



CANopen Serial Gateway Software Specifications

Revision	Description	Date
0.1A	Initial Specification	10/21/2003
1.0A	First Release	01/20/2004

Table of Contents

Revision.....	2
Table of Contents.....	3
Overview:	5
Physical.....	6
Network Switch Settings:.....	6
Bit Timing Settings.....	6
LED Indicators:	6
Supported SDO abort codes	8
Supported Emergency Error Codes	9
Emergency Object Data.....	10
Pre-defined connection set	11
Identifier allocation scheme for the pre-defined connection set.....	11
Broadcast Objects of the Pre-defined Connection Set.....	11
Peer-to-Peer Objects of the Pre-defined Connection Set.....	11
Assigning COB-IDS.....	11
Object Dictionary Data Types	12
Object Dictionary Structure.....	13
Object Dictionary Overview.....	14
Object Dictionary Entries for Communication.....	14
Standard Objects.....	14
Server SDO Parameter.....	14
Client SDO Parameter	14
Receive PDO Communication Parameter.....	15
Receive PDO Mapping Parameter.....	15
Transmit PDO Communication Parameter.....	15
Transmit PDO Mapping Parameter	15
Detailed Specification of Communication Profile Specific Objects.....	16
Object 1000h: Device Type.....	16
Object 1001h: Error Register.....	17
Object 1002h: Manufacturer Status Register.....	18
Object 1003h: Pre-defined Error Field	19
Object 1005h: COB-ID SYNC message.....	20
Object 1008h: Manufacturer Device Name.....	21
Object 1009h: Manufacturer Hardware Version	22
Object 100Ah: Manufacturer Software Version.....	23
Object 100Ch: Guard Time	24
Object 100Dh: Life Time Factor	25
Object 1010h: Store parameters.....	26
Object 1011h: Restore default parameters.....	29
Object 1012h: COB-ID Time Stamp Object.....	31
Object 1014h: COB-ID Emergency Object.....	32
Object 1015h: Inhibit Time EMCY.....	34
Object 1016h: Consumer Heartbeat Time	35
Object 1017h: Producer Heartbeat Time	36
Object 1018h: Identity Object	37
Object 1020h: Verify Configuration.....	39
Object 1029: Error behavior object	40
Object 1200h-1201h: Server SDO Parameter.....	41
Object 1400h – 141Eh: Receive PDO Communication Parameter.....	43
Object 1600h – 161Eh: Receive PDO Mapping Parameter.....	47
Object 1800h – 1803h: Transmit PDO Communication Parameter.....	49

Object 1A00h – 1A1Fh: Transmit PDO Mapping Parameter	51
Multiplexed PDOs	52
MPDO Protocol	52
Destination Address Mode (DAM)	52
Source Address Mode (SAM)	52
Object dictionary entries	53
PDO Mapping Record	53
Additional error code meanings	54
Predefinitions	54
Introduction	54
PDO Mapping	55
6.2.1 1st RPDO mapping (Memory Access)	55
6.2.2 1st TPDO mapping (Memory Access)	55
Manufacturer Specific Objects	56
Overview:	56
Serial Objects	56
Serial Configuration Objects	56
CANopen Objects	56
Manufacturer Specific Object Details:	57
Serial Objects	57
Object 2000h: Serial Transmit Data	57
Object 2001h: Serial Receive Data	58
Object 2002h: Serial Request Fragment	59
Object 2003h: Serial Response Fragment	61
Object 2004h: Serial Control Byte	63
Object 2005h: Serial Status Byte	64
Serial Configuration Objects	65
Object 2100h: Serial Mode	65
Object 2101h: Serial Flow Control	66
Object 2102h: Serial Baud	67
Object 2103h: Serial Data Width	68
Object 2104h: Serial Stop Bits	69
Object 2105h: Serial Parity	70
Object 2106h: Serial Char Delay	71
Object 2107h: Serial Start String	72
Object 2108h: Serial Error String	73

Overview:

The CANopen serial gateway is a bridge between a CANopen network and serial ports. The gateway has two serial ports that can be selected at purchase time. The supported serial types are RS232, RS485, and CAN. Other protocols such as Modbus, J1939, and J1708/1587 are available in different models (see vehicle gateway). The serial gateway can be operated in a passive mode, where the user sends the number of bytes and the raw data to transmit and the device transmits the information to the serial port. The data received on the serial port is converted to the network protocol and the number of bytes and raw data received are available on the network. A second mode is also supported in which the gateway can be configured by the user to send and receive formatted messages. This allows the conversion to be accomplished in the gateway instead of in the controller. The serial data can be transferred using SDOs or PDOs. There are 4 Tx PDOS and 4 Rx PDOS supported.

Physical

Network Switch Settings:

SW1 and SW2

Node ID	Meaning
0x00	Re-Configure Module (Does Not got Online)
0x01-0x7F	Use this Node Address
0x80-0xFF	Use value stored in NV

SW3

Baud Rate	Meaning
0x0	Auto-Baud
0x1	10k bps
0x2	20k bps
0x3	50k bps
0x4	125k bps
0x5	250k bps
0x6	500k bps
0x7	800k bps
0x8	1M bps
0x9-0xF	Use value stored in NV

Bit Timing Settings

Bit rate Bus length ⁽¹⁾	Nominal bit time ^{tb}	Number of time quanta per bit	Length of time quantum ^{tq}	Location of sample point
1 Mbit/s 25 m	1 us	8	125 ns	6 ^{tq} (750 ns)
800 kbit/s 50 m	1,25 us	10	125 ns	8 ^{tq} (1 us)
500 kbit/s 100 m	2 us	16	125 ns	14 ^{tq} (1,75 us)
250 kbit/s 250 m ⁽²⁾	4 us	16	250 ns	14 ^{tq} (3,5 us)
125 kbit/s 500 m ⁽²⁾	8 us	16	500 ns	14 ^{tq} (7 us)
50 kbit/s 1000 m ⁽³⁾	20 us	16	1,25 us	14 ^{tq} (17,5 us)
20 kbit/s 2500 m ⁽³⁾	50 us	16	3,125 us	14 ^{tq} (43,75 us)
10 kbit/s 5000 m ⁽³⁾	100 us	16	6,25 us	14 ^{tq} (87,5 us)

LED Indicators:

Two LEDs are used to indicate the status of the module, these are the CAN and MOD leds.

There are two additional LEDs for each serial port, one for the transmit status and one for the receive status. The LED is green when transmitting or receiving. The LED is red when there is an error such as an overflow or parity error. It may be possible to have an amber color when the port has an error and is transmitting or receiving.

Supported Codes

Supported SDO abort codes

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	Out 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 length 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.

The abort codes not listed here are reserved.

Supported Emergency Error Codes

Error Code (hex)	Meaning
00xx	Error Reset or No Error
10xx	Generic Error
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
8150	Transmit COB-ID
82xx	Protocol Error
8210	PDO not processed due to length error
8220	PDO length exceeded
FFxx	Device specific

Emergency Object Data

The Emergency Telegram consists of 8 bytes with the data as shown in

Emergency Object

Byte	0	1	2	3	4	5	6	7
Content	Emergency Error Code (see Table)		Error register (Object 1001H)	TBD	TBD	TBD		

Pre-defined connection set

Bit Number: COB-Identifier

10	9	8	7	6	5	4	3	2	1	0
Function Code				Node-ID						

Identifier allocation scheme for the pre-defined connection set

Broadcast Objects of the Pre-defined Connection Set

Object	function code(binary)	resulting COB-ID	Communication Parameters at Index
NMT	0000	0	-
SYNC	0001	128 (80h)	1005h,1006h,1007h
TIME STAMP	0010	256 (100h)	1012h, 1013h

Peer-to-Peer Objects of the Pre-defined Connection Set

object	function code (binary)	Resulting COB-IDs	Communication Parameters at Index
EMERGENCY	0001	129 (81h) - 255 (FFh)	1014h, 1015h
PDO1 (tx)	0011	385 (181h) - 511 (1FFh)	1800h
PDO1 (rx)	0100	513 (201h) - 639 (27Fh)	1400h
PDO2 (tx)	0101	641 (281h) - 767 (2FFh)	1801h
PDO2 (rx)	0110	769 (301h) - 895 (37Fh)	1401h
PDO3 (tx)	0111	897 (381h) - 1023 (3FFh)	1802h
PDO3 (rx)	1000	1025 (401h) - 1151 (47Fh)	1402h
PDO4 (tx)	1001	1153 (481h) - 1279 (4FFh)	1803h
PDO4 (rx)	1010	1281 (501h) - 1407 (57Fh)	1403h
SDO (tx)	1011	1409 (581h) - 1535 (5FFh)	1200h
SDO (rx)	1100	1537 (601h) - 1663 (67Fh)	1200h
NMT Error Control	1110	1793 (701h) - 1919 (77Fh)	1016h, 1017h

Seen from the device's point of view.

The pre-defined connection set always applies to the standard CAN frame with 11-bit Identifier, even if extended CAN frames are present in the network.

Assigning COB-IDS

When Assigning COB-Ids to SDO and PDO objects, the user should use care. If the user selects a COB-ID for a PDO that is already assigned to the display's SDO by the pre-defined connection set, the RTC/Memory Module will accept the COB-ID, but the module may not function as the user desires. The COB-ID acceptance is done in order to provide greater flexibility for the advanced user.

Object Dictionary Data Types

Index	Object	Name
0001	DEFTYPE	BOOLEAN
0002	DEFTYPE	INTEGER8
0003	DEFTYPE	INTEGER16
0004	DEFTYPE	INTEGER32
0005	DEFTYPE	UNSIGNED8
0006	DEFTYPE	UNSIGNED16
0007	DEFTYPE	UNSIGNED32
0008	DEFTYPE	REAL32
0009	DEFTYPE	VISIBLE_STRING
000A	DEFTYPE	OCTET_STRING
000B	DEFTYPE	UNICODE_STRING
000C	DEFTYPE	TIME_OF_DAY
000D	DEFTYPE	TIME_DIFFERENCE
000E	reserved	-
000F	DEFTYPE	DOMAIN
0010	DEFTYPE	INTEGER24
0011	DEFTYPE	REAL64
0012	DEFTYPE	INTEGER40
0013	DEFTYPE	INTEGER48
0014	DEFTYPE	INTEGER56
0015	DEFTYPE	INTEGER64
0016	DEFTYPE	UNSIGNED24
0017	reserved	-
0018	DEFTYPE	UNSIGNED40
0019	DEFTYPE	UNSIGNED48
001A	DEFTYPE	UNSIGNED56
001B	DEFTYPE	UNSIGNED64
001C-001F	reserved	-
0020	DEFSTRUCT	PDO_COMMUNICATION_PARAMETER
0021	DEFSTRUCT	PDO_MAPPING
0022	DEFSTRUCT	SDO_PARAMETER
0023	DEFSTRUCT	IDENTITY
0024-003F	reserved	-
0040-005F	DEFSTRUCT	Manufacturer Specific Complex Data Types
0060-007F	DEFTYPE	Device Profile (0) Specific Standard Data Types
0080-009F	DEFSTRUCT	Device Profile (0) Specific Complex Data Types
00A0-00BF	DEFTYPE	Device Profile 1 Specific Standard Data Types
00C0-00DF	DEFSTRUCT	Device Profile 1 Specific Complex Data Types
00E0-00FF	DEFTYPE	Device Profile 2 Specific Standard Data Types
0100-011F	DEFSTRUCT	Device Profile 2 Specific Complex Data Types
0120-013F	DEFTYPE	Device Profile 3 Specific Standard Data Types
0140-015F	DEFSTRUCT	Device Profile 3 Specific Complex Data Types
0160-017F	DEFTYPE	Device Profile 4 Specific Standard Data Types
0180-019F	DEFSTRUCT	Device Profile 4 Specific Complex Data Types
01A0-01BF	DEFTYPE	Device Profile 5 Specific Standard Data Types
01C0-01DF	DEFSTRUCT	Device Profile 5 Specific Complex Data Types
01E0-01FF	DEFTYPE	Device Profile 6 Specific Standard Data Types
0200-021F	DEFSTRUCT	Device Profile 6 Specific Complex Data Types
0220-023F	DEFTYPE	Device Profile 7 Specific Standard Data Types
0240-025F	DEFSTRUCT	Device Profile 7 Specific Complex Data Types

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	Standardized Device Profile Area
A000-FFFF	Reserved for further use

Object Dictionary Overview

Object Dictionary Entries for Communication

Standard Objects

Index(hex)	Object (Symbolic Name)	Name	Type	Acc. 1	M/O
1000	VAR	device type	UNSIGNED32	ro	M
1001	VAR	error register	UNSIGNED8	ro	M
1002	VAR	Manufacturer status register	UNSIGNED32	ro	O
1003	VAR	Pre-defined error field	UNSIGNED32	Ro	O
1005	VAR	COB-ID SYNC	UNSIGNED32	rw	O
1008	VAR	Manufacturer device name	Vis-String	const	O
1009	VAR	Manufacturer hardware version	Vis-String	const	O
100A	VAR	Manufacturer software version	Vis-String	const	O
100C	VAR	guard time	UNSIGNED16	rw	O
100D	VAR	life time factor	UNSIGNED8	rw	O
1010	ARRAY	store parameters	UNSIGNED32	rw	O
1011	ARRAY	restore default parameters	UNSIGNED32	rw	O
1012	VAR	COB-ID TIME	UNSIGNED32	rw	O
1014	VAR	COB-ID EMCY	UNSIGNED32	rw	O
1015	VAR	Inhibit Time EMCY	UNSIGNED16	rw	O
1016	ARRAY	Consumer heartbeat time	UNSIGNED32	RW	O
1017	VAR	Producer heartbeat time	UNSIGNED16	rw	O
1018	RECORD	Identity Object	Identity (23h)	ro	M
1020	ARRAY	Verify Configuration	UNSIGNED32	rw	O
1029	ARRAY	Error Behavior	UNSIGNED8	rw	O

Server SDO Parameter

1200	RECORD	1 st Server SDO parameter	SDO Parameter (22h)	ro	O
1201	RECORD	2 st Server SDO parameter	SDO Parameter (22h)	rw	O

Client SDO Parameter

Not Supported

Receive PDO Communication Parameter

1400	RECORD	1 st receive PDO Parameter	PDO CommPar (20h)	rw	M/O*
1401	RECORD	2 nd receive PDO Parameter	PDO CommPar (20h)	rw	M/O*
.....
1403	RECORD	4 th receive PDO Parameter	PDO CommPar (20h)	rw	M/O*

Receive PDO Mapping Parameter

1600	RECORD	1 st receive PDO mapping	PDO Mapping (21h)	rw	M/O*
1601	RECORD	2 nd receive PDO mapping	PDO Mapping (21h)	rw	M/O*
.....
1603	RECORD	4 th receive PDO mapping	PDO Mapping (21h)	rw	M/O*

Transmit PDO Communication Parameter

1800	RECORD	1 st transmit PDO Parameter	PDO CommPar (20h)	rw	M/O*
1801	RECORD	2 nd transmit PDO Parameter	PDO CommPar (20h)	rw	M/O*
.....
1803	RECORD	4 th transmit PDO Parameter	PDO CommPar (20h)	rw	M/O*

Transmit PDO Mapping Parameter

1A00	RECORD	1 st transmit PDO mapping	PDO Mapping (21h)	rw	M/O*
1A01	RECORD	2 nd transmit PDO mapping	PDO Mapping (21h)	rw	M/O*
.....
1A03	RECORD	4 th transmit PDO mapping	PDO Mapping (21h)	rw	M/O*

Ranges 1600-1603h and 1A00-1A03h can also be used to map multiplexed PDOs. See specification below.

Detailed Specification of Communication Profile Specific Objects

Object 1000h: Device Type

Contains information about the device type. The object at index 1000h describes the type of 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 optional functionality of the device. The Additional Information parameter is device profile specific.

OBJECT DESCRIPTION

INDEX	1000h
Name	device type
Object Code	VAR
Data Type	UNSIGNED32
Category	Mandatory

ENTRY DESCRIPTION

Access	Ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

Byte:

MSB	LSB
Additional Information	Device Profile Number
0x0000	0x0000

Object 1001h: Error Register

This object is an error register for the device. The device can map internal errors in this byte. This entry is mandatory for all devices. It is a part of an Emergency object.

OBJECT DESCRIPTION

INDEX	1001h
Name	error register
Object Code	VAR
Data Type	UNSIGNED8
Category	Mandatory

ENTRY DESCRIPTION

Access	ro
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	No

Structure of the Error Register

Bit	M/O	Supported	Meaning
0	M	Yes	generic error
1	O	No	Current
2	O	Yes	Voltage
3	O	No	Temperature
4	O	Yes	communication error (overrun, error state)
5	O	No	device profile specific
6	O	No	Reserved (always 0)
7	O	Yes	manufacturer specific

If a bit is set to 1 the specified error has occurred. The generic error is signaled at any error situation.

Object 1002h: Manufacturer Status Register

This object is a common status register for manufacturer specific purposes. In this document only the size and the location of this object is defined.

OBJECT DESCRIPTION

INDEX	1002h
Name	manufacturer status register
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Access	ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

MSB				LSB

Object 1003h: Pre-defined Error Field

The object at index 1003h holds the errors that have occurred on the device and have been signaled via the Emergency Object. In doing so it provides an error history.

1. The entry at sub-index 0 contains the number of actual errors that are recorded in the array starting at sub-index 1.
2. Every new error is stored at sub-index 1, the older ones move down the list.
3. Writing a "0" to sub-index 0 deletes the entire error history (empties the array). Values higher than 0 are not allowed to write. This has to lead to an abort message (error code: 0609 0030h).
4. The error numbers are of type UNSIGNED32 (see Table 7-18) and are composed of a 16 bit error code and a 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). The additional information consists of the information contained in the Error register Object (1001h) The length entry on sub-index 0h and at least one error entry at sub-index 1H.

Byte: MSB LSB

Additional Information	Error code
------------------------	------------

Structure of the pre-defined error field

OBJECT DESCRIPTION

INDEX	1003h
Name	pre-defined error field
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Sub-Index	0h
Description	number of errors
Entry Category	Mandatory
Access	rw
PDO Mapping	No
Value Range	0 - 10
Default Value	0

Sub-Index	1h
Description	standard error field
Entry Category	Optional
Access	ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2h – 10
Description	standard error field
Entry Category	Optional
Access	ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

Object 1005h: COB-ID SYNC message

Index 1005h defines the COB-ID of the Synchronization Object (SYNC). Further, it defines whether the device generates the SYNC.

UNSIGNED32

MSB LSB

bits	31	30	29	28-11	10-0
11-bit-ID	X	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	X	0/1	1	29-bit Identifier	

Structure of SYNC COB-ID entry

Description of SYNC COB-ID entry

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	bits 10-0 of SYNC-COB-ID

Bits 29, 30 may be static (not changeable). If a device is not able to generate SYNC messages, an attempt to set bit 30 is responded with an abort message (abort code: 0609 0030h). Devices supporting the standard CAN frame type only either ignore attempts to change bit 29 or respond with an abort message (abort code: 0609 0030h). The first transmission of SYNC object starts within 1 sync cycle after setting Bit 30 to 1. It is not allowed to change Bit 0-29, while the objects exist (Bit 30=1).

OBJECT DESCRIPTION

INDEX	1005h
Name	COB-ID SYNC
Object Code	VAR
Data Type	UNSIGNED32
Category	Conditional;Mandatory, if PDO communication on a synchronous base is supported

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	80h or 8000 0080h

Object 1008h: Manufacturer Device Name

Contains the manufacturer device name.

OBJECT DESCRIPTION

INDEX	1008h
Name	manufacturer device name
Object Code	VAR
Data Type	Visible String
Category	Optional

ENTRY DESCRIPTION

Access	const
PDO Mapping	No
Value Range	No
Default Value	Serial Gateway

Object 1009h: Manufacturer Hardware Version

Contains the manufacturer hardware version description.

OBJECT DESCRIPTION

INDEX	1009h
Name	manufacturer hardware version
Object Code	VAR
Data Type	Visible String
Category	Optional

ENTRY DESCRIPTION

Access	const
PDO Mapping	No
Value Range	No
Default Value	01

Object 100Ah: Manufacturer Software Version

Contains the manufacturer software version description.

OBJECT DESCRIPTION

INDEX	100Ah
Name	Manufacturer software version
Object Code	VAR
Data Type	Visible String
Category	Optional

ENTRY DESCRIPTION

Access	Const
PDO Mapping	No
Value Range	No
Default Value	1.01

Object 100Ch: Guard Time

The objects at index 100Ch and 100Dh 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.

OBJECT DESCRIPTION

INDEX	100Ch
Name	guard time
Object Code	VAR
Data Type	UNSIGNED16
Category	Conditional; Mandatory, if heartbeat is not supported

ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	0

Object 100Dh: Life Time Factor

The life time factor multiplied with the guard time gives the life time for the node guarding protocol. It is 0 if not used.

OBJECT DESCRIPTION

INDEX	100Dh
Name	life time factor
Object Code	VAR
Data Type	UNSIGNED8
Category	Conditional; Mandatory, if heartbeat is not supported

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Object 1010h: Store parameters

This object supports the saving of parameters in non volatile memory. By read access the device provides information about its saving 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 specific communication parameters).

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

At Sub-Index 4 - 127 manufacturers may store their choice of parameters individually.

Sub-Index 128 - 254 are reserved for future use.

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 MSB LSB

ISO 8859

("ASCII")

hex

E	V	a	s
65h	76h	61h	73h

Storage write access signature

On reception of the correct signature in the appropriate sub-index the device stores the parameter and then confirms the SDO transmission (initiate download response). If the storing failed, the device responds with an Abort SDO Transfer (abort code: 0606 0000h).

If a wrong signature is written, the device refuses to store and responds with Abort SDO Transfer (abort code: 0800 002xh).

On read access to the appropriate Sub-Index the device provides information about its storage functionality with the following format:

UNSIGNED32

MSB

LSB

31-2	1	0
reserved (=0)	0/1	0/1

Storage read access structure

Structure of read access

bit number	Value	meaning
31-2	0	<i>reserved (=0)</i>
1	0	Device does not save parameters autonomously
	1	Device saves parameters autonomously
0	0	Device does not save parameters on command
	1	Device saves parameters on command

Autonomous saving means that a device stores the storable parameters in a non-volatile manner without user request.

OBJECT DESCRIPTION

INDEX	1010h
Name	store parameters
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Sub-Index	0h
Description	largest subindex supported
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1h - 7Fh
Default Value	4

Sub-Index	1h
Description	Save all parameters
Entry Category	Mandatory
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 55 for write access; Figure 56 for read access)
Default Value	01h

Sub-Index	2h
Description	save communication parameters
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 55 for write access; Figure 56 for read access)
Default Value	01h

Sub-Index	3h
Description	Save application parameters
Entry Category	Optional
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 55 for write access; Figure 56 for read access)
Default Value	02h

Sub-Index	4h
Description	Save I/O Configuration and Node Address
Entry Category	Optional
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 55 for write access; Figure 56 for read access)
Default Value	01h

Note Application parameters are stored autonomously. Communication parameters must be manually stored.

Object 1011h: Restore default parameters

With 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 distinguished:

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).

At Sub-Index 4 - 127 manufacturers may restore their individual choice of parameters.

Sub-Index 128 - 254 are reserved for future use.

In order to avoid the restoring of default parameters by mistake, restoring is only executed when a specific signature is written to the appropriate sub-index. The signature is "load".

Signature MSB LSB

ASCII

Hex

d	a	o	l
64h	61h	6Fh	6Ch

Figure 57: Restoring write access signature

On reception of the correct signature in the appropriate sub-index the device restores the default parameters and then confirms the SDO transmission (initiate download response). If the restoring failed, the device responds with an Abort SDO Transfer (abort code: 0606 0000h). If a wrong signature is written, the device refuses to restore the defaults and responds with an Abort SDO Transfer (abort code: 0800 002xh).

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.

On read access to the appropriate sub-index the device provides information about its default parameter restoring capability with the following format:

UNSIGNED32

MSB	LSB
31-1	0
Reserved (=0)	0/1

Figure 59: Restoring default values read access structure

Table 50: Structure of restore read access

bit number	value	meaning
31-1	0	<i>reserved (=0)</i>
0	0	Device does not restore default parameters
	1	Device restores parameters

OBJECT DESCRIPTION

Index	1011h
Name	restore default parameters
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Sub-Index	0h
Description	largest subindex supported
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1h- 7Fh
Default Value	4

Sub-Index	1h
Description r	Restore all default parameters
Entry Category	Mandatory
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 57)
Default Value	01h

Sub-Index	2h
Description	Restore communication default parameters
Entry Category	Optional
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 57)
Default Value	01h

Sub-Index	3h
Description	Restore application default parameters
Entry Category	Optional
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 57)
Default Value	01h

Sub-Index	4h
Description	Restore Manufacturer Specific parameters
Entry Category	Optional
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 57)
Default Value	01h

The number and type of physically attached modules control most of the manufacturer specific parameters, therefore Sub-Index 4 is reserved for future use.

Object 1012h: COB-ID Time Stamp Object

Index 1012h defines the COB-ID of the Time-Stamp Object (TIME). Further, it defines whether the device consumes the TIME or whether the device generates the TIME. The structure of this object is shown in Figure 60 and Table 51.

UNSIGNED32
MSB LSB

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 0 0	11-bit Identifier
29-bit-ID	0/1	0/1	1	29-bit Identifier	

Figure 60: Structure of TIME COB-ID entry

Table 51: Description of TIME COB-ID entry

bit number	value	meaning
31 (MSB)	0	Device does not consume TIME message
	1	Device consumes TIME message
30	0	Device does not produce TIME message
	1	Device produces TIME message
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-TIME-COB-ID
10-0 (LSB)	X	bits 10-0 of TIME-COB-ID

Bits 29, 30 may be static (not changeable). If a device is not able to generate TIME messages, an attempt to set bit 30 is responded with an abort message (abort code: 0609 0030h). Devices supporting the standard CAN frame type only, an attempt to set bit 29 is responded with an abort message (abort code: 0609 0030h). It is not allowed to change Bits 0-29, while the object exists (Bit 30=1).

OBJECT DESCRIPTION

INDEX	1012h
Name	COB-ID time stamp message
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	100h

Object 1014h: COB-ID Emergency Object

Index 1014h defines the COB-ID of the Emergency Object (EMCY). The structure of this object is shown in Figure 61.

UNSIGNED32

MSB LSB

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

Figure 61: Structure of the EMCY Identifier entry

Table 52: Description of EMCY COB-ID entry

bit number	value	Meaning
31 (MSB)	0	EMCY exists / is valid
	1	EMCY does not exist / is not valid
30	0	reserved (always 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-COB-ID
10-0 (LSB)	X	bits 10-0 of COB-ID

Devices supporting the standard CAN frame type only, an attempt to set bit 29 is responded with an abort message (abort code: 0609 0030h). It is not allowed to change Bits 0-29, while the object exists (Bit 31=0).

OBJECT DESCRIPTION

INDEX	1014h
Name	COB-ID Emergency message
Object Code	VAR
Data Type	UNSIGNED32
Category	Conditional;Mandatory, if Emergency is supported

ENTRY DESCRIPTION

Access	ro;optional rw
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	80h + Node-ID

Object 1015h: Inhibit Time EMCY

The inhibit time for the EMCY message can be adjusted via this entry. If this entry exists it must be writeable in the object dictionary. The time has to be a multiple of 100ms.

OBJECT DESCRIPTION

INDEX	1015h
Name	Inhibit Time EMCY
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	0

Object 1016h: Consumer Heartbeat Time

The consumer heartbeat time defines the expected heartbeat cycle time and thus has to be higher than the corresponding producer heartbeat time configured on the device producing this heartbeat.

Monitoring starts after the reception of the first heartbeat. If the consumer heartbeat time is 0 the corresponding entry is not used. The time has to be a multiple of 1ms.

UNSIGNED32

MSB

LSB

Bits	31-24	23-16	15-0
Value	reserved (value: 00h)	Node-ID	heartbeat time
Encoded as	-	UNSIGNED8	UNSIGNED16

Table 62: Structure of Consumer Heartbeat Time entry

At an attempt to configure several consumer heartbeat times unequal 0 for the same Node-ID the device aborts the SDO download with abort code 0604 0043h

OBJECT DESCRIPTION

INDEX	1016h
Name	Consumer Heartbeat Time
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Sub-Index	0h
Description	number entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1 - 127
Default Value	4

Sub-Index	1h
Description	Consumer Heartbeat Time
Entry Category	Mandatory
Access	rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 62)
Default Value	0

Sub-Index	2h - 4h
Description	Consumer Heartbeat Time
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 62)
Default Value	No

Object 1017h: Producer Heartbeat Time

The producer heartbeat 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 1ms.

OBJECT DESCRIPTION

INDEX	1017h
Name	Producer Heartbeat Time
Object Code	VAR
Data Type	UNSIGNED16
Category	Conditional; Mandatory if guarding not supported

ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	0

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 each 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. The major revision number identifies a specific CANopen behavior. If the CANopen functionality is expanded, the major revision has to be incremented. The minor revision number identifies different versions with the same CANopen behavior.

31	16 15	0
major revision number		minor revision number
MSB		LSB

Structure of Revision number

The manufacturer-specific Serial number (sub-index 4h) identifies a specific device.

OBJECT DESCRIPTION

INDEX	1018h
Name	Identity Object
Object Code	RECORD
Data Type	Identity
Category	Mandatory

ENTRY DESCRIPTION

Sub-Index	0h
Description	number of entries
Entry Category	Mandatory
Access	Ro
PDO Mapping	No
Value Range	1 .. 4
Default Value	4

Sub-Index	1h
Description	Vendor ID
Entry Category	Mandatory
Access	Ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0x00000141

Sub-Index	2h
Description	Product code
Entry Category	Optional
Access	Ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0x000003E9

Sub-Index	3h
Description	Revision number
Entry Category	Optional
Access	Ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0x00010001

Sub-Index	4h
Description	Serial number
Entry Category	Optional
Access	Ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

Object 1020h: Verify Configuration

If a device supports the saving of parameters in non-volatile memory, a network configuration tool or a CANopen manager can use this object to verify the configuration after a device's reset and to check if a reconfiguration is necessary. The configuration tool shall store the date and time in that object and shall store the same values in the DCF. Now the configuration tool lets the device save its configuration by writing to index 1010h Sub-Index 1h the signature "save". After a reset the device shall restore the last configuration and the signature automatically or by request. If any other command changes boot-up configuration values, the device shall reset the object Verify Configuration to 0.

The Configuration Manager compares signature and configuration with the value from the DCF and decides if a reconfiguration is necessary or not.

Index	Object	Name	Type	Attr.	M/O
1020h	ARRAY	Verify Configuration	Unsigned32	rw	0

The sub-objects for the Verify Configuration Object are:

Index	Sub-Index	Field in Configuration Verify	Data Type
1020h	0h	Number of supported entries	Unsigned8
	1h	Configuration date	Unsigned32
	2h	Configuration time	Unsigned32

Configuration date shall contain the number of days since January 1, 1984. Configuration time shall be the number of ms after midnight.

Application hint: The usage of this object allows a significant speed-up of the boot-up process. If it is used, the system integrator has to consider that a user may change a configuration value and afterwards activate the command store configuration 1010h without changing the value of 1020h. So the system integrator has to ensure a 100% consequent usage of this feature.

Object 1029: Error behavior object

If a serious device failure is detected in Operational State, the module shall enter by default autonomously the pre-operational state. If object 1028h (Error Behavior) is implemented, the device can be configured to enter alternatively the stopped state or remain in the current state in case of a device failure. Device failures shall include the following communication errors:

- Bus-off conditions of the CAN interface
- Life guarding event with the state 'occurred'
- Heartbeat event with state 'occurred'

Serious device errors also can be caused by device internal failures.

The value of the Error Classes is as follows:

0 = pre-operational (only if current state is operational)

1 = no state change

2 = stopped

3 .. 127 = reserved

OBJECT DESCRIPTION

INDEX	1029h
Name	Error Behavior
Object Code	ARRAY
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Sub-Index	0h
Description	No. of Error Classes
Entry Category	Mandatory
Access	Ro
PDO Mapping	No
Value Range	1h
Default Value	1

Sub-Index	1h
Description	Communication Error
Entry Category	Mandatory
Access	rw
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Object 1200h-1201h: Server SDO Parameter

In order to describe the SDOs used on a device the data type SDO Parameter is introduced. The data type has the index 22h in the Object Dictionary. The structure is described in 9.5.4.

The number of supported entries in the SDO object record is specified at sub-index 0h. The values at 1h and 2h specify the COB-ID for this SDO. Sub-index 3 gives the server of the SDO in case the record describes an SDO for which the device is client and gives the client of the SDO if the record describes an SDO for which the device is server.

UNSIGNED32

MSB

LSB

bits	31	30	29	28-11	10-0
11-bit-ID	0/1	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 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	29-bit Identifier	

Figure 64: Structure of SDO COB-ID entry

Table 53: Description of SDO COB-ID entry

bit number	value	Meaning
31 (MSB)	0	SDO exists / is valid
	1	SDO does not exist / is not valid
30	0	reserved (always 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-COB-ID
10-0 (LSB)	X	bits 10-0 of COB-ID

An SDO is only valid if both SDO-valid-bits are 0. Devices supporting the standard CAN frame type only, an attempt to set bit 29 is responded with an abort message (abort code: 0609E0030h).

These objects contain the parameters for the SDOs for which the device is the server. If a device handles more than one server SDO the default SDO must be located at index 1200h as the first server SDO. This entry is read only 2. All additional server SDOs are invalid by default (invalid bit - see Table 53), there description is located at subsequent indices. It is not allowed to change the COB-ID while the SDO exists.

The description of the Client of the SDO (sub-index 3h) is optional. It is not available for the default SDO (no Sub-index 3h at Index 1200h), as this entry is read only.

OBJECT DESCRIPTION

INDEX	1200-1201h
Name	Server SDO parameter
Object Code	RECORD
Data Type	SDO Parameter
Category	Conditional Index 1200h: Optional Index 1201h - 127Fh: Mandatory for each additionally supported server SDO

ENTRY DESCRIPTION

Sub-Index	0h
Description	number of entries
Entry Category	Mandatory
Access	Ro
PDO Mapping	No
Value Range	Index 1200h: 2 Index 1201h - 127F: 2 – 3
Default Value	No

Sub-Index	1h
Description	COB-ID Client->Server (rx)
Entry Category	Mandatory
Access	Index 1200h: ro, Index 1201h-127Fh: rw
PDO Mapping	No
Value Range	UNSIGNED32 (Table 53)
Default Value	Index 1200h: 600h+Node-ID, Index 1201h-127Fh: No

Sub-Index	2h
Description	COB-ID Server -> Client (tx)
Entry Category	Mandatory
Access	Index 1200h: ro Index 1201-127Fh: rw
PDO Mapping	No
Value Range	UNSIGNED32 (Table 53)
Default Value	Index 1200h: 580h+Node-ID, Index 1201h-127Fh: No

Object 1400h – 141Eh: Receive PDO Communication Parameter

Contains the communication parameters for the PDOs the device is able to receive. The type of the PDO communication parameter (20h) is described in 9.5.4. The sub-index 0h contains the number of valid entries within the communication record. Its value is at least 2. If inhibit time supported the value is 3. 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 (CAN 2.0A) as well as for 29-bit CAN identifiers (CAN 2.0B).

The entry has to be interpreted as defined in Figure 65 and Table 54.

UNSIGNED32

MSB				LSB	
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 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	

Figure 65: Structure of PDO COB-ID entry

Table 54: Description of PDO COB-ID entry

bit number	value	Meaning
31 (MSB)	0	PDO exists / is valid
	1	PDO does not exist / is 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	bits 10-0 of COB-ID

The PDO valid/not valid allows to select which PDOs are used in the operational state. There may be PDOs fully configured (e.g. by default) but not used, and therefore set to "not valid" (deleted). The feature is necessary for devices supporting more than 4 RPDOs or 4 TPDOs, because each device has only default identifiers for the first four RPDOs/TPDOs. Devices supporting the standard CAN frame type only or do not support Remote Frames, an attempt to set bit 29 to 1 or bit 30 to 0 is responded with an abort message (abort code: 0609E0030h). It is not allowed to change bit 0-29 while the PDO exists (Bit 31=0).

The transmission type (sub-index 2) defines the transmission/reception character of the PDO (see 9.2.1.1). Table 55 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: 0609E0030h) is generated.

transmission 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) means 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 type 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.

Receive PDOs are always triggered by the following SYNC upon reception of data independent of the transmission types 0 - 240.

The transmission types 252 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 transmission type 253 the data is updated at the reception of the remote transmission request (hardware 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 application 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 100ms. It is not allowed to change the value while the PDO exists (Bit 31 of sub-index 1 is 0).

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 occurrence of the events set the timer. Independent of the transmission type the RPDO event timer is used recognize the expiration of the RPDO.

OBJECT DESCRIPTION

INDEX	1400h – 141Fh
Name	receive PDO parameter
Object Code	RECORD
Data Type	PDO CommPar
Category	Conditional; Mandatory for each supported PDO

ENTRY DESCRIPTION

Sub-Index	0h
Description	largest sub-index supported
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2 – 5

Sub-Index	1h
Description	COB-ID used by PDO
Entry Category	Mandatory
Access	ro; rw if variable COB-ID is supported
PDO Mapping	No
Value Range	UNSIGNED32 (Table 54)
Default Value	Index 1400h: 200h + Node-ID, Index 1401h: 300h + Node-ID, Index 1402h: 400h + Node-ID, Index 1403h: 500h + Node-ID, Index 1404h: 15FFh: disabled

Sub-Index	2h
Description	transmission type
Entry Category	Mandatory
Access	ro;rw if variable transmission type is supported
PDO Mapping	No
Value Range	UNSIGNED8 (Table 55)
Default Value	(Device Profile dependent)

Sub-Index	3h
Description	inhibit time (not used for RPDO)
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	No

Sub-Index	4h
Description	compatibility entry
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	No

Sub-Index	5h
Description	event timer

Entry Category	Optional (not used for RPDO)
Access	rw
PDO Mapping	No
Value Range	0 = not used UNSIGNED16
Default Value	No

Object 1600h – 161Eh: Receive PDO Mapping Parameter

Contains the mapping for the PDOs the device is able to receive. The type of the PDO mapping parameter (21h) is described in 9.5.4. 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/received 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 (Figure 66). All three values are hexadecimal coded. The length 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		LSB
index (16 bit)	sub-index (8 bit)	object length (8 bit)

Figure 66: Structure of PDO Mapping Entry

If the change of the PDO mapping cannot be executed (e.g. the PDO length is exceeded 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 the PDO has to be deleted, the sub-index 0 must be set to 0 (mapping is deactivated). Then the objects can be remapped. When a new object is mapped by writing a subindex between 1 and 64, the device may check whether the object specified by index / sub-index 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. Finally the PDO will be created by writing to its communication parameter COB-ID. 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 Abort SDO Transfer Service with one of the abort codes 0602 0000h, 0604 0041h or 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 utilizing a part of the PDO. It is not possible to create a dummy mapping for a TPDO.

OBJECT DESCRIPTION

INDEX	1600h – 161Fh
Name	receive PDO mapping
Object Code	RECORD
Data Type	PDO Mapping
Category	Conditional; Mandatory for each supported PDO

ENTRY DESCRIPTION

Sub-Index	0h
Description	number of mapped application objects in PDO
Entry Category	Mandatory
Access	ro; rw if dynamic mapping is supported
PDO Mapping	No
Value Range	0: deactivated 1 - 64: activated
Default Value	(device profile dependent)

Sub-Index	1h - 40h
Description	PDO mapping for the nth application object to be mapped
Entry Category	Conditional depends on number and size of object be mapped
Access	rw
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	(device profile dependent)

Object 1800h – 1803h: Transmit PDO Communication Parameter

Contains the communication parameters for the PDOs the device is able to transmit. The type of the PDO communication parameter (20h) is described in 9.5.4. A detailed description of the entries is done in the section for the Receive PDO Communication Parameter (1400h – 141Fh).

OBJECT DESCRIPTION

INDEX	1800h – 181Fh
Name	transmit PDO parameter
Object Code	RECORD
Data Type	PDO CommPar
Category	Conditional; Mandatory for each supported PDO

ENTRY DESCRIPTION

Sub-Index	0h
Description	largest sub-index supported
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2 – 5

Sub-Index	1h
Description	COB-ID used by PDO
Entry Category	Mandatory
Access	ro; rw if COB-ID can be configured
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 65)
Default Value	Index 1800h: 180h + Node-ID, Index 1801h: 280h + Node-ID, Index 1802h: 380h + Node-ID, Index 1803h: 480h + Node-ID, Index 1804h - 18FFh: disabled

Sub-Index	2h
Description	transmission type
Entry Category	Mandatory
Access	ro; rw if transmission type can be changed
PDO Mapping	No
Value Range	UNSIGNED8 (Table 54)
Default Value	(device profile dependent)

Sub-Index	3h
Description	inhibit time
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	No

Sub-Index	4h
-----------	----

Description	reserved
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	No

Sub-Index	5h
Description	event timer
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	0 = not used UNSIGNED16
Default Value	No

Object 1A00h – 1A1Fh: Transmit PDO Mapping Parameter

Contains the mapping for the PDOs the device is able to transmit. The type of the PDO mapping parameter (21h) is described in 9.5.4. A detailed description of the entries is done in the section for the Receive PDO Mapping Parameter (1600h – 161Fh).

OBJECT DESCRIPTION

INDEX	1A00h – 1A1Fh
Name	transmit PDO mapping
Object Code	RECORD
Data Type	PDO Mapping
Category	Conditional; Mandatory for each supported PDO

ENTRY DESCRIPTION

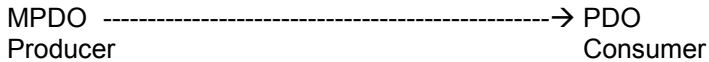
Sub-Index	0h
Description	number of mapped application objects in PDO
Entry Category	Mandatory
Access	ro; rw if dynamic mapping is supported
PDO Mapping	No
Value Range	0: deactivated 1 - 64: activated
Default Value	(device profile dependent)

Sub-Index	1h - 40h
Description	PDO mapping for the n-th application object to be mapped
Entry Category	Conditional; depends on number and size of objects to be mapped
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	(device profile dependent)

Multiplexed PDOs

MPDO Protocol

This protocol is used to implement MPDO services. The MPDO producer sends data and the multiplexor indicating the source or destination address.



Byte	0		1-3	4-8
Meaning	f	Addr	m	D

'd' shall contain the data to be transferred. The value always shall be filled up to 32 bit. ³
 'm' shall contain the multiplexor (Index and Sub-Index) of the variable in the object dictionary.
 The MSB 'f' of the first byte shall be a format flag, and 'addr' shall be an address field, which may be used in the following combinations:

f	addr	Usage
0	0	Reserved
0	1-127	Source addressing. addr is a single producer's Node ID. Multiplexor is index and sub-index of the object dictionary of the producer.
1	0	Destination addressing. The consumer is a group.
1	1-127	Destination addressing. addr is a single consumer's Node ID. Multiplexor is index and sub-index of the object dictionary of the consumer.

Destination Address Mode (DAM)

The *addr* and the *m* field of the MPDO refers to the consumer. This allows access to the consumer's Object Dictionary in an SDO-like manner. With *addr* = 0, it allows multicasting and broadcasting, to write into the Object Dictionaries of more than one node simultaneously, without having a PDO for each single object.

Initiating a DAM-MPDO is application-dependent, like it is for SDOs.

Source Address Mode (SAM)

The *addr* and the *m* field of the MPDO refers to the producer. Only one producer MPDO of this type is allowed for each node.

Transmission type has to be 254 or 255.

The producer uses an Object Scanner List in order to know, which objects are to send. The consumer uses an Object Dispatcher List as a 'cross reference'. ³ The restriction about using 32-bit transfers only will not present problems in practice since all of the participating devices know the data types (and sizes) of their related objects.

Object dictionary entries

PDO Mapping Record

The meaning of Sub-Index 0 (number of mapped objects is extended. The valid range for non-multiplexed PDOs is 0 to 64. A value of 255 indicates a DAM-MPDO, a value of 254 indicates an SAM-MPDO.

For SAM, the further entries in the MR are don't care.

For DAM the first object describes the local object (there can be mapped only one object into an MPDO).

Index	Object	Name	Type
16XXh-1AXXh	0h	Number of mapped objects in the PDO: 0 .. 64: Valid range for number of mapped objects 254: formatted as SAM- MPDO 255: formatted as DAM- MPDO	Unsigned 8

This leaves open the possibility for further alternative PDO formats.

Additional error code meanings

Error Code Meaning

TBD

Predefinitions

Introduction

All TPDOs with transmission type 255 shall be transmitted when entering the OPERATIONAL state.

PDO Mapping

6.2.1 1st RPDO mapping (Memory Access)

This RPDO receives asynchronously the objects that control the memory module access. The default transmission type shall be 255. The default values of the mapped outputs are described in the Default State objects. *Note:* After power-on and application reset these default objects are valid.

Index	Sub-Index	Comment	Default Value
1600h	0h	number of mapped objects	1
1600h	1h	1st object to be mapped	3110 00 40h

6.2.2 1st TPDO mapping (Memory Access)

The default transmission type shall be 255; the default values for inhibit and event timer are 0. If one of the values changes its value, this PDO shall be transmitted immediately.

Index	Sub-Index	Comment	Default Value
1A00h	0h	number of mapped objects	1
1A00h	1h	1st object to be mapped	3111 00 40h

Manufacturer Specific Objects

Overview:

Serial Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2000	ARRAY	Serial Transmit Data	OCTET_STRING	Ro	N	V
2001	ARRAY	Serial Receive Data	OCTET_STRING	Ro	N	V
2002	ARRAY	Serial Request Fragment	SER_REQ_FRAGMENT	Rww	Y	V
2003	ARRAY	Serial Response Fragment	SER_RSP_FRAGMENT	Ro	Y	V
2004	ARRAY	Serial Control Byte	UNSIGNED8	Rw	Y	V
2005	ARRAY	Serial Status Byte	UNSIGNED8	Ro	Y	V

Serial Configuration Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2100	ARRAY	Serial Mode	UNSIGNED8	Rw	N	NV
2101	ARRAY	Serial Flow Control	UNSIGNED8	Rw	N	NV
2102	ARRAY	Serial Baud	UNSIGNED32	Rw	N	NV
2103	ARRAY	Serial Data Bits	UNSIGNED8	Rw	N	NV
2104	ARRAY	Serial Stop Bits	UNSIGNED8	Rw	N	NV
2105	ARRAY	Serial Parity	UNSIGNED8	Rw	N	NV
2106	ARRAY	Serial Char Delay	UNSIGNED8	Rw	N	NV
2107	ARRAY	Serial Start String	OCTET_STRING	Rw	N	NV
2108	ARRAY	Serial Error String	OCTET_STRING	Rw	N	NV

CANopen Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
4000	VAR	CANopen Baud	UNSIGNED32	Rw	N	NV
4001	VAR	CANopen NodeID	UNSIGNED8	Rw	N	NV

Manufacturer Specific Object Details:

Serial Objects

Object 2000h: Serial Transmit Data

This object provides a way for the user to transmit data out of the serial port

Object Description

INDEX	2000h
Name	Serial Transmit Data
Object Code	Array
Data Type	OCTET_STRING
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Transmit Data
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	OCTET_STRING
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Transmit Data
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	OCTET_STRING
Default value	No

The first byte must contain the number of bytes to transmit, followed by the data. It cannot be more than 255 or specify more bytes than is sent in the SDO.

Object 2001h: Serial Receive Data

This object provides a way for the user to receive data in from the serial port

Object Description

INDEX	2001h
Name	Serial Receive Data
Object Code	Array
Data Type	OCTET_STRING
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Receive Data
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	OCTET_STRING
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Receive Data
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	OCTET_STRING
Default value	No

The first byte contains the number of bytes received, followed by the data.

Object 2002h: Serial Request Fragment

This object provides a way for the user to send and receive serial data through process data.

Object Description

INDEX	2002h
Name	Serial Request Fragment
Object Code	Array
Data Type	SER_REQ_FRAGMENT
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Request Fragment
Access	Rww
Entry Category	Mandatory
PDO Mapping	YES
Value Range	SER_REQ_FRAGMENT
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Request Fragment
Access	Rww
Entry Category	Optional
PDO Mapping	YES
Value Range	OCTET_STRING
Default value	No

Format of Fragmented Serial Output Data:

Byte 1: Control Byte

Byte 2: Data-block, if necessary

...

Byte 8: Data-block, if necessary

Control Byte

Byte 1 of the Output Data contains the following

7	6	5	4	3	2	1	0
Frag. Rx Enable	Reset Request	Transmit More	Transmit Request	Receive Ack.	Number of Data Bytes To Transmit		

Bit 7 Reserved:

Bit 6 Reserved:

Bit 5 Receive Ack (RxAck):

This bit must be set to the value of the Receive Request bit (RxReq) in order for the module to receive more data. This tells the module that the “master” has received the data and has processed it.

Bit 4 Transmit Request:

This bit must be toggled to signal the module that there is new data to transmit. On devices that cannot ensure data consistency, the user should first set the number of bytes and place the proper data into the TxData mapping before toggling this bit.

Bit 3 Reset Request:

When this bit is set, all errors are cleared, the buffers are flushed, and the RxReq and TxAck bits are cleared. This allows re-sync of the protocol.

Bit 2:0 Number of Bytes To Transmit:

This tells the module how many bytes of the data are valid and should be transmitted.

Object 2003h: Serial Response Fragment

This object provides a way for the user to send and receive serial data through process data.

Object Description

INDEX	2002h
Name	Serial Response Fragment
Object Code	Array
Data Type	SER_RSP_FRAGMENT
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Response Fragment
Access	Ro
Entry Category	Mandatory
PDO Mapping	YES
Value Range	SER_RSP_FRAGMENT
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Response Fragment
Access	Ro
Entry Category	Optional
PDO Mapping	YES
Value Range	OCTET_STRING
Default value	No

Format of Fragmented Serial Response:

Response:

Byte 1: Status Byte

Byte 2: Data-block, if necessary

...

Byte 8: Data-block, if necessary

Status Byte

Byte 1 of the Input Data contains the following

7	6	5	4	3	2	1	0
Rx Enabled	Reset Ack	Error	Transmit Ack.	Receive Request	Number of Data Bytes Received		

Bit 7 Fragmented Rx Enabled:

When this bit is set the gateway automatically receives data from the queue and starts transfer using process data.

Bit 6: Reset Ack:

If a Reset has been requested, this value will be set to 1 to indicate that the serial port has been reset and the buffers have been flushed. This causes the TxAck and RxReq to be reset to 0 allowing re-sync of protocol.

Bit 5 Error

When this value is set, an error has occurred such as parity or overrun. The user should query the status parameter for more information.

Bit 4: Transmit Ack:

When this value is equal to the Transmit Request, it indicates that the output data has been queued into the output buffer. Once they are equal the user can then send more data.

Bit 3 Receive Request:

This value is toggled to indicate new data has been received. The user must acknowledge the reception of data by echoing back this value in the Receive Ack bit.

Bits 2:0 Number of Bytes Received:

Indicates the number of valid data bytes that are in the data section of the input data. 0-7

Object 2004h: Serial Control Byte

This object provides a way for the user to control the port on the serial gateway, such as flow control etc.

Object Description

INDEX	2004h
Name	Serial Control Byte
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Control Byte
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Control Byte
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	No

Object 2005h: Serial Status Byte

This object provides a way for the user to get the status of the port on the serial gateway, such as parity error etc.

Object Description

INDEX	2005h
Name	Serial Status Byte
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Status Byte
Access	Ro
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Status Byte
Access	Ro
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	No

Serial Configuration Objects

Object 2100h: Serial Mode

This object provides a way for the user to set the operating mode of the serial port on the serial gateway.

Object Description

INDEX	2100h
Name	Serial Mode
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Mode
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Mode
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	No

0=Pass Through

1=Translate

Object 2101h: Serial Flow Control

This object provides a way for the user to set the flow control of the serial port on the serial gateway.

Object Description

INDEX	2101h
Name	Flow Control
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Flow Control
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Flow Control
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	No

0=None

1=CTS/RTS

2=XON/XOFF

Object 2102h: Serial Baud

This object provides a way for the user to set the baud rate of the serial port on the serial gateway.

Object Description

INDEX	2102h
Name	Serial Baud
Object Code	Array
Data Type	UNSIGNED32
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Baud
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED32
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Baud
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED32
Default value	No

The gateway supports the standard serial baud rates (300,600,1200,2400,4800,9600,19200,38400,57600). The user has the option of entering off standard baud rates and the gateway will do its best to match the specified baud rate. These values should be used at the users responsibility.

Object 2103h: Serial Data Width

This object provides a way for the user to set the data width of the serial port on the serial gateway.

Object Description

INDEX	2103h
Name	Serial Data Width
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Data Width
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Data Width
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	No

Supported values:

7

8

Object 2104h: Serial Stop Bits

This object provides a way for the user to set the stop bits of the serial port on the serial gateway.

Object Description

INDEX	2104h
Name	Serial Stop Bits
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Stop Bits
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Stop Bits
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	No

Supported Values

- 1
- 2

Object 2105h: Serial Parity

This object provides a way for the user to set the parity of the serial port on the serial gateway.

Object Description

INDEX	2105h
Name	Serial Parity
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Parity
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Parity
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	No

Supported Values:

- 0=None
- 1=Odd
- 2=Even
- 3=Mark
- 4=Space

Object 2106h: Serial Char Delay

This object provides a way for the user to set the inter-character transmit delay of the serial port on the serial gateway.

Object Description

INDEX	2106h
Name	Serial Char Delay
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Character Delay
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Character Delay
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	No

Object 2107h: Serial Start String

This object provides a way for the user to set the start string of the serial port on the serial gateway.

Object Description

INDEX	2107h
Name	Serial Start String
Object Code	Array
Data Type	OCTET_STRING
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	OCTET_STRING
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Start String
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	OCTET_STRING
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Start String
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	OCTET_STRING
Default value	No

The string defined here will be transmitted out the corresponding port when the module first boots up.

Object 2108h: Serial Error String

This object provides a way for the user to set the error string of the serial port on the serial gateway.

Object Description

INDEX	2108h
Name	Serial Error String
Object Code	Array
Data Type	OCTET_STRING
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Serial Ports
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	OCTET_STRING
Default Value	2

Sub-Index	1h
Description	Serial Port 1 Error String
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	OCTET_STRING
Default Value	No

To

Sub-Index	NUMBER OF SERIAL PORTS
Description	Serial Port N Error String
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	OCTET_STRING
Default value	No

The string defined here will be transmitted out the corresponding port when the module has a CAN error such as a busoff or Node Guard failure.