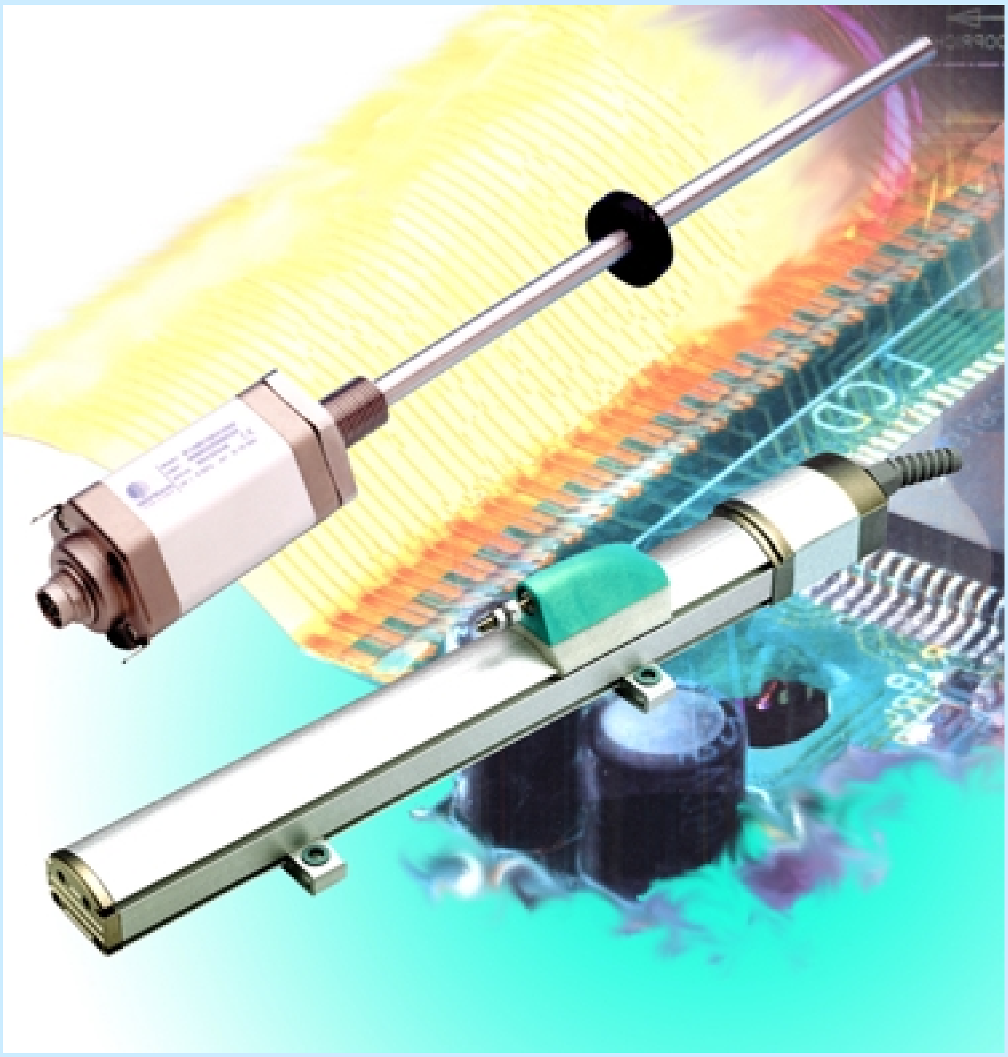
CANopen



Ver. 02.05

The MAB (MAZ) position transducer is a slave device of a CAN BUS network, having the functionality complying with CANopen standard protocol, provided by C.i.A. (Can in Automation) and proposed by "CANopen Application Layer and Communication Profile DS 301 v. 4.0" document of 16 – 06 – 1999 and by other documents as mentioned below.

To allow the application with less recent CANopen networks, the compatibility with "CANopen Communication Profile DS 301 v. 3.0" document, as well as with CAN Application Layer (CAL) DS 201 documents is ensured.

Being the MAB (MAZ) transducer similar to an encoder, it fully complies with the specific Device Profile Standard DS 406 v. 2.0 of 11–05–1998.

This document describes the functions of the CANopen standard available in the device. It is addressed to system integrators of CAN networks and to developers of CANopen devices, who already know the content of the above mentioned standards provided by CiA. The detailed analysis of what defined by the CANopen is not one of the purposes of this text.

1.1 Definitions

CAN: Controller Area Network.

It describes a serial communication bus that implements layer 1 , "physical", and layer 2, "data link", of ISO/OSI reference model.

CAL: CAN Application Layer. It describes CAN implementation in layer 7, "application" of ISO/OSI reference model, from which the CAN open derives.

CMS: CAN Message Specification. CAL service element. It defines the CAN Application Layer for the different industrial applications.

COB: Communication Object. Data transfer unit in a CAN network (a CAN message). A max. of 2048 COB is available in a CAN network; each COB can carry from 0 up to 8 bytes max.

COB-ID: COB Identifier. The identifier that distinguishes and determines a COB priority in case of concurrence of messages in the network.

D0 - D7: Data from 0 to 7. Data range bytes of a CAN message.

DBT: Distributor Service Specification. CAL service element. It defines the identifier dynamic distribution.

DLC: Data Length code. Number of bytes transmitted in a single frame.

ISO: International Standard Organization. International body that provides standard rules in different commodity sectors.

LMT: Layer Management. CAL service element that allows to set the communication parameters of a local module.

LSS: Layer Service Setting.

Service element allowing to set the communication parameters of a CANopen module.

NMT: Network Management.

CAL service element. It describes how to configure, initialise, manage errors, control and enable to work network devices.

OSI: Open Systems Interconnection. Adopted as an ISO rule, it describes, through 5 functional "layers", the characteristics that data communication systems must have to be defined "open".

PDO: Process Data Object. Process data communication objects (with high priority).

RTR: Remote Transmit Request. Data transmission request message, sent to a remote device.

RXPDO: Receive PDO. PDO objects received by the remote device.

RXSDO: Receive SDO. SDO objects received by the remote device.

SDO: Service Data Object. Communication objects of service data having low priority. The value of these data is contained in the "Dictionary of Objects" of every device in the CAN network.

TXPDO: Transmit PDO. PDO objects transmitted by the remote device.

TXSDO: Transmit SDO. SDO objects transmitted by the remote device.

N.B.: In digital communications, numbers are transferred via a binary codification and it is often useful to express values in hexadecimal radix. For this reason, **if not otherwise specified, the value contained in a message is always in hexadecimal radix.** However, in the following tables, an explanatory suffix has been added to each digit, where possible. "hex" ("h") suffix = hexadecimal value; "bin" ("b") suffix = binary value; "dec" ("d") suffix = decimal value.

1.2 COMMINICATION OBJECT

CANopen Data Protocol

SOF	Arbitration		Control	Data Field	CRC	ACK			EOF	Interframe Space
1	11	1	6	0-8 Bytes	15	1	1	1	7	$\geq$ 3Bits

DATA FIELD: Byte zero is transmitted first, byte seven is transmitted last. Within a byte, bit zero is the least significant bit, bit seven is the most significant bit.

The terms 'LSB' and 'MSB' stand for 'least significant byte' and 'most significant byte' respectively and are used to define how an integer number is represented in more than one byte for the LSS Protocol. The order of significance is increasing from LSB to MSB.

Data communication objects in a CAN network are described by means of "services" and "protocols" expressed by the C.A.L.; each of the 2048 "COB" has got a peculiar meaning to allow to exchange data between devices of different types.

To make the configuration of a network easier, you have to use a table of pre-defined objects (Pre-Defined Connection Set) where every 11 bits the COB-ID is divided into 4 bits of "FUNCTION CODE" (which identifies the peculiar message) and 7 bits of "NODE-ID" (which univocally identifies the involved node).

When "COB-ID" equals 0, the message is "broadcasted", which means sent to all the devices connected to the CAN network.

bit	10	9	8	7	6	5	4	3	2	1	0
COB - ID	X	X	X	X	X	X	X	X	X	X	X
	FUNCTION CODE				NODE - ID						

OBJECT	FUNCTION CODE (bin)	COB – ID (hex)	(dec)
NMT	0000	000	0
SYNC	0001	080	128
EMERGENCY	0001	081 – 0FF	129 – 255
TIME STAMP	0010	100	256
TXPD01	0011	181 – 1FF	385 – 511
RXPD01	0100	201 – 27F	513 – 639
TXPD02	0101	281 – 2FF	641 – 767
RXPD02	0110	301 – 37F	769 – 895
TXPD03	0111	381 – 3FF	897 – 1023
RXPD03	1000	401 – 47F	1025 – 1151
TXPD04	1001	481 – 4FF	1153 – 1279
RXPD04	1001	501 – 57F	1281 – 1407
TXSD0	1011	581 – 5FF	1409 – 1535
RXSD0	1100	601 – 67F	1537 – 1663
NODE GUARDING	1110	701 – 77F	1793 – 1919
LMT - LSS	1111	7E4 - 7E5	2020 - 2021

The CAN communication bus is a multimaster type, so that all messages can be seen by any node.

In the definition made by the CAL, some messages, such as LMT, LSS and NMT, are typically "broadcasted", that is, they are addressed to all devices connected to the CAN network. If, on the contrary, you want to make a selection, you can send them to a single device, entering its specific address in the message data blank field.

COB-IDs related to "SYNC", "TIME STAMP", "EMERGENCY" , "TXPDOn" and "RXPDOn" objects can be modified through the SDO objects in the Dictionary of Objects "Communication Profile" of the Slave node (Index 1005hex, 1012hex, 1014hex, 1800hex and 1400hex, respectively).

1.2.1 NMT (Network Management Object)

To configure the CAN bus, which is typically a Multimaster, a Master/Slave communication has to be defined.

Each Slave device is identified by the NODE-ID (from 1 up to 127). The network Master, through the NMT services, can initialize, control and enable Slave devices.

Provided operating statuses:

. INITIALIZATION

. PREOPERATIONAL

. OPERATIONAL

. STOPPED

The type of message communicated by the slave node depends on its operating status, as shown in the following table:

MESSAGE	OPERATING STATUS			
	INITIALIZATION	PREOPERATIONAL	OPERATIONAL	STOPPED
PDO			X	
SDO		X	X	
SYNC		X	X	
TIME STAMP		X	X	
EMERGENCY		X	X	
NMT		X	X	X
NODE GUARDING		X	X	X
LMT- LSS		X	X	X
BOOT - UP	X			

In **"INITIALIZATION"** status the slave device can be controlled in three different ways, to which the following sub-status corresponds (see the diagram at page 5).

• **INIT** = Condition that the device takes upon Power-on or as a reply to a hardware reset command. In this phase, the hardware devices available in the module are activated and the operating variables stored on EEPROM are loaded on RAM. All the writing operations on "Objects of the Dictionary" are made on a RAM. For their permanent storage in EEPROM you have to use the saving commands specified at 1010hex Index of the Communication Profile. Once the **INIT** phase is over, the node sends a **"BOOT-UP"** message to the Master and automatically sets on **"PREOPERATIONAL"** mode**.**

2) RESET COMMUNICATION = Status that the node takes in reply to a Master message. The command to transfer data related to the Communication Profile, from EEPROM into RAM is activated. At the end of this operation, the node sends a **"BOOT-UP"** message to the Master and automatically sets on **"PREOPERATIONAL"** status**.**

3) RESET APPLICATION = Status that the node takes in reply to a Master message. The command to transfer data, related to the "Device Profile" and to the "Manufacturer Profile", from EEPROM into RAM is activated. At the end of this operation, the node automatically sets on **"RESET COMMUNICATION"** status.

In **"PREOPERATIONAL"** status, the communication with the slave node can only occur either through the SDO objects, or the NMT and NODE GUARDING network services, to allow the Master to configure the Slave variables.

In **"OPERATIONAL"** status, all communications are active and the slave node is completely operating.

In **"STOPPED"** status, the communication with the slave node can occur only through NMT and NODE GUARDING network services.

To modify the operating status of a single slave node or of all of them at the same time, the Master sends the following message to the CAN network:

MASTER → **SLAVE**

COB -ID	RTR	DLC	D0	D1	D2	D3	D4	D5	D6	D7
0000	0	2	CS	ID	X	X	X	X	X	X

CS, "Command Specifier", is the code corresponding to the operating status. **ID** is NODE-ID of the slave device to which the message is addressed; **if ID = 0,** all the slaves have to take the operating status specified in **CS** according to the following codes:

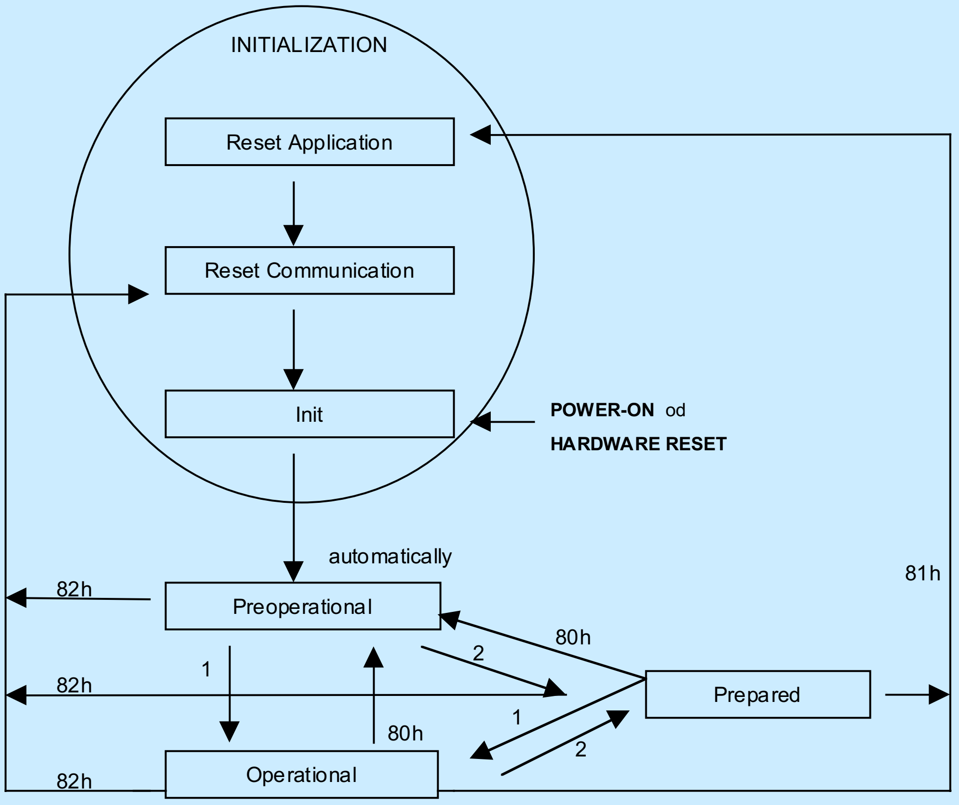
CS=1	**OPERATIONAL**
CS=2	**STOPPED**
CS=80h (128d)	**PREOPERATIONAL**
CS=81h (129d)	**RESET APPLICATION**
CS=82h (130d)	**RESET COMMUNICATION**

The **"BOOT-UP"** is activated only once, by the slave node, at the end of the INITIALIZATION status to signal to the Master the transition to the PREOPERATIONAL status and the pre-arrangement to communication. The message format is as follows:

MASTER → **SLAVE**

COB -ID	RTR	DLC	D0	D1	D2	D3	D4	D5	D6	D7
700h + NODE-ID (1792d+NODE-ID)	0	1	0	X	X	X	X	X	X	X

The operating statuses that the node can take are listed in the following diagram:



1.2.2 NODE GUARDING and HEARTBEAT

The **"NODE GUARDING"** protocol is used by the Master of the network to continuously check the correct working of the Slave nodes. The Master cyclically sends the following message:

MASTER → SLAVE

COB - ID	RTR	DLC	D0	D1	D2	D3	D4	D5	D6	D7
700+NODE (1792d +NODE-ID)	1	0	X	X	X	X	X	X	X	X

The queried Slave node has to reply with**:**

MASTER ← SLAVE

COB - ID	RTR	DLC	D0	D1	D2	D3	D4	D5	D6	D7
700+NODE-ID (1792d +NODE-ID)	0	2	t	s	x	x	x	x	x	X

"**t**" =TOGGLE BIT. It continuously changes the value from 0 to 1 at every NODE GUARDING request by the Master. With "**s**" the Slave node shows its own operating status by means of the following codes:

s= 0	**BOOT-UP**
s= 4	**STOPPED**
s= 5	**OPERATIONAL**
s= 127	**PREOPERATIONAL**

The interval between two consecutive NODE GUARDING queries made by the Master is called **"GUARD TIME".** It is expressed in milliseconds and, through the SDO, it can be stored by the Master in the dictionary of objects, Communication Profile, of the Slave node (Index 100C h).

In the dictionary there is also the "**LIFE TIME FACTOR** " (Index 100D hex), a number that, multiplied by the GUARD TIME, specifies the "**NODE LIFE TIME**", that is the max. time allowed to update the NODE GUARDING.

NODE LIFE TIME [msec] = LIFE TIME FACTOR * GUARD TIME [msec]

When the LIFE TIME FACTOR is other than 0, the slave node verifies if, inside the NODE LIFE TIME interval, the Master has sent a NODE GUARDING request. If not, the Slave node automatically takes the PREOPERATIONAL status.

As alternative to the NODE GUARDING protocol by the Master, the slave node can cyclically send a message called "**HEARTBEAT**" to the Master to indicate its correct way of working, as well as its operating status. The time interval between two consecutive transmissions can be stored by the Master in the dictionary of objects, Communication Profile, of the Slave node (Index 1017 hex). The message format of the Heartheat protocol is the following one:

MASTER ← SLAVE

COB - ID	RTR	DLC	D0	D1	D2	D3	D4	D5	D6	D7
700+NODE-ID (1792d+NODE-ID)	0	1	s	x	x	x	x	x	x	X

"**s**" specifies the operating status of the Slave node. If the HEARTBEAT time is 0, the protocol is not active. In the other cases, the cyclic transmission of the operating condition will start immediately after the BOOT-UP message.

NODE GUARDING and HEARTBEAT protocols cannot be present at the same time.

1.2.3 SYNC (Synchronisation Object)

The **"SYNC"** protocol is used by the Master to synchronize the transmission of PDO messages (TXPDO and RXPDO) sent by the nodes.

The **SYNC** is activated through a broadcast message and has the following format:

MASTER → SLAVE

COB -ID	RTR	DLC	D0	D1	D2	D3	D4	D5	D6	D7
80 h (128 d)	0	0	x	x	x	x	x	x	x	x

Only the Slave nodes reply to such a message, being their "TRANSMISSION TYPE" parameter of the "Transmit PDO Communication Parameter" objects (Index 1800hex, 1801hex of the Communication Profile) a numeric value between 0 and 240, as requested to enable the asynchronous communication.

1.2.4 P.D.O. (Process Data Object)

Data and information exchange is obtained by means of the PDOs. They are active only when the node is in OPERATIONAL status. Transmitted data type and order are freely configurable.

The standard configuration of MK2/IK2 transducer, part of the slave node, provides the transmission of cam position, speed and status. Two PDOs have been defined: one for each cursor (two cursors available). When fully operating, both PDOs are active: the first one sends the data relating to the second cursor. When using a single cursor, the second PDO is normally deactivated, even if it is available for any further configuration.

PDO structure:

MASTER ← SLAVE

COB -ID	RTR	DLC	D0	D1	D2	D3	D4	D5	D6	D7
180+NODE-ID (384d+NODE-ID)	0	X	X	X	X	X	X	X	X	X

MASTER ← SLAVE

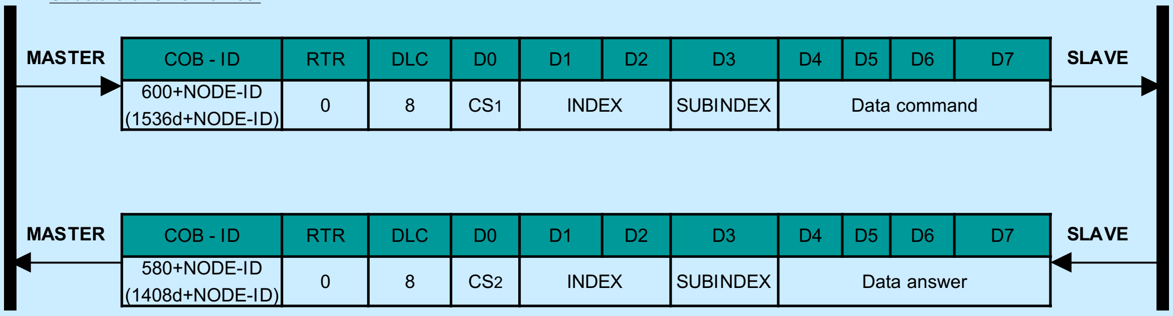
COB -ID	RTR	DLC	D0	D1	D2	D3	D4	D5	D6	D7
280+NODE-ID (640d+NODE-ID)	0	X	X	X	X	X	X	X	X	X

1.2.5 S.D.O (Service Data Object)

According to the objects (as defined in the dictionary), an access communication is activated through the SDOs, to allow them to be read and changed. The request is made through SDO-rx; while the response is carried out by means of SDO-tx.

They are specified as follows:

Structure of SDO frames:



COB - ID	RTR	DLC	D0	D1	D2	D3	D4	D5	D6	D7
600+NODE-ID (1536d+NODE-ID)	0	8	CS1	INDEX		SUBINDEX	Data command			

COB - ID	RTR	DLC	D0	D1	D2	D3	D4	D5	D6	D7
580+NODE-ID (1408d+NODE-ID)	0	8	CS2	INDEX		SUBINDEX	Data answer			

Command Specifier (CS), is 8 bits with the following mean:

		Byte 0 (CS)							
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MASTER WRITES									
	Master request	**Css = 1**			**x**	**n**		e=1	**s**
	Node response	**Scs =3**							
MASTER READS									
	Master request	**Scs = 2**			**x**				
	Node response	**Ccs = 2**			**x**	**n**		**E = 1**	**s**
Abort Transfer									
	request	**Cs = 4**			**x**				

• **cs:** command specifier

• **ccs:** client (master) command specifier

• **scs:** server (generic node) command specifier

• **n:** only valid if e =1 and s =1, otherwise 0. It indicates the number of no_significant bytes in field data (bytes [4, 7] that do not contain data. The bytes [8-n , 7] not contain segment data.

• **e:** transfer type:

0 → normal transfer

1 → expedited transfer.

This device implements the expedited transfer because the max data length is 4 bytes.

• **s:** size indicator:

0 → data set size is not indicated

1 → data set size is indicated

• **x:** not used, always 0

MASTER WRITES IN THE NODE'S DICTIONARY:

Master CS =	22	it isn't specified the number of transmitted data bytes
	23	4 transmitted data bytes
	27	3 transmitted data bytes
	2B	2 transmitted data bytes
	2F	1 transmitted data bytes
Slave CS =	60	

MASTER READS IN THE NODE'S DICTIONARY :

Master CS =	40	
Slave CS =	42	it isn't specified the number of transmitted data bytes
	43	4 transmitted data bytes
	47	3 transmitted data bytes
	4B	2 transmitted data bytes
	4F	1 transmitted data bytes

Abort Domain Transfer:

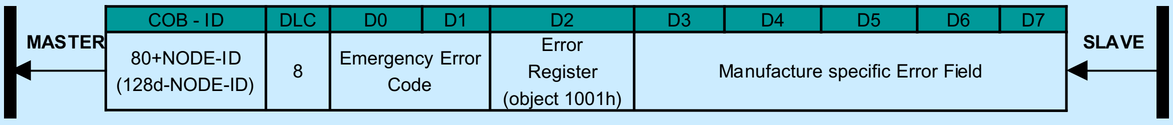
Slave CS =80

Abort code	Description
0503 0000h	Toggle bit not alternated
0504 0000h	SDO protocol timed out
0504 0001h	Client/server command specifier not valid or unknown
0504 0002h	Invalid block size (block mode only)
0504 0003h	Invalid sequence number (block mode only)
0504 0004h	CRC error (block mode only)
0504 0005h	Ouf of memory
0601 0000h	Unsupported access to an object
0601 0001h	Attempt to read a write only object
0601 0002h	Attempt to write a read only object
0602 0000h	Object does not exist in the object dictionary
0604 0041h	Object cannot be mapped to the PDO
0604 0042h	The number and lenght of the objects to be mapped would exceed PDO length
0604 0043h	General parameter incompatibility reason
0604 0047h	General internal incompatibility in the device
0606 0000h	Access failed due to an hardware error
0607 0010h	Data type does not match, length of service parameter does not match
0607 0012h	Data type does not match, length of service parameter too high
0607 0013h	Data type does not match, length of service parameter too low
0609 0011h	Sub-index does not exist
0609 0030h	Value range of parameter exceeded (only for write access)
0609 0031h	Value of parameter written too high
0609 0032h	Value of parameter written too low
0609 0036h	Maximum value is less than minimum value
0800 0000h	general error
0800 0020h	Data cannot be transferred or stored to the application
0800 0021h	Data cannot be transferred or stored to the application because of local control
0800 022h	Data cannot be transferred or stored to the application because of the present device state
0800 0023h	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error)

1.2. 7 EMERGENCY OBJECT

Emergency Object Data

The Emergency Telegram is 8 bytes long, as shown by the Figure:



COB - ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
80+NODE-ID (128d-NODE-ID)	8	Emergency Error Code		Error Register (object 1001h)	Manufacture specific Error Field				

Emergency Object Usage

Emergency objects are triggered by the occurrence of the device internal error. Emergency objects are suitable for interrupt type error alerts. An emergency object is transmitted only one time for every error event. As long as no new errors occur on a device, no further emergency objects are transmitted.

The following tables xx and yy indicate the means of the used code.

Table xx

Emergency Error Code (hex)	Meaning
00xx	Error Reset or No Error
10xx	Generic Error
20xx	Current
21xx	Current, device input side
22xx	Current, inside the device
23xx	Current, device output side
30xx	Voltage
31xx	Mains Voltage
32xx	Voltage inside the device
33xx	Output Voltage
40xx	Temperature
41xx	Ambient Temperature
42xx	Device Temperature
50xx	Device Hardware
60xx	Device Software
61xx	Internal Software
63xx	User Software
64xx	Data Set
70xx	Additional Modules
80xx	Monitoring
81xx	Communication
8110	CAN Overrun (Objects lost)
8120	CAN in Error Passive Mode
8130	Life Guard Error or Heartbeat Error
8140	recovered from bus off
82xx	Protocol Error
8210	PDO not processed due to length error
8220	PDO length exceeded
90xx	External Error
F0xx	Additional Functions
FFxx	Device specific

The device may be in one of two emergency states (Figure zz). Dependent on the transitions emergency, different objects will be transmitted.

0. After initialization the device enters the error free state if no error is detected. No error message is sent.

- The device detects an internal error located in the first three bytes of the emergency message (error code and error register). The device enters the error state. An emergency object with the appropriate error code and error register is transmitted. The error code is filled in at the location of object 1003h (pre-defined error field).
- One, but not all error reasons, are gone. An emergency message containing error code 0000 (Error reset) may be transmitted together with the remaining errors in the error register and in the manufacturer specific error field.
- A new error occurs on the device. The device remains in error state and transmits an emergency object with the appropriate error code. The new error code is filled in at the top of the array of error codes (1003h). It has to be guaranteed that the error codes are sorted in a timely manner (oldest error - highest sub-index, see Object 1003h).
- All errors are repaired. The device enters the error free state and transmits an emergency object with the error code ‘reset error / no error’.

Table yy

Error Register code	
Bit	Meaning
0	Generic errror
1	Current
2	Voltage
3	Temperature
4	Communication error (overrun, error state)
5	Device profile specific
6	Reserved (always 0)
7	Manufacturer specific

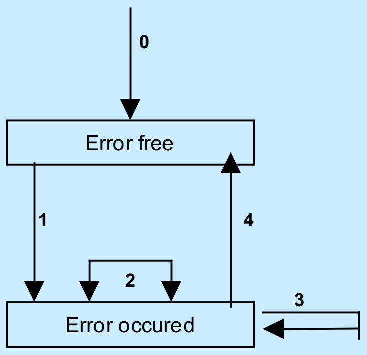


Figure zz: Emergency State Transition Diagram

Device Error messages

The following table indicate the Device Emergency Message code (code FFxx table xx):

Additional error code (hex)	Error code (Hex)	Meaning
0006	9000	No Cursors in measuring range, or incorrect number of Cursors
0007	9000	Cursor 1 velocity too high (values invalid)
0107	9000	Cursor 2 velocity too high (values invalid)
0008	9000	Cursor 1 outside working range
0108	9000	Cursor 2 outside working range

Example:

The transducer has the Cursor 2 outside working *range (there aren't other error states) :*

MASTER ← SLAVE

COB - ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
80+NODE-ID (128d+NODE-ID)	8	00	90	20	80	01	00	00	00

The Cursor 2 back in working range *(there aren't other error states)* :

MASTER ← SLAVE

COB - ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
80+NODE-ID (128d+NODE-ID)	8	00	00	00	80	01	00	00	00

Cam Emergencies

Cam Emergencies

Trasmission of the Emergency Object for the Cam emergencies (see Fig. 3–1) is done when a Cam status change.

This ensures that any overtravel, of a enabled Cam switchpoint, is immediately reported to the controller.

In the byte 3 there is indicated the Cam state Cursor 1;

In the byte 4 there is indicated the Cam state Cursor 2;

In the byte 6 there is indicated the number of Cursors involved in the changing: (0= Cursor _1;   1= Cursor _2);

In the byte 7 there is indicated the number of Cam involved in the changing: (1= Cam_1; 2=Cam_2; 4=Cam_3; 8=Cam_4) ;

MASTER ← ← **SLAVE**

COB - ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
80+NODE-ID (128d+NODE-ID)	8	00	F0	Error Field	Cam State 1	Cam State 2	00	Cursor	Cam

Fig. 3-1 :Organization of Cam status Emergency Messages

Example:

The Cursor 1 go beyond the Cam 2. The Cam 4 of the Cursor 3 is active

MASTER ← ← **SLAVE**

COB - ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
80+NODE-ID (128d+NODE-ID)	8	00	F0	20	02	08	00	00	02

The Cursor 1 go out the Cam 2. The Cam 4 of the Cursor 3 is active

MASTER ← ← **SLAVE**

COB - ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
80h+NODE-ID (120d+NODE-ID)	8	00	F0	00	02	00	00	00	02

1.2.8 Layer Setting Service (L.S.S)

LSS enable the inquiry and the change of the values of some parameters of the local layers on a CANopen module. The CANopen module activate these operations with LSS Master and LSS Slave capabilities.

The LSS affect the following parameters:

. Node-ID of the CANopen Slave

. Bit timing parameters of the pysical layer (baud rate)

. LSS address (Object Dictionary, Index 1018H)

The module that sets other modules via a CANopen network is called the **LSS Master**. There may be only one LSS Master in a network. The LSS Master has no attributes.

The module that is setted by the LSS Master via a CANopen network is called the **LSS Slave**. The number of the LSS Slaves in a network is unlimited.

The LSS Slave has the following attributes:

<u>a) LSS ADDRESS:</u>

An LSS Slave is identified by an LSS Address. An LSS Address is formed to a vendor-id, a product-code, a revision-number and a serial-number. They adhere to the following syntax:

<LSS-ADDRESS>	= <vendor-id><product-code><revision-number><serial number>
<vendor-id>	= 'UNSIGNED32'
<product-code>	= 'UNSIGNED32'
<revision-number>	= 'UNSIGNED32'
<serial-number>	= 'UNSIGNED32'

The <vendor-id> is assigned to module suppliers by CiA.

The <product-code>, is a manufacturer specific code

The <revision-number> consist of two 16 bit numerical values. The higher 16 bit represent the revision of the CANopen parts of the software and the lower 16 bits represent the general firmware release

The <serial number> is also manufacturer specific and contain the coded date assigned by the module manufacturer.

The LSS-Address must met the following conditions :

- The LSS address is the same of the CANopen identity object
- The LSS address of a LSS Slave can be inquired
- There exists no other LSS Slave in the world with the same <LSS-Address>

<u>b) LSS MODE</u>

To activate the LSS functionality, all devices of the network must be set to the stop state. The LSS-Master have to locate on the same device where reside in the NMT-Master. The LSS mode operate in a different manner when the module is set to configuration phase or to operation phase. In operation mode is available only the LSS switch mode service.

The module, not clearly put to configuration mode, is in operation mode.

LSS MODES AND SERVICES

LSS services can be functionally grouped in three areas:

1 - The switch mode service provide a way to logically connect the LSS Master and LSS Slave(s) for configuration purposes. They change the LSS mode attribute of the LSS Slave (see fig A)

2 - The configuration services perform the actual task of configuring the layer parameters of an LSS Slave. The configuration services are only available in configuration mode.

3 - The inquiry services provide a way for the LSS Master to determine layer parameters. The inquiry services are available only in configuration mode.

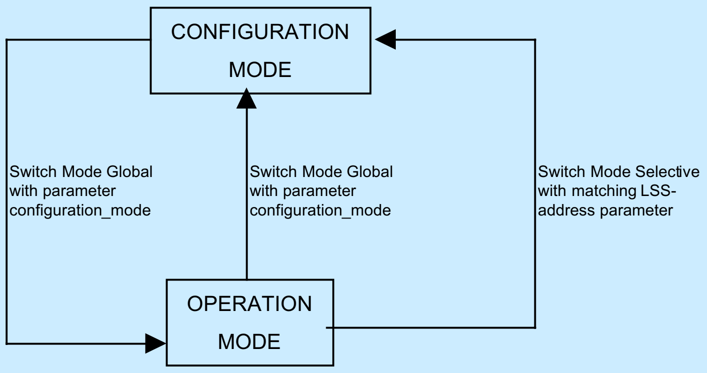

CONFIGURATION MODE
OPERATION MODE
Switch Mode Global with parameter configuration_mode
Switch Mode Global with parameter configuration_mode
Switch Mode Selective with matching LSS-address parameter

Fig. A: LSS modes and switching procedure

SWITCH MODE SERVICES

The Switch Mode Services control the mode attribute of a LSS Slave. The LSS provides two ways to put a LSS Slave into configuration mode : the Switch Mode Global and the Switch Mode Selective.

SWITCH MODE GLOBAL

This service is used to switch to configuration mode all the LSS Slaves in the network.

SWITCH MODE SELECTIVE

This service is used to switch to configuration mode, all the LSS Slaves whose the LSS address attribute equals the LSS_address.

CONFIGURATION SERVICES

The Configuration Services are available only in configuration mode. Some of the services require that only one LSS Slave must be set to configuration mode.

CONFIGURE NODE-ID

Through this service the LSS Master configures the NMT-address parameter of a LSS Slave.The allowed NODE-ID range is 1..7F h (1..127 d).

This service allows only one LSS Slave to configuration mode. The remote result parameter confirms the success or failure of the service. Optionally, in case of a failure, the mistake reason is explained.

CONFIGURE BIT TIMING PARAMETERS

Through this service, the LSS Master Configure Bit Timing Parameters table on a LSS Slave.

By means of the table_selector the bit timing parameter table to be used is specified. The table define the used parameters for different baud rates.

With table_selector value ‘0’ the followed standard CiA bit timing parameter table is referenced.

Table Index	Baud Rate	Nominal Bit time	Max
0	1 Mbit/s	1µs	25 m
1	800 kbit/s	1.25 µs	50 m
2	500 kbit/s	2 µs	100 m
3	250 kbit/s	4 µs	250 m
4	125 kbit/s	8 µs	500 m
5	100 kbit/s	10 µs	700 m
6	50 kbit/s	20 µs	1,0 km
7	20 kbit/s	50 µs	2,5 km
8	10 kbit/s	100 µs	5,0 km

This service allows all LSS Slaves in configuration mode. The service has to be followed by an Activate Bit Timing Parameters service to enable the configured parameters. After execution of the Configure Bit Timing Parameters service, the node may not execute any remote LSS service, except the services Configure Bit Timing Parameters, Activate Bit Timing Parameters and Switch Mode.

The remote result parameter confirms the success or failure of the service. Optionally, in case of a failure, the mistake reason is explained.

ACTIVATE BIT TIMING PARAMETERS

Through the Activate Bit Timing Parameters service the LSS Master activates the bit timing as defined by the Configure Bit Timing Parameters service.

The switch_delay parameter specifies the length (milliseconds) of two delay periods of equal length, which are necessary to avoid operating the bus with incorrect Bit Timing Parameters. The first delay period define the waiting time of node before to switch to new parameters. After the switching, the second delay period define the waiting time of node before to transmit any message with the new Bit Timing Parameters.

STORE CONFIGURED PARAMETERS

The Store Configured Parameters service is used to store the active configured parameters into non-volatile memory.

The remote result parameter confirms the success or failure of the service. Optionally, in case of a failure, the mistake reason is explained.

INQUIRY SERVICES

This services are available only in configuration mode.

INQUIRE LSS ADDRESS

This service allows to determine the LSS-address parameters of a LSS Slave in configuration mode.

Exactly one LSS Slave may be in configuration mode when this service is executed. The remote result parameter confirms the LSS address of the LSS Slave in configuration mode or the failure of the service. Optionally, in case of a failure, the mistake reason is explained.

IDENTIFICATION SERVICES

LSS IDENTIFY REMOTE SLAVES

Through this service, the LSS Master consults all LSS slaves, whose LSS address meets the LSS_Address_sel, for identify themselves through the 'LSS Identify Slave' service. **LSS_Address_sel** consists of a fixed manufacturer and product name, and a span of serial numbers. This service is unconfirmed.

LSS IDENTIFY SLAVE

Through this service, an LSS Slave reply back to the Master message, that it is a Slave with an LSS address within the LSS_Address_sel of the 'LSS Identify Remote Slave' service. This service is unconfirmed.

LSS SLAVE SYNCHRONISATION

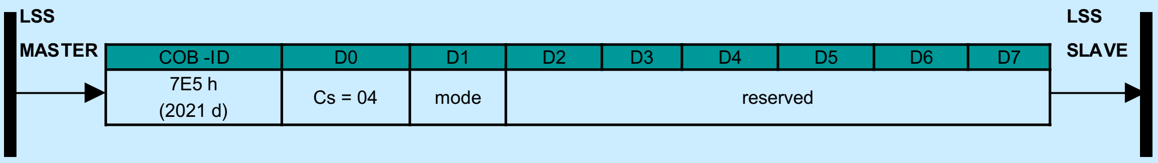
Since, in the LSS Protocol, all LSS Slaves use the same COB to send information to the LSS Master, there must be only one LSS Slave at a time that communicates with the LSS Master. For all protocols the LSS Master takes the initiative, and a LSS Slave is only allowed to transmit within a confirmed service, after it has been uniquely switched into configuration mode. Since there can be at least one confirmed LSS service outstanding at a time, the synchronization is established.

SWITCH MODE PROTOCOLS

SWITCH MODE GLOBAL

This protocol is used to implement the 'Switch Mode Global' service.

SWITCH MODE GLOBAL



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 04	mode	reserved					

. Cs

LSS command specifier

04 for Switch Mode Global

. mode

The LSS mode to switch to:

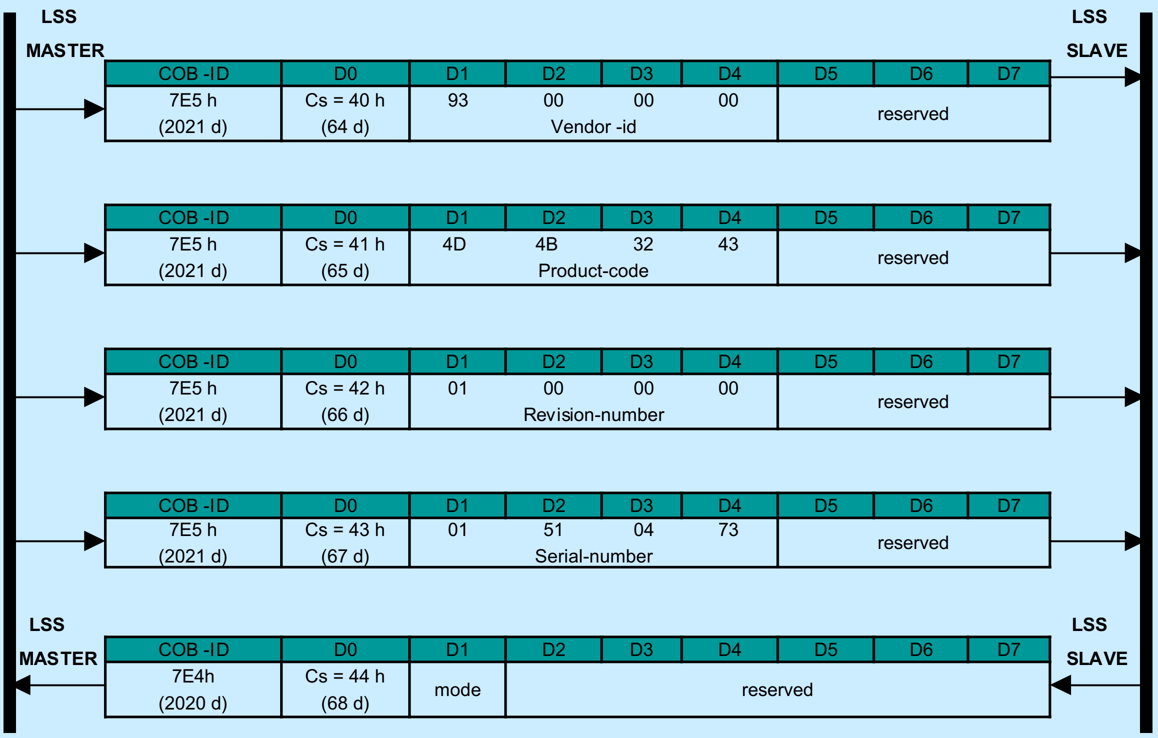
0: switches to operation mode

1: switches to configuration mode

. Reserved

reserved for further use by CiA

SWITCH MODE SELECTIVE This protocol is used to implement the ‘Switch Mode Selective’ service.



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 40 h (64 d)	93	00	00	00	reserved		
		Vendor -id						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 41 h (65 d)	4D	4B	32	43	reserved		
		Product-code						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 42 h (66 d)	01	00	00	00	reserved		
		Revision-number						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 43 h (67 d)	01	51	04	73	reserved		
		Serial-number						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4h (2020 d)	Cs = 44 h (68 d)	mode	reserved					

. **Cs**: LSS command specifiers;

64d(40 h) to 68d (44 h) for Switch Mode Selective

. **Vendor-id**: Vendor name part of the LSS address, with respect to index 1018h, subindex 1

. **Product-code**: Product name part of the LSS address, with respect to index 1018h, subindex 2

. **Revision-number**: Revision part of the LSS address, with respect to index 1018h, subindex 3

. **Serial_number**: Serial number part of the LSS address, with respect to index 1018h, subindex 4

. **Mode**: The actual LSS mode of the Slave

0: operation mode

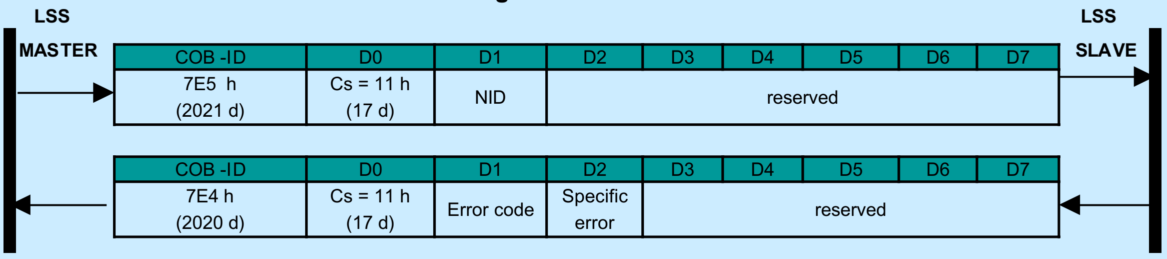
1: configuration mode

CONFIGURATION PROTOCOL

CONFIGURE NODE-ID PROTOCOL

This protocol is used to implement the ‘Configure Node-id’ service for the Node-Id part of the NMT address

Configure Node-ID



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 11 h (17 d)	NID	reserved					

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 11 h (17 d)	Error code	Specific error	reserved				

. **Cs**: LSS command specifiers;

17d (11 h) for Configure Node-ID

. **NID:** The new Node-ID to configure

. **Error-code**:

0:	protocol successfully completed
1:	Node-ID out of range
2 … 254:	reserved for further use by CiA
255:	implementation specific error occured

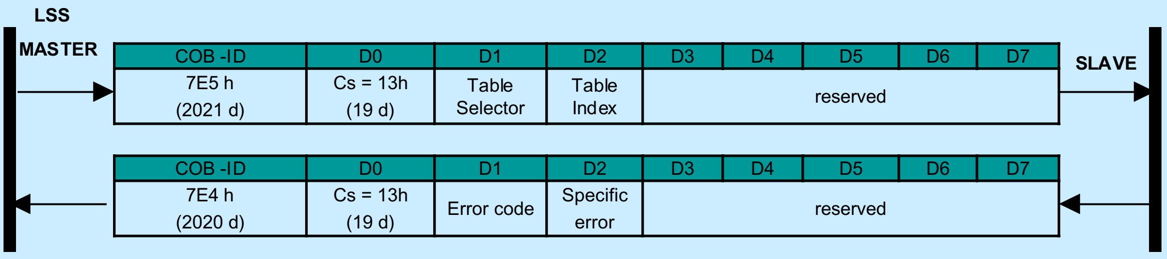
. **Specific_error_code**: If error_code equals 255, specific_error_code gives an implementation specific error code, otherwise it is reserved for further use by CiA

. **Reserved:** reserved for further use by CiA

Configure Bit Timing Parameters Protocol

This protocol is used to implement the ‘Configure Bit Timing Parameters’ service

Configure Bit Timing Parameters



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 13h (19 d)	Table Selector	Table Index	reserved				

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 13h (19 d)	Error code	Specific error	reserved				

. **Cs**: LSS command specifiers;

19d (13 h) for Configure Bit Timing Parameters

. **Table-selector:**

selects which bit timing parameters table

0: standard CiA bit timing table (see /4/)

1…127: reserved for further use by CiA

. **Table_index**:

Selects the entry (bit timeing parameters) in the selected table; for valid indices when using the standard CiA bit timings (table_selector =0) see configure bit timing parameters

. **Error_code:**

0:	protocol successfully completed
1:	bit timing not supported
2 … 254:	reserved for further use by CiA
255:	implementation specific error occured

. **Specific_error_code:**

If error_code equals 255, specific_error_code gives an implementation specific error code, otherwise it is reserved for further use by CiA

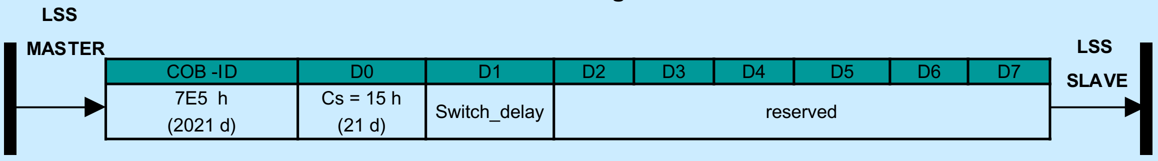
. **Reserved:**

reserved for further use by CiA

Activate bit timing parameters protocol.

This protocol is used to implement the ‘Activate Bit Timing Parameters’ service.

Activate Bit Timing Parameters



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 15 h (21 d)	Switch_delay	reserved					

. **Cs:**

LSS command specifier

21d (15 h) for Activate Bit Timing Parameters

. **Switch_delay:**

The duration of the two periods of time to wait until the bit timing parametrs switch is done (first period) and before transmitting any CAN message with the new bit timing parameters after performing the switch (second period). The time unit of switch delay is 1 ms.

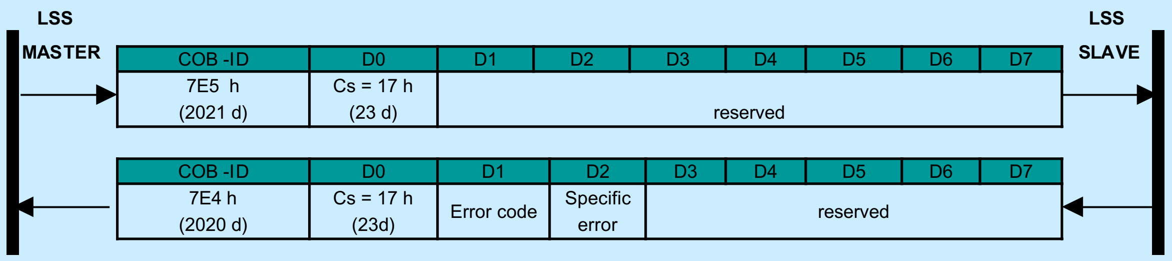
. **Reserved:**

reserved for further use by CiA

Store Configuration Protocol:

This protocol is used to implement the 'Store Configured Parameters' service.

Store Configuration



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 17 h (23 d)	reserved						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 17 h (23d)	Error code	Specific error	reserved				

. Cs:

LSS command specifier

23d (17 h) for Store Configuration

. Error_code:

0:	protocol successfully completed
1:	store configuration is not supported
2:	storage media access error
3 … 254:	reserved for further use by CiA
255:	implementation specific error occured

. Specific_error_code.

If error_code equals 255, specific_error_code gives an implementation specific error code otherwise it is reverved for further use by CiA

. Reserved:

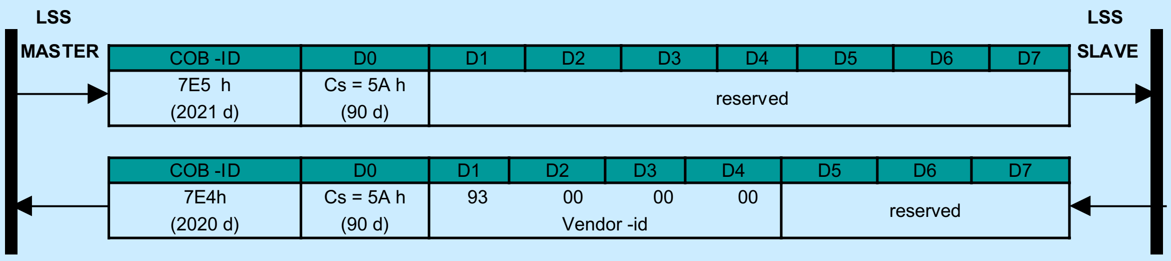
reserved for further use by CiA

INQUIRY PROTOCOLS

Inquire LSS Address Protocols

These protocols are used to implement the 'inquire LSS Address' service. To implement the service, each of the following three protocols has to be executed

Inquire Identity Vendor-ID



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 5A h (90 d)	reserved						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4h (2020 d)	Cs = 5A h (90 d)	93	00	00	00	reserved		
		Vendor -id						

. Cs:

LSS command specifier

90d (5Ah) for Inquire Manufacturer Name

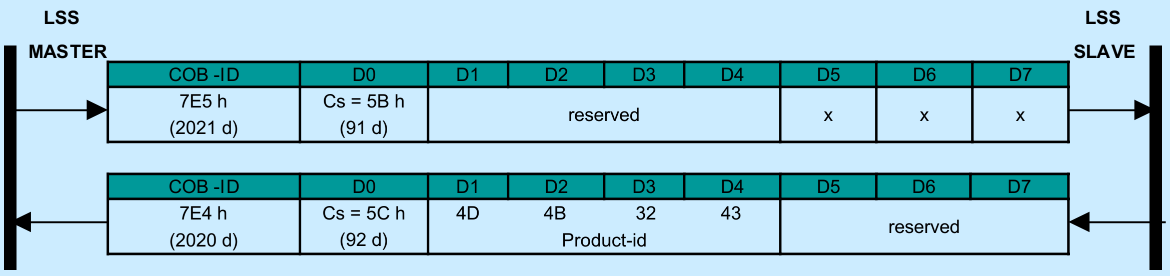
. Vendor-id

The vendor-id of the selected module

. Reserved:

reserved for further use by CiA

Inquire Identity Product-Code



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 5B h (91 d)	reserved				x	x	x

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 5C h (92 d)	4D	4B	32	43	reserved		
		Product-id						

. Cs:

LSS command specifier

91d (5B h) for Inquire Product Name

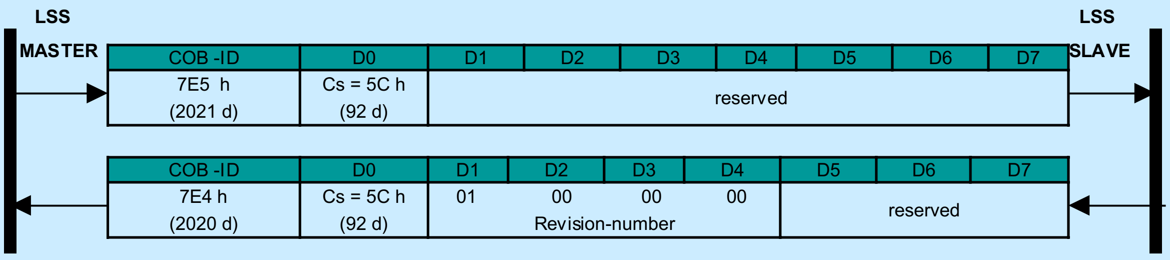
. Reserved:

reserved for further use by CiA

. Product-code:

The product-code of the selected module

Inquire Identity Revision-Number



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 5C h (92 d)	reserved						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 5C h (92 d)	01	00	00	00	reserved		
		Revision-number						

. Cs:

LSS command specifie 92d (5C h) for Inquire Serial name.

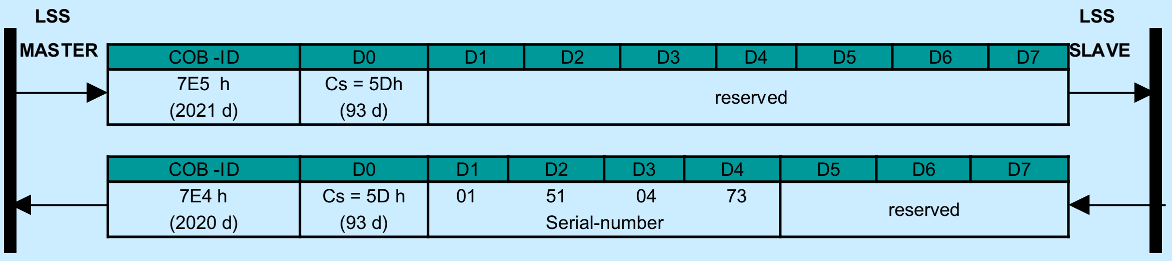
. Revision-number:

The revision-number (see/2/) of the selected module

. Reserved:

reserved for further use by CiA

Inquire Identity Serial-Number



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 5Dh (93 d)	reserved						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 5D h (93 d)	01	51	04	73	reserved		
		Serial-number						

. Cs:

LSS command specifier

93d (5D h) for Inquire Serial Number

. Serial-Number:

The serial-number of the selected module

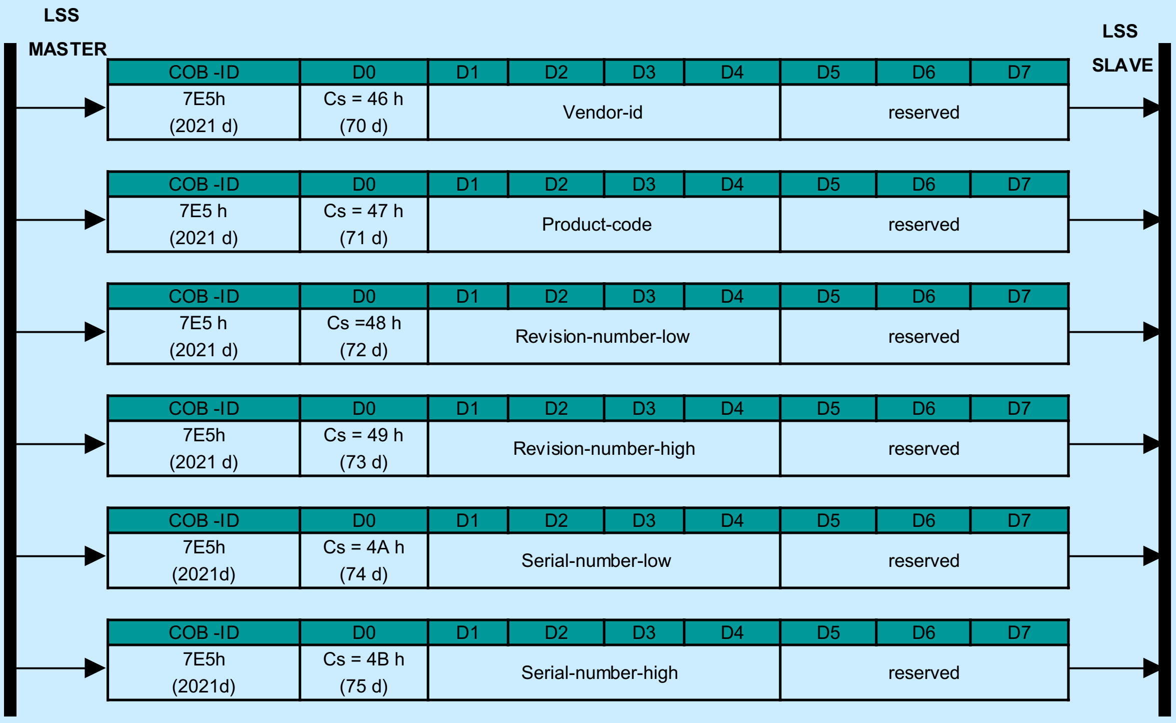
. Reserved:

reserved for further use by CiA

IDENTIFICATION PROTOCOLS

LSS Identify Remote Slaves:

This protocol is used to implement the ‘LSS identify Remote Slave’ service.



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5h (2021 d)	Cs = 46 h (70 d)	Vendor-id				reserved		

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 47 h (71 d)	Product-code				reserved		

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs =48 h (72 d)	Revision-number-low				reserved		

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5h (2021 d)	Cs = 49 h (73 d)	Revision-number-high				reserved		

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5h (2021d)	Cs = 4A h (74 d)	Serial-number-low				reserved		

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5h (2021d)	Cs = 4B h (75 d)	Serial-number-high				reserved		

. Cs:

LSS command specifier

70d (46 h) to 75d (4B h) for Inquire Serial Number

. Vendor.id:

the manufacturer name part of the LSS Address

. Product-code:

The product name part of the LSS Address

. Revision-number-low:

The lower boundary of the requested revision numbers range. The minore range must be to 0000h

. Revision-number-high:

The higher boundary of the requested revision numbers range. The minor range must be set to FFFFh

.Serial-number-low:

The lower boundary

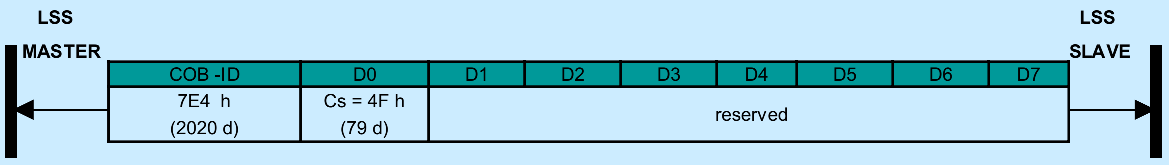
. Serial-number-high:

The higher boundary of the requested serial numbers range

The boundaries are included in tre interval. All LSS Slaves with matching vendor-id and product-code whose major revision-number and serial-number lie within the given ranges, are requested to identify themselves with the LSS Identify Slave service.

LSS Identify Slave Protocol

This protocol is used to implement the ‘LSS identify Slave’ service.



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 4F h (79 d)	reserved						

. Cs:

LSS command specifiers

79d (4F h) for Identify Slave

. Reserved:

All bytes set to ‘0’

LMT Prespective

LMT offers di possibility to inquire and change the settings of the local layers on a CAN module. The following parameters can be inquired and/or changed by the use of LMT:

. NMT-address of the NMT Service Element

. Bit timing parameters of the physical layer

. LMT address

LMT Master and Slave Object

The module that configures other modules via a CAN network is called the **LMT Master**. There may be only one LMT Master in a network. The LMT Master has no attributes.

The module that is configured by the LMT Master via a CAN Network is called the **LMT Slave**. The number of LMT Slaves in a network is unlimited.

The **LMT address** uniquely identifies a LMT Slave.

LMT MODES AND SERVICES

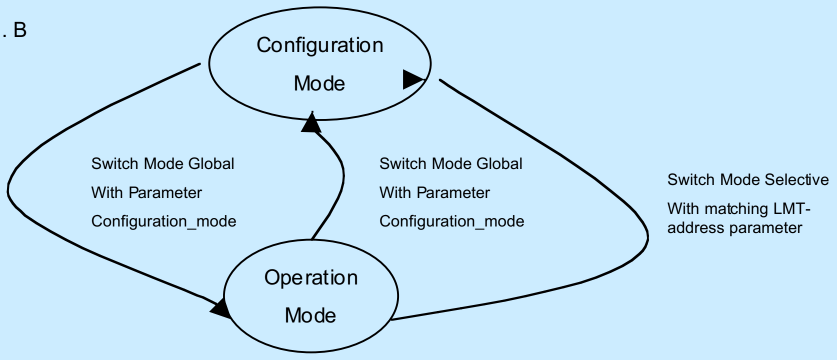
LMT services can be functionally prouped in three areas:

1 - The switch mode service provide a way to logically connect the LMT Master and LMT Slave(s) for configuration purposes. They change the LMT mode attribute of the LMT Slave (see fig B)

2 - The configuration services perform the actual task of configuring the layer parameters of an LMT Slave. The configuration services are only available in configuration mode.

3 - The inquiry services provide a way for the LMT Master to determine layer parameters. The inquiry services are available only in configuration mode.

Fig. B



LMT Slave Synchronisation

Since in the LMT Protocol all LMT slave use the same COB to send information to the LMT Master, there must be only one LMT Slave at the time that communicates with the LMT Masters.

1.2.9 Layer Management service (L.M.T.)

SWITCH MODE SERVICES

The switch Mode Services control the mode attribute of a LMT Slave. LMT provides two way to put a LMT Slave into configuration mode, Switch Mode Global and Switch Mode Selective. Switch Mode Selective switches one LMT-Slave into configuration mode.

Switch Mode Global switches all LMT Slaves into configuration mode.

Some LMT configuration and inquiry services require that only one LMT Slave is in configuration mode.

SWITCH MODE GLOBAL

This service is used to switch all LMT Slaves in the network, between operation mode and configuration mode. This service is unconfirmed.

SWITCH MODE SELECTIVE

This service is used to switch the Slave, whose LMT address attribute equals LMT_address, into configuration Mode. This service is unconfirmed.

CONFIGURATION SERVICES

The Configuration Services are available only in configuration mode. Some of this services require that only one LMT Slave must be set to configuration mode.

CONFIGURE NMT ADDRESS

Through this service the LMT Master configures the NMT-address parameter of a LMT Slave.

This service allows only one LMT Slave in configuration mode. This service is confirmed. The remote result parameter confirms the success or failure of the service. Optionally, in case of a failure, the mistake reason is explained.

CONFIGURE BIT TIMING PARAMETERS

Through the Configure Bit Timing Parameters service the LMT Master sets the new bit timing on a LMT Slave.

With table_selector value '0' the standard CiA bit timing parameter table is referenced.

Exactly one LMT slave may be in configuration mode when this service is executed. The service has to be followed by an Activate Bit Timing Parameters service to activate the configured parameters.

This service is confirmed. The remote result parameter confirms the success or failure of the service. Optionally, in case of failure, the mistake reason is explained.

The maximum possible **baud rate** is determined by the total length of the CAN bus cable.

Table Index	Baud Rate	Nominal Bit time	Max
0	1 Mbit/s	1µs	25 m
1	800 kbit/s	1.25 µs	50 m
2	500 kbit/s	2 µs	100 m
3	250 kbit/s	4 µs	250 m
4	125 kbit/s	8 µs	500 m
5	100 kbit/s	10 µs	700 m
6	50 kbit/s	20 µs	1,0 km
7	20 kbit/s	50 µs	2,5 km
8	10 kbit/s	100 µs	5,0 km

ACTIVATE BIT TIMING PARAMETERS

Through the Activate Bit Timing Parameters service the LMT Master activates the bit timing as defined by the Configure Bit Timing Parameters service.

The switch_delay parameter specifies the length (milliseconds) of two delay periods of equal length, which are necessary to avoid operating the bus with incorrect Bit Timing Parameters. The first delay period define the waiting time of node before to switch to new parameters. After the switching, the second delay period define the waiting time of node before to transmit any message with the new Bit Timing Parameters.

This service is unconfirmed

STORE CONFIGURED PARAMETERS

The Store Configured Parameters service is used to store the active configured parameters into non-volatile memory.

The remote result parameter confirms the success or failure of the service. Optionally, in case of a failure, the mistake reason is explained.

INQUIRY SERVICES

The inquiry services are available only in configuration mode.

INQUIRY LMT ADDRESS

This service allows to determine the LMT-address parameters of a LMT Slave in configuration mode.

This service is confirmed. The remote result parameter confirms the LMT address of the LMT Slave in configuration mode or the failure of the service. In case of a failure optionally the reason is confirmed.

IDENTIFICATION SERVICES

LMT IDENTIFY REMOTE SLAVES

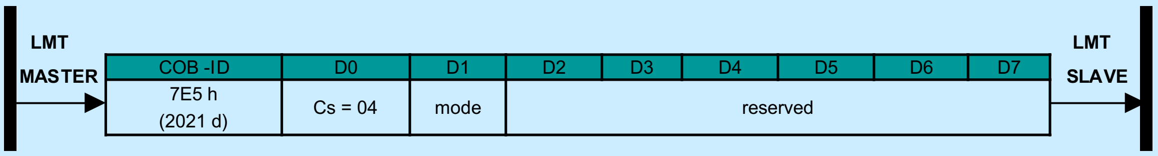
Through this service, the LMT Master inquires all LMT slaves, whose LMT address meets the LMT_Address-selection to identify themselves through the 'LMT Identify Slave' service. LMT_Address _sel consists of a fixed manufacturer and product name and a span of serial numbers. This service is unconfirmed and mandatory for LMT Nodes with Class 2.

LMT IDENTIFY SLAVE

Through this service, an LMT Slave indicates, that is a Slave with an LMT address within the LMT_Address_sel of an 'LMT Identify Remote Slave' service executed prior to this service. The service is unconfirmed.

SWITCH PROTOCOLS

Switch Mode Global



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 04	mode	reserved					

. **Cs:** LMT command specifier

04d for Switch Mode Global

. **Mode**: The LMT mode to switch to

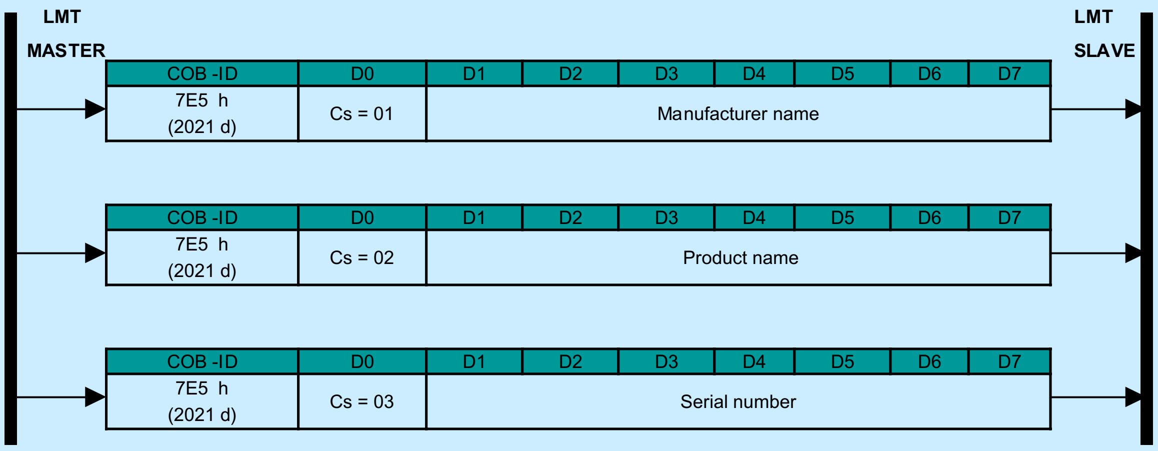
0: switches to operation mode

1: switches to configuration mode

. **Reserved**: reserved for further use by CiA

Switch Mode Selective

This protocol is used to implement the Switch Mode Selective service



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 01	Manufacturer name						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 02	Product name						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 03	Serial number						

. Cs:

LMT command specifiers

01d to 03 d for Switch Mode Selective

. Manufacturer_name: The manufacturer name part of the LMT address

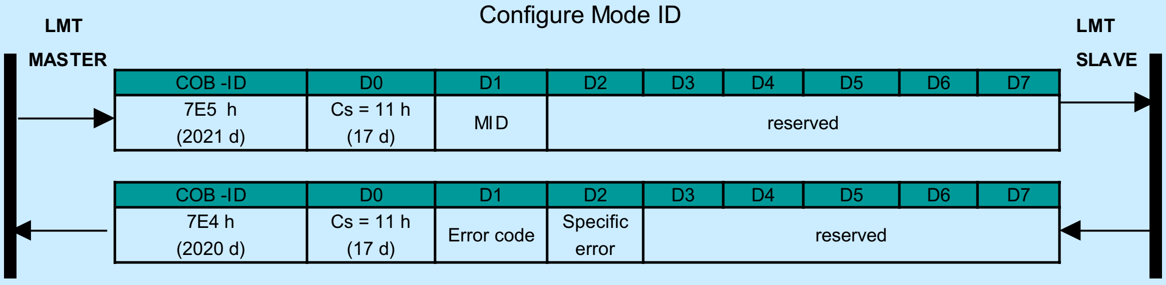
. Product_name: The product name part of the LMT address

Serial_number: The serial number part of the LMT address

CONFIGURATION PROTOCOLS

Configure NMT Address Protocols

Configure Mode ID



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 11 h (17 d)	MID	reserved					

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 11 h (17 d)	Error code	Specific error	reserved				

. Cs:

LMT command specifier

17d (11 h) for Configure Mode ID

. MID: The new module_id to configure

. Error_code:

0:	protocol seccessfully completed
1 ... 254:	reserved for further use by CiA
255:	implementation specific error occured

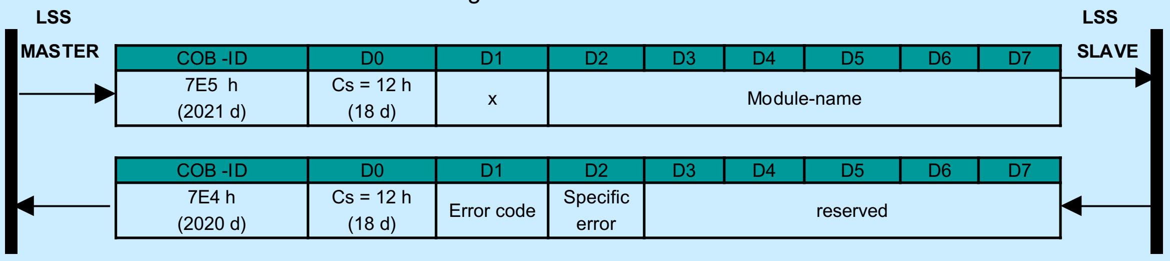
. Specific_error_code: If error_code equals 255, specific_error_code gives an implementation specific error code, otherwise it is reserved for further use by CiA

. Reserved: reserved for further use by CiA

Configure Module Name Protocol:

This protocol is used to implement the configure NMT Address service for the module-name part of the NMT address.

Configure Module Name



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 12 h (18 d)	x	Module-name					

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 12 h (18 d)	Error code	Specific error	reserved				

. Cs:

LMT command specifier

18d (12 h) for Configure Mode Name

. Module_name: the new module_name to configure

. Table_idnex: index for the bit timing to use

. Error_code:

0: protocol successfully completed

1 …254: reserved for further use by Cia

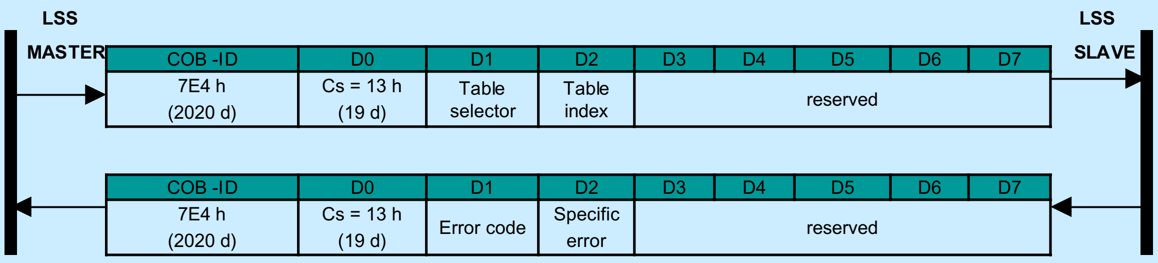
255: implementation specific error occured

. Specific _error_code: If error_code equals 255, specific_error_code gives an implementation specific error code, otherwise it is reserved for further use by CiA

. Reserved: reserved for further use by CiA

Configure Bit Timing Parameters Protocol

This protocol is used to implement the 'Configure Bit Timing Parameters' service



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 13 h (19 d)	Table selector	Table index	reserved				

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 13 h (19 d)	Error code	Specific error	reserved				

. Cs:

LMT command specifier

19d (13 h) for Configure Bit Timing Parameters

. Table_selector: selects which bit timing parameters table has to be used

0:	standard CiA bit timing table
1…127:	reserved for further use by CiA
128…255:	may be used for manufacturer specific bit timings

. Table_index: selects the entry (bit timing parameters) in the selected table; see/4/ for valid indices when using the standard CiA bit timings (table_selector ='0')

. Error_code:

0: protocol successfully completed

1: bit timing not supported

2…254: reserved for further use bu CiA

255: implementation specific error occured

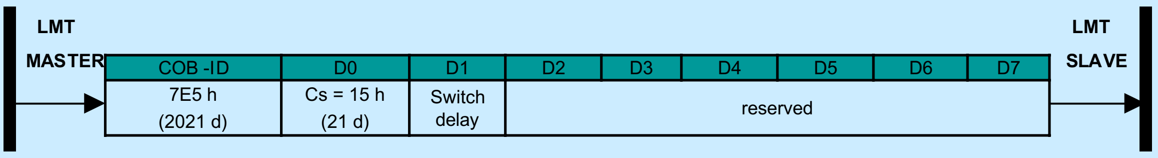
. Specific_error_code: If error_code equals, specific_error_code gives an implementation specific error code, otherwise it is reserved for further use by CiA

. Reserved: reserved for further use by CiA

Activate Bit Timing Parameters Protocol

This protocol is used to implement the Activate Bit Timing Parameters service

Activate Bit Timing Parameters



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 15 h (21 d)	Switch delay	reserved					

. Cs:

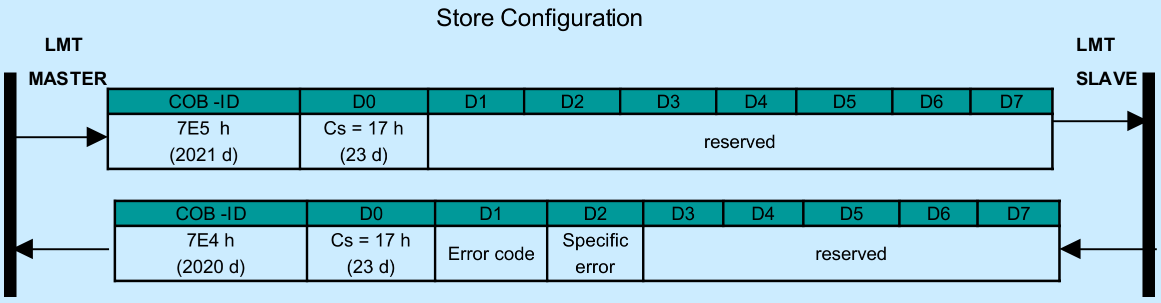
LMT command specifier

21d (15 h) for Activate Bit Timing Parameters

. Reserved: reserved for further use by CiA

. Switch_delay: the duartion of the two periods of time to wait until the bit timing parameters switch is done (first period) and before transmitting any CAN message with the new bit timing parameters after performing the switch (second period) The time unit of switch delay is 1 ms.

Store Configuration Protocol



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 17 h (23 d)	reserved						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 17 h (23 d)	Error code	Specific error	reserved				

. **Cs:** LMT command specifier

23d (17 h) for Store Configuration

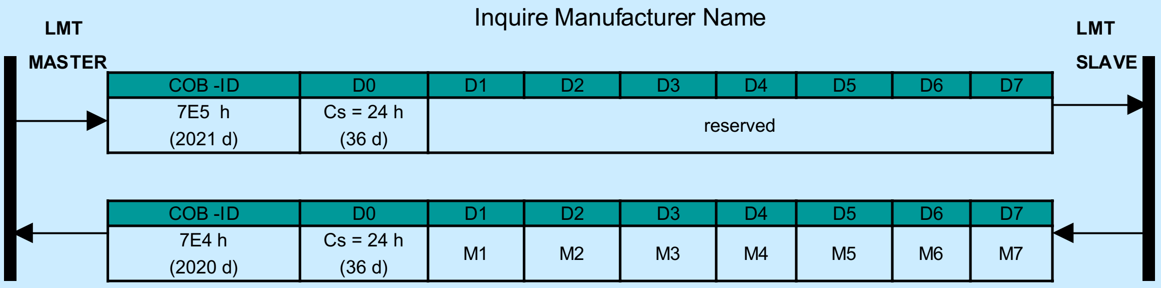
. **Error_code:**

0:	protocol successfully completed
1:	store configuration is not supported
2…254:	reserved for further use by CiA
255:	implementation specific error occured

. **Specific_error_code**: If error_code equals 255, specific_error_code gives an implementation specific error code, otherwise it is reserved for further use by CiA

. **Reserved**: reserved for further use by CiA

Inquire Manufacturer Name Protocol



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 24 h (36 d)	reserved						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 24 h (36 d)	M1	M2	M3	M4	M5	M6	M7

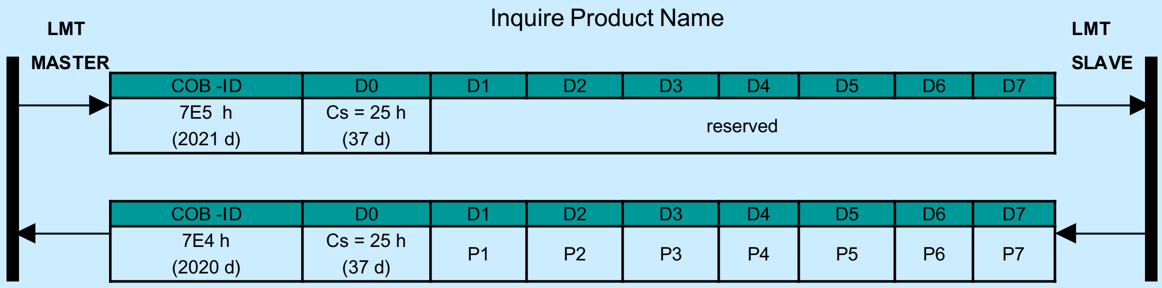
. **Cs:** LMT command specifier

36d (24 h) for Inquire Manufacturer Name

. **M1-M7**: The manufacturer_name of the selected module or error code.If M1 is a valid<alpha-num>, the response contains che name. If M1 is 0ffh, M2 contains the error code, M3 contains the reason if valid for the error code.

. **Reserved**: reserved for further use by CiA

Inquire Product Name Protocol



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 25 h (37 d)	reserved						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 25 h (37 d)	P1	P2	P3	P4	P5	P6	P7

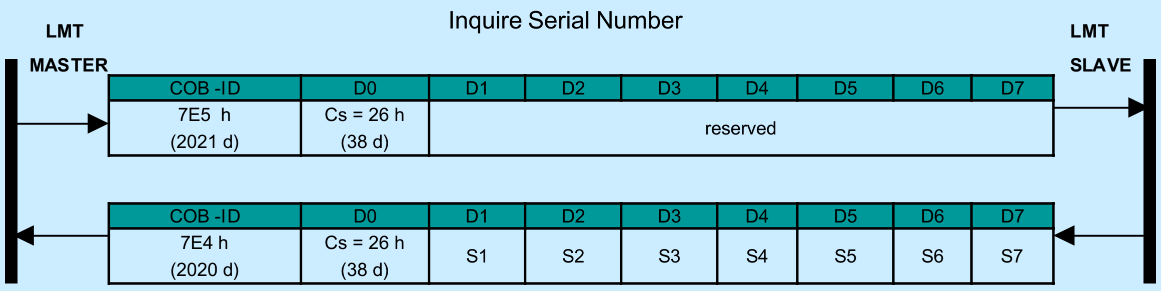
. **Cs:** LMT command specifier

37d (25 h) for Inquire Product Name

. **P1 - P7**: The product_name of the selected module or error code.If P1 is a valid<alpha-num>, the response contains che name. If P1 is 0ffh, P2 contains the error code, P3 contains the reason if valid for the error code.

. **Reserved**: reserved for further use by CiA

Inquire Serial-Number Protocol



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 26 h (38 d)	reserved						

COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E4 h (2020 d)	Cs = 26 h (38 d)	S1	S2	S3	S4	S5	S6	S7

. **Cs:** LMT command specifier

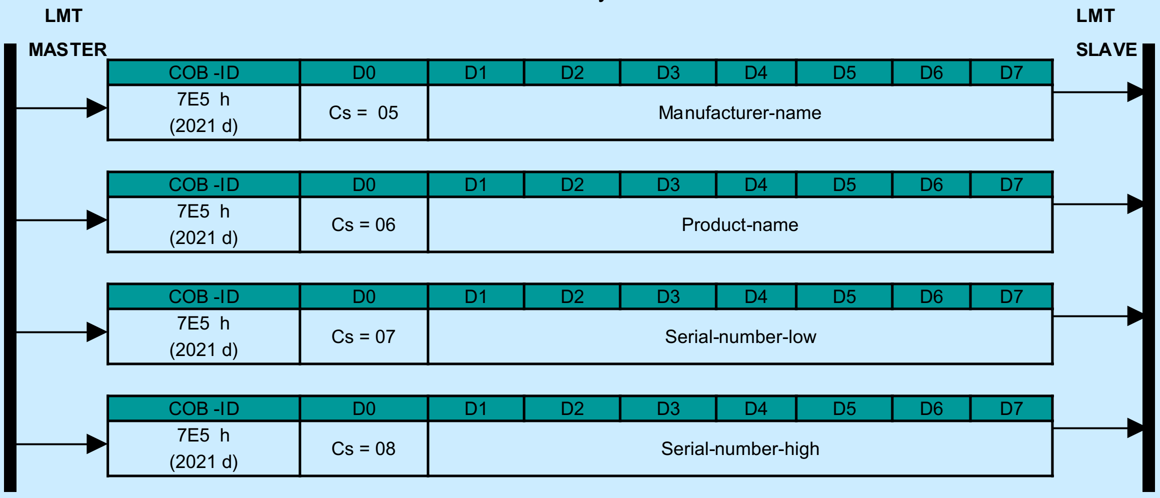
38d (26 h) for Inquire Serial number

. **S1 - S7**: The product_name of the selected module or error code.If S1 contains a valid BCD-pair, the response contains the serial number. If S1 is 0ffh, S2 contains the error code, S3 contains the reason if valid for the error code.

. **Reserved**: reserved for further use by CiA

LMT Identity Remote Slaves

LMT Identity Remote Slaves



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 05	Manufacturer-name						
7E5 h (2021 d)	Cs = 06	Product-name						
7E5 h (2021 d)	Cs = 07	Serial-number-low						
7E5 h (2021 d)	Cs = 08	Serial-number-high						

. Cs:

LMT command specifier

05d to 08d for LMT Identify Remote Slaves

. Manufacture_name:

the manufacturer name part of the LSS Address

. Product_name:

The product name part of the LMT Address

. serial-number-low:

The lower boundary of the requested serial numbers range.

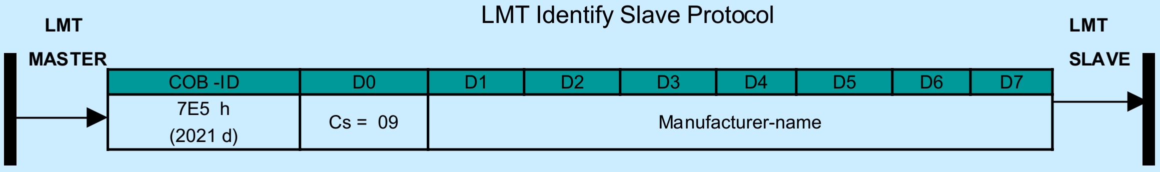
. Serial-number-high:

The higher boundary of the requested serial numbers range

The boundaries are included in the interval. All LMT Slaves with matching manufacturer name and product name whose serial numbers lie within this range, are requested to identify themselves with the LMT identify Slave service.

LMT Identify Slave Protocol

LMT Identify Slave Protocol



COB -ID	D0	D1	D2	D3	D4	D5	D6	D7
7E5 h (2021 d)	Cs = 09	Manufacturer-name						

. Cs:

LMT command specifier

09d for Identify Slaves

. Reserved: all bytes set to ‘0’

Example

Here are some examples on how you can configure the network nodes.

a) A new node is entered into the network

The master requires the three identification parameters of the bus nodes.

The “MK2C000” transducer has to be entered into the network using “01 48 20 45 00 00 00” serial number (default Node-ID =127d).

	COB -ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7	
Master	7E5	8	04	01	x	x	x	x	x	x	Change to Configuration Mode
Master	7E5	8	24	01	x	x	x	x	x	x	Request for the Manufacturer’s Name
Slave	7E4	8	24	47	45	46	52	41	4E	73	Answer: GEFRANs
Master	7E5	8	25	01	x	x	x	x	x	x	Request for the Device Name
Slave	7E4	8	25	4D	4B	32	43	30	30	30	Answer: MK2C000
Master	7E5	8	26	01	x	x	x	x	x	x	Request for the Serial Number
Slave	7E4	8	26	01	48	20	45	00	00	00	Answer: serial number
Master	7E5	8	04	00	x	x	x	x	x	x	Change to Operating Mode

b) The master assigns a Node-ID=3 to the new transducer.

	COB -ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7	
Master	7E5	8	01	47	45	46	52	41	4E	73	Identification of GEFRAN network devices
Master	7E5	8	02	4D	4B	32	43	30	30	30	Identification of the devices having MK2C000 as name
Master	7E5	8	02	01	48	20	45	00	00	00	The device with this serial number changes to configuration mode
Master	7E5	8	11	3	x	x	x	x	x	x	The device is assigned node number 3
Slave	7E4	8	11	0	0	x	x	x	x	x	The device replies er=0 (positive outcome)
Master	7E5	8	17	X	x	x	x	x	x	x	Storage command
Slave	7E4	8	17	0	0	x	x	x	x	x	The device replies er=0 (positive outcome)
Master	7E5	8	04	00	x	x	x	x	x	x	Back to the operating mode

c) The Master wants to change the bus baud rate (bit-time) to/into 100Kbit/sec.

After selecting the baud rate, a time beyond which the new bit time automatically activates can be selected (without waiting for the power on).

	COB -ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7	
Master	7E5	8	04	01	x	x	x	x	x	x	All nodes change into the configuration mode
Master	7E5	8	13	00	05	x	x	x	x	x	Request for a new baud rate
Slave	7E4	8	13	00	00	x	x	x	x	x	The device replies er=0 (positive outcome)
Master	7E5	8	17	x	x	x	x	x	x	x	Storage command
Slave	7E4	8	17	0	0	x	x	x	x	x	The device replies er=0 (positive outcome)
Master	7E5	8	15	30	75	x	x	x	x	x	30-sec wait to activate the new baud rate
Master	7E5	8	04	00	x	x	x	x	x	x	All nodes jump back to the operating mode

1.3.0 Trasmitting data via the PDO

The PDO is a message through which you can access the transducer information. It is active when in OPERATIONAL status.

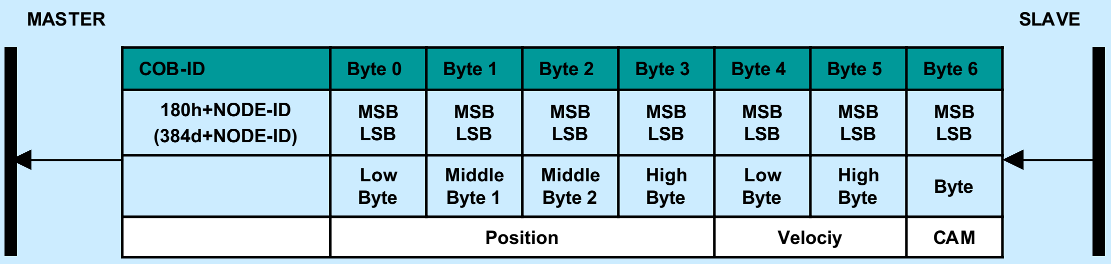
The PDO allows a real time transfer of some information available in the Dictionary of the Objects.

The Objects of the Dictionary transmitted via the PDO are defined by means of the mapping procedure (object 1A00h and 1A01h).

Through 1800h and 1801h objects you can define the PDO communication parameters: COB-ID, as well as transmission type and time.

Data format:

Default mapping provides the transmission of position, 32 integer bit, speed, 16 integer bit, and of cam statuses, 8 unsigned bit.
A PDO is active for each of the available cursors.



COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
180h+NODE-ID (384d+NODE-ID)	MSB LSB	MSB LSB	MSB LSB	MSB LSB	MSB LSB	MSB LSB	MSB LSB
	Low Byte	Middle Byte 1	Middle Byte 2	High Byte	Low Byte	High Byte	Byte
	Position				Velociy		CAM

Fig. 2–1: Default structure of the PDO1, first cursor.

Cam status in the PDO

The position of the cursor as to the areas identified by the Cams can be controlled through this Status byte.

"State_register (6300h), Enable_register (6301h), Polarity_ _register (6302h), Low and High_Limit_register " registers, defining Cam functional parameters, can be read and set via the SDOs :

The Status byte in the State_register is configured as follows: (see table)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	CAM_No_4	CAM_No_3	CAM_No_2	CAM_No_1
-	-	-	-	State	State	State	State

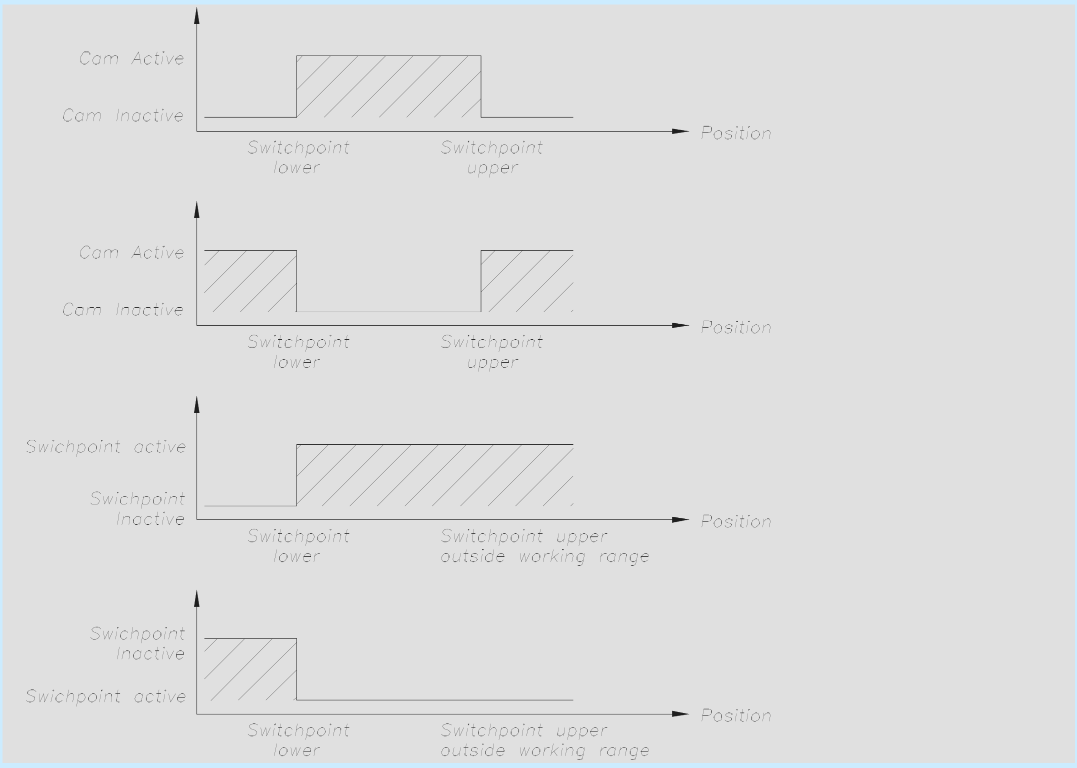
Bits 0 to 3 contain the current switching state of the respective cam; depending on the configuration, the active state of the cam is indicated by a 0 or by a 1 :

0 = Low state

1 = High state

Any change in the active cam leads to a change of the cam status in the Emergency Object.

The State_register, Enable_register and Polarity_register have the same structure



1.4.0 Object Directory

The most important part of a device profile is the Object Dictionary description.
The Object Dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. Each object within the dictionary is addressed using a 16-bit index.
The overall layout of the standard Object Dictionary is shown below.

Table 1: Object Dictionary Structure

INDEX (HEX)	OBJECT
0000	not used
0001-001F	Static Data Types
0020-003F	Complex Data Types
0040-005F	Manufacturer Specific Complex Data Types
0060-007F	Device Profile Specific Static Data Types
0080-009F	Device Profile Specific Complex Data Types
00A0-0FFF	Reserved for further use
1000-1FFF	Communication Profile Area
2000-5FFF	Manufacturer Specific Profile Area
6000-9FFF	Standardised Device Profile Area
A000-FFFF	Reserved for further use

1.4.1 Communication Profile

The parameters which are critical for communication are determined in the communication profile.

ro = The parameter cam be read only
rw = The parameter can be read and also configured with a writen

Idex	Su b- ide x	Name	Type	Attr- bute	Default- value	Meaning
1000	0	Device type	Unsigned32	ro	0008 0196 000A 0196	D. profile 406, absolute linear encoder with one cursor D. profile 406, absolute linear encoder with two cursors
1001	0	Error register	Unsigned8	ro	0	No error
1003	0	Predefined error field	Unsigned8	ro	0	Number of different errors which occurred
	1		Unsigned32	ro	no	
	2		Unsigned 32	ro	no	
	…					
	08		Unsigned32	ro	No	
1004	0	Number of PDOs supported	Unsigned32	ro	2	Total number of PDOs supported
	1		Unsigned32	ro	2	Number of synchronous PDOs
	2		Unsigned32	ro	2	Number of asynchronous PDOs
1005	0	COB-ID Sync message	Unsigned32	rw	80h	COB-ID of the SYNC object
1008	0	Manufacturer device name	Visible String[4]	ro	MK2C	Device name of the communication module
1009	0	Manufacturer hardware version	Visible String[4]	ro	0001	Hardware version number
100A	0	Manufacturer software version	Visible String[4]	ro	0001	Software version number
100B	0	Node-ID	Unsigned32	ro	7Fh	Node number ; can be set using the LMT and LSS

Structure of the communication profile

Idex	Sub - idex	Name	Type	Attri - bute	Default-value	Meaning
100C	0	Guard time	Unsigned 16	rw	0	Cycle time in ms
100D	0	Life time factor	Unsigned 8	rw	0	Guard time multiplier
100E	0	Node guarding identifier	Unsigned 32	rw	700h+ Nodel ID	Identifier for node guarding.
100F	0	Numbers of SDOs	Unsigned 32	ro	1	Numbers of SDOs supported.
1010	0	Store parameter	Unsigned 8	ro	4	Numbers of store options.
	1	Store all parameters	Unsigned 32	Ro	No	By writting the signature 'save'
	2	Store communication parameters	Unsigned 32	Ro	No	0x65766173, all setting are saved
	3	Store device parameter	Unsigned 32	Ro	No	on the module.
	4	Store Manufacture parameters	Unsigned 32	Ro	no	
1011	0	Restore parameter	Unsigned 8	ro	4	Number of Restore options.
	1	Restore all parameters	Unsigned 32	Ro	No	By writing the signature 'load' 0x64616F6C the factory default setting are loaded.
	2	Restore communication parameters	Unsigned 32	Ro	No	
	3	Restore device parameter	Unsigned 32	Ro	No	
	4	Restore Manufacture parameters	Unsigned 32	Ro	no	
1014	0	COB-ID Emergency Message	Unsigned 32	rw	80h+ Nobel ID	COB-ID of the Emergency Object.
1017	0	Producer heartbeat time	Unsigned 16	rw	0	Defines the cycle time of the heartbeat. It is 0 if it is not used. The time has to be a multiple of 1 ms.
1018	4	Identity object	Unsigned 8	Ro	4	General information about the device
	1	Vendor ID	Unsigned 32	Ro	00000093	
	2	Product code	Unsigned 32	Ro	4D4B3243	
	3	Revision number	Unsigned 32	Ro	00000001	
	4	Serial Number	Unsigned 32	Ro	Xxxxxxxx	
1200	0	Server SDOs	Unsigned 8	ro	1	Numbers of server SDOs.
	1	COB-ID SDO-rx	Unsigned 32	ro	600h+ Nodel ID	COB-ID for request.
	2	COB-ID SDOtx	Unsigned 32	ro	580h+Node-ID	COB-ID for response
1800	0	Numbers of elements	Unsigned 8	ro	5	Communication parameters of 1st Trasmit PDO.
	1	COB-ID	Unsigned 32	rw	180h+ Nodel ID	CANopen minimum system ID assignment
	2	Trasmission Type	Unsigned 8	rw	FEh	See table 6- 2c
	3	Inhibit Time	Unsigned 16	Rw	0	
	4	Reserved	-	-	-	
	5	Event timer	Unsigned 16	Rw	0Ah	

Structure of the Communication Profile

Idex	Sub-idex	Name	Type	Attri-bute	Default-value	Meaning
1801	0	Numbers of elements	Unsigned 8	ro	5	Communication parameters of 2nd Trasmit PDO.
	1	COB-ID	Unsigned 32	rw	180h+ Nodel ID	CANopen minimum system ID assignment
	2	Trasmission Type	Unsigned 8	rw	FEh	See table 6- 2c
	3	Inhibit Time	Unsigned 16	Rw	0	
	4	Reserved	-	-	0	
	5	Event timer	Unsigned 16	Rw	0A	
1A00	0	Numbers of elements	Unsigned 8	rw	8	Mapping parameters of the 1st Trasmit-PDO
	1	1st object	Unsigned 32	rw	60200120h	Position 1
	2	2nd object	Unsigned 32	rw	60300110h	Velocity 1
	3	3rd object	Unsigned 32	rw	63000108h	CAM State Channel 1
	4	4rd object	-	rw	-	
	5	5rd object	-	rw	-	
	6	6rd object	-	rw	-	
	7	7rd object	-	rw	-	
	8	8rd object	-	Rw	-	
1A01	0	Numbers of elements	Unsigned 8	rw	3	Mapping parameters of the 2nd Trasmit-PDO
	1	1st object	Unsigned 32	rw	60200220h	Position 2
	2	2nd object	Unsigned 32	rw	60300210h	Velocity 2
	3	3rd object	Unsigned 32	rw	63000208h	CAM State Channel 2
	4	4rd object	-	rw	-	
	5	5rd object	-	rw	-	
	6	6rd object	-	rw	-	
	7	7rd object	-	rw	-	
	8	8rd object	-	Rw	-	

Structure of the Communication Profile

Object 1000h – Device Type

Describes the type of the device and its functionality. It is composed of a 16-bit field which describes the device profile that is used and a second 16-bit field which gives additional information about option functionality of the device.

16 bit field = 2 bytes (2^7-2^0||2^{15}-2^8)	16 bit field = 2 bytes (2^{23}-2^{16}||2^{31}-2^{24})
Device Profile Number	Additional information

Structure of the Device Type Parameter

Example:Reading object (unsigned 32):

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CS	Index		Sub-index	Data			
Master	40	00	10	00				
Slave	43	00	10	00	96	01	08	00
					Device Profile Number		**Additional Information**	

The master inquires the device type, the trasducer answers 0196 0008h:

0196h =406d = "Device Profile for Encoder"

0008h =008d = absolute linear encoder

Object 1001h – Error Register

Is an error register for the device. The device map internal errors in this byte. It is part of the Emergency object. If a bit is set to 1 the specific error has occured. The generic error is signaled at any error situation.

Example: Reading Object (unsigned 8)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CS	**Index**		**Sub-index**	**Data**			
Master	40	01	10	00				
Slave	4F	01	10	00	XX			
					Register			

The master inquires the error register, the trasducer answers the internal error map

Error Register Bits: 0	Generic error
1	Current
2	Voltage
3	Temperature
4	Communication error
5	Device profile specific
6	Reserved
7	Manufacturer specific

Structure of the Error Register

Object 1003h – Pre-defined Error Field

Holds the errors that have occured on the device and have been signaled via the emergency Object Object. In doing so it provides an error history.

- The entry at sub-index 0 contains the number of actual errors that are recorded in the array starting at sub-index 1.
- Every new error is stored at sub-index 1, older ones move down the list.
- Writing a “0” to sub-index 0 deletes the entire error history (empties the array).
- The error numbers are of the type unsigned 32 and are composed of a 16 bit error code and 16 bit additional error information field which is manufacturer specific. The error code is contained in the lower 2 bytes (LSB) and the additional information is included in the upper 2 bytes (MSB)

A maximum of 8 different errors is managed.

16 bit field = 2 bytes (2^7-2^0||2^{15}-2^8)	16 bit field = 2 bytes (2^{23}-2^{16}||2^{31}-2^{24})
Additional information	Error code
Structure of the pre-defined error field	

Example Reading object (unsigned 8)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub-index**	**Data**			
Master	40	03	10	00				
Server	4F	03	10	00	0			

The master inquires the number of different errror occurred; the transducer answers 0.

Object 1004h – Number of PDOs supported

Contain information about the maximum number of PDOs supported by the device. It distinguished between input and output PDOs and between synchonous and asynchonous trasmission. Sub-index 0 describes the overall number of the PDOs supported. Sub-index 1 describes the number of syncronous PDOs that is supported by the device. Sub-index 2 the number of asyncronous PDOs that is supported.

Example Reading object (unsigned 8)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub-index**	**Data**			
Master	40	04	10	00				
Server	4F	05	10	00	2			

The master inquires the maximum number of PDOs supported; the transducer answers 2.

Sub-index	Byte MSB	Byte LSB
0	No. of Receive PDOs supported	No. of Trasmit PDOs supported
1	No. of synchonous Receive PDOs	No. of synchonous Trasmit PDOs
2	No. of asynchonous Receive PDOs	No. of asynchonous Trasmit PDOs
Structure of entry 1004h		

Object 1005h – COB-ID SYNC message

Defines the COB-ID of the Syncrhonisation Object

Bit	31 (MSB)	30	29	28-11	10-0 (LSB)
11-bit-ID	X	0/1	0	000000000000000000	11-bit Identifier
29-bit-ID	X	0/1	1	29 bit Identifier	
Structure of SYNC COB-ID					

Bit number	value	Meaning
31 (MSB)	X	Do not care
30	0 1	Device does not generate SYNC message Device generates SYNC message
29	0 1	11-bit ID (CAN 2.0A) 29-bit ID (CAN 2.0B)
28-11	0 X	If bit 29=0 If bit 29=1: bits 28-11 of 29-bit-SYNC-COB-ID
10-0(LSB)	X	Bit 10-0 of SYNC-COB-ID
Description of SYNC COB-ID entry		

Example Reading object (unsigned 32):

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub-index**	**Data**			
Master	40	05	10	00	0-10 bits	11 – 30 bits		31
Server	43	05	10	00	80	00	0	0
					80 h		**0**	0

The master inquires the SYNC COB-ID; the transducer answers 80h,

The 31st bit is 0 if the transducer doesn't consume SYNC message

1 if the transducer consumes SYNC message

COB – ID SYNC message, writing (unsigned 32):

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Subindex**	**Data**			
Master	22	05	10	00	0-10 bits	11-30 bits		31
Server	60	05	10	00	20	01		01

The master sets a new value of SYNC COB-ID = 120h and the transducer must consume the SYNC message, the transducer answers with an acknowledge.

Index 1008h; sub – index 00

Manufacturer Device Name: Contain the manufactur device name.

Reading (visible string):

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub-index**	**Data**			
Master	40	08	10	00				
Server	43	08	10	00	43	32	4B	4D
					“C”	**“2”**	**“K”**	**“M”**

The master inquires the device name, the transducer answers: MK2C.

Index 1009h; sub – index 00

Manufacturer Hardware Version: contain the manufacturer hardware version description.

Reading (visible string):

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub-index**	**Data**			
Master	40	09	10	00				
Server	43	09	10	00	31	30	30	30
					“1”	**“0”**	**“0”**	**“0”**

The master inquires the harware version, the transducer answers: 00 01

Index 100Ah; sub – index 00

Manufacturer Software Version:contain the manufacturer software version description.

Reading (visible string):

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub-index**	**Data**			
Master	40	0A	10	00				
Server	43	0A	10	00	31	30	30	30
					“1”	“0”	“0”	“0”

The master inquires the harware version, the transducer answers: 00 01

Index 100Bh; sub – index 00

Contain the Node-ID. The node_ID entry has the access type “read only” as it can not be changing using SDO services. However, it is feasible to change it via LMT and LSS. Reading (visible string):

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub-Index**	**Data**			
Master	40	0B	10	00				
Server	4F	0B	10	00	7F			
					Node - ID			

The master inquires the Node-ID, the transducer answers: 7Fh (127d).

Index 100Ch; sub – index 00

Guard Time:The Objects at index 100C and 100D include the guard time in milliseconds and the life time factor. The life time factor multiplied with the guard time gives the life time for the life Guarding Protocol. It is 0 if not used.

Reading (unsigned 16):

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub-index**	**Data**			
Master	40	0C	10	00				
Server	4F	0C	10	00	00			

When the master request the guard time, the transducer sends 00h

Guard Time: writing object (unsigned 16)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub-index**	**Data**			
Master	22	0C	10	00	05			
Server	60	0C	10	00				

The master sets a new value of guard time = 05 msec, the transducers answers with an acknowledge.

The guard time multipled with the life time factor gives the life time for the Node Guarding Protocol

Index 100Dh; sub – index 00

Life Time Factor:The life time factor multiplied with che guard time gives the life time for the node guarding protocol. It is 0 if not used(read, unsigned 16)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	40	0D	10	00				
Server	4F	0D	10	00	00			

When the master inquires the life time factor, the transducer sends 00h

Life Time Factor: read, unsigned 16

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	22	0D	10	00	01	A0		
Server	60	0D	10	00				

The master sets a new value of guard time = A001h, the transducers answers with an acknowledge.

The life time factor multipled with the guard time gives the life time for the Node Guarding Protocol

Index 100Eh; sub – index 00

Node Guarding Identifier NG - ID:

The identifier used for node guarding and life guarding procedure. The structure is to object 1005h.

read, unsigned 32

Bit	31 (MSB)	30	29	28-11	10-0 (LSB)
11-bit-ID	reserved		0	000000000000000000	11-bit Identifier
29-bit-ID	reserved		1	29 bit Identifier	
Structure of Node Guarding Identifier entry					

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	40	0E	10	00				
					0-10 bits	11-31 bits		
Server	43	0E	10	00	Node-ID	07	0	0
					700h + node - ID			

When che master inquires the NG – ID, the transducer sends 77Fh.

The transducer accepts only 11 bit identifier format (CAN 2.0A)

Object 100Fh – Number of SDOs supported

Contain information about the number of SDOs supported by the device.

Object 1010h – Store parameters

This object supports the saving of parameters in non volatile memory. By read access the device provides information about its savign capabilities. Several parameter groups are distinguished:

Sub-Index 0 contains the largest Sub-Index that is supported.

Sub-Index 1 refers to all parameters that can be stored on the device.

Sub-Index 2 refers to communication related parameters (Index 1000h – 1FFFh manufacturer speciffic communication parameters)

Sub-Index 3 refers to application related parameters (Index 6000h – 9FFFh manufacturere specific application parameters)

In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate sub-index. The signature is “save” .

Signature ISO 8859 (“ASCII”) hex	MSB	Byte LSB	Byte LSB	Byte LSB
	e	v	a	s
	65h	76h	61h	73h
Storage writen access signature				

Index 1010h; sub – index 00

(read, unsigned 8)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
		Index		**Sub. 1**	**Data**			
Master	40	10	10	00				
Server	4F	10	10	00	4			

When the master inquires the largest Sub-index that is supported, the transducer sends 4.

The device support 4 sub-index

- Save all parameter in the object dictionary
- Save the communication parameter (Index 1000h – 1FFFh)
- Save the standardised Device Profile parameters (Index 6000h – 9FFFh)
- Save the Manufacture Specific Profile parameters (Index 2000h – 5FFFh)

Index 1010h; sub – index 01/02/03/04

Store parameter: read, unsigned 32

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
		Index		**Sub. 1**	**Data**			
Master	40	10	10	1/2/3/4				
Server	43	10	10	1/2/3/4	0	0	0	0

When the master inquires information about saving capabilities, the transducers sends 0.

The device saves parameter on command and does not save parameters autonomously

Store parameter: read, unsigned 32

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	22	10	10	1/2/3/4	73	61	76	65
					“s”	“a”	“v”	“e”
Server	60	10	10	1/2/3/4				

The master stores the actual parameter in the eprom of the device, the transducer answers with an acknowledge.

If a wrong signature is written, the device refuses to store and responds with abort domain transfer.

1011h – 00 Restore Default Parameter: Whith this object the default values of parameters according to the communication or device profile are restored By read access the device provides information about its capabilities to restore these values.

Several parameter groups are distinguisched:

Sub-Index 0 contains the largest Sub-Index that is supported.

Sub-Index 1 refers to all parameters that can be restored.

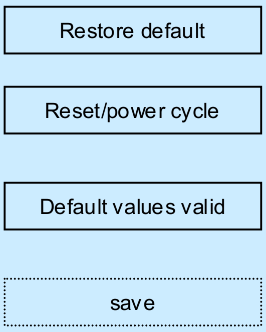
Sub-Index 2 refers to communication related parameters (Index 1000h – 1FFFh manufacturer specific communication parameters)

Sub-Index 3 refers to application related parameters (Index 6000h – 9FFFh manufacturer specific application parameters)

Signature ISO 8859 ("ASCII") hex	MSB	Byte LSB	Byte LSB	Byte LSB
	d	a	o	l
	64h	61h	6Fh	6Ch
Restoring writen access signature				

The default values are set valid after the device is reset (reset node for sub-index 1h – 7Fh, reset communication for sub-index 2h) or power cycled. If the device requires storing on command (see Object 1010h), the appropriate command has to be executed after the reset if the default parameters are to be stored permanently.

(read only, unsigned 32)



Index 1011h; sub – index 00

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	40	11	10	00				
Server	4F	11	10	00	3			

When the master inquires the largest Sub – Index that is supported, the transducer sends 3.

The device support 3 sub – index:

- Restores all default parameter in the object dictionary
- Restore the Communication parameter (Index 1000h – 1FFFh)
- Restore the standardised Device Profile parameters (Index 6000h – 9FFFh)
- Restore the Manufacture Specific Profile parameters (Index 2000h – 5FFFh)

Index 1011h; sub – index 01/02/03/04

Restore Default Parameter: read, unsigned 32

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	40	11	10	1/2/3/4				
Server	4F	11	10	1/2/3/4	1	0	0	0

When the master inquires information about capabilities to restore, the transducer sends 1.

The device has the capability to restore the dafault parameter.

Restore Default Parameter: write, unsigned 32

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	22	11	10	1/2/3/4	6C	6F	61	64
					"l"	"o"	"a"	"d"
Server	60	11	10	1/2/3/4				

The master restores the default parameter, the transducer answers with an acknowledge.

If a wrong signature is written, the device refuses to store and responds with abort domain transfer.

The default values are set valid after a new power-on or after the device is reset:

- reset node (see NMT messages) for restore of sub index 1, 2, 3, and 4

-- reset comminication (see NMT messages) for store of only sub index 2.

Object 1014h: COB – ID Emergency Object

Index 1014h defines the COB-ID of the Emergency Object (EMCY).

Bit	31 (MSB)	30	29	28-11	10-0 (LSB)
11-bit-ID	Reserved (=0)		0	000000000000000000	11-bit Identifier
29-bit-ID	Reserved (=0)		1	29 bit Identifier	

Structure of the EMCY Identifier entry

Object 1017h: Producer Heartbeat Time

The producer hartbeat time defines the cycle time of the heartbeat. The producer heartbeat time is 0 if it not used. The time has to be a multiple of 1 ms.

Object 1018h : Identity Object

The object at index 1018h contains general information about the device.

The Vendor ID (sub-index 1h) contains a unique value allocated to eache manufacturer.

The manufacturer specific Product code (sub-index 2h) identifies a specific device version.

The manufacturer specific Revision number (sub-index 3h) consists of a major revision number and a minor revision number.

Bit 31 (MSB) bit 16	bit 15 bit 0 (LSB)
Major revision number	Minor revision number

Structure of Revision number

Object 1200h – 127Fh: Server SDO Parameter

In order to describe the SDOs used on a device the data type SDO Parameter is introduced.

Bit	31 (MSB)	30	29	28-11	10-0 (LSB)
11-bit-ID	0/1	0	0	000000000000000000	11-bit Identifier
29-bit-ID	0/1	0	1	29 bit Identifier	

Structure of SDO COB-ID entry

Bit number	value	Meaning
31 (MSB)	0	SDO valid
	1	SDO not valid
30	0	Reserved (=0)
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
28-11	0	If bit 29=0
	X	If bit 29=1: bits 28-11 of 29-bit-SYNC-COB-ID
10-0(LSB)	X	Bit 10-0 of SYNC-COB-ID

Description of SDO COB-ID entry

Object 1800h – 1801h: Transmit PDO Comminication Parameter

Contains the comminication parameters for the PDOs che device is able to transmit. The sub-index 0h contains the number of valid entries within the communication record. Its value is 5. At sub-index 1h resides the COB-ID of the PDO. This entry has been defined as UNSIGNED32 in order to cater for 11-bit CAN identifiers (CAN2.0A) as well as for 29-bit CAN identifiers (CAN 2.0B) The entry has to be interpreted as defined in Fig. A and table a

UNSIGNED32

Bits	31	30	29	28-11	10-0
11-bit-ID	0/1	0/1	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-bit identifier
29-bit-ID	0/1	0/1	1	29-bit identifier	

Fig. A: Structure of PDO COB-ID entry

Table a: Description of PDO COB-ID entry

Bit number	Value	Meaning
31(MSB)	0	PDO valid
	1	PDO not valid
30	0	RTR allowed on this PDO
	1	No RTR allowed on this PDO
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
28 – 11	0	If bit 29=0
	X	If bit 29=1; bits 28-11 of 29-bit-COB-ID
10-0 (LSB)	X	Bit 10-0 of COB-ID

The PDO valid/not valid allows to select which PDOs are used in the operational state.

The transmission type (sub-index 2) defines the transmission character of the PDO. Table b describes the usage of this entry. On an attempt to change the value of the transmission type to a value that is not supported by the device an abort message (abort code: 0609 0030h) is generated.

Table b: Description of transmission type

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

Synchronous (transmission types 0 – 240 and 252) neans that the transmission of the PDO shall be related to the SYNC object as described in 9.3. Preferably the devices use the SYNC as a trigger to output or actuate based on the previous synchronous Receive PDO respectively to update the data transmitted at the following synchronous Transmit PDO. Details of this mechanism depend on the device types and are defined in the device profile if applicable.

Asynchronous means that the transmission of the PDO is not related to the SYNC object.

A transmission type of zero means that the message shall be transmitted synchronously with the SYNC object but not periodically.

A value between 1 and 240 means that the PDO is transferred synchronously and cyclically, the transmission type indicating the number of SYNC which are necessary to trigger PDO transmissions/receptions.

The transmission types252 and 253 mean that the PDO is only transmitted on remote transmission request.

At transmission type 252, the data is updated (but not sent) immediately after reception of the SYNC object.

At trasmission type 253 the data is updated at the reception of the remote transmission request (harware and software restrictions may apply). These value are only possible for TPDOs.

For TPDOs transmission type 254 means, the application event is manufacturer specific (manufacturer specific part of the Object Dictionary), transmission type 255 means, the appliaction event is defined in the device profile. RPDOs with that type trigger the update of the mapped data with the reception.

Sub-index 3h contains the inhibit time. This time is a minimum interval for PDO transmission.

The value is defined as multiple of 100µs.

Sub-index 4h is reserved. It does not have to be implemented, in this case read or write access leads to Abort SDO Transfer (abort code: 0609 0011h):

In mode 254/255 additionally an event time can be used for TPDO. If an event timer exists for a TPDO (value not equal to 0) the elapsed timer is considered to be an event. The event timer elapses as multiple of 1 ms of the entry in sub-index 5h of the TPDO. This event will cause the transmission of this TPDO in addition to otherwise defined events. The occurence of the events set the timer. Indipendent of the transmission type the RPDO event timer is used recognize the expiration of the RPDO.

Index 1800h; sub – index 01

Transmit PDO-1 Communication Parameter, COB – ID: Contains the communication parameters for the PDOs the device is able to transmit. The type of the PDO communication parameter (20h) is described
(read write, unsigned 8)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	40	00	18	01				
					0 – 10 bits	11 – 30 bits		31
Server	43	00	1801	01	FF	01	0	0
					180h + node-ID		0	0

When the master inquires the number of entries, the transducer sends 5.

Index 1800h; sub – index 01

Transmit PDO-1 Communication Parameter, COB – ID: Contains the communication parameters for the PDOs the device is able to transmit. The type of the PDO communication parameter (20h) is described
(read write, unsigned 8)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	40	00	18	01				
					0 – 10 bits	11 – 30 bits		31
Server	43	00	1801	01	FF	01	0	0
					180h + node-ID		0	0

When the master inquires the COB-ID used by PD01, the transducer sends 180h + node-ID

The 31st bit is 0 if PDO-1 valid

1 if PDO-1 not valid

RTR allowed on this PDO

The transducer accepts only 11 bit identifier format (CAN 2.0A)

Transmit PDO-1 Communication Parameter, COB - ID: write, unsigned 32

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
					0-10 bits	11-30 bits		31
Master	22	00	18	01	A5	06	0	00
Server	60	00	18	01				

The master sets a new value of PDO COB-ID=6A5h and the PDO valid, the transducer answers with an acknowledge.

The transducer accepts only 11 bit identifier format (CAN 2.0A).

Index 1800h; sub – index 02

Transmit PDO-1 Communication Parameter, transmission type: read, unsigned 8

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	40	00	18	02				
Server	4F	00	18	02	FE			

When the master inquires the transmission type, the transducer sends FE h.

Is possible change the transmission type of PDO-1

TRANS. TYPE	MEANING
0	Non implemented
1 – 240	The PDO is transmitted synchronously with the SYNC message and cyclically after to have received a number 1 – 240 SYNC objects.
241 - 251	Reserved
252	The PDO is transmitted after to have received a RTR. The Data is updated but not sent immediately after reception of the SYNC object. Value
253	When the device receives a RTR the Data is updated, and the PDO is transmitted
254	The PDO is transmitted synchronously with the internal timer of device and cyclically with a settable transmission rate, 1 msec to about 1 minute (see index 2004)
255	Not implemented

(see table 10 – 7: Description of transmission type of CiA DS301 page 10 – 6)

Transmit PDO-1 Communication Parameter, transmission type:write, unsigned 8

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	22	00	18	02	0A			
Server	60	00	18	02				

The master sets a new value of transmission type =0Ah, the transducer answers with an acknowledge.

Type 0A the device send the PDO-1 after receive 10 SYNC message

It is necessary to active the SYNC message consumer (see 1005h).

Index 1800h; sub – index 05

Transmit PDO-1 Communication Parameter, event timer:read, unsigned 16

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	40	00	18	05				
Server	4B	00	18	05	0A	00		

When the master inquires the event timer, the transducer sends 000Ah.

So if the Transmission type=Feh the device send the PDO-1 with a period of 10msec.

PDO-1 Transmission Rate: write, unsigned 16

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	22	0A	75	01	64	00		
Server	60	0A	75	01				

The master sets PDO-1 Transmission Rate =0064h so id ghe Transmission type =Feh the device send the PDO-1 with a period of 100 msec.

Transmission rate for transmit PDO: < 1 to 2 ^ 16 msec >.

Index 1801h

Transmit PDO- 2 Communication Parameter

It define the parameter of the second PDO like for Index 1800.

It is possible to set the two Event Timer also by the object dictionary 6200

Object 1A00h – 1A01h: Transmit PDO Comminication Parameter

The sub-index 0h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be transmitted with the corresponding PDO.

The sub-indices from 1h to number of entries contain the information about the mapped application variables.

These entries describe the PDO contents by their index, sub-index and length. All three values are hexadecimal coded. The lengh entry contains the length of the object in bit (1..40h).This parameter can be used to verify the overall mapping length. It is mandatory.

The structure of the entries from sub-index 1h – 40h is as follows:

Byte: MSB

Index (16 bit)	Sub-index (8 bit)	Object length (8 bit)

Structure of PDO Mapping Entry

If the change of the PDO mapping cannot be executed (e.g. the PDO length is exceede or the SDO client attempts to map an object that cannot be mapped) the device responds with an Abort SDO Transfer Service.

Subindex 0 determines the valid number of objects that have been mapped. For changing the PDO mapping first subindex 0 must be set to 0 (mapping is deactivated). Then the objects can be remapped. When a new object is mapped by wrinting a subindex between 1 and 64, the device may check whether the object specified by index/subindex exists. If the object does not exist or the object cannot be mapped, the SDO transfer must be aborted with the Abort SDO Transfer Service with one of the abort codes 0602 0000h or 0604 0041h.

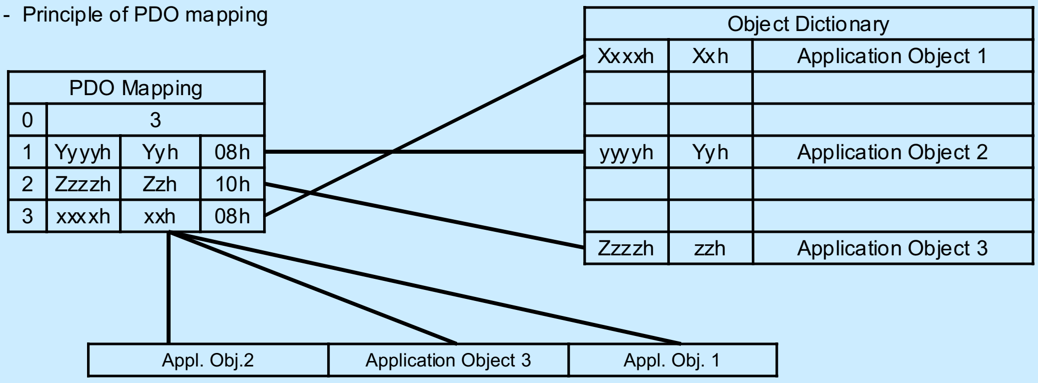
After all objects are mapped subindex 0 is set to the valid number of mapped objects. When subindex 0 is set to a value>0 the device may validate the new PDO mapping before transmitting the response of the SDO service. If an error is detected the device has to transmit the Arbort SDO Transfer Service with one of the abort codes 0602 0000h, 0604 0042h.

When subindex 0 is read the actual number of valid mapped objects is returned.

If data types (Index 1h-7h) are mapped they serve as “dummy entries”. The corresponding data in the PDO is not evaluated by the device. This optional feature is useful e.g. to transmit data to several devices using one PDO, each device only utilising a part of the PDO. It is not possible to create a dummy mapping for a TPDO.

A device that supports dynamic mapping of PDOs must support this during the state PREOPERATIONAL state. If dynamic mapping during the state OPERATIONAL is supported, the SDO client is responsible for data consistency.

- Principle of PDO mapping



Contains the communication parameters for the PDOs the device is able to transmit. The type of the PDO communication parameter (20h) is described.

Contains the mapping for the PDOs the device is able to transmit

Index 1A00h; sub index 00

Transmit PDO- 1 Mapping Parameter, number of mapped application object:Contains the mapping for the PDOs the device is able to transmit. (read only, unsigned 8)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	40	00	1A	00				
Server	4F	00	1A	00	8			

When the master inquires the number of entries, the transducer sends 8

Index 1A00h; sub index 01

Transmit PDO- 1 Mapping Parameter, 1st. Application object: read only, unsigned 32

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	40	00	1A	01				
Server	43	00	1A	01	20	01	20	60
					length	Sub index	Index	

Index 1A00h; sub index 02

Transmit PDO- 1 Mapping Parameter, 2nd. Application object: read only, unsigned 32

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	40	00	1A	02				
Server	43	00	1A	02	10	01	30	60
					length	Sub index	Index	

Index 1A00h; sub index 03

Transmit PDO- 1 Mapping Parameter, 3rd. Application object: read only, unsigned 32

When the master inquires 2nd object mapping of PDO-1, the transducer sends the index 6030 and sub index 01, that correspond to Speed Value cursor 1 and its length, 16 bits.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Cs	**Index**		**Sub. 1**	**Data**			
Master	40	00	1A	03				
Server	43	00	1A	03	08	01	00	63
					length	Sub index	Index	

When the master inquires 3rd object mapping of PDO – 1, the transducer sends the index 6300 and sub index 01, that correspond to CAM State cursor I and its length, 8 bits.

Index 1A01h; sub index 00/01/02/03

Transmit PDO- 2 Mapping Parameter: read only

It is the same of index 1A00h

1.5.0 Encoder Profile DS 406

This Device profile represents the CANopen device profile for incremental and absolute, linear encoders. Besides position and velocyity output possibility complete cam functionality is covered.

For fast comparison of the switch-points and the working range, all values are stored un_signed as absolute values with 5 µm resolution in the object directory.

RO = read only; RW= read and write

Index	Sub-Index	Name	Type	Attri-bute	Default-value	Meaning
6000	0	Operating Parameter	Unsigned 16	rw	0x04H	Operating parameter
6003	0	Preset value	Unsigned 32	rw	00	Set nullpoint for one cursors (User Offset)
6004	0	Position value	Integer 32	ro	None	Transducer position for one cursor
6005	0	Measuring step settings	Unsigned 8	ro	02	No. of entries
	1	Position steps	Unsigned 32	rw	1388h	Position values in 0.001 µm steps 1388h ≤5000d ≤ 5µm
	2	Speed steps	Unsigned 32	rw	0Ah	Velocity in 0.01 mm/s steps 0Ah = 10d = 5 mm/s
6010	0	Preset value	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Unsigned 32	rw	0	Set nullpoint Cursor 1 (User-Offset)
	2	Channel 2	Unsigned 32	Rw	0	Set nullpoint Cursor 2 (User-Offset)
6020	0	Position values	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Integer 32	ro	None	Position Cursor 1
	2	Channel 2	Integer 32	ro	None	Position Cursor 2
6030	0	Speed values	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Integer 16	ro	None	Velocity Cursor 1
	2	Channel 2	Integer 16	ro	None	Velocity Cursor 2
6200	0	Cyclic Timer	Unsigned 16	rw	0Ah	Cyclic sending of position value (Default 10 ms)
6300	0	Cam State Register	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Unsigned 8	ro	None	Cam Status Cursor 1
	2	Channel 2	Unsigned 8	ro	None	Cam Status Cursor 2
6301	0	Cam Enable Register	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Unsigned 8	rw	0	Cam Enable Cursor 1
	2	Channel 2	Unsigned 8	rw	0	Cam Enable Cursor 2
6302	0	Cam Polarity Register	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Unsigned 8	rw	0	Cam Polarity Cursor 1
	2	Channel 2	Unsigned 8	rw	0	Cam Polarity Cursor 2
6310	0	Cam 1 Low Limit	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Signed 32	rw	None	Cam 1 Low Limit Cursor 1
	2	Channel 2	Signed 32	rw	None	Cam 1 Low Limit Cursor 2
6311	0	Cam 2 Low Limit	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Signed 32	rw	None	Cam 2 Low Limit Cursor 1
	2	Channel 2	Signed 32	rw	None	Cam 2 Low Limit Cursor 2

Organization of the Device Profile DS 406

Encoder Profile DS 406 (Tables, cont.)

Index	Sub-Index	Name	Type	Attri-bute	Default-value	Meaning
6312	0	Cam 3 Low Limit	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Signed 32	rw	None	Cam 3 Low Limit Cursor 1
	2	Channel 2	Signed 32	rw	None	Cam 3 Low Limit Cursor 2
6313	0	Cam 4 Low Limit	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Signed 32	rw	None	Cam 4 Low Limit Cursor 1
	2	Channel 2	Signed 32	rw	None	Cam 4 Low Limit Cursor 2
6320	0	Cam 1 High Limit	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Signed 32	rw	None	Cam 1 High Limit Cursor 1
	2	Channel 2	Signed 32	rw	None	Cam 1 High Limit Cursor 2
6321	0	Cam 2 High Limit	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Signed 32	rw	None	Cam 2 High Limit Cursor 1
	2	Channel 2	Signed 32	rw	None	Cam 2 High Limit Cursor 2
6322	0	Cam 3 High Limit	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Signed 32	rw	None	Cam 3 High Limit Cursor 1
	2	Channel 2	Signed 32	rw	None	Cam 3 High Limit Cursor 2
6323	0	Cam 4 High Limit	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Signed 32	rw	None	Cam 4 High Limit Cursor 1
	2	Channel 2	Signed 32	rw	None	Cam 4 High Limit Cursor 2
6330	0	Cam 1 Hysteresis	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Unsigned 16	rw	None	Cam 1 Hysteresis Cursor 1
	2	Channel 2	Unsigned 16	rw	None	Cam 1 Hysteresis Cursor 2
6331	0	Cam 2 Hysteresis	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Unsigned 16	rw	None	Cam 2 Hysteresis Cursor 1
	2	Channel 2	Unsigned 16	rw	None	Cam 2 Hysteresis Cursor 2
6332	0	Cam 3 Hysteresis	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Unsigned 16	rw	None	Cam 3 Hysteresis Cursor 1
	2	Channel 2	Unsigned 16	rw	None	Cam 3 Hysteresis Cursor 2
6333	0	Cam 4 Hysteresis	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Unsigned 16	rw	None	Cam 4 Hysteresis Cursor 1
	2	Channel 2	Unsigned 16	rw	None	Cam 4 Hysteresis Cursor 2
6400	0	Working range State	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Unsigned 8	ro	None	Status Working range Cursor 1
	2	Channel 2	Unsigned 8	ro	None	Status Working range Cursor 2
6401	0	Working range State Low Limit	Unsigned 8	ro	2	No. of entries
	1	Channel 1	Signed 32	rw	00	Working range Low Limit Cursor 1
	2	Channel 2	Signed 32	rw	00	Working range Low Limit Cursor 2 (Value referenced to user nullpoint)

Organization of the Device Profile DS 406

Encoder Profile DS 406 (Tables, cont)

Index	Sub-Index	Name	Type	Attri-bute	Default-value	Meaning
6402	0	Working range High Limit	Unsigned8	ro	2	No. of entries
	1	Channel 1	Signed32	rw	Range-Max	Working range High Limit Cursor 1
	2	Channel 2	Signed32	rw	Range-Max	Working range High Limit Cursor 2 (Value referenced to user nullpoint)
6500	0	Operating status	Unsigned16	ro	-	Current operating parameters
6501	0	Measuring steps	Unsigned32	ro	0x1388h	Output measuring step
6503	0	Alarms	Unsigned16	ro	0	Additional alarms
6504	0	Supported alarms	Unsigned16	ro	0xf001 h	Additional alarms supported Bit 0: Position error Bit 12 -15: Manufacturer Specific
6505	0	Warnings	Unsigned16	ro	0	Alarms
6506	0	Supported warnings	Unsigned16	ro	0xf004h	Supported alarms Bit 2 : watchdog status Bit 12 -15: Manufacturer Specific
6507	0	Profile and Software version	Unsigned32	ro	0x000200 01h	Profile and software version
650A	0	Manufacturer offset value	Signed32	ro	0	Manufacturer Offset
	1	Manufacturer min position	Signed32	ro	0	Minimum position (manufacturer)
	2	Manufacturer max position	Signed32	ro	Full length sensor	Maximum position (manufacturer)
650B	0	Serial number	Unsigned32	ro	XXXXXX	Serial number (manufacturer)

Organization of the Encoder Profile DS 406

Encoder Profile DS 406 (Tables, cont)

All these objects can be read and set with SDO, as in the examples for Communication Profile

Object 6000h – Operating Parameters

The operating Parameters contain the functions for code sequence, Commissioning diagnostic control and scaling function control.

Object 6003h – Preset value

The Preset functions supports adaption of the encoder zero point to the mechanical zero point of the system.

For multi-sensor devices (2 cursors) refer to object 6010h.

The output position value is set to the parameter “Preset value” and the offset from the position value is calculated and stored in the encoder.

STRUCTURE OF PARAMETER

Table 01

Preset value			
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

Object 6004h – Position value

The object 6004h “Position value” defines the output position value for the communication objects 1800h and 1801h. For multi-sensor devices (2 cursors) refer to object 6020h.

STRUCTURE OF PARAMETER = see **Table 01**

Object 6005h – Linear encoder measuring step setting

The parameter “Linear encoder measurung step setting” defines the measuring step setting for the position value(s) and the speed value(s) for linear encoder.

Sub-Index 01h: Position measuring step in 0.001 µm

Sub-Index 02h : Speed measuring step in 0.01 mm/s

Object 6010h – Preset value for multi-sensor devices

The parameter “Preset value for multi-sensor devices” (2 cursors) is similar to object 6003h. In sub-index 00h the number of supported channels is defined.

The preset function supports adaption of the encoder zero point to the mechanical zero point of the system.

The output position values in the sub-indices of object 6020h are set to the sub-indices of the parameter “Preset value” in object 6010h, accordingly. The offset from the position value is calculated and stored in the encoder.

Sub-Index 01h: preset value channel 1

Sub-Index 02h: preset value channel 2

STRUCTURE OF PARAMETER = see **Table 01**

Object 6020h – Position value for multi-sensor devices

Similar to object 6004h the parameter “Position value for multi-sensor devices” defines the output position value(s) for the communication objects 1800h and 1801h.

Sub-Index 01h: preset value channel 1

Sub-Index 02h: preset value channel 2

STRUCTURE OF PARAMETER = see **Table 01**

Object 6030h – Speed value

The parameter “Speed value” defines the output speed value(s). The speed measuring step is defined in object 6005h, sub-index 02h.

Sub-Index 01h: preset value channel 1

Sub-Index 02h: preset value channel 2

STRUCTURE OF PARAMETER

Preset value	
Byte 0	Byte 1
2^7 to 2^0	2^{15} to 2^8

Table 02

Object 6200h – Cyclic timer

Object 6200h contains the parameter “Cyclic timer”. The cyclic timer defines the transmission period for all asynchronous PDOs.

A cyclic transmission of the position value is set, when the cyclic timer is programmed >0. Value between 1 ms and 65535 ms can be selected.

e.g.: 1 msec = 1h

256 msec = 100h

La nuova versione del **Communication Profile DS 301 v. 4.0** prevede la possibilità di stabilire differenti Cyclic timers per i differenti PDO, vedi object 1800 e 1801 sub-index 5. Il valore scritto nell’oggetto 6200h viene copiato anche negli oggetti 1800h e 1801 sub-index 5; i valori scritti negli oggetti 1800 e 1801 sub-index 5 ignorano l’oggetto 6200h. Per memorizzare correttamente in EEPROM il nuovo valore di Cycle time si deve eseguire una operazione che comprende i parametri del communication profile.

Object 6300h – Cam state register

The parameter “Cam state register” defines the status bit of the cam in a cam channel. The status bit set to 1 defines “cam active”. The status bit set to 0 defines “cam inactive”. If the polarity bit of a cam is set (refer to index 6302h) the actual cam state will be inverted. (see figure page 19.

Sub-Index 01h: cam state channel 1

Sub-Index 02h: cam state channel 2

STRUCTURE OF PARAMETER

Cam Enable							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				CAM_No_4	CAM_No_3	CAM_No_2	CAM_No_1
				State	State	State	State

Table 03

Object 6301h – Cam enable

The two "Cam_enable_channel" contain the calculation state for 4 cam's for one position channel. If the enable bit is set to 1, the cam state will be calculated by the device. In the other case the cam state of the related cam will be set permanently to 0.

Sub-Index 01h: cam_enable_channel 1

Sub-Index 02h: cam_enable_channel 2

STRUCTURE OF PARAMETER

Cam Enable							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				CAM_No_4	CAM_No_3	CAM_No_2	CAM_No_1
				Enable	Enable	Enable	Enable

Table 04

Object 6302h – Cam polarity

The Two Cam_polarity_channel contains the actual polarity settings for 4 cam's for one position channel. If the polarity bit is set to 1, the cam state of an active cam will signal by setting the related cam state bit to zero. In the other case the cam state of the related cam will not be inverted. (see figure page 19)

Sub-Index 01h: Cam_polarity_ channel 1

Sub-Index 02h: Cam_polarity_channel 2

STRUCTURE OF PARAMETER

Cam Enable							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				CAM_No_4	CAM_No_3	CAM_No_2	CAM_No_1
				Polarity	Polarity	Polarity	Polarity

Table 05

Object 6310h (6311; 6312; 6313) – Cam 1(2; 3; 4) low limit

The Two Cam_low_channel contains the switch point for the lower limit setting for the 4 cam's for one position channel.

Sub-Index 01h: Cam_low_limit_channel 1

Sub-Index 02h: Cam_low_limit_channel 2

Object 6320h (6321; 6322; 6323) – Cam 1(2; 3; 4) high limit

The Two Cam_high_channel contains the switch point for the higher limit setting for the 4 cam's for one position channel.

Sub-Index 01h: Cam_high_limit_channel 1

Sub-Index 02h: Cam_high_limit_channel 2

WORK AREA SUPERVISION

It is defined a so called user defined working area. The actual work area information with work area low limit and work area high limit can be stored in objects 6401h and 6402h respectively. This way object 6400h can also be used as software limit switches.

Object 6400h – Area state register

The object "area state register" contains the actual area status of the encoder position. If the position is out of range, a bit will be set in the related position line. If the position is lowet than the position value set in object 6401h "work area low limit" then bit 2 flags the underflow. If the position is higher than the position value set in object 6402h "work area high limit" then bit 1 flags the overflow. If the manufacturer minimum position value or the manufacturer maximum position value (refer to object 650Ah "Module identification") is reached, bit 0 flags "out of range"

Sub-Index 01h: work_area_state_channel_1

Sub-Index 02h: work_area_state_channel_2

STRUCTURE OF PARAMETER

Work_area_state							
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

Table 06

Object 6401h – Work area low limit

The Object "work area low limit" contains the position value, at witch bit 2 of the according work_area_state_channel in object 6400h flags the underflow of the related work area.

Sub-Index 01h: Work_area_low_limit_channel_1

Sub-Index 02h: Work_area_low_limit_channel_2

STRUCTURE OF PARAMETER = See **Table 01**

Object 6402h – Work area high limit

The Object "work area high limit" contains the position value, at witch bit 1 of the according work_area_state_channel in object 6400h flags the overrflow of the related work area.

Sub-Index 01h: Work_area_high_limit_channel_1

Sub-Index 02h: Work_area_high_limit_channel_2

STRUCTURE OF PARAMETER = See **Table 01**

ENCODER DIAGNOSTICS

All encoder diagnostics are read from securely stored parameters.

Object 6402h – Operating status

The Object contains the operating status of the encoder. If gives information on encoder internal programmed parameters

STRUCTURE OF PARAMETER

Bit	Function	Bit = 0	Bit = 1
0	Code Sequence	CW	CCW
1	Commissioning Diagnostic Control	Not supp.	Supp.
2	Scaling function control	Disa.	Enab.
3 – 11	Reserved for further use		
12 – 15	Manufacturer specific functions		

Table 07

Object 6501h – Linear encoders

It indicates the measuring step that is output by the encoder.

The measuring step is given in nm (0.001 µm).

E.g: 1 µm = 00 00 03 E8h

STRUCTURE OF PARAMETER = See **Table 01**

Object 6503h – Alarms

Additionally to the emergency messages, object 6503h provides further alarm messages. An alarm is set if a malfunction in the encoder could lead to incorrect position value. If an alarm occurs, the according bit is set to logical high until the alarm is cleared and the encoder is able to provide an accurate position value.

STRUCTURE OF PARAMETER

Bit	Function	Bit = 0	Bit = 1
0	Position error	No	Yes
1	Commissioning diagnostics	OK	Error
2 - 11	Reserved to further use		
12 - 15	Manufacturer specific functions		

Table 08

Object 6504h – Supported alarms

Object 6504h contains the information on supported alarms by the encoder.

STRUCTURE OF PARAMETER

Bit	Function	Bit = 0	Bit = 1
0	Position error	No	Yes
1	Commissioning diagnostics	No	Yes
2 - 11	Reserved to further use		
12 - 15	Manufacturer specific functions		

Table 09

Object 6505h – Warnings

Warnings indicate that tolerances for certain internal parameters of the encoder have been exceeded. In contrast to alarm and emergency messages warnings do not imply incorrect position values. All warnings are cleared if the tolerances are again within normal parameters. For the operating time limit warning (bit 3) the warning is only set again after a power-on sequence

STRUCTURE OF PARAMETER

Bit	Function	Bit = 0	Bit = 1
0	Frequency exceeded	No	Yes
1	Light control reserve	Not reached	Error
2	CPU watchdog stauts	Ok	Reset generated
3	Operating time limit warning	No	Yes
4	Battery charge	Ok	Too low
5	Reference point	Reached	Not reached
6 – 11	Reserved for further use		
12 - 15	Manufacturer specific functions		

Table 10

Object 6506h – Supported warnings

Object 6506h contains the information on supported warnings by the encoder.

STRUCTURE OF PARAMETER

Bit	Function	Bit = 0	Bit = 1
0	Frequency exceeded	Not supp.	Supported
1	Light control reserve	Not supp	Supported
2	CPU watchdog status	Not supp	Supported
3	Operating time limit warning	Not supp	Supported
4	Battery charge	Not supp	Supported
5	Reference point	Not supp	Supported
6 – 11	Reserved for further use		
12 - 15	Manufacturer specific functions		

Table 11

Object 6507h – Profile and software version

This object contains in the 16 bits the profile version whiche is implemented in the encoder. It is combined to a revision number and an index.

E.g: Profile version:	1.40	
Binary code:	00000001	01000000
Hexadecimal:	1h	40h

The 2nd 16 bits contains the software version which is implemented in the encoder. It is combined to a revision number and an index.

E.g: Software version:	1.40	
Binary code:	00000001	01000000
Hexadecimal:	1h	40h

STRUCTURE OF PARAMETER

Profile version		Software version	
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^7 to 2^0	2^{15} to 2^8

Table 12

Object 650Ah – Module identification

Object 650Ah contains the parameter “Module identification”: manufacturer offset value, manufacturer minimum position value, manufacturer maximum position value.

In sub-index 00h, the manufacturer offset value is stored. This value gives information on the shift of the zero point in the number of positions from the physical zero point of the encoder disk.

In sub-index 01h and 02h the manufacturer minimum and maximum position value is stored, respectively.

All three values are given in number of steps according to the basic resolution of the encoder and are located in write protected memory area only changeable by the encoder manufacturer

Sub-Index 00h: manufacturer_offset_value

Sub-Index 01h: manufacturer_min_position_value

Sub-Index 02h: manufacturer_max_position_value

Object 650Bh – Serial number

Object 650Bh contains the encoder serial number. It is given as an unsigned32 binary value. If the parameters serial number is not used the value is set to maximum value FF FF FF FFh by the encoder manufacturer.

1.5.1 MK2C Profile

RO = read only; **RW** = read / write

Idex	Sub-idex	Name	Type	Atteri-bute	Default-value	Meaning
2000	0	Field value	Unsigned 08	RO	2	
	1	Channel 1	Unsigned 32	RO	None	Unscaled readings from analogue digital converter
	2	Channel 2	Unsigned 32	R0	None	
2001	0	Measuring status	Unsigned 08	RO	None	
2002	0	Scaling Factor	Unsigned 24	RO	xxxxx	The field value si riscaled with factor xxxx
2003	0	Number of cursors	Unsigned 08	RW	1	E' attivo un cursore
					2	Sono attivi 2 cursori

Additional usable functions from the Gefran Profile

Object 2000h – Field Value

Object 2000h contains the field value that is the unscaled reading of the position.

Sub-Index 01h: work_area_state_channel_1

Sub-Index 02h: work_area_state_channel_2

Object 2001h – Measuring Status

Object 2001h contains additional information about the correct measuring. If all bit are =0 the measure sistem is all OK.

Object 2002h – Scaling Factor

Object 2002h contains the value of the scaling factor of the the Field value.

Object 2003h – Number of the cursors

Object 2003h contains the parameter “Number of cursors”. Writing 1 or 2 in this object it is possible to set the number of the active cursors. The setting is automactly stored in eeprom and it is active with the next power on. When it is set to 1 the Transmit PDO 2 is not active and it is available for particular application.