the CAN bus with the CANopen protocol. The communications profile integrated is the CiA DS 301, and the device profile integrated is the CiA DSP 406 for encoders.

11.2 Features

The features of the ATM60-Cxx in summary:

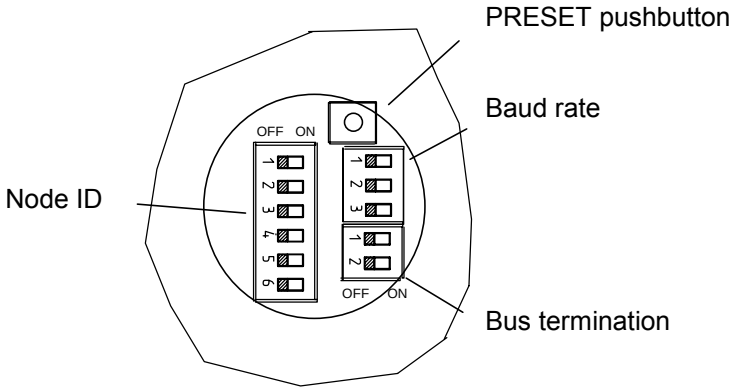
- Robust magnetic pick-up system.
- 26-bit a multi-turn absolute multi-turn rotary encoder.
- Basic resolution 8192 steps per revolution, 8192 revolutions.
- CAN high speed bus access coupling (ISO 11898).
- DC isolation of the bus interface via optocouplers.
- CANopen communications profile DS 301.
- Device profile DSP 406 Class 2.
- CAN controller status LED.

11.3 Technical data

Technical data and characteristics	Clamp and servo flange	Hollow-shaft design	Units
Dimensions	see drawing		
Mass	about 590		g
Moment of inertia of the rotor	35	55	gcm^2
Measuring step	0.043		degrees
Max. number of steps per revolution	8192		
Max. number of revolutions	8192		
Error limits	±0.25		degrees
Repeatability	0.1		degrees
Operating speed	6000	3000	min^{-1}
Position-forming time	0.15		ms
Max. angular acceleration	5×10^5		red/s^2
Operating torque with shaft sealing ring	1.8	0.8	Ncm
Operating torque without shaft sealing ring	0.3	-	Ncm
Starting torque with shaft sealing ring	2.5	1.2	Ncm
Starting torque without shaft sealing ring	0.5	-	Ncm
Permissible shaft loading radial/axial	300/50	-	N
Permissible shaft movement of the drive element			
Radial static/dynamic	-	±0.3/±0.1	mm
Axial static/dynamic	-	±0.5/±0.2	mm
Bearing service life	3.6×10^9		revolutions
Working temperature range	-20 ... +85		°C
Operating temperature range	-40 ... +100		°C
Storage temperature range	-40 ... +100		°C
Permissible relative humidity Condensation permissible (on request)	98		%
EMC to EN 50081 Part 2 and EN 50082 Part 2			
Shock resistance (DIN IEC 68 Parts 2-27)	100/ (6 ms)		g
Vibration resistance (DIN IEC 68 Parts 2-6)	20/ (10 ... 2000 Hz)		g
Protection class to DIN VDE 0470 Part 1			
with shaft sealing ring	IP 67	IP 67	
without shaft sealing ring, not sealed at flange	IP 43	IP 43	
without shaft sealing ring, sealed at flange	IP 66	-	
Operating voltage range	10 ... 32		V
Max. power consumption	2.0		W
Initialization time	1250		ms
CANopen bus interface			
Electrical interface	To ISO 11898, CAN specification 2.0 B, DC isolated		
Protocol	Communications profile DS 301 Device profile DSP 406 Class 2		
Address setting, node ID	via DIP switches or CANopen		
Data transfer rate, Baud rate	10k, 20k, 50k, 125k, 250k, 500k, 1M via DIP switches or CANopen		
Electronic adjustment (number SET)	via SET pushbutton or CANopen		
Bus termination	via DIP switch or external		

12. Connecting the encoder

Specific features of the ATM60-Cxx may be configured using setting switches on the hardware. These are the node ID, the baud rate and the terminating resistance for the CAN network. In addition, by operating a pushbutton, the encoder can be set to a defined value (PRESET function). In order to implement these actions, the PG plug on the rear of the encoder has to be removed. Underneath it, three DIP switches and one micro pushbutton become visible.



12.1 Setting the node ID

The node ID can be set via the 6-pole DIP switch. Ex works, it is set to 1. The coding of the DIP switch is binary, beginning with DIP switch 1 = 2^0 up to DIP switch 6 = 2^5.

DIP 6 2^5	DIP 5 2^4	DIP 4 2^3	DIP 3 2^2	DIP 2 2^1	DIP 1 2^0	Address
0	0	0	0	0	0	0
0	0	0	0	0	1	1
...	...	...	...	...	...	...
1	1	1	1	1	1	63

0 = DIP switch OFF,
1 = DIP switch ON.

Node ID 0 may not be set, since this is reserved in a CANopen system.

The setting of the node ID to 0 will be converted internally into node ID 1.

A new node ID is accepted at a power-on of the encoder.

12.2 Setting the baud rate

The baud rate can be set by using the three-pole DIP switch.

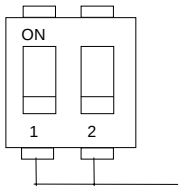
DIP 3	DIP 2	DIP 1	Baud rate
0	0	0	10 kBaud
0	0	1	20 kBaud
0	1	0	50 kBaud
0	1	1	125 kBaud
1	0	0	250 kBaud
1	0	1	500 kBaud
1	1	0	1 MBaud
1	1	1	20 kBaud

0 = DIP switch OFF
1 = DIP switch ON

A new baud rate is accepted at a power-on of the encoder.

12.3 Setting the bus termination

Using the two-pole DIP switch, a bus terminating resistance network can be connected in. For this purpose, both slide switches have to be switched to the “ON” position. The standard ex-works setting is “OFF”.



Bus termination
1 and 2 OFF: not active
1 and 2 ON: active

12.4 Preset function

When the PRESET function is executed (by pushbutton or CANopen), the ATM60-Cxx is set to a specific, predefined value, which then applies to the position currently occupied. This predefined value is either already stored in the Eeprom or it is changed by the object 6003 (preset value) and stored permanently in an Eeprom. The PRESET value preset at the works is 0.

The PRESET function was not conceived for dynamic configuration operations but as an electronic adjustment function for commissioning, in order to assign a specific value to the mechanical angular position of the ATM60-Cxx.

When the PRESET value is set via CANopen, the value must lie within the currently set total working range (steps per revolution x number of revolutions).

It should be noted that the number of write cycles of the Eeprom is restricted to 1 million cycles. This means that the memory location can become defective if 1 million write cycles are exceeded.

12.5 Screening

In an interference-free environment, unscreened cable is permitted. The following reasons support the use of screened cable in every case:

- An interference-free space exists at best in the interior of screening switch cabinets. However, as soon as relays are also located in the cabinet, freedom from interference is no longer ensured.
- The use of unscreened cable demands additional protective measures on the bus signal inputs against overvoltages.

For this reason, it is fundamentally recommended to use screened leads for the bus cable. This recommendation also extends to any supply cables which may be needed for external voltage supplies.

Double-screened leads are particularly suitable for environments subjected to high EMC. In order to ensure optimal protection, in this case the outer braided screen and the inner foil screen at both cable ends should be connected to protective earth over a large area, using an earthing clamp.

12.5.1 Screening rules

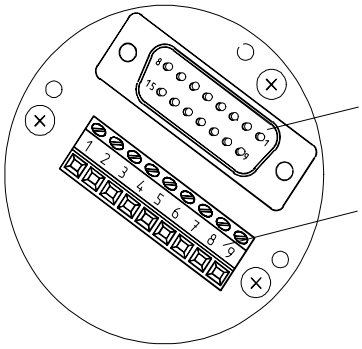
When a screened bus cable is used, we recommend that the screen be connected to the protective earth at both ends with low inductance. This achieves the most optimal EMC. One exception relates to isolated potentials (e.g. in refineries); here, as a rule only earthing on one side is permitted.

In the sense of optimal EMC, the connection between the cable screen and the protective earth is carried out via a metallic device casing and the screw connection of the circular plug connector or PG screw fittings.

12.6 Pin allocation

12.6.1 Bus connector with PG screw fittings

In order to connect the bus lines and the supply, the bus connector must be removed. For this purpose, the three screws on the rear of the bus connector have to be unscrewed. The terminal strip described below becomes accessible.

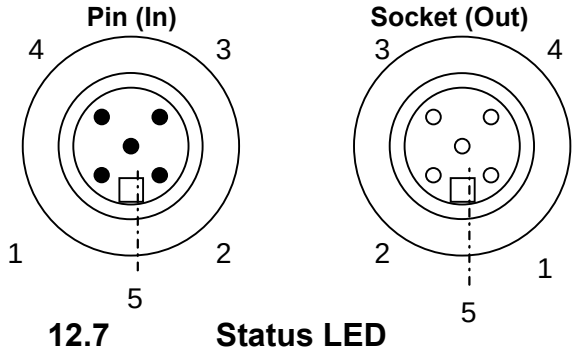


9-way terminal strip	
Signal	Explanation
US	Supply voltage 10 .. 32V
GND	Earth
CAN_L	CAN bus signal low
CAN_H	CAN bus signal high
CAN_L	CAN bus signal low
CAN_H	CAN bus signal high
US	Supply voltage 10 .. 32V
GND	Earth
Shield	Screen connection

After the leads have been connected, the bus connector should be mounted on the encoder housing again, attention having to be paid to the correct seat of the seal (O-ring on the encoder housing and sealing washers under the housing screws).

12.6.2 Bus connector with 5-pole micro circular plug connectors

Bus connectors of the type BC 6 SRx CO have M 12 circular plug connectors (5-pin micro style). In the case of this device type, it is not necessary for the bus connector to be unscrewed completely from the complete device in order to connect the bus and supply lines. The connection is made directly via the device connectors.



5-pole micro circular plug connector	
Pin	Meaning
1	Screen connection
2	Supply voltage 10 .. 32V
3	Earth
4	CAN bus signal high
5	CAN bus signal low

12.7 Status LED

In the bus connector of the ATM60-Cxx there is a bicolour LED (red/green) which reproduces the state of the CAN controller.

LED	Status	Meaning
Off	no power	No supply voltage applied to the encoder.
Green	Error Active	No error has occurred in the CAN controller
Flashing green	Error Passive	CAN controller has registered a transmission error - receive error counter >=96 or - transmit error counter >=96
Red	Bus off	CAN controller has registered a transmission error and isolated the connection from the bus - transmit error counter > 255

13. Commissioning

Before the encoder is switched on, the following settings must have been made:

- Node ID
- Baud rate
- Bus terminating resistors

The procedure for these settings is described in the chapter on “Connecting the encoder”.

13.1 Switch on encoder

After applying the supply voltage, the encoder runs through an initialization phase and then changes automatically to the “pre-operational” state.

Within the initialization phase, data are loaded from the Eeprom into the RAM (non-volatile memory) of the encoder, and the objects in the object directory are initialized.

If no data have been saved to the Eeprom, default values are stored in the Eeprom. If data have been saved to the Eeprom, then the objects in the object directory are set to the saved values.

In the “pre-operational” state, only SDO connections are active. The PDO connections are inactive and are activated in the “operational” state.

In order to change from the “pre-operational” state to the “operational” state, it is merely necessary for the “start remote node” command to be transmitted to the encoder. If no configuration of objects in the encoder is needed, then the encoder can be set ready to operate by transmitting this command.

“Start remote node” command:

COB-ID	Command	Node ID		Data				
11 bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0	1_{hex}	Node ID of the encoder		00_{hex}	00_{hex}	10_{hex}	00_{hex}	00_{hex}

13.2 Configure encoder

If any configuration of objects in the encoder is needed, it is recommended to carry out this configuration in the “pre-operational” state. In this state, only the SDO connections are active, PDO connections become active only in the “operational” state. This means that the bus activity of the encoder is low in the “pre-operational” state.

13.2.1 Example of PDO mapping

The data which are transmitted in a PDO connection are determined by what is known as PDO mapping (see object $1A00_{hex}$ “Transmit PDO 1, mapping” and $1A02_{hex}$ “Transmit PDO2, mapping”). Within one PDO, a maximum of 8 bytes of data can be transmitted.

The position value is always contained in sub-index 1. Sub-index 1 cannot be altered. This means that, in PDO 1 and PDO 2, the position value is always transmitted. This means that a maximum of 4 bytes of a PDO can be filled up with objects which are permitted for PDO mapping.

In the following example, the position value (object 6003_{hex}) and the status register of the cams (object 6300_{hex}) are transmitted in the Transmit PDO 1. For this purpose, the following entries in the following sequence have to be made in the mapping of the Transmit PDO 1:

- 1. Set sub-index 0 to 0:
- 2. Enter 63000108_{hex} under sub-index 2
- 3. Set sub-index 0 to 2. This means that the sub-indices 1 and 2 will be taken into account in the composition of the PDO mapping. In this case, a check on the PDO mapping over a total length of 8 bytes is made.

13.2.2 Example of changing the object “Measuring steps per revolution”

The measuring steps per revolution are to be changed to 4096 steps.

The master sends the following Initiate Download Request to the encoder.

SDO-ID	Command	Index		Sub-index	4-byte data			
11 bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
COB-ID for SDO(rx)	22 hex	60 hex	01 hex	00 hex	00 hex	10 hex	00 hex	00 hex

The encoder replies with the following Initiate Download Response to the Master:

SDO-ID	Command	Index		Sub-index	4-byte data			
11 bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
COB-ID for SDO(tx)	60_{hex}	60 hex	01 hex	00 hex	reserved			

13.2.3 Write all objects into EEPROM

Objects which have been changed via an SDO connection are saved in the RAM (volatile memory of the encoder). In order to save these objects in the Eeprom (non-volatile memory) of the encoder, the command “save” must be written to the sub-index 1 of object 1010_{hex} (save parameters). (See also object 1010_{hex}, save parameters).

The Master sends the following Initiate Download Request to the encoder.

SDO-ID	Command	Index		Sub-index	4-byte data			
11 bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
COB-ID for SDO(rx)	22 hex	10 hex	10 hex	01 hex	73 hex ASCII s	61 hex ASCII a	76 hex ASCII v	65 hex ASCII e

The encoder replies with the following Initiate Download Response to the Master:

SDO-ID	Command	Index		Sub-index	4-byte data			
11 bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
COB-ID for SDO(tx)	60_{hex}	10 hex	10 hex	01 hex	reserved			

13.2.4 Load all objects with default values

In order to set all the objects to their default values, the command “load” must be written to sub-index 1 of object 1011_{hex} (see also object 1011_{hex}, load default parameters).

The Master sends the following Initiate Download Request to the encoder.

SDO-ID	Command	Index		Sub-index	4-byte data			
11 bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
COB-ID for SDO(rx)	22 $_{hex}$	10 $_{hex}$	11 $_{hex}$	01 $_{hex}$	6C $_{hex}$ ASCII l	6F $_{hex}$ ASCII o	61 $_{hex}$ ASCII a	64 $_{hex}$ ASCII d

The encoder replies with the following Initiate Download Response to the Master:

SDO-ID	Command	Index		Sub-index	4-byte data			
11 bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
COB-ID for SDO(tx)	60_{hex}	10 $_{hex}$	10 $_{hex}$	01 $_{hex}$	reserved			

With the “load” command, the default values for the respective objects are saved in the RAM (volatile memory) of the encoder. This means that after the encoder has been switched off and on, these default values are lost again. In order to save the default values in the Eeprom (non-volatile memory) of the encoder, it is additionally necessary for the “save” command to be written to sub-index 1 of object 1010_{hex}.

13.3 EDS file

The EDS file is the electronic description of the object directory of the ATM60-Cxx. The commissioning instructions represent the description of the object directory in written form.

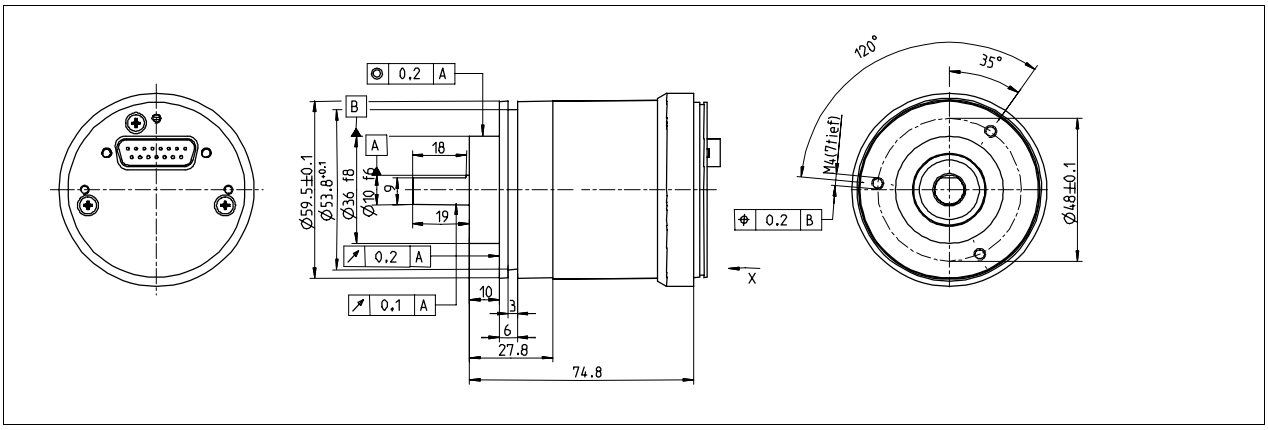
The description of the object directory is needed for the configuration and operation of the ATM60-Cxx.

The EDS file is read in by a configuration tool. Using the entries in the EDS file, the configuration tool provides the user with all the available objects in this device.

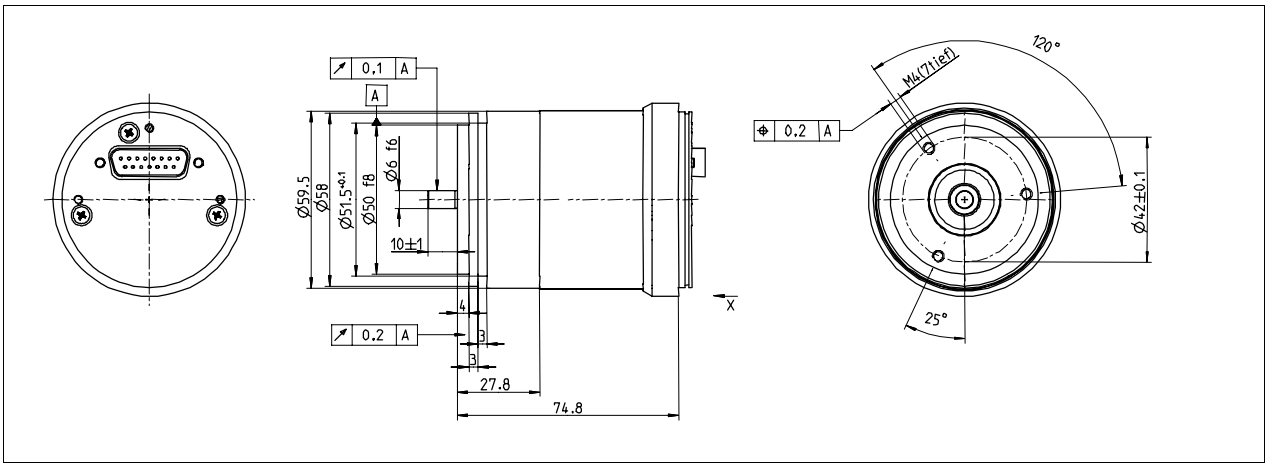
As a result, a device can be configured easily without any prior knowledge of its object directory.

14. Dimensional drawing

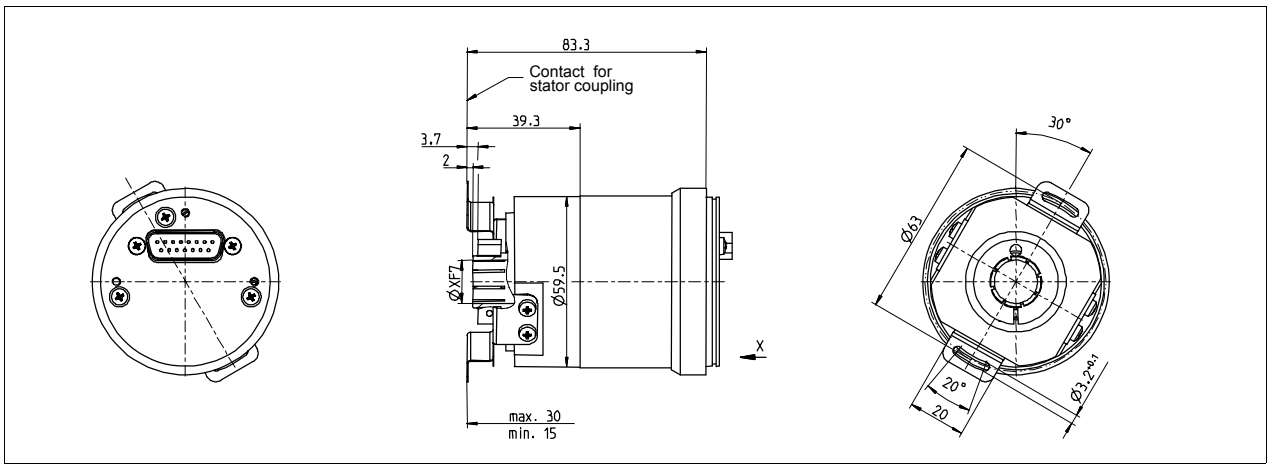
ATM60-C4H13X13 (with face mount flange)



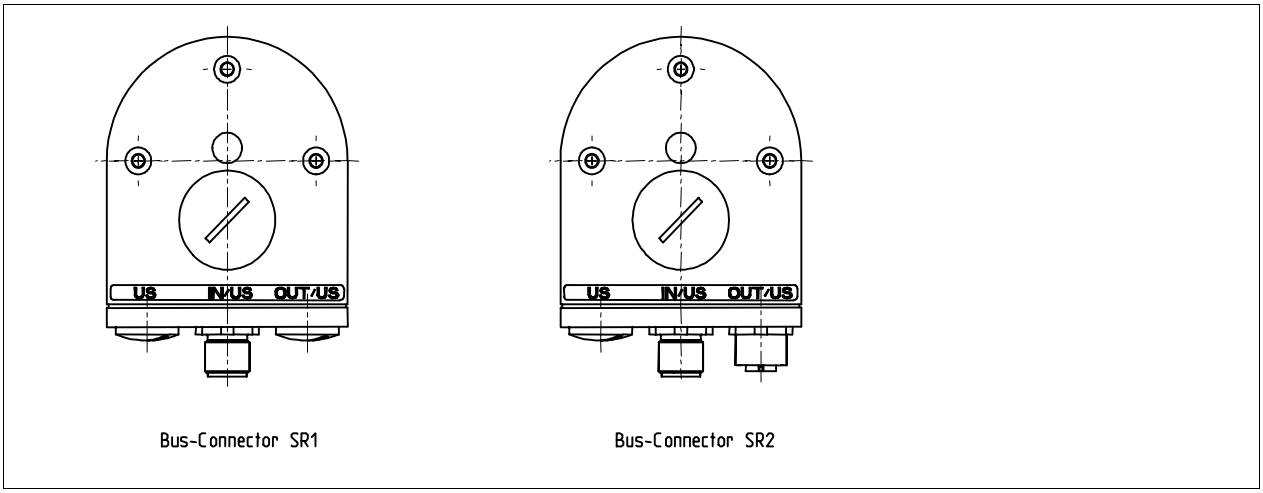
ATM60-C1H13X13 (with servo flange)



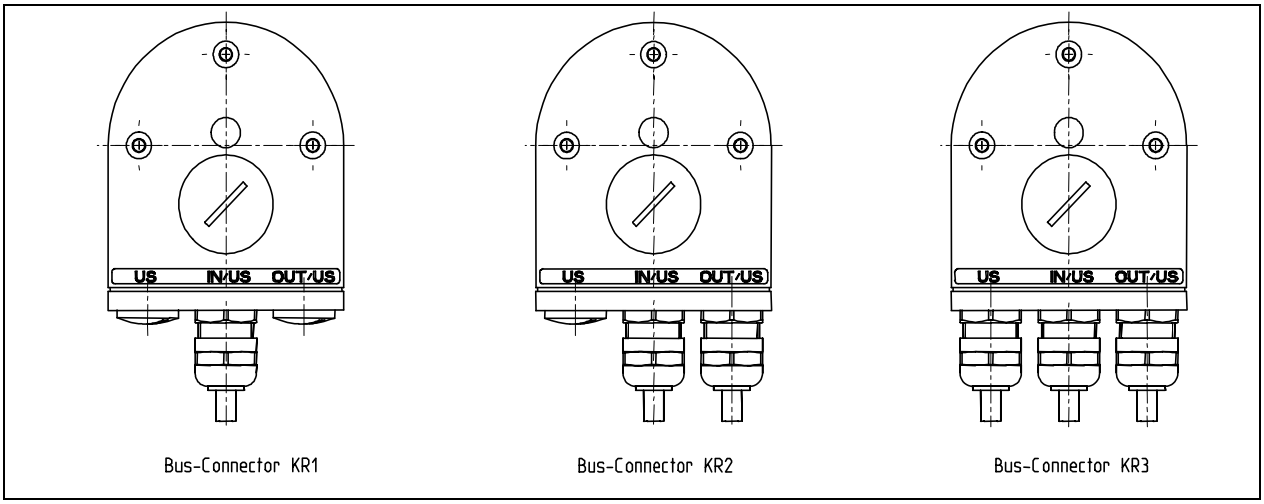
ATM60-CAH13X13 (with blind hollow shaft)



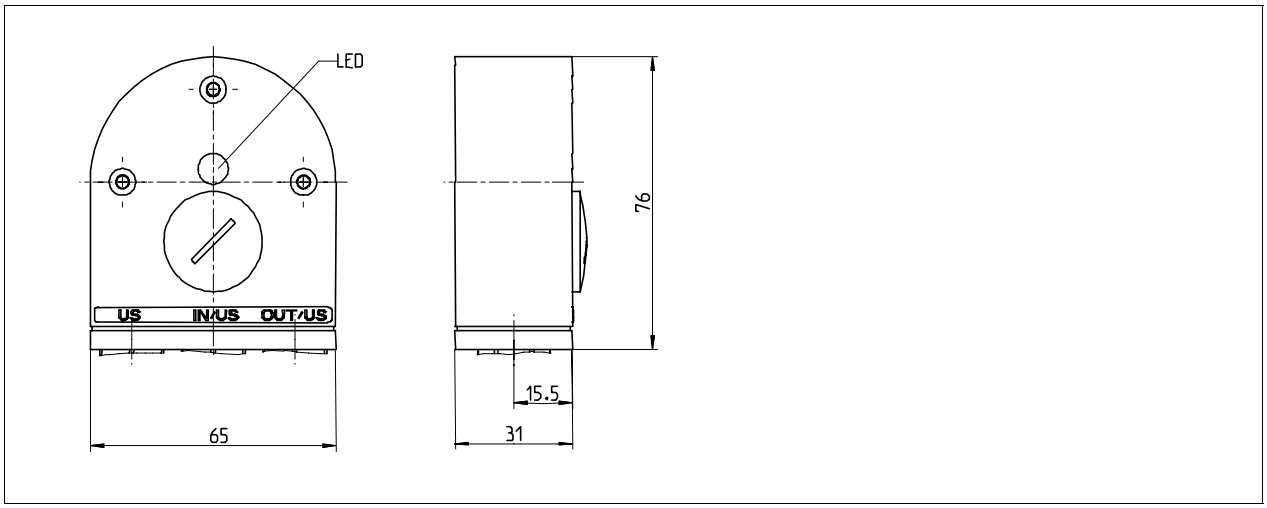
Bus connector BC 6 with 5-pole micro circular plug-in connectors



Bus connector BC 6 with PG for cable connection



General view for bus connector



15. Scope of supply

ATM60-Cxx without cable or mating connector.
Floppy disk with the EDS file. This file contains device-specific features of the ATM60-Cxx and can be read by a configuration tool.
Commissioning instructions for ATM60-Cxx.

Note

Further literature:

CANopen - Communications profile DS 301 V 4.0
CANopen - Device profile DS 406 V 2.0, profile for encoders

Addresses:

CAN in Automation (CiA)
Am Weichselgarten 26
D 91058 Erlangen
Tel.: (49) 9131 - 6 90 86-0
Fax: (49) 9131 - 6 90 86-79
Web: http://www.can-cia.de

16. Ordering information

Encoder without bus connector

Ordering designation	Explanation
ATM60-C4H13X13 (Item n°1030024)	Face mount flange (shaft ∅ 10 mm)
ATM60-C1H13X13 (Item n°1030025)	Servo flange (shaft ∅ 6 mm)
ATM60-CAH13X13 (Item n°1030026)	Blind hollow shaft (different Ø via inserts)

Shaft inserts for hollow shaft

The shaft inserts for the blind hollow shaft must be specified separately in the order. The following diameters are available:
6 mm, ¼", 8 mm, 3/8", 10 mm, 12 mm, ½", 15 mm (no shaft insert necessary)

Bus connector

Ordering designation	Explanation
AD-ATM60-SR1CO (Item n°2021686)	Radial, with 1x M12 (5-pin), pin
AD-ATM60-SR2CO (Item n°2020935)	Radial, with 2x M12 (5-pin), pin, socket
AD-ATM60-KR1CO (Item n°2029230)	Radial, cable outlet with 1 PG
AD-ATM60-KR2CO (Item n°2029231)	Radial, cable outlet with 2 PG
AD-ATM60-KR3CO (Item n°2029232)	Radial, cable outlet with 3 PG

Ordering example

ATM60-CAH13X13 :	ATM60 with blind hollow shaft and CANopen interface
SPZ-010-AD-A:	Shaft insert for blind hollow shaft, 10 mm
AD-ATM60-KR3CO:	Bus connector, radial, cable outlet with 3 PG for CANopen interface.

17. Notes

2021-12-16 · LF_07 · A4 4c int49

Australia
Phone +61 (3) 9457 0600
1800 33 48 02 – tollfree
E-Mail sales@sick.com.au

Austria
Phone +43 (0) 2236 62288-0
E-Mail office@sick.at

Belgium/Luxembourg
Phone +32 (0) 2 466 55 66
E-Mail info@sick.be

Brazil
Phone +55 11 3215-4900
E-Mail comercial@sick.com.br

Canada
Phone +1 905.771.1444
E-Mail cs.canada@sick.com

Czech Republic
Phone +420 234 719 500
E-Mail sick@sick.cz

Chile
Phone +56 (2) 2274 7430
E-Mail chile@sick.com

China
Phone +86 20 2882 3600
E-Mail info.china@sick.net.cn

Denmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Finland
Phone +358-9-25 15 800
E-Mail sick@sick.fi

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Germany
Phone +49 (0) 2 11 53 010
E-Mail info@sick.de

Greece
Phone +30 210 6825100
E-Mail office@sick.com.gr

Hong Kong
Phone +852 2153 6300
E-Mail ghk@sick.com.hk

Hungary
Phone +36 1 371 2680
E-Mail ertekesites@sick.hu

India
Phone +91-22-6119 8900
E-Mail info@sick-india.com

Israel
Phone +972 97110 11
E-Mail info@sick-sensors.com

Italy
Phone +39 02 27 43 41
E-Mail info@sick.it

Japan
Phone +81 3 5309 2112
E-Mail support@sick.jp

Malaysia
Phone +603-8080 7425
E-Mail enquiry.my@sick.com

Mexico
Phone +52 (472) 748 9451
E-Mail mexico@sick.com

Netherlands
Phone +31 (0) 30 229 25 44
E-Mail info@sick.nl

New Zealand
Phone +64 9 415 0459
0800 222 278 – tollfree
E-Mail sales@sick.co.nz

Norway
Phone +47 67 81 50 00
E-Mail sick@sick.no

Poland
Phone +48 22 539 41 00
E-Mail info@sick.pl

Romania
Phone +40 356-17 11 20
E-Mail office@sick.ro

Russia
Phone +7 495 283 09 90
E-Mail info@sick.ru

Singapore
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Slovakia
Phone +421 482 901 201
E-Mail mail@sick-sk.sk

Slovenia
Phone +386 591 78849
E-Mail office@sick.si

South Africa
Phone +27 10 060 0550
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321/4
E-Mail infokorea@sick.com

Spain
Phone +34 93 480 31 00
E-Mail info@sick.es

Sweden
Phone +46 10 110 10 00
E-Mail info@sick.se

Switzerland
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Taiwan
Phone +886-2-2375-6288
E-Mail sales@sick.com.tw

Thailand
Phone +66 2 645 0009
E-Mail marcom.th@sick.com

Turkey
Phone +90 (216) 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates
Phone +971 (0) 4 88 65 878
E-Mail contact@sick.ae

United Kingdom
Phone +44 (0)17278 31121
E-Mail info@sick.co.uk

USA
Phone +1 800.325.7425
E-Mail info@sick.com

Vietnam
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Detailed addresses and further locations at **www.sick.com**

SICK
Sensor Intelligence.