EPOS2 P

Programmable Positioning Controllers

Firmware Specification

Document ID: rel4619

PLEASE READ THIS FIRST

These instructions are intended for qualified technical personnel. Prior commencing with any activities …

- *you must carefully read and understand this manual and*
- *you must follow the instructions given therein.*

We have tried to provide you with all information necessary to install and commission the equipment in a secure, safe and time-saving manner. Our main focus is …

- to familiarize you with all relevant technical aspects,
- to let you know the easiest way of doing,
- to alert you of any possibly dangerous situation you might encounter or that you might cause if you do not follow the description,
- to write as little and to say as much as possible and
- not to bore you with things you already know.

Likewise, we tried to skip repetitive information! Thus, you will find things mentioned just once. If, for example, an earlier mentioned action fits other occasions you then will be directed to that text passage with a respective reference.

Follow any stated reference – observe respective information – then go back and continue with the task!

PREREQUISITES FOR PERMISSION TO COMMENCE INSTALLATION

The EPOS2 P Programmable Positioning Controllers are considered as partly completed machinery according to EU's directive 2006/42/EC, Article 2, Clause (g) and therefore are intended to be incorporated into or assembled with other machinery or other partly completed machinery or equipment.

You must not put the device into service, …

- *unless you have made completely sure that the other machinery – the surrounding system the device is intended to be incorporated to – fully complies with the requirements stated in the EU directive 2006/42/EC!*
- *unless the surrounding system fulfills all relevant health and safety aspects!*
- *unless all respective interfaces have been established and fulfill the stated requirements!*

TABLE OF CONTENTS

1 About this Document

1.1 Intended Purpose

The purpose of the present document is to familiarize you with the described equipment and the tasks on safe and adequate installation and/or commissioning.

Observing the described instructions in this document will help you …

- to avoid dangerous situations,
- to keep installation and/or commissioning time at a minimum and
- to increase reliability and service life of the described equipment.

Use for other and/or additional purposes is not permitted. maxon motor, the manufacturer of the equipment described, does not assume any liability for loss or damage that may arise from any other and/or additional use than the intended purpose.

1.2 Target Audience

This document is meant for trained and skilled personnel working with the equipment described. It conveys information on how to understand and fulfill the respective work and duties.

This document is a reference book. It does require particular knowledge and expertise specific to the equipment described.

1.3 How to use

Take note of the following notations and codes which will be used throughout the document.

Notation	Explanation
➔	denotes “see”, “see also”, “take note of” or “go to”

Table 1-1 Notations used in this Document

1.4 Symbols and Signs

Requirement / Note / Remark
Indicates an action you must perform prior continuing or refers to information on a particular item.

Best Practice
Gives advice on the easiest and best way to proceed.

Material Damage
Points out information particular to potential damage of equipment.

1.5 Sources for additional Information

For further details and additional information, please refer to below listed sources:

#	Reference
[1]	CiA 301 Communication Profile for Industrial Systems www.can-cia.org
[2]	CiA 302 Framework for CANopen Managers and Programmable CANopen Devices www.can-cia.org (section accessible for CiA members only)
[3]	CiA 405 Interface and Device Profile for IEC 61131-3 Programmable Devices www.can-cia.org
[4]	CiA 306 Electronic Data Sheet Specification www.can-cia.org
[5]	Konrad Etschberger: Controller Area Network ISBN 3-446-21776-2
[6]	maxon motor: EPOS2 Firmware Specification EPOS Positioning Controller DVD or www.maxonmotor.com

Table 1-2 Sources for additional Information

1.6 Trademarks and Brand Names

For easier legibility, registered brand names are listed below and will not be further tagged with their respective trademark. It must be understood that the brands (the below list is not necessarily concluding) are protected by copyright and/or other intellectual property rights even if their legal trademarks are omitted in the later course of this document.

Brand Name	Trademark Owner
CANopen® CiA®	© CiA CAN in Automation e.V, DE-Nuremberg
Windows®	© Microsoft Corporation, USA-Redmond, WA

Table 1-3 Brand Names and Trademark Owners

1.7 Copyright

maxon motor ag
Brünigstrasse 220
P.O.Box 263
CH-6072 Sachseln
Switzerland

Phone +41 41 666 15 00
Fax +41 41 666 16 50

www.maxonmotor.com

2 Introduction

2.1 Important Notice: Prerequisites for Permission to commence Installation

The EPOS2 P Programmable Positioning Controllers are considered as partly completed machinery according to EU's directive 2006/42/EC, Article 2, Clause (g) and therefore are intended to be incorporated into or assembled with other machinery or other partly completed machinery or equipment.

WARNING

Risk of Injury
Operating the device without the full compliance of the surrounding system with the EU directive 2006/42/EC may cause serious injuries!

- *Do not operate the device, unless you have made sure that the other machinery fulfills the requirements stated in EU's directive!*
- *Do not operate the device, unless the surrounding system fulfills all relevant health and safety aspects!*
- *Do not operate the device, unless all respective interfaces have been established and fulfill the stated requirements!*

2.2 General Information

The present document provides you with the firmware details on the EPOS2 P Programmable Positioning Controllers. It contains descriptions of the architecture, error handling and object dictionary.

maxon motor control's EPOS2 P is a small-sized, full digital and free programmable positioning control units. Due to its flexible and high efficient power stage, the EPOS2 P drives brushed DC motors with digital encoder as well as brushless EC motors with digital Hall sensors and encoder.

The sinusoidal current commutation by space vector control offers to drive brushless EC motors with minimal torque ripple and low noise. The integrated position, velocity and current control functionality allows sophisticated positioning applications. The EPOS2 P is programmable with a very efficient software tool. The programming languages are according to IEC 61131-3 standard. The built-in CANopen interface allows the design of an easy-to-use standalone multiple axis system, particularly with standard maxon EPOS controllers. Thanks to the communication interface to the superior process level, supervisory control and data acquisition via RS232; USB 2.0 or CANopen is possible.

Find the latest edition of the present document, as well as additional documentation and software to the EPOS2 P Programmable Positioning Controllers also on the Internet: ➔www.maxonmotor.com

2.3 Documentation Structure

The present document is part of a documentation set. Please find below an overview on the documentation hierarchy and the interrelationship of its individual parts:

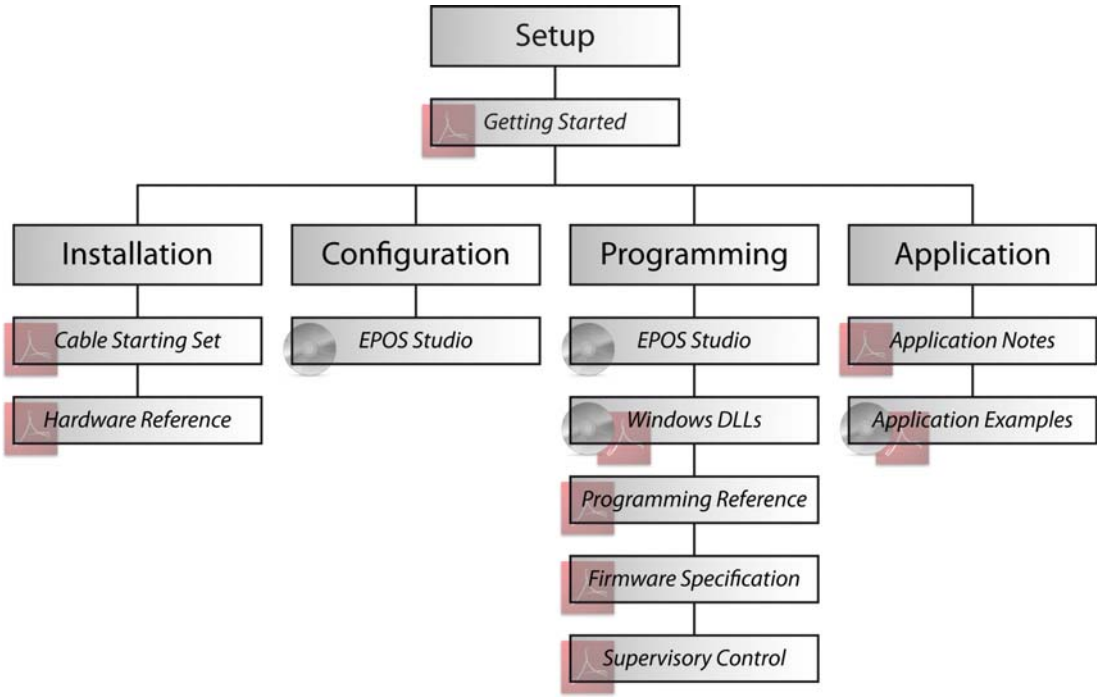

Figure 2-1 Documentation Structure

3 Architecture

3.1 Device Architecture

The communication interface of the EPOS2 P follows the CANopen specifications. Note that numbers in brackets refer to respective items listed on page 1-8:

- CiA 301 V4.2
  Application Layer and Communication Profile (➔[1])
- CiA 302 V3.3
  Framework for CANopen Managers and Programmable CANopen Devices (➔[2])
- CiA 405 V3.0
  Interface and Device Profile or IEC 61131-3 Programmable Devices (➔[3]).
- CiA 306 V1.3
  Electronic Data Sheet Specification(➔[4])

The communication interface and the behavior of the EPOS [Internal] is described in separate document «EPOS2 Firmware Specification», edition April 2010 or later (➔ [6]).

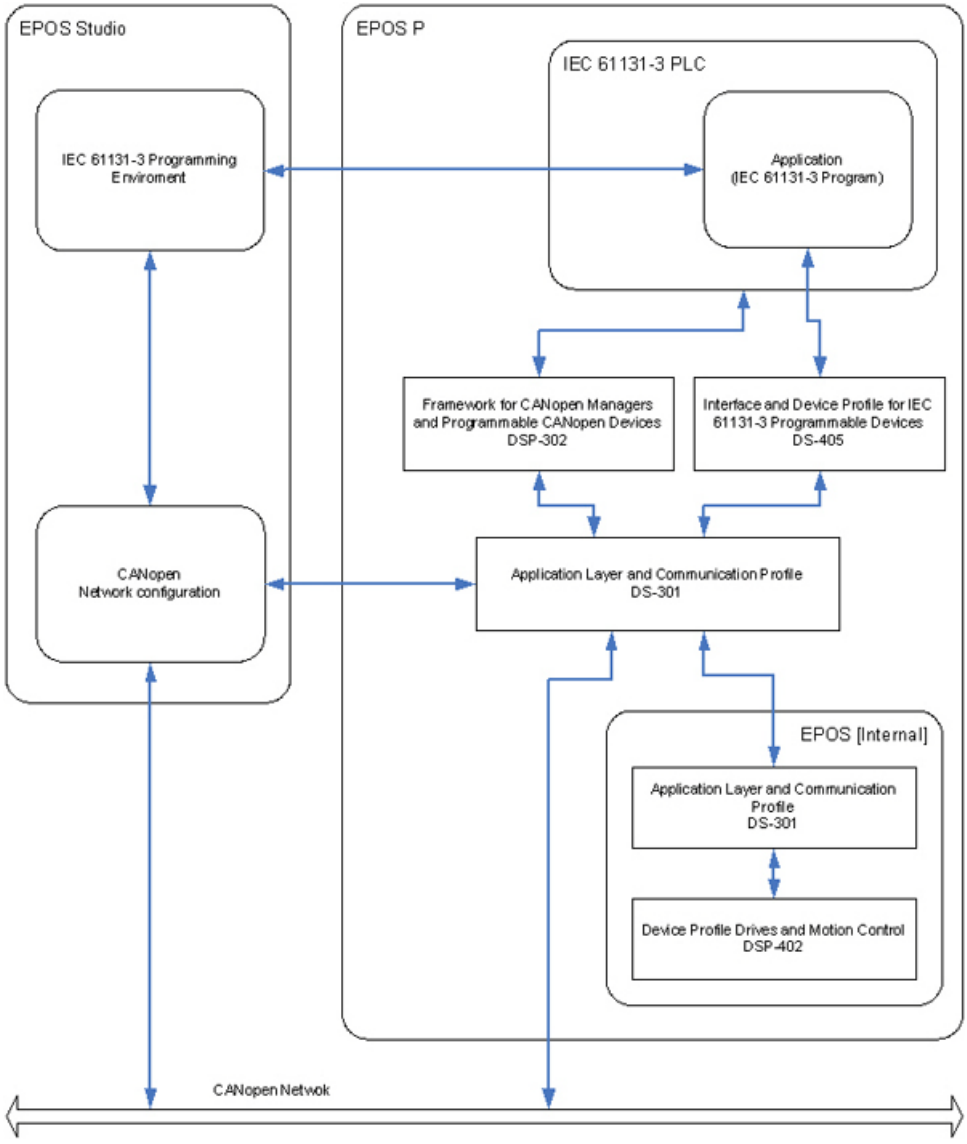

Figure 3-2 Communication Architecture

3.2 System Architecture

3.2.1 Functionalities

The EPOS2 P is designed as a CANopen manager. Therefore, the following functionalities have been implemented:

- Configuration Manger
  The functionality that provides mechanisms for checking the configuration of nodes in a system during boot-up.
- Nmt Master
  The network management (Nmt) provides services for controlling the network behavior of the nodes as defined in CiA 301.
- SYNC Producer
  A functionality responsible for transmitting the SYNC object.

3.2.2 Boot-Up Procedure

The Configuration Manager has the task of booting all assigned slaves and checking their configuration. At each cold or warm start of the application program, the CANopen boot-up procedure will be executed before the EPOS2 P switches the state from Pre-Operational to Operational. The main flow resumes in figure below.

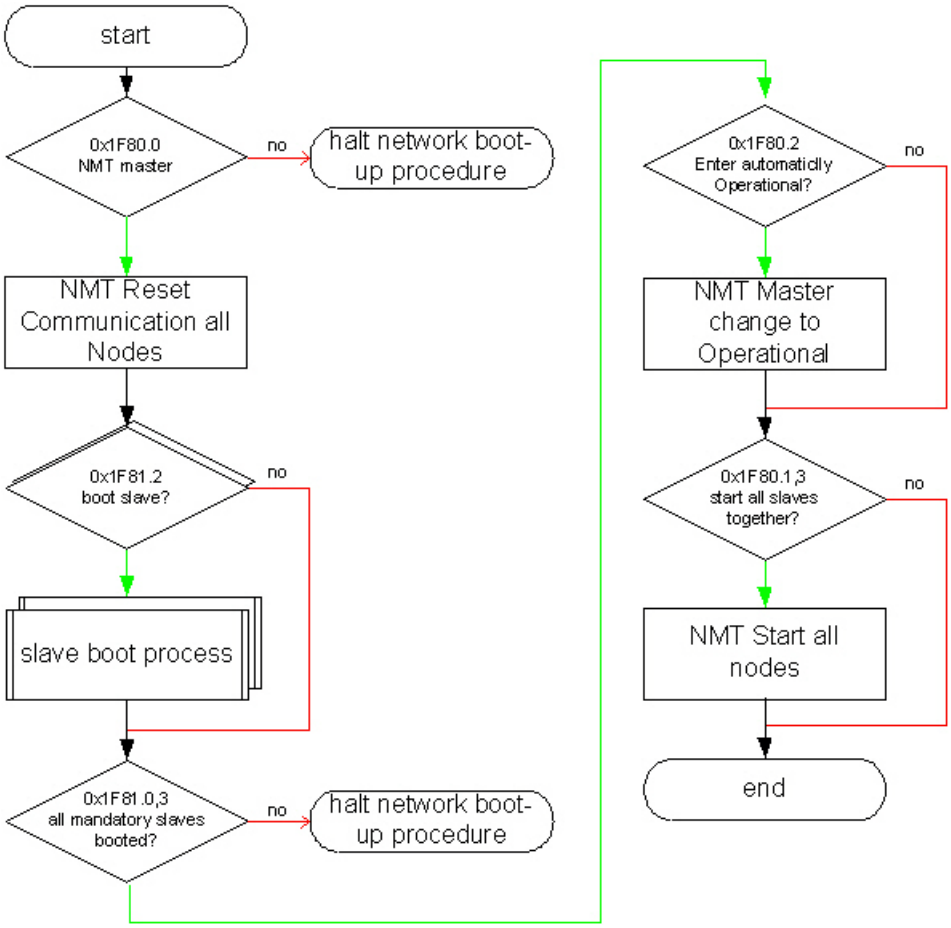

Figure 3-3 CANopen Boot-up Procedure – Main Flow

The slave boot process is given by figure below. Checks the stored slave configurations in the objects 0x1F84 to 0x1F89, 0x1F26 and 0x1F27 and produces an error if a mandatory slave does not match.

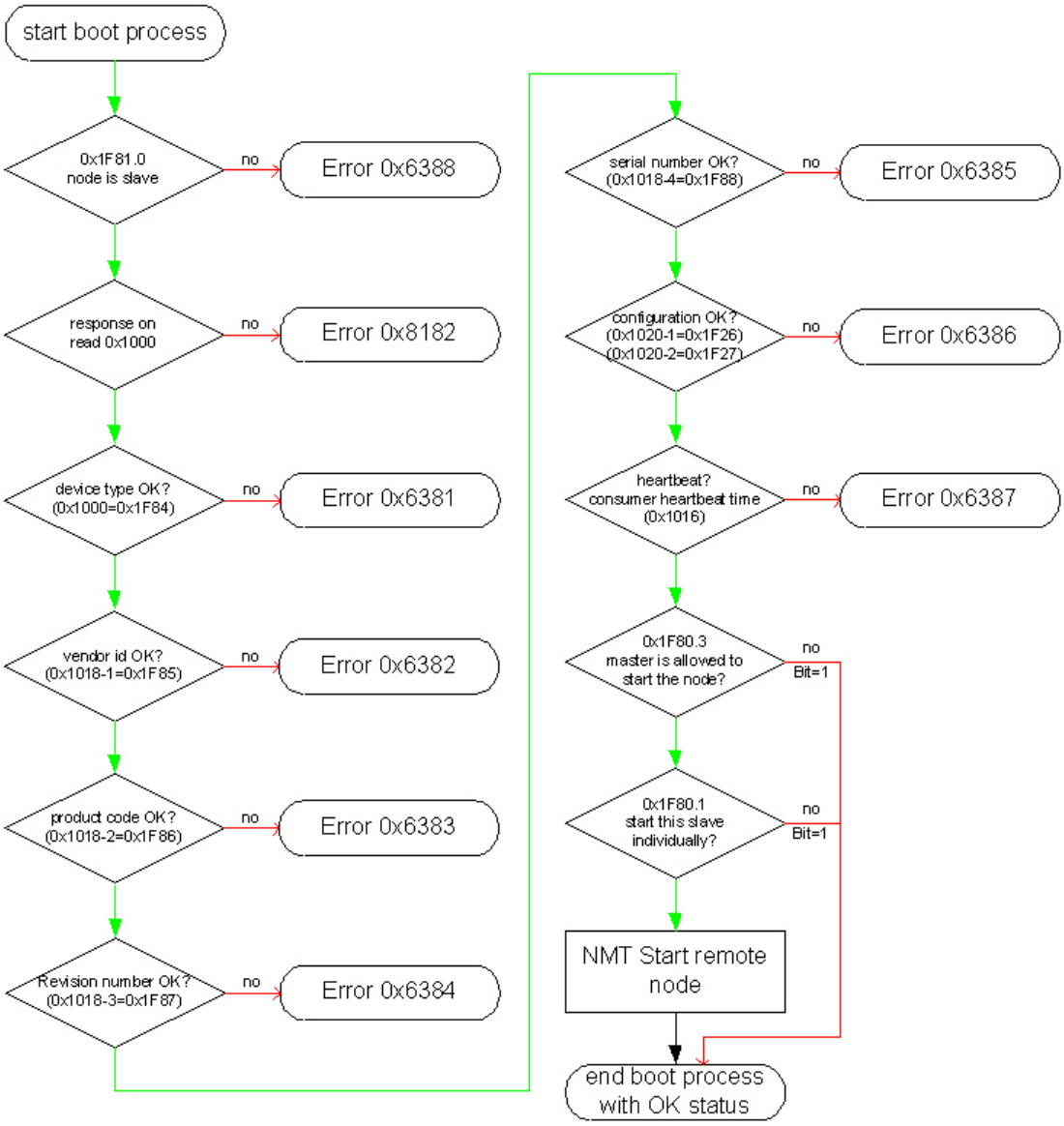

Figure 3-4 CANopen Boot-up Procedure – Slave Boot Process

••page intentionally left blank••

4 Communication

The EPOS2 supports RS232, USB and CANopen communication profiles (for details ➔separate document «EPOS2 Communication Guide»).

Most important communication objects are as follows:

➔“RS232 Baudrate” on page 6-84
➔“RS232 Frame Timeout” on page 6-87
➔“CAN Bitrate” on page 6-83
➔“CAN Bitrate Display” on page 6-88
➔“Consumer Heartbeat Time” on page 6-52
➔“Producer Heartbeat Time” on page 6-53
➔“Identity Object” on page 6-53
➔“Node ID” on page 6-82
➔“COB-ID EMCY Message” on page 6-51
➔“COB-ID SYNC Message” on page 6-47
➔“Receive PDO Parameter” on page 6-60
➔“Receive PDO Mapping” on page 6-61
➔“Transmit PDO Parameter” on page 6-63
➔“Transmit PDO Mapping” on page 6-66

4.1 CANopen Node Identification

Within the CANopen network, a unique Node Identification Number (Node ID) is allocated to each individual CANopen device. The EPOS2 P Node ID can be set by either Hardware Switches or software (➔“Node ID” on page 6-82).

4.2 CAN Bitrate

Within a CANopen network it is of importance that all devices communicate with the very same bit rate. To change bit rate ➔“CAN Bitrate” on page 6-83.

4.3 CANopen Network Management (NMT)

CANopen network management follows a master/slave structure and is node-oriented. It requires one device in the network, which fulfills the function of the NMT Master. The other nodes (as well as the EPOS2 P) are NMT Slaves.

Each NMT slave device features an implemented state machine, which arranges the allowed type of communication with the device.

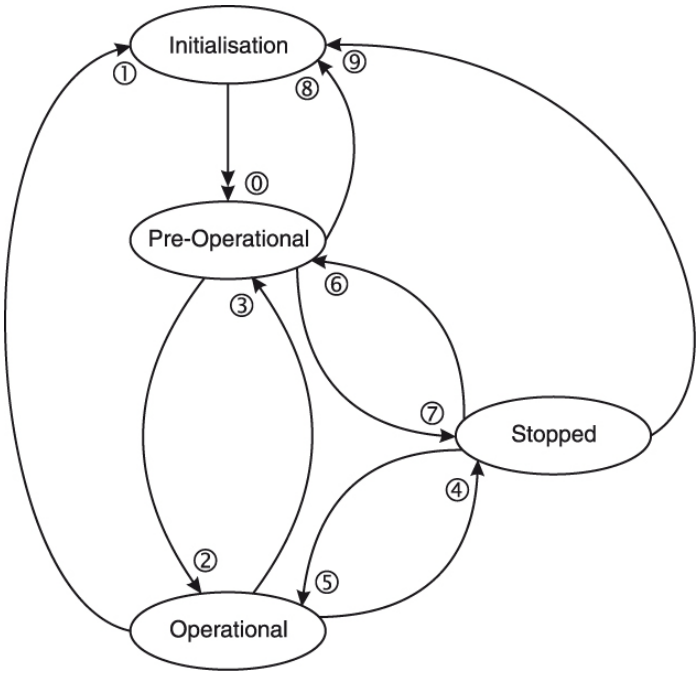

Figure 4-5 NMT Slave – State

REMARKS

*1) Command may be sent with Network Management (NMT) Protocol.
*2) This Transition is generated automatically by the EPOS2 P after initialization is completed. After initialization a Boot-Up message is send.
*3) Remote flag Bit 9 of the ➔"Axis 0 to 31 Statusword" on page 6-91.

Service *1	Transi-tion	NMT State after Command	Remote *3	Functionality
–*2	0	Pre-Operational	FALSE	Communication: – Service Data Objects (SDO) Protocol – Emergency Objects – Network Management (NMT) Protocol
Enter Pre-Operational	3, 6	Pre-Operational	FALSE	
Reset Communi-cation	1, 8, 9	Initialization (Pre-Operational)	FALSE	Calculates SDO COB-IDs. Setup Dynamic PDO-Mapping and calculates PDO COB-IDs. Communication: – While initialization is active, no communication is supported. – Upon completion, a boot-up message will be sent to the CAN bus.
Reset Node	1, 8, 9	Initialization (Pre-Operational)	FALSE	Generates a general reset of EPOS2 P software having same effect as turning off and on the supply voltage. Not saved parameters will be overwritten with values saved to the EEPROM using «Save all Parameters».

Service *1	Transi-tion	NMT State after Command	Remote *3	Functionality
Start Remote Node	2, 5	Operational	TRUE	Communication: – Service Data Objects (SDO) Protocol – Process Data Objects (PDO) Protocol – Emergency Objects – Network Management (NMT) Protocol
Stop Remote Node	4, 7	Stopped	FALSE	Communication: – Network Management (NMT) Protocol – Layer setting services (LSS) – Lifeguarding (Heartbeating)

Table 4-4 NMT Slave – Commands, Transitions and States

4.3.1 Enter Pre-Operational

Used to change NMT state of a particular or of all NMT slaves to “Pre-Operational”.
In state “Pre-Operational”, PDO communication may be configured by the following objects:

- ➔“Receive PDO Parameter” on page 6-60
- ➔“Transmit PDO Parameter” on page 6-63
- ➔“Receive PDO Mapping” on page 6-61
- ➔“Transmit PDO Mapping” on page 6-66.

cs	0x80	(NMT command specifier NMT command Enter Pre-Operational)
Node ID	1…127 0	NMT slave with given Node ID will enter NMT state Pre-Operational All NMT Slaves will enter NMT state Pre-Operational

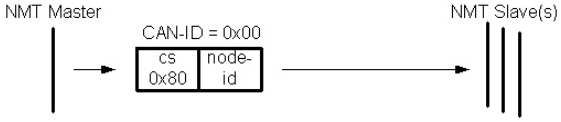

Figure 4-6 NMT Enter Pre-Operational

4.3.2 Reset Communication

Used to reset communication of a particular or of all NMT slaves.
After state “Initialization”, NMT slave changes automatically to “Pre-Operational”.

cs	0x82	(NMT command specifier NMT command Reset Comminution)
Node ID	1…127 0	NMT slave with given Node ID will reset communication All NMT Slaves will reset communication

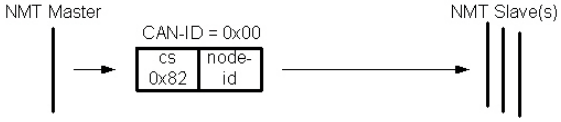

Figure 4-7 NMT Reset Communication

4.3.3 Reset Node

Used to reset a particular or all NMT slaves.
After state “Initialization”, NMT slave changes automatically to “Pre-Operational”.

cs	0x81	(NMT command specifier NMT command Reset Node)
Node ID	1…127 0	Reset of NMT slave with given Node ID Reset of all NMT Slaves

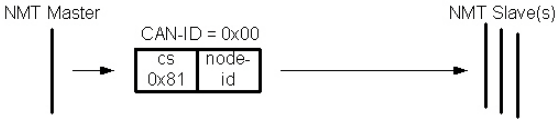

Figure 4-8 NMT Reset Node

4.3.4 Start Remote Node

Used to change NMT state of one or all NMT slave to “Operational”.
In state “Initialization”, all communication protocols are permitted, especially PDO communication.

cs	0x01	(NMT command specifier NMT command State Remote Node)
Node ID	1…127 0	Start of NMT slave with given Node ID Start of all NMT Slaves

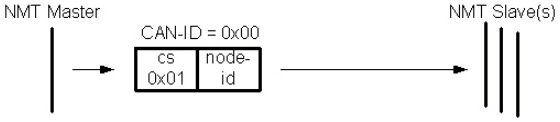

Figure 4-9 NMT Start Remote Node

4.3.5 Stop Remote Node

Used to change the NMT state of only one or all NMT slave to “Stopped”.
In state “Stopped”, only Network Management, Lifeguarding, Heartbeating and ➔Layer Setting Services (LSS) are permitted.

REMARKS

➔“COB-ID EMCY Message” on page 6-51 will not be launched in this state.

cs	0x02	(NMT command specifier NMT command Stop Remote Node)
Node ID	1…127 0	Stop of NMT slave with given Node ID Stop of all NMT Slaves

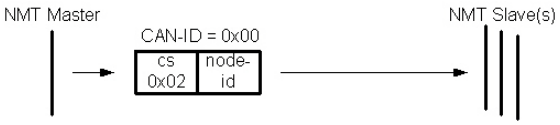

Figure 4-10 NMT Stop Remote Node

4.4 Layer Setting Services (LSS)

By using layer setting services and protocols, a LSS Slave may be configured via CAN network without using DIP switches for setting the Node ID and bit timing parameters.The CANopen device that can configure other devices via CANopen network is called «LSS Master». There must be only one (active) LSS master in a network.

The CANopen device that will be configured by the LSS Master via CANopen network is called «LSS Slave».

An LSS Slave can be identified by its worldwide (at least network-wide) unique LSS address. The LSS address consists of the sub objects «Vendor ID», «Product Code», «Revision Number» and «Serial Number» of the CANopen ➔"Identity Object" on page 6-53. In the network, there must not be other LSS Slaves possessing the same LSS address.
With this unique LSS address an individual CANopen device can be allocated within the network. The Node ID is valid if it is in the range of 0x01…0x7F, values 0xFF and 0x00 identify not configured CANopen devices.

Communication between LSS Master and LSS Slaves is accomplished by LSS protocols which use only two COB-IDs:

- LSS master message from LSS Master to LSS Slaves (COB-ID 0x7E5)
- LSS slave message from the LSS Slaves to LSS Master (COB-ID 0x7E4).

Layer Setting Services are only accessible in NMT slave state "Stopped". To enter Stopped state, the ➔"Stop Remote Node" on page 4-18 is used.

4.4.1 Overview

The table below provides an overview on the LSS commands, including details on whether they may used in states "Waiting" and "Configuration". To change the LSS state, the LSS commands ➔Switch State Global or ➔Switch State Selective may be used.

Command Specifier	LSS Command	LSS State Waiting	LSS State Configuration
0x04	➔Switch State Global	yes	yes
0x40…0x43	➔Switch State Selective	yes	no
0x11	➔Configure Node ID	no	yes
0x13	➔Configure Bit Timing Parameters	no	yes
0x15	➔Activate Bit Timing Parameters	no	yes
0x17	➔Store Configuration Protocol	no	yes
0x5A	➔Inquire Identity Vendor ID	no	yes
0x5B	➔Inquire Identity Product Code	no	yes
0x5C	➔Inquire Identity Revision Number	no	yes
0x5D	➔Inquire Identity Serial Number	no	yes
0x5E	➔Inquire Identity Node ID	no	yes
0x46…0x4B	➔Identify Remote Slave	yes	yes
0x4C	➔Identify non-configured Remote Slave	yes	yes

Table 4-5 LSS Commands – Overview

4.4.2 LSS Commands

4.4.2.1 Switch State Global

Changes state of all connected LSS Slaves to “Configuration” or back to “Waiting”. Thereby, particular LSS commands are not permitted (➔Table 4-5).

cs	0x04	LSS command specifier 4 or switch state global
mode	0 1	switch to LSS state waiting switch to LSS state configuration

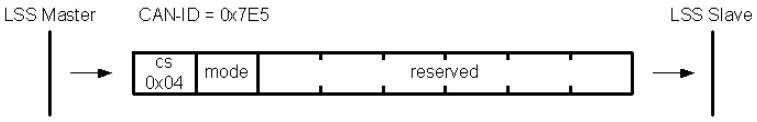

Figure 4-11 LSS – Switch State Global

4.4.2.2 Switch State Selective

Changes state of one LSS Slave from “Waiting” to “Configuration”.

LSS command specifier…

- 0x40 is used to submit the Vendor ID,
- 0x41 to submit the Product Code,
- 0x42 to submit the Revision Number,
- 0x43 to submit the Serial Number (of the ➔“Identity Object” on page 6-53).

Then, the single addressed LSS Slave changes to configuration state and answers by sending a command specifier 0x44 response.

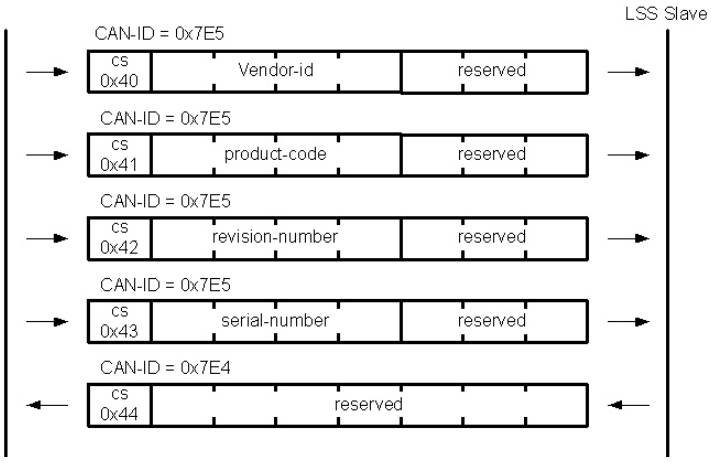

Figure 4-12 LSS – Switch State Selective

4.4.2.3 Configure Node ID

Configures the Node ID (of value 1…127).

The LSS Master must determine the LSS Slave's Node ID in LSS configuration state. The LSS Master is responsible to switch a single (only one!) LSS Slave into LSS state "Configuration" (➔Switch State Selective) before requesting this service.

cs	0x11	LSS Slave answers with error code and specific error
error code	0 1	protocol successfully completed Node ID out of value range
specific error	always 0	

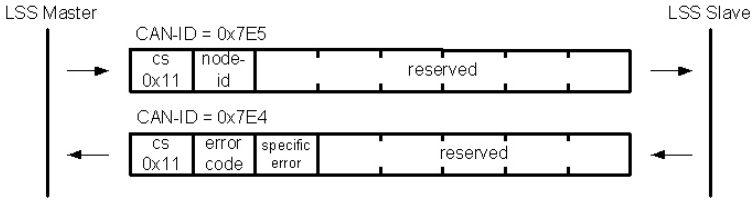

Figure 4-13 LSS – Configure Node ID

4.4.2.4 Configure Bit Timing Parameters

By means of the service configure bit timing parameters, the LSS Master must configure new bit timing on a single LSS Slave. The new bit timing will be active not before receiving ➔Store Configuration Protocol and ➔Activate Bit Timing Parameters.

table selector	always 0	
table index	CAN bit rate codes	
error code	0 1	protocol successfully completed bit timing not supported
specific error	always 0	

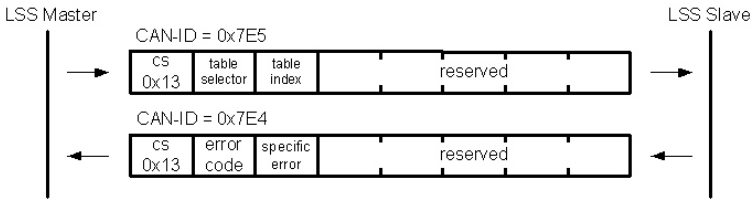

Figure 4-14 LSS – Configure Bit Timing Parameters

4.4.2.5 Activate Bit Timing Parameters

Activates bit timing parameters selected with ➔Configure Bit Timing Parameters.

switch delay	The duration [ms] of the two periods 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).

Upon receiving an activate bit timing command, the LSS Slave stops communication on old (actual) bit rate. After the first switch delay, communication is switched to new bit rate, after a second switch delay, the LSS Slave is permitted to communicate with new bit rate.

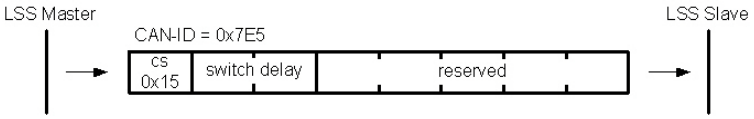

Figure 4-15 LSS – Activate Bit Timing Parameters

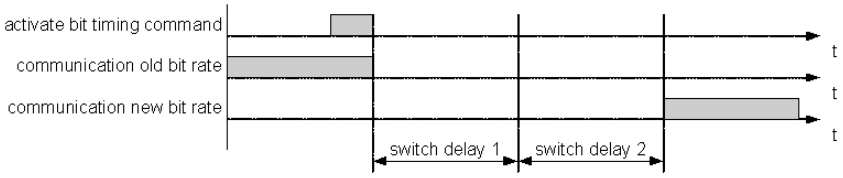

Figure 4-16 LSS – Switch Delay

4.4.2.6 Store Configuration Protocol

Stores all parameter in non-volatile memory. The functionality is equal to the store function commanded in ➔“Store Parameters” on page 6-50.

error code	0 1 2	protocol successfully completed store configuration is not supported storage media access error
specific error	always 0	

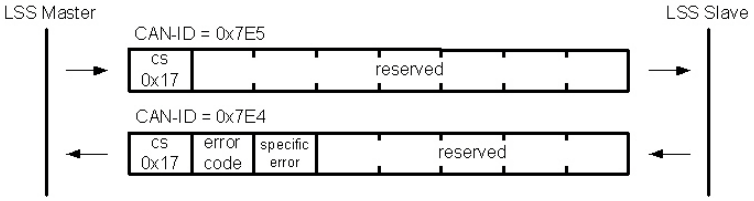

Figure 4-17 LSS – Store Configuration

4.4.2.7 Inquire Identity Vendor ID

Reads «Vendor ID» of a LSS Slave (➔“Identity Object” on page 6-53).

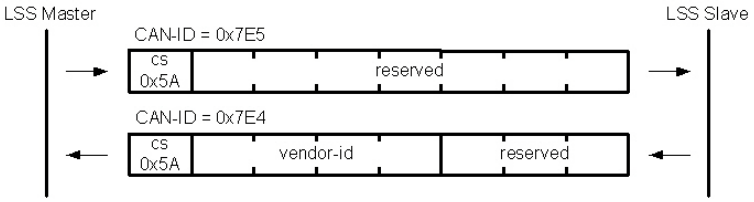

Figure 4-18 LSS – Inquire Identity Vendor ID

4.4.2.8 Inquire Identity Product Code

Reads «Product Code» of a LSS Slave (➔“Identity Object” on page 6-53).

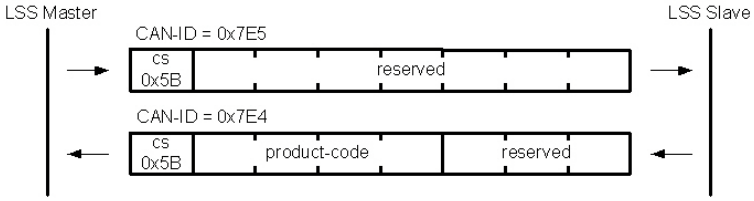

Figure 4-19 LSS – Inquire Identity Product Code

4.4.2.9 Inquire Identity Revision Number

Reads «Revision Number» of a LSS Slave (➔“Identity Object” on page 6-53).

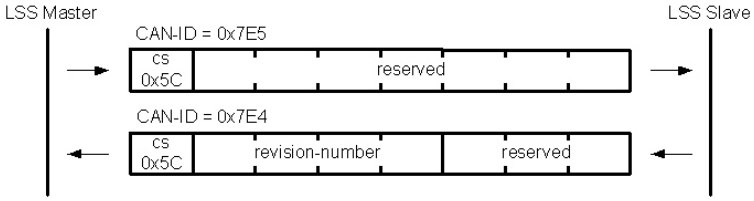

Figure 4-20 LSS – Inquire Identity Revision Number

4.4.2.10 Inquire Identity Serial Number

Reads «Serial Number» of a LSS Slave (➔“Identity Object” on page 6-53).

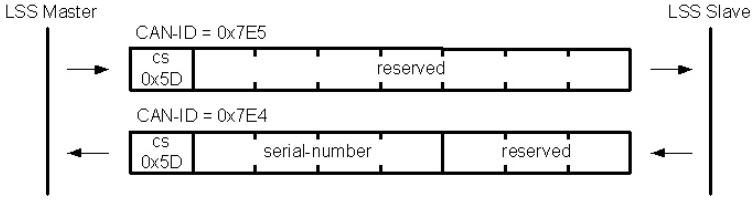

Figure 4-21 LSS – Inquire Identity Serial Number

4.4.2.11 Inquire Identity Node ID

Reads «Node ID» of a LSS Slave (➔“Node ID” on page 6-82).

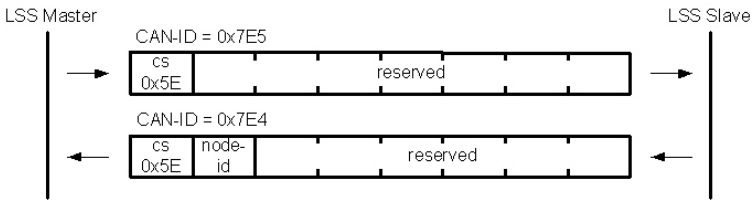

Figure 4-22 LSS – Inquire Identity Node ID

4.4.2.12 Identify Remote Slave

Detects LSS Slaves in the CAN network. Thereby, the LSS Master sends an identify remote slave request with a single Vendor ID, a single Product Code and a span of Revision Numbers and Serial Numbers determined by a low and a high number to the LSS Slaves. All LSS Slaves which meet this LSS address range (inclusive boundaries) answer by a identify slave response (cs = 0x4F).

Along with this protocol, a binary network search can be implemented for the LSS Master. This method sets the LSS address range to the full address area first, then requests the identify remote slave. The range (which comprises one or more responded LSS Slaves) will be split in two sub-areas. The request to the sub-areas will be repeated until each LSS Slave has been identified (➔"Identity Object" on page 6-53, «➔Vendor ID», «➔Product Code», «➔Revision Number» and «➔Serial Number»).

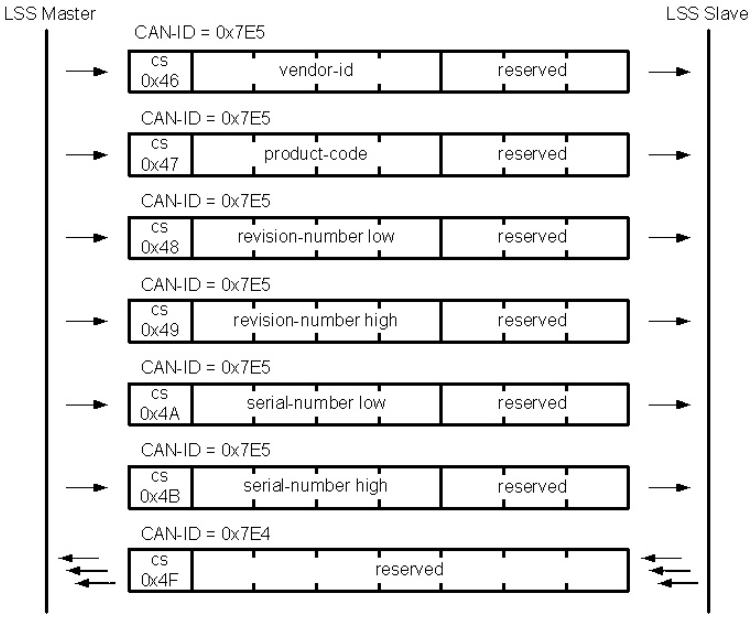

Figure 4-23 LSS – Identify Remote Slave

4.4.2.13 Identify non-configured Remote Slave

Allows the LSS Master to detect presence of a non-configured device in the network. All LSS Slaves without configured Node ID (0xFF or 0x00) will answer with a command specifier 0x50 response.

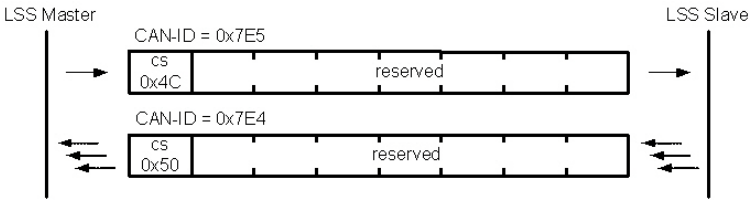

Figure 4-24 LSS – Identify non-configured Remote Slave

5 Error Handling

5.1 EPOS2 P Error Codes

5.1.1 Overview

The error detected by the EPOS2 P will be stored in the Error History (➔page 6-46).

Error Code	Error Register	Name	Comment
0x5530	1000 0001b	➔Eeprom Error	
0x6100	1000 0001b	➔Internal Software Error	
0x6181	1000 0001b	➔Interpreter Error	
0x6182	1000 0001b	➔Native Code Error	
0x6183	1000 0001b	➔Timer Task Error	
0x6184	1000 0001b	➔Interrupt Task Error	
0x6185	1000 0001b	➔Data Access Error	
0x6281	1000 0001b	➔Stack Overflow Error	
0x6310	0000 0000b	➔Retain Variable Error	Warning only
0x6381	1000 0001b	➔Device Type Error	
0x6382	1000 0001b	➔Vendor-Id Error	
0x6383	1000 0001b	➔Product Code Error	
0x6384	1000 0001b	➔Revision Number Error	
0x6385	1000 0001b	➔Serial Number Error	
0x6386	1000 0001b	➔Configuration Time and Date Error	
0x6387	1000 0001b	➔Error Control Error	
0x6388	1000 0001b	➔Node is not Slave Error	
0x8110	0000 0000b	➔CAN Overrun Error	Warning only
0x8120	0001 0001b	➔CAN Passive Error	Warning or Error
0x8130	0001 0001b	➔CAN Lifeguard Error	
0x8181	0001 0001b	➔CAN Controller Error	
0x8182	0001 0001b	➔CAN No Answer Error	Warning or Error
0x81FC	0000 0000b	➔CAN Warning	Warning only
0x81FD	0001 0001b	➔CAN Bus Off Error	
0x8210	0001 0001b	➔CAN PDO Length Error	
0xFF81	0001 0001b	➔Serial Overrun Error	

Table 5-6 EPOS2 P Error Codes

5.1.2 Eeprom Error

Error Code	0x5530
Error Register	1000 0001b
Cause	No proper access to eeprom memory while booting the device.
Effect	Application program not available.
Error Recovery	Reset the device.

5.1.3 Internal Software Error

Error Code	0x6100
Error Register	1000 0001b
Cause	Internal software error occurred.
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.4 Interpreter Error

Error Code	0x6181
Error Register	1000 0001b
Cause	Error occurred at interpreting code.
Effect	Application program in error state.
Error Recovery	Improve programming of the application.

5.1.5 Native Code Error

Error Code	0x6182
Error Register	1000 0001b
Cause	Error occurred upon executing native code.
Effect	Application program in error state.
Error Recovery	Improve programming of the application.

5.1.6 Timer Task Error

Error Code	0x6183
Error Register	1000 0001b
Cause	Error occurred at starting of a timer task since previous call has not yet started.
Effect	Application program in error state.
Error Recovery	Improve programming of the application.

5.1.7 Interrupt Task Error

Error Code	0x6184
Error Register	1000 0001b
Cause	Error occurred at starting of an interrupt/event task since previous call has not yet started.
Effect	Application program in error state.
Error Recovery	Improve programming of the application.

5.1.8 Data Access Error

Error Code	0x6185
Error Register	1000 0001b
Cause	Data or program access exception detected.
Effect	Application program in error state.
Error Recovery	Improve programming of the application.

5.1.9 Stack Overflow Error

Error Code	0x6281
Error Register	1000 0001b
Cause	Stack overflow in application task detected.
Effect	Application program in error state.
Error Recovery	Improve programming of the application.

5.1.10 Retain Variable Error

Error Code	0x6310
Error Register	0000 0000b
Cause	Restoring of retain variables failed at application program warmstart/hotstart.
Effect	Coldstart of application program. Error is registered in the error history.
Error Recovery	Clear error history.

5.1.11 Device Type Error

Error Code	0x6381
Error Register	1000 0001b
Cause	Network bootup failed. Node with inaccurate device type detected in the network. • Wrong network configuration. • Incorrect device connected to the network.
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.12 Vendor-Id Error

Error Code	0x6382
Error Register	1000 0001b
Cause	Network bootup failed. Node with inaccurate vendor-Id detected in the network. • Wrong network configuration. • Incorrect device connected to the network.
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.13 Product Code Error

Error Code	0x6383
Error Register	1000 0001b
Cause	Network bootup failed. Node with inaccurate product code detected in the network. • Wrong network configuration. • Incorrect device connected to the network.
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.14 Revision Number Error

Error Code	0x6384
Error Register	1000 0001b
Cause	Network bootup failed. Node with inaccurate revision number detected in the network. • Wrong network configuration. • Incorrect device connected to the network.
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.15 Serial Number Error

Error Code	0x6385
Error Register	1000 0001b
Cause	Network bootup failed. Node with inaccurate serial number detected in the network. • Wrong network configuration. • Incorrect device connected to the network.
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.16 Configuration Time and Date Error

Error Code	0x6386
Error Register	1000 0001b
Cause	Network bootup failed. Node with inaccurate configuration of date/time detected in the network. • Wrong network configuration. • Device with incorrect or changed configuration connected to the network.
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.17 Error Control Error

Error Code	0x6387
Error Register	1000 0001b
Cause	Network bootup failed. CAN life guarding procedure (heartbeat) has failed. • Heartbeat producer is not enabled properly. • Hardware wiring of CAN bus not correct. • Wrong network configuration.
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.18 Node is not Slave Error

Error Code	0x6388
Error Register	1000 0001b
Cause	Network bootup failed. Mandatory node with inaccurate configuration detected in the network. • Wrong configuration, node is not defined as slave device. • Wrong network configuration.
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.19 CAN Overrun Error

Error Code	0x8110
Error Register	0000 0000b
Cause	CAN controller reports an overflow caused by too high communication rate (frames lost).
Effect	Error is registered in the error history.
Error Recovery	Clear error history.

5.1.20 CAN Passive Error

Error Code	0x8120
Error Register	0001 0001b
Cause	Device changed to CAN passive Mode due to… • too high communication failure rate, • incorrect CAN baud rate of one CAN node in network, • CAN network not connected, • hardware wiring of CAN bus not correct.
Effect	Application program in error state.
Error Recovery	Restart application program.

Remark
If no Node is configured as mandatory device, the error is only indicated as a warning. The error will not have any practical effect but will be registered in the error history.

5.1.21 CAN Lifeguard Error

Error Code	0x8130
Error Register	0001 0001b
Cause	CAN life guarding procedure (heartbeat) has failed due to… • heartbeat producer is not configured properly, • hardware wiring of CAN bus not correct, • wrong network configuration.
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.22 CAN Controller Error

Error Code	0x8181
Error Register	0001 0001b
Cause	CAN controller error due to wrong configuration or malfunctioning hardware.
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.23 CAN No Answer Error

Error Code	0x8182
Error Register	0001 0001b
Cause	Network bootup failed. Mandatory node does not answer/is not detected in the network. • Hardware wiring of CAN bus not correct. • Wrong network configuration.
Effect	Application program in error state.
Error Recovery	Restart application program.

Remark
If the node is not configured as mandatory device, the error is only indicated as a warning. The error will not have any practical effect but will be registered in the error history.

5.1.24 CAN Warning

Error Code	0x81FC
Error Register	0000 0000b
Cause	CAN controller has detected frame warning level caused by high communication failure rate.
Effect	Error is registered in the error history.
Error Recovery	Clear error history.

5.1.25 CAN Bus Off Error

Error Code	0x81FD
Error Register	0001 0001b
Cause	CAN Controller has entered CAN bus off state due to too many communication failures.
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.26 CAN PDO Length Error

Error Code	0x8210
Error Register	0001 0001b
Cause	Received PDO was not processed due to length error (to short).
Effect	Application program in error state.
Error Recovery	Restart application program.

5.1.27 Serial Overrun Error

Error Code	0xFF81
Error Register	0001 0001b
Cause	Serial communication interface buffer overflow.
Effect	Serial communication aborted.
Error Recovery	Connect to the device again. Clear error history.

5.2 Communication Errors (Abort Codes)

An abort object will be sent over the CANopen network instead of a response to a SDO request if the request failed. The same abort code will be sent as part of the response to the RS232 and USB transfer request.

The following Abort Codes are defined by CANopen specification CiA 301 Communication Profile (the codes greater then 0x0F00 0000 are maxon-specific).

Abort Code	Description	Comment
0x0000 0000	No Communication Error	The communication was successful
0x0503 0000	Toggle Error	Toggle bit not alternated
0x0504 0000	SDO Time Out	SDO protocol timed out
0x0504 0001	Client/Server Specifier Error	Client/server command specifier not valid or unknown
0x0504 0005	Out of Memory Error	Out of memory
0x0601 0000	Access Error	Unsupported access to an object
0x0601 0001	Write Only	Read command to a write only object
0x0601 0002	Read Only	Write command to a read only object
0x0602 0000	Object does not exist Error	The last read or write command had a wrong object index or -subindex
0x0604 0041	PDO mapping Error	The object is not mappable to the PDO
0x0604 0042	PDO Length Error	The number and length of the objects to be mapped would exceed PDO length
0x0604 0043	General Parameter Error	General parameter incompatibility
0x0604 0047	General Internal Incompatibility Error	General internal incompatibility in device
0x0606 0000	Hardware Error	Access failed due to an hardware error
0x0607 0010	Service Parameter Error	Data type does not match, length or service parameter does not match
0x0607 0012	Service Parameter too Long Error	Data type does not match, length of service parameter too high
0x0607 0013	Service Parameter too Short Error	Data type does not match, length of service parameter too low
0x0609 0011	Object Subindex Error	The last read or write command had a wrong object subindex
0x0609 0030	Value Range Error	Value range of parameter exceeded
0x0609 0031	Value too High Error	Value of parameter written too high
0x0609 0032	Value too Low Error	Value of parameter written too low
0x0609 0036	Maximum less Minimum Error	Maximum value is less than minimum value
0x0800 0000	General Error	General error
0x0800 0020	Transfer or Store Error	Data cannot be transferred or stored
0x0800 0021	Local control Error	Data cannot be transferred or stored to application because of local control
0x0800 0022	Wrong Device State	Data cannot be transferred or stored to application because of the present device state
0x0A00 0001	Network Id unknown	Network Id is unknown (does not exist in routing list)

Abort Code	Description	Comment
0x0A00 0002	Node ID unknown	Node ID is unknown
0x0F00 FFBF	Illegal Command Error	The serial communication interface command is illegal (does not exist)
0x0F00 FFC0	Wrong NMT State Error	The device is in wrong NMT state
0x0F00 FFC2	Segmented Transfer required	Segmented transfer required (Initialization is already done)
0x0F01 0110	Program flashing error	For details ➔“Flash Status Identification” on page 6-75
0x0FFF FFF0	Communication Sequence Error	Error during function block execution
0x0FFF FFF1	Communication aborted	Communication has been aborted
0x0FFF FFF2	Communication Buffer Overflow	Communication buffer overflow
0x0FFF FFF9	Segmented Transfer Communication Error	Segmented transfer communication error
0x0FFF FFFA	Wrong Axis Number	The axis number was not within 0…32
0x0FFF FFFB	Wrong CAN Device	The can device number was not within 0…127
0x0FFF FFFC	Wrong CAN Port	Can Port is not valid (not 1 or 2)
0x0FFF FFFD	Wrong Parameter	Internal function calling parameters wrong
0x0FFF FFFE	General Communication Error	General communication error
0x0FFF FFFF	Communication Timeout	Communication timeout occurred

Table 5-7 Abort Codes

••page intentionally left blank••

6 Object Dictionary

6.1 Overview

6.1.1 Object Dictionaries and Addressing Scheme

The master controller EPOS2 P features three object dictionaries.

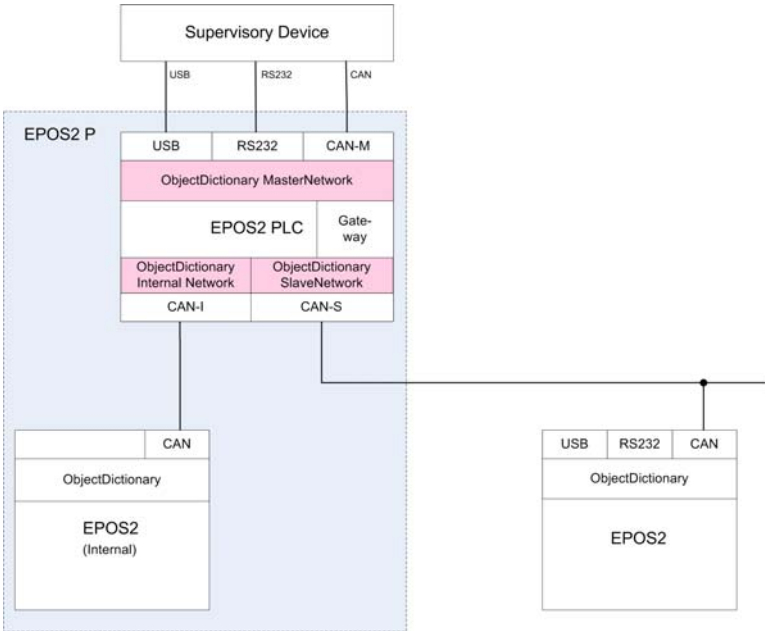

Figure 6-25 Object Dictionaries

Object Dictionary	Description
Master Network	contains objects for CAN-M, USB, RS232: communication to the supervisor
Internal Network	contains Objects for CAN-I internal: used only for communication to the internal EPOS2
Slave Network	contains Objects for CAN-S: Used for external slaves

Table 6-8 Object Dictionaries

The different object dictionaries can be accessed with the combination of Network ID and Node ID.

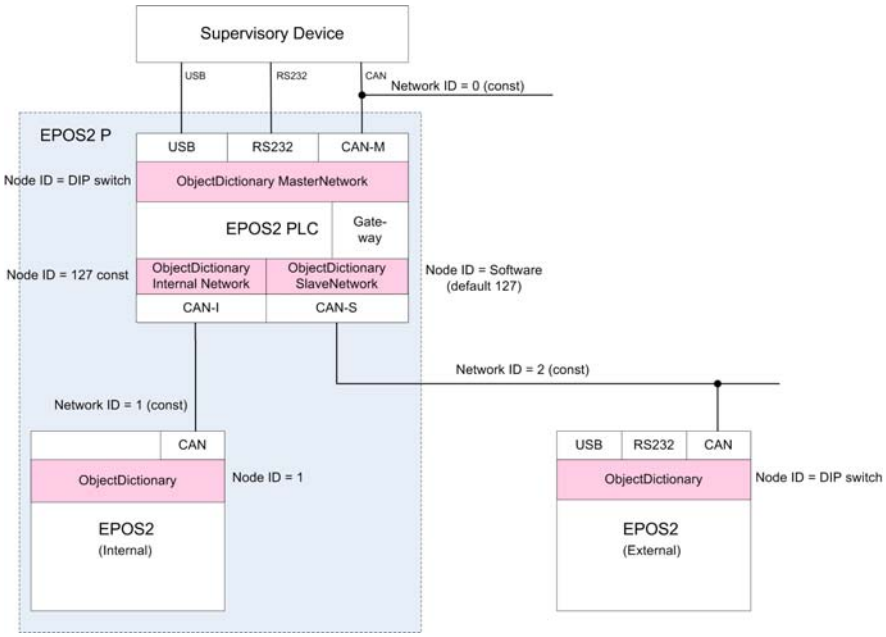

Figure 6-26 Addressing Scheme

Object Dictionary	Network ID (const)	Node ID	Note to Node ID
Master Network	0	DIP switch	configurable
Internal Network	0	127	constant
Slave Network	0	Software	configurable
EPOS 2 (internal)	1	1	constant
EPOS 2 (external)	2	(DIP switch)	configurable

Table 6-9 Addressing Scheme

6.1.2 Master Network Object Dictionary

Index	# of Sub-idx.	Name	Data Type	Access Type	Subindex of Parameter	
					configu-rable	PDO mapable
0x1000		➔Device Type	UNSIGNED32	RO		
0x1001		➔Error Register	UNSIGNED8	RO		
0x1002		➔Manufacturer Status Register	UNSIGNED32	RO		0x00
0x1003	16	➔Error History	ARRAY	RW/RO		
0x1005		➔COB-ID SYNC Message	UNSIGNED32	CONST		
0x1008		➔Manufacturer Device Name	VISIBLE_STRING	CONST		
0x1010	1	➔Store Parameters	ARRAY	RW		
0x1011	2	➔Restore Default Parameters	ARRAY	RW		
0x1014		➔COB-ID EMCY Message	UNSIGNED32	RO		
0x1016	2	➔Consumer Heartbeat Time	ARRAY	RW	0x01, 0x02	
0x1017		➔Producer Heartbeat Time	UNSIGNED16	RW	0x00	
0x1018	4	➔Identity Object	RECORD	RO		
0x1020	2	➔Verify Configuration	ARRAY	RW		
0x1200	2	➔SDO Server Parameter	RECORD	RO		
0x1400…0x1403	2	➔Receive PDO Parameter	RECORD	RW	0x01, 0x02	
0x1600…0x1603	8	➔Receive PDO Mapping	RECORD	RW	0x00…0x08	
0x1800…0x1803	5	➔Transmit PDO Parameter	RECORD	RW	0x01…0x03, 0x05	
0x1A00…0x1A03	8	➔Transmit PDO Mapping	RECORD	RW	0x00…0x08	
0x1F2A		➔Lock Current Configuration	UNSIGNED32	RW		
0x1F2B		➔Local Network ID	UNSIGNED16	CONST		
0x1F2C	2	➔Remote Network Routing List	ARRAY	RW		
0x1F2E		➔Functional Elements and Error State	UNSIGNED8	RO		
0x1F50	1	➔Program Data	ARRAY	WO		
0x1F51	1	➔Program Control	ARRAY	RW		
0x1F56	1	➔Application Software Identification	ARRAY	RO		
0x1F57	1	➔Flash Status Identification	ARRAY	RO		
0x2000		➔Node ID	UNSIGNED8	RW		
0x2001		➔CAN Bitrate	UNSIGNED16	RW		
0x2002		➔RS232 Baudrate	UNSIGNED16	RW		
0x2003	4	➔EPOS2 P Version Numbers	ARRAY	RO		
0x2004	4	➔Module Version Numbers	ARRAY	RO		

Index	# of Sub-idx.	Name	Data Type	Access Type	Subindex of Parameter	
					configu-rable	PDO mapable
0x2005		➔RS232 Frame Timeout	UNSIGNED16	RW		
0x2006		➔USB Frame Timeout	UNSIGNED16	RW		
0x2007		➔CAN SDO Frame Timeout	UNSIGNED16	RW		
0x200A		➔CAN Bitrate Display	UNSIGNED16	RW		
0x200B		➔Serial Number	UNSIGNED64	RO		
0x2F51		➔Startup Program Control	UNSIGNED16	RW	0x00	
0x2F70	3	➔Process Image	RECORD	RW		
0x5280…0x529F	1	➔Axis Data	RECORD	RW	0x01	
0xA000	16	➔Process Input Integer 8	ARRAY	RW		0x01…0x10
0xA040	16	➔Process Input Unsigned 8	ARRAY	RW		0x01…0x10
0xA0C0	16	➔Process Input Integer 16	ARRAY	RW		0x01…0x10
0xA100	16	➔Process Input Unsigned 16	ARRAY	RW		0x01…0x10
0xA1C0	16	➔Process Input Integer 32	ARRAY	RW		0x01…0x10
0xA200	16	➔Process Input Unsigned 32	ARRAY	RW		0x01…0x10
0xA480	16	➔Process Output Integer 8	ARRAY	RW		0x01…0x10
0xA4C0	16	➔Process Output Unsigned 8	ARRAY	RW		0x01…0x10
0xA540	16	➔Process Output Integer 16	ARRAY	RW		0x01…0x10
0xA580	16	➔Process Output Unsigned 16	ARRAY	RW		0x01…0x10
0xA640	16	➔Process Output Integer 32	ARRAY	RW		0x01…0x10
0xA680	16	➔Process Output Unsigned 32	ARRAY	RW		0x01…0x10

Table 6-10 Master Network Object Dictionary – Overview

6.1.3 Internal Network Object Dictionary

Index	# of Sub-idx.	Name	Data Type	Access Type	Subindex of Parameter	
					configu-rable	PDO mapable
0x1000		➔Device Type	UNSIGNED32	RO		
0x1001		➔Error Register	UNSIGNED8	RO		
0x1005		➔COB-ID SYNC Message	UNSIGNED32	RW		
0x1006		➔Communication Cycle Period	UNSIGNED32	RO		
0x1007		➔Synchronous Window Length	UNSIGNED32	RO		
0x1016	1	➔Consumer Heartbeat Time	ARRAY	RW	0x01	
0x1017		➔Producer Heartbeat Time	UNSIGNED16	RW	0x00	
0x1018	4	➔Identity Object	RECORD	RO		
0x1019		➔Synchronous Counter Overflow Value	UNSIGNED8	RW		
0x1280	3	➔SDO Client Parameter	RECORD	RO		
0x1400…0x1403	2	➔Receive PDO Parameter	RECORD	RW	0x01, 0x02	
0x1600…0x1603	8	➔Receive PDO Mapping	RECORD	RW	0x00…0x08	
0x1800…0x1803	5	➔Transmit PDO Parameter	RECORD	RW	0x01…0x03, 0x05	
0x1A00…0x1A03	8	➔Transmit PDO Mapping	RECORD	RW	0x00…0x08	
0x1F26	1	➔Expected Configuration Date	ARRAY	RW		
0x1F27	1	➔Expected Configuration Time	ARRAY	RW		
0x1F2A		➔Lock Current Configuration	UNSIGNED32	RW		
0x1F2B		➔Local Network ID	UNSIGNED16	CONST		
0x1F2C	1	➔Remote Network Routing List	ARRAY	RW		
0x1F2E		➔Functional Elements and Error State	UNSIGNED8	RO		
0x1F80		➔NMT Startup	UNSIGNED32	RW		
0x1F81	1	➔NMT Slave Assignment	ARRAY	RW		
0x1F82	2	➔Request NMT	ARRAY	RW/WO		
0x1F84	1	➔Device Type Identification	ARRAY	RW		
0x1F85	1	➔Vendor Identification	ARRAY	RW		
0x1F86	1	➔Product Code	ARRAY	RW		
0x1F87	1	➔Revision Number	ARRAY	RW		
0x1F88	1	➔Serial Number Node	ARRAY	RW		
0x1F89		➔Boot Time	UNSIGNED32	RW		
0x5000		➔Node ID	UNSIGNED8	RO		
0x5001		➔CAN Bitrate	UNSIGNED16	RO		

Index	# of Sub-idx.	Name	Data Type	Access Type	Subindex of Parameter	
					configu-rable	PDO mapable
0x5280…0x529F	4	➔Axis Data	RECORD	RO/RW		0x02, 0x04
0xA000	4	➔Process Input Integer 8	ARRAY	RW		0x01…0x04
0xA040	4	➔Process Input Unsigned 8	ARRAY	RW		0x01…0x04
0xA0C0	4	➔Process Input Integer 16	ARRAY	RW		0x01…0x04
0xA100	4	➔Process Input Unsigned 16	ARRAY	RW		0x01…0x04
0xA1C0	4	➔Process Input Integer 32	ARRAY	RW		0x01…0x04
0xA200	4	➔Process Input Unsigned 32	ARRAY	RW		0x01…0x04
0xA400	2	➔Process Input Integer 64	ARRAY	RW		0x01, 0x02
0xA440	2	➔Process Input Unsigned 64	ARRAY	RW		0x01, 0x02
0xA480	4	➔Process Output Integer 8	ARRAY	RW		0x01…0x04
0xA4C0	4	➔Process Output Unsigned 8	ARRAY	RW		0x01…0x04
0xA540	4	➔Process Output Integer 16	ARRAY	RW		0x01…0x04
0xA580	4	➔Process Output Unsigned 16	ARRAY	RW		0x01…0x04
0xA640	4	➔Process Output Integer 32	ARRAY	RW		0x01…0x04
0xA680	4	➔Process Output Unsigned 32	ARRAY	RW		0x01…0x04
0xA880	2	➔Process Output Integer 64	ARRAY	RW		0x01, 0x02
0xA8C0	2	➔Process Output Unsigned 64	ARRAY	RW		0x01, 0x02

Table 6-11 Internal Network Object Dictionary – Overview

6.1.4 Slave Network Object Dictionary

Index	# of Sub-idx.	Name	Data Type	Access Type	Subindex of Parameter	
					configu-rable	PDO mapable
0x1000		➔Device Type	UNSIGNED32	RO		
0x1001		➔Error Register	UNSIGNED8	RO		
0x1005		➔COB-ID SYNC Message	UNSIGNED32	RW		
0x1006		➔Communication Cycle Period	UNSIGNED32	RW		
0x1007		➔Synchronous Window Length	UNSIGNED32	RW		
0x1016	127	➔Consumer Heartbeat Time	ARRAY	RW	0x01	
0x1017		➔Producer Heartbeat Time	UNSIGNED16	RW	0x00	
0x1018	4	➔Identity Object	RECORD	RO		
0x1019		➔Synchronous Counter Overflow Value	UNSIGNED8	RW		
0x1280…0x12FE	3	➔SDO Client Parameter	RECORD	RW		
0x1400…0x141F	2	➔Receive PDO Parameter	RECORD	RW	0x01, 0x02	
0x1600…0x161F	8	➔Receive PDO Mapping	RECORD	RW	0x00…0x08	
0x1800…0x181f	5	➔Transmit PDO Parameter	RECORD	RW	0x01…0x03, 0x05	
0x1A00…0x1A1F	8	➔Transmit PDO Mapping	RECORD	RW	0x00…0x08	
0x1F26	127	➔Expected Configuration Date	ARRAY	RW		
0x1F27	127	➔Expected Configuration Time	ARRAY	RW		
0x1F2A		➔Lock Current Configuration	UNSIGNED32	RW		
0x1F2B		➔Local Network ID	UNSIGNED16	CONST		
0x1F2C	1	➔Remote Network Routing List	ARRAY	RW		
0x1F2E		➔Functional Elements and Error State	UNSIGNED8	RO		
0x1F80		➔NMT Startup	UNSIGNED32	RW		
0x1F81	127	➔NMT Slave Assignment	ARRAY	RW		
0x1F82	128	➔Request NMT	ARRAY	RW/WO		
0x1F84	127	➔Device Type Identification	ARRAY	RW		
0x1F85	127	➔Vendor Identification	ARRAY	RW		
0x1F86	127	➔Product Code	ARRAY	RW		
0x1F87	127	➔Revision Number	ARRAY	RW		
0x1F88	127	➔Serial Number Node	ARRAY	RW		
0x1F89		➔Boot Time	UNSIGNED32	RW		
0x5000		➔Node ID	UNSIGNED8	RW		
0x5001		➔CAN Bitrate	UNSIGNED16	RW		

Index	# of Sub-idx.	Name	Data Type	Access Type	Subindex of Parameter	
					configu-rable	PDO mapable
0x5280…0x529F	4	➔Axis Data	RECORD	RO/RW		0x02, 0x04
0xA000	64	➔Process Input Integer 8	ARRAY	RW		0x01…0x40
0xA040	64	➔Process Input Unsigned 8	ARRAY	RW		0x01…0x40
0xA0C0	64	➔Process Input Integer 16	ARRAY	RW		0x01…0x40
0xA100	64	➔Process Input Unsigned 16	ARRAY	RW		0x01…0x40
0xA1C0	64	➔Process Input Integer 32	ARRAY	RW		0x01…0x40
0xA200	64	➔Process Input Unsigned 32	ARRAY	RW		0x01…0x40
0xA400	32	➔Process Input Integer 64	ARRAY	RW		0x01…0x20
0xA440	32	➔Process Input Unsigned 64	ARRAY	RW		0x01…0x20
0xA480	64	➔Process Output Integer 8	ARRAY	RW		0x01…0x40
0xA4C0	64	➔Process Output Unsigned 8	ARRAY	RW		0x01…0x40
0xA540	64	➔Process Output Integer 16	ARRAY	RW		0x01…0x40
0xA580	64	➔Process Output Unsigned 16	ARRAY	RW		0x01…0x40
0xA640	64	➔Process Output Integer 32	ARRAY	RW		0x01…0x40
0xA680	64	➔Process Output Unsigned 32	ARRAY	RW		0x01…0x40
0xA880	32	➔Process Output Integer 64	ARRAY	RW		0x01…0x20
0xA8C0	32	➔Process Output Unsigned 64	ARRAY	RW		0x01…0x20

Table 6-12 Slave Network Object Dictionary – Overview

6.1.5 Object Data Types

Type	Description	Size [Bits]	Range
INTEGER8	Signed Integer	8	-128…+127
INTEGER16	Signed Integer	16	-32 768…+32 767
INTEGER32	Signed Integer	32	-2 147 483 648…+2 147 483 647
INTEGER64	Signed Integer	64	-9 223 372 036 854 775 808… +9 223 372 036 854 775 807
UNSIGNED8	Unsigned Integer	8	0…255
UNSIGNED16	Unsigned Integer	16	0…65 535
UNSIGNED32	Unsigned Integer	32	0…4 294 967 295
UNSIGNED64	Unsigned Integer	64	0…18 446 744 073 709 551 615
VISIBLE_STRING	Array of (8-bit) characters	n * 8	–
ARRAY	Array of other type	n * size	Range of type
RECORD	Structure of other types	–	–

Table 6-13 Object Data Types

6.1.6 Object Attributes

Attribute	Description
RW	read and write access
RO	read only access
WO	write only access
CONST	read only access, value is constant

Table 6-14 Object Attributes

6.2 Objects

Note
For easier legibility, the object dictionaries for the different network types will be abbreviated as follows:

- *Object Dictionary Master Network: MasterNet*
- *Object Dictionary Internal Network: InternalNet*
- *Object Dictionary Slave Network: SlaveNet*

6.2.1 Device Type

DESCRIPTION

Describes the device type. The lower word of the device type stands for the supported device profile number. The value 0x0195 (405) mean that the device follows the CiA 405, Interface and Device Profile or IEC 61131-3 Programmable Devices.

Name	Device Type	
Index	0x1000	
Subindex	0x00	
Type	UNSIGNED32	
Access	RO	
Default Value	0x00000195	
Value Range	–	–

6.2.2 Error Register

DESCRIPTION

The device's error register where internal errors are mapped to.

Name	Error Register	
Index	0x1001	
Subindex	0x00	
Type	UNSIGNED8	
Access	RO	
Default Value	0x00	
Value Range	–	–

Bit	Description
0	Generic
1	Current (not used)
2	Voltage (not used)
3	Temperature (not used)
4	Communication
5	Device profile-specific (not used)
6	reserved (not used)
7	manufacturer-specific

Table 6-15 Error Register – Description

6.2.3 Manufacturer Status Register

DESCRIPTION

An overview of all Error Register Flags of all connected slaves as well as the Master Errors and Warnings. Any set bit represents an error/warning.

Name	Manufacturer Status Register	
Index	0x1002	
Subindex	0x00	
Type	UNSIGNED32	
Access	RO	
Default Value	0	
Value Range	–	–

Bit	Description
0	one of the connected slaves signals a Generic Error Bit in Error Register
1	one of the connected slaves signals a Current Error Bit in Error Register
2	one of the connected slaves signals a Voltage Error Bit in Error Register
3	one of the connected slaves signals a Temperature Error Bit in Error Register
4	one of the connected slaves signals a Communication Error Bit in Error Register
5	one of the connected slaves signals a Device Profile-specific Error Bit in Error Register
6	reserved
7	one of the connected slaves signals a manufacturer-specific Error Bit in Error Register
8…15	copy of Error Register (➔page 6-44)
16	Master Generic Warning
17…19	not used
20	Master Communication Warning
21…22	not used
23	Master manufacturer-specific Warning
24…31	not used

Table 6-16 Manufacturer Status Register – Description

6.2.4 Error History

DESCRIPTION

Holds the errors that occurred on the device.

Name	Error History
Index	0x1003
Number of entries	0x10

DESCRIPTION

Contains the number of actual errors that are recorded in the array starting at subindex 1. Writing “0” (zero) will delete the error history (empties the array). Writing a value higher then zero is not permitted.

Name	Number of Errors	
Index	0x1003	
Subindex	0x00	
Type	UNSIGNED8	
Access	RW	
Default Value	0x00	
Value Range	–	–

DESCRIPTION

Every new error code is stored at subindex 1, the older ones move down the list. For the structure of an error entry ➔Table 6-17.

Name	Error History (1) to (16)	
Index	0x1003	
Subindex	0x01…0x10	
Type	UNSIGNED32	
Access	RO	
Default Value	0	
Value Range	–	–

Bit 31	Bit 30…16	Bit 15…0
warning	additional error information	error code

Table 6-17 Error History – Structure

Bit		Description
warning	0 1	Error Warning only
additional error information		Will be interpreted by «EPOS Studio».
error code		CAN Master Error Codes (➔page 5-25)

Table 6-18 Error History – Description

6.2.5 COB-ID SYNC Message

DESCRIPTION

If the generate flag (bit 30) is set, the communication object identifier will be produced in an interval defined by ➔"Communication Cycle Period" on page 6-48.

Name	COB-ID SYNC Message	
Index	0x1005	
Subindex	0x00	
Type	UNSIGNED32	
Access	MasterNet: InternalNet: SlaveNet:	CONST RW RW
Default Value	0x00000080	
Value Range	–	–

Bit 31	Bit 30	Bit 29…11	Bit 10…0
do not care	generate	0 (CAN base frame)	11-bit CAN-ID = 0x80

Table 6-19 COB-ID SYNC – Structure

6.2.6 Communication Cycle Period

DESCRIPTION

Provides the communication cycle period, which defines the SYNC interval, given in microseconds (µs). The value 0 disables the SYNC messages.

REMARKS

It is in the system or network designer's responsibility to choose a communication cycle time, which allows transmission of all cyclic data together with the communication base load. The "Communication Cycle Period should be balanced to the cycle time used by the application software.

For the correlation of ➔"Communication Cycle Period" and ➔"Synchronous Window Length" ➔Figure 6-27.

RELATED OBJECTS

- ➔"COB-ID SYNC Message" on page 6-47
- ➔"Synchronous Window Length" on page 6-49

Name	Communication Cycle Period	
Index	0x1006	
Subindex	0x00	
Type	UNSIGNED32	
Access	InternalNet: SlaveNet:	RO RW
Default Value	0x00000000	
Value Range	1000	4 294 967 295

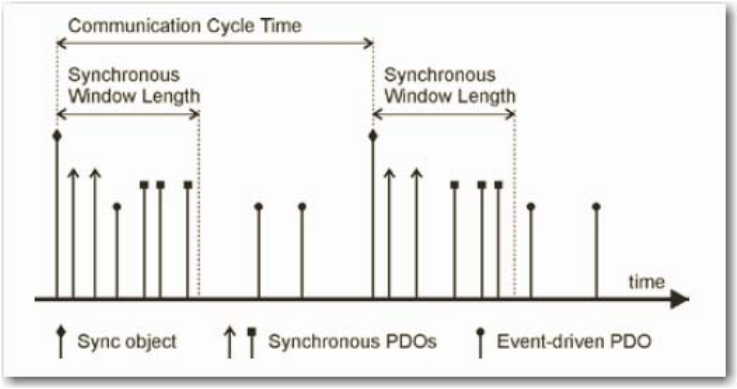

Figure 6-27 Synchronous Communication Cycle

6.2.7 Synchronous Window Length

DESCRIPTION

Indicates the configured time window length for the synchronous PDOs, given in microseconds (µs).

REMARKS

For the correlation of ➔“Synchronous Window Length” and ➔“Communication Cycle Period” ➔Figure 6-27.

RELATED OBJECTS

- ➔“COB-ID SYNC Message” on page 6-47
- ➔“Communication Cycle Period” on page 6-48

Name	Synchronous Window Length	
Index	0x1007	
Subindex	0x00	
Type	UNSIGNED32	
Access	InternalNet: RO SlaveNet: RW	
Default Value	0x00000000	
Value Range	500	4 294 967 295

6.2.8 Manufacturer Device Name

DESCRIPTION

The name of the device is “EPOS2 P”.

Name	Manufacturer Device Name	
Index	0x1008	
Subindex	0x00	
Type	VISIBLE_STRING	
Access	CONST	
Default Value	“EPOS2 P”	
Value Range	–	–

6.2.9 Store Parameters

DESCRIPTION

All parameters of the device will be stored in a non-volatile memory, if the signature “save” is written to this object.

Name	Store Parameters
Index	0x1010
Number of entries	0x01

Name	Save all Parameters	
Index	0x1010	
Subindex	0x01	
Type	UNSIGNED32	
Access	RW	
Default Value	–	
Value Range	–	–

Byte	MSB			LSB
Character	‘e’	‘v’	‘a’	‘s’
Hex value	0x65	0x76	0x61	0x73

Table 6-20 Storage Write Access Signature

6.2.10 Restore Default Parameters

DESCRIPTION

All parameters of the device will be restored with default values, if the signature “load” is written to this object.

Name	Restore Default Parameters
Index	0x1011
Number of entries	0x02

Name	Restore all Default Parameters	
Index	0x1011	
Subindex	0x01	
Type	UNSIGNED32	
Access	RW	
Default Value	–	
Value Range	–	–

Byte	MSB			LSB
Character	‘d’	‘a’	‘o’	‘l’
Hex value	0x64	0x61	0x6F	0x6C

Table 6-21 Restore Default Write Access Signature

DESCRIPTION

The application program will be deleted by writing the signature “eras” to this object.

Name	Delete Application Program from Persistence Memory	
Index	0x1011	
Subindex	0x02	
Type	UNSIGNED32	
Access	RW	
Default Value	–	
Value Range	–	–

Byte	**MSB**			**LSB**
Character	‘s’	‘a’	‘r’	‘e’
Hex value	0x73	0x61	0x72	0x65

Table 6-22 Erase Application Program Write Access Signature

6.2.11 COB-ID EMCY Message

DESCRIPTION

Communication Object Identifier for the emergency object.

Name	COB-ID EMCY Message	
Index	0x1014	
Subindex	0x00	
Type	UNSIGNED32	
Access	RO	
Default Value	0x00000080 + Node ID	
Value Range	–	–

6.2.12 Consumer Heartbeat Time

DESCRIPTION

Provides the expected heartbeat cycle times, given in milliseconds (ms). They are higher than the corresponding producer heartbeat times configured on the CAN device producing this heartbeat.

If the heartbeat time or the node ID are “0” (zero), the corresponding heartbeat consumer is inactive.

Name	Consumer Heartbeat Time	
Index	0x1016	
Number of entries	MasterNet: InternalNet: SlaveNet:	2 1 127

Name	Consumer Heartbeat Time 1 to Number of Entries	
Index	0x1016	
Subindex	1…number of entries	
Type	UNSIGNED32	
Access	RW	
Default Value	0x00000000	
Value Range	–	–

Bit 31…24	Bit 23…16	Bit 15…0
reserrved	node-Id	heartbeat time

Table 6-23 Consumer Heartbeat Time – Structure

Bits		Description
node-Id	1…127	producer node-Id
heartbeat time		maximal expected heartbeat time [ms]

Table 6-24 Consumer Heartbeat Time – Description

6.2.13 Producer Heartbeat Time

DESCRIPTION

Indicates the configured cycle time of the heartbeat, given in multiples of 1 ms. The value 0 disables the heartbeat producer.

Name	Producer Heartbeat Time	
Index	0x1017	
Subindex	0x00	
Type	UNSIGNED16	
Access	RW	
Default Value	InternalNet: 2000 SlaveNet: 0	
Value Range	–	–

6.2.14 Identity Object

DESCRIPTION

The CANopen vendor identification of “maxon motor ag” defined by CiA is 0x000000FB.

Name	Identity Object	
Index	0x1018	
Number of entries	0x04	

Name	Vendor ID	
Index	0x1018	
Subindex	0x01	
Type	UNSIGNED32	
Access	RO	
Default Value	0x000000FB	
Value Range	–	–

DESCRIPTION

Contains the hardware-specific type of the CANopen device.

RELATED OBJECTS

- ➔“EPOS2 P Version Numbers” on page 6-85

Name	Product Code	
Index	0x1018	
Subindex	0x02	
Type	UNSIGNED32	
Access	RO	
Default Value	–	
Value Range	–	–

DESCRIPTION

Contains the software-specific revision number.

RELATED OBJECTS

- ➔“EPOS2 P Version Numbers” on page 6-85

Name	Revision Number	
Index	0x1018	
Subindex	0x03	
Type	UNSIGNED32	
Access	RO	
Default Value	–	
Value Range	–	–

DESCRIPTION

Contains a (unique) serial number.

Name	Serial Number	
Index	0x1018	
Subindex	0x04	
Type	UNSIGNED32	
Access	RO	
Default Value	–	
Value Range	–	–

6.2.15 Synchronous Counter Overflow Value

DESCRIPTION

Indicates the highest value the counter can support. The synchronous counter is mapped into the SYNC message. The value of the counter will be incremented by one with any request of the service SYNC write. When the highest value is reached, the counter will recommence to count from “1”.

RELATED OBJECTS

- ➔“COB-ID SYNC Message” on page 6-47
- ➔“Communication Cycle Period” on page 6-48

Name	Synchronous Counter Overflow Value
Index	0x1019
Subindex	0x00
Type	UNSIGNED8
Access	RW
Default Value	0x00
Value Range	➔Table 6-25

Value	Description
0	The SYNC message will be transmitted as a CAN message of data length 0.
1	reserved
2…240	The SYNC message will be transmitted as a CAN message of data length 1. The data byte contains the counter.
241…255	reserved

Table 6-25 Synchronous Counter – Description

6.2.16 Verify Configuration

DESCRIPTION

Indicates the downloaded configuration date/time and may be used by a network configuration tool or the CANopen manager to verify the EPOS2 P's configuration.

The configuration tool stores date/time in the object as well, for example, in the DCF file. Then, the configuration tool stores the EPOS2 P's configuration parameters with the object ➔"Store Parameters" on page 6-50. If any other command changes the configuration parameters, the EPOS2 P will reset the object "Verify Configuration" to "0" (zero).

A CANopen master can verify that the device configuration has not changed by checking configuration date and time for the correct value. The column "Subindex of configurable Parameters" in the object dictionary overview (➔as from page 6-35) indicates objects which will be checked by the firmware. Any change to them will reset the configuration time and date.

Name	Verify Configuration
Index	0x1020
Number of entries	0x02

Description

Configuration date must contain the number of days since January 1, 1984.

Name	Configuration Date	
Index	0x1020	
Subindex	0x01	
Type	UNSIGNED32	
Access	RW	
Default Value	0x00000000	
Value Range	–	–

Description

Configuration time must contain the number of milliseconds [ms] after midnight.

Name	Configuration Time	
Index	0x1020	
Subindex	0x02	
Type	UNSIGNED32	
Access	RW	
Default Value	0x00000000	
Value Range	–	–

6.2.17 SDO Server Parameter

DESCRIPTION

Contains the parameters for the SDO clients for which the EPOS2 P is the SDO server.

Name	SDO Server Parameter
Index	0x1200
Number of entries	0x02

DESCRIPTION

Specifies the COB-ID for the communication from the EPOS2 P as client to the associated SDO server.

Name	COB-ID Client to Server (SDO server rx)	
Index	0x1200	
Subindex	0x01	
Type	UNSIGNED32	
Access	RO	
Default Value	0x00000600 + Node ID	
Value Range	–	–

DESCRIPTION

Specifies the COB-ID or the communication from the associated SDO server to the EPOS2 P.

Name	COB-ID Server to Client (SDO server tx)	
Index	0x1200	
Subindex	0x02	
Type	UNSIGNED32	
Access	RO	
Default Value	0x00000580 + Node ID	
Value Range	–	–

6.2.18 SDO Client Parameter

DESCRIPTION

Contains the parameters for the SDO servers for which the EPOS2 P is the SDO client.

Name	InternalNet SlaveNet	SDO Client Parameter SDO 1 to 127 Client Parameter
Index	InternalNet: SlaveNet:	0x1280 0x1280…0x12FE
Number of entries	0x03	

DESCRIPTION

Specifies the COB-ID for the communication from the EPOS2 P as client to the associated SDO server.

Name	InternalNet: SlaveNet:	COB-ID Client to Server (SDO tx) COB-ID Client to Server (SDO 1…127 tx)
Index	InternalNet: SlaveNet:	0x1280 0x1280…0x12FE
Subindex	0x01	
Type	UNSIGNED32	
Access	InternalNet: SlaveNet:	CONST RW
Default Value	0x600 + server Node ID	
Value Range	–	–

DESCRIPTION

Specifies the COB-ID or the communication from the associated SDO server to the EPOS2 P.

Name	InternalNet: SlaveNet:	COB-ID Server to Client (SDO rx) COB-ID Server to Client (SDO 1…127 rx)
Index	InternalNet: SlaveNet:	0x1280 0x1280…0x12FE
Subindex	0x02	
Type	UNSIGNED32	
Access	InternalNet: SlaveNet:	CONST RW
Default Value	0x580 + server Node ID	
Value Range	–	–

DESCRIPTION

Specifies the node-Id of the associated sever.

REMARKS

Not changeable

Name	InternalNet: SlaveNet:	Server Node ID (SDO) Server Node ID (SDO 1…127)
Index	InternalNet: SlaveNet:	0x1280 0x1280…0x12FE
Subindex	0x03	
Type	UNSIGNED8	
Access	InternalNet: SlaveNet:	RO RW
Default Value	InternalNet: SlaveNet:	0x1280: 0x01 0x1281: 0x01, 0x1282: 0x02, 0x1282: 0x03, …, 0x12FE: 0xFE
Value Range	–	–

6.2.19 Receive PDO Parameter

DESCRIPTION

Contains the communication parameters for the PDOs the CANopen device is able to receive.

RELATED OBJECTS

- ➔"Receive PDO Mapping" on page 6-61

Name	MasterNet: InternalNet: SlaveNet:	Receive PDO 1 to 4 Parameter Receive PDO 1 to 4 Parameter Receive PDO 1 to 32 Parameter
Index	MasterNet: InternalNet: SlaveNet:	0x1400…0x1403 0x1400…0x1403 0x1400…0x141F
Number of entries	0x02	

DESCRIPTION

Communication Object Identifier of receive process data object.

REMARKS

Changes in mapping are only possible in Nmt state Pre-Operational

Name	MasterNet: InternalNet: SlaveNet:	COB-ID used by RxPDO 1 to 4 COB-ID used by RxPDO 1 to 4 COB-ID used by RxPDO 1 to 32
Index	MasterNet: InternalNet: SlaveNet:	0x1400…0x1403 0x1400…0x1403 0x1400…0x141F
Subindex	0x01	
Type	UNSIGNED32	
Access	RW	
Default Value	MasterNet: InternalNet: SlaveNet:	0xC0000000 COB-ID used by RxPDO 1: 0x00000181, others 0xC0000000 0xC0000000
Value Range	➔Table 6-27	–

Bit 31	Bit 30	Bit 29…11	Bit 10…0
valid	RTR	0 (CAN base frame)	11-bit CAN-ID

Table 6-26 COB-ID RxPDO – Structure

Bit		Description	
valid	0b 1b	PDO exists / is valid PDO does not exist / is not valid	
RTR	0b 1b	RTR allowed on this PDO no RTR allowed on this PDO	
11-bit CAN-ID		11-bit CAN-ID of the CAN base frame	
		Value range:	0x181…0x57F; 0x000 (if valid = 1)

Table 6-27 COB-ID RxPDO – Description

DESCRIPTION

The transmission type describes how PDO communication works.

REMARKS

Changes in mapping are only possible in Nmt state Pre-Operational.

Name	MasterNet: InternalNet: SlaveNet:	Transmission Type RxPDO 1 to 4 Transmission Type RxPDO 1 to 4 Transmission Type RxPDO 1 to 32
Index	MasterNet: InternalNet: SlaveNet:	0x1400…0x1403 0x1400…0x1403 0x1400…0x141F
Subindex	0x02	
Type	UNSIGNED8	
Access	RW	
Default Value	255	
Value Range	1: synchronous 255: asynchronous	–

6.2.20 Receive PDO Mapping

DESCRIPTION

Contains the number of valid object entries within the mapping record.

REMARKS

Changes in mapping are only possible in Nmt state Pre-Operational.

Name	MasterNet: InternalNet: SlaveNet:	Receive PDO 1 to 4 Mapping Receive PDO 1 to 4 Mapping Receive PDO 1 to 32 Mapping
Index	MasterNet: InternalNet: SlaveNet:	0x1600…0x1603 0x1600…0x1603 0x1600…0x161F
Number of entries	8	

Name	MasterNet: InternalNet: SlaveNet:	Number of Mapped Application Objects in RxPDO 1 to 4 Number of Mapped Application Objects in RxPDO 1 to 4 Number of Mapped Application Objects in RxPDO 1 to 32
Index	MasterNet: InternalNet: SlaveNet:	0x1600…0x1603 0x1600…0x1603 0x1600…0x161F
Subindex	0x00	
Type	UNSIGNED8	
Access	RW	
Default Value	MasterNet: InternalNet: SlaveNet:	0x00 Number of Mapped Application Objects in RxPDO 1: 0x02 0x00
Value Range	0: PDO is disabled 1…8: 1 to 8 objects are mapped	–

DESCRIPTION

The objects in the next table are supported to map.

REMARKS

Changes in mapping are only possible in Nmt state Pre-Operational.

To change a mapped object it is necessary to disable PDO by writing 0 to the number of mapped PDO objects.The maximal length of a process data object (PDO) is 64 bit.

Name	MasterNet: InternalNet: SlaveNet:	1st…8th mapped Object in RxPDO 1 to 4 1st…8th mapped Object in RxPDO 1 to 4 1st…8th mapped Object in RxPDO 1 to 32
Index	MasterNet: InternalNet: SlaveNet:	0x1600…0x1603 0x1600…0x1603 0x1600…0x161F
Subindex	0x01…0x08	
Type	UNSIGNED32	
Access	RW	
Default Value	MasterNet: InternalNet: SlaveNet:	0x00000000 1st mapped Object in RxPDO 1: 0x52800210 2nd mapped Object in RxPDO 1: 0x52800408 0x00000000
Value Range	➔Table 6-28	–

Entries marked in *grey/italic* may also be mapped, but it might not really make sense to do so.

Bit 31…16 Index	Bit 15…8 Subindex			Bit 7…0 Length [bit]	Description
	MasterNet	InternalNet	SlaveNet		
0x1002	*0x00*	*–*	*–*	*0x20*	*Manufacturer Status Register*
0x5280…0x529F	–	0x02	0x02	0x10	Axis 0 to 31 Statusword
0x5280…0x529F	–	0x04	0x04	0x08	Axis 0 to 31 Modes of Operation
0xA000	0x01…0x10	0x01…0x04	0x01…0x40	0x08	Process Image Input Integer 8
0xA040	0x01…0x10	0x01…0x04	0x01…0x40	0x08	Process Image Input Unsigned 8
0xA0C0	0x01…0x10	0x01…0x04	0x01…0x40	0x10	Process Image Input Integer 16
0xA100	0x01…0x10	0x01…0x04	0x01…0x40	0x10	Process Image Input Unsigned 16
0xA1C0	0x01…0x10	0x01…0x04	0x01…0x40	0x20	Process Image Input Integer 32
0xA200	0x01…0x10	0x01…0x04	0x01…0x40	0x20	Process Image Input Unsigned 32
0xA400	–	0x01…0x02	0x01…0x20	0x40	Process Image Input Integer 64
0xA440	–	0x01…0x02	0x01…0x20	0x40	Process Image Input Unsigned 64
0xA480	*0x01…0x10*	*0x01…0x04*	*0x01…0x40*	*0x08*	*Process Image Output Integer 8*
0xA4C0	*0x01…0x10*	*0x01…0x04*	*0x01…0x40*	*0x08*	*Process Image Output Unsigned 8*
0xA540	*0x01…0x10*	*0x01…0x04*	*0x01…0x40*	*0x10*	*Process Image Output Integer 16*
0xA580	*0x01…0x10*	*0x01…0x04*	*0x01…0x40*	*0x10*	*Process Image Output Unsigned 16*
0xA640	*0x01…0x10*	*0x01…0x04*	*0x01…0x40*	*0x20*	*Process Image Output Integer 32*
0xA680	*0x01…0x10*	*0x01…0x04*	*0x01…0x40*	*0x20*	*Process Image Output Unsigned 32*

Bit 31…16 Index	Bit 15…8 Subindex			Bit 7…0 Length [bit]	Description
	MasterNet	InternalNet	SlaveNet		
0xA880	–	*0x01…0x02*	*0x01…0x20*	*0x40*	*Process Image Output Integer 64*
0xA8C0	–	*0x01…0x02*	*0x01…0x20*	*0x40*	*Process Image Output Unsigned 64*

Table 6-28 Receive PDO Mapping Objects – Value Definition

6.2.21 Transmit PDO Parameter

DESCRIPTION

Contains the communication parameters for the PDOs the CANopen device is able to transmit.

RELATED OBJECTS

➔"Receive PDO Mapping" on page 6-61

Name	MasterNet: InternalNet: SlaveNet:	Transmit PDO 1 to 4 Parameter Transmit PDO 1 to 4 Parameter Transmit PDO 1 to 32 Parameter
Index	MasterNet: InternalNet: SlaveNet:	0x1800…0x1803 0x1800…0x1803 0x1800…0x181F
Number of entries	0x05	

DESCRIPTION

Communication Object Identifier of transmit process data object.

REMARKS

Changes in mapping are only possible in Nmt state Pre-Operational.

Name	MasterNet: InternalNet: SlaveNet:	COB-ID used by TxPDO 1 to 4 COB-ID used by TxPDO 1 to 4 COB-ID used by TxPDO 1 to 32
Index	MasterNet: InternalNet: SlaveNet:	0x1800…0x1803 0x1800…0x1803 0x1800…0x181F
Subindex	0x01	
Type	UNSIGNED32	
Access	RW	
Default Value	0xC0000000	
Value Range	➔Table 6-30	–

Bit 31	Bit 30	Bit 29…11	Bit 10…0
valid	RTR	0 (CAN base frame)	11-bit CAN-ID

Table 6-29 COB-ID TxPDO – Structure

Bit		Description	
valid	0b 1b	PDO exists / is valid PDO does not exist / is not valid	
RTR	0b 1b	RTR allowed on this PDO no RTR allowed on this PDO	
11-bit CAN-ID		11-bit CAN-ID of the CAN base frame	
		Value range:	0x181…0x57F; 0x000 (if valid = 1)

Table 6-30 COB-ID TxPDO – Description

DESCRIPTION

The transmission type describes how PDO communication works.

REMARKS

The transmission type 253 means that the PDO is only transmitted on remote transmission request (RTR). If transmission type 255 is selected the PDO is transmitted if the data's change its values. The inhibit time defines a minimum interval therefore.

Changes in mapping are only possible in Nmt state Pre-Operational.

Name	MasterNet: InternalNet: SlaveNet:	Transmission Type TxPDO 1 to 4 Transmission Type TxPDO 1 to 4 Transmission Type TxPDO 1 to 32
Index	MasterNet: InternalNet: SlaveNet:	0x1800…0x1803 0x1800…0x1803 0x1800…0x181F
Subindex	0x02	
Type	UNSIGNED8	
Access	RW	
Default Value	255	
Value Range	➔Table 6-31	–

Value	Description
1	synchronous
253	asynchronous on RTR only
254	asynchronous on event
255	asynchronous on change

Table 6-31 Transmit PDO – Transmission Types

DESCRIPTION

Represents the minimum interval for event-triggered PDO transmission. The value is defined as multiple of 100 µs.

REMARKS

Event-triggered PDOs can generate a huge CAN bus load and device load also, especially if the inhibit time of different PDOs are set to a small value.

Name	MasterNet: InternalNet: SlaveNet:	Inhibit Time TxPDO 1 to 4 Inhibit Time TxPDO 1 to 4 Inhibit Time TxPDO 1 to 32
Index	MasterNet: InternalNet: SlaveNet:	0x1800…0x1803 0x1800…0x1803 0x1800…0x181F
Subindex	0x03	
Type	UNSIGNED16	
Access	RW	
Default Value	10	
Value Range	–	–

DESCRIPTION

Represents the maximum interval for event-triggered PDO transmission if the Transmission Type is set to “255”. The value is defined as multiple of 1 ms. The value of 0 disables the Event Timer.

Name	MasterNet: InternalNet: SlaveNet:	Event Timer TxPDO 1 to 4 Event Timer TxPDO 1 to 4 Event Timer TxPDO 1 to 32
Index	MasterNet: InternalNet: SlaveNet:	0x1800…0x1803 0x1800…0x1803 0x1800…0x181F
Subindex	0x05	
Type	UNSIGNED16	
Access	RW	
Default Value	0	
Value Range	–	–

6.2.22 Transmit PDO Mapping

DESCRIPTION

Contains the number of valid object entries within the mapping record.

REMARKS

Changes in mapping are only possible in Nmt state Pre-Operational.

Name	MasterNet: InternalNet: SlaveNet:	Transmit PDO 1 to 4 Mapping Transmit PDO 1 to 4 Mapping Transmit PDO 1 to 32 Mapping
Index	MasterNet: InternalNet: SlaveNet:	0x1A00…0x1A03 0x1A00…0x1A03 0x1A00…0x1A1F
Number of entries	8	

Name	MasterNet: InternalNet: SlaveNet:	Number of Mapped Application Objects in TxPDO 1 to 4 Number of Mapped Application Objects in TxPDO 1 to 4 Number of Mapped Application Objects in TxPDO 1 to 32
Index	MasterNet: InternalNet: SlaveNet:	0x1A00…0x1A03 0x1A00…0x1A03 0x1A00…0x1A1F
Subindex	0x00	
Type	UNSIGNED8	
Access	RW	
Default Value	0x00	
Value Range	0: PDO is disabled 1…8: 1 to 8 objects are mapped	–

DESCRIPTION

The objects in next table are supported to map.

REMARKS

Changes in mapping are only possible in Nmt state Pre-Operational.

To change a mapped object it is necessary to disable PDO by writing 0 to the number of mapped PDO objects.The maximal length of a process data object (PDO) is 64 bit.

Name	MasterNet: InternalNet: SlaveNet:	1st to 8th Mapped Object in TxPDO 1 to 4 1st to 8th Mapped Object in TxPDO 1 to 4 1st to 8th Mapped Object in TxPDO 1 to 32
Index	MasterNet: InternalNet: SlaveNet:	0x1A00…0x1A03 0x1A00…0x1A03 0x1A00…0x1A1F
Subindex	0x01…0x08	
Type	UNSIGNED32	
Access	RW	
Default Value	0x00000000	
Value Range	➔Table 6-32	–

Entries marked in *grey/italic* may also be mapped, but it might not really make sense to do so.

Bit 31…16 Index	Bit 15…8 Subindex			Bit 7…0 Length [bit]	Description
	MasterNet	InternalNet	SlaveNet		
0x1002	0x00	–	–	0x20	Manufacturer Status Register
0x5280…0x529F	*–*	*0x02*	*0x02*	*0x10*	*Axis 0 to 31 Statusword*
0x5280…0x529F	*–*	*0x04*	*0x04*	*0x08*	*Axis 0 to 31 Modes of Operation*
0xA000	*0x01…0x10*	*0x01…0x04*	*0x01…0x40*	*0x08*	*Process Image Input Integer 8*
0xA040	*0x01…0x10*	*0x01…0x04*	*0x01…0x40*	*0x08*	*Process Image Input Unsigned 8*
0xA0C0	*0x01…0x10*	*0x01…0x04*	*0x01…0x40*	*0x10*	*Process Image Input Integer 16*
0xA100	*0x01…0x10*	*0x01…0x04*	*0x01…0x40*	*0x10*	*Process Image Input Unsigned 16*
0xA1C0	*0x01…0x10*	*0x01…0x04*	*0x01…0x40*	*0x20*	*Process Image Input Integer 32*
0xA200	*0x01…0x10*	*0x01…0x04*	*0x01…0x40*	*0x20*	*Process Image Input Unsigned 32*
0xA400	*–*	*0x01…0x02*	*0x01…0x20*	*0x40*	*Process Image Input Integer 64*
0xA440	*–*	*0x01…0x02*	*0x01…0x20*	*0x40*	*Process Image Input Unsigned 64*
0xA480	0x01…0x10	0x01…0x04	0x01…0x40	0x08	Process Image Output Integer 8
0xA4C0	0x01…0x10	0x01…0x04	0x01…0x40	0x08	Process Image Output Unsigned 8
0xA540	0x01…0x10	0x01…0x04	0x01…0x40	0x10	Process Image Output Integer 16
0xA580	0x01…0x10	0x01…0x04	0x01…0x40	0x10	Process Image Output Unsigned 16
0xA640	0x01…0x10	0x01…0x04	0x01…0x40	0x20	Process Image Output Integer 32
0xA680	0x01…0x10	0x01…0x04	0x01…0x40	0x20	Process Image Output Unsigned 32
0xA880	–	0x01…0x02	0x01…0x20	0x40	Process Image Output Integer 64
0xA8C0	–	0x01…0x02	0x01…0x20	0x40	Process Image Output Unsigned 64

Table 6-32 Transmit PDO Mapping Objects – Value Definition

6.2.23 Expected Configuration Date

DESCRIPTION

Contains the value of the object ➔"Verify Configuration" on page 6-56 (0x1020sub1) of the corresponding slave. This will allow verification of the connected slave's correct configuration upon bootup. A value of "0" (zero) will disable the check.

Name	Expected Configuration Date	
Index	0x1F26	
Number of entries	InternalNet: SlaveNet:	1 127

Name	Expected Configuration Date Node 1 to 128	
Index	0x1F26	
Subindex	1…number of entries	
Type	UNSIGNED32	
Access	RW	
Default Value	0x00000000	
Value Range	–	–

6.2.24 Expected Configuration Time

DESCRIPTION

Contains the value of the object ➔"Verify Configuration" on page 6-56 (0x1020sub2) of the corresponding slave. This will allow verification of the connected slave's correct configuration upon bootup. A value of "0" (zero) will disable the check.

Name	Expected Configuration Time	
Index	0x1F27	
Number of entries	InternalNet: SlaveNet:	1 127

Name	Expected Configuration Time Node 1 to 128	
Index	0x1F27	
Subindex	1…number of entries	
Type	UNSIGNED32	
Access	RW	
Default Value	0	
Value Range	–	–

6.2.25 Lock Current Configuration

DESCRIPTION

Locks/unlocks the current configuration. If the CANopen interface is locked, the communication parameter objects in the range of 0x1F2B to 0x1F4F must not be writeable.

REMARKS

This object has no effect since the communication parameter objects are locked anyway.

Name	Lock Current Configuration	
Index	0x1F2A	
Subindex	0x00	
Type	UNSIGNED32	
Access	RW	
Default Value	00x65657266 (“FREE”)	
Value Range	➔Table 6-34	–

Bit 31…24	Bit 23…16	Bit 15…8	Bit 7…0
Character 4	Character 3	Character 2	Character 1

Table 6-33 Lock Current Configuration – Structure

Character 1	Character 2	Character 3	Character 4	Description
L (0x6C)	O (0x6F)	C (0x63)	K (0x6B)	Some entries are not writeable
F (0x66)	R (0x72)	E (0x65)	E (0x65)	All read/write entries are writeable

Table 6-34 Lock Current Configuration – Description

6.2.26 Local Network ID

DESCRIPTION

Sets the ID of the local network.

Name	Local Network ID	
Index	0x1F2B	
Subindex	0x00	
Type	UNSIGNED16	
Access	CONST	
Default Value	MasterNet: 32 InternalNet: 1 SlaveNet: 2	
Value Range	–	–

6.2.27 Remote Network Routing List

DESCRIPTION

Indicates the remote networks that are accessible via the local CANopen interface. The supported networks are directly or indirectly linked to the remote interfaces of this router.

REMARKS

If the value of the network-ID field is 0x0000 this shall indicate that no network-ID has been assigned.

The router node-ID field shall indicate the configured node-ID of the router in the same CANopen network. The value of 0x00 shall indicate that no node-ID has been assigned.

The port number shall be unique for each CANopen interface (1 to 32). The value of 0xFF shall indicate an invalid port number.

Name	Remote Network Routing List	
Index	0x1F2C	
Number of entries	MasterNet: InternalNet: SlaveNet:	2 1 1

Name	Routing 1	
Index	0x1F2C	
Subindex	0x01	
Type	UNSIGNED32	
Access	RW	
Default Value	MasterNet: InternalNet: SlaveNet	0x00010001 0x000100FF 0x000100FF
Value Range	–	–

Name	Routing 2	
Index	0x1F2C	
Subindex	0x02	
Type	UNSIGNED32	
Access	RW	
Default Value	0x00020002	
Value Range	–	–

Bit 31…16	Bit15…8	Bit 7…0
Network ID	Router Node ID	Port number

Table 6-35 Remote Network Routing List – Structure

6.2.28 Functional Elements and Error State

DESCRIPTION

Indicates the functional elements (FE) provided by a device and the error state.

REMARKS

The error state functionality is not supported. Therefore, bit 4 is always “0”.

Name	Functional Elements and Error State	
Index	0x1F2C	
Subindex	0x02	
Type	UNSIGNED32	
Access	RW	
Default Value	0x00020002	
Value Range	–	–

Bit	Value Definition
7	0: remote SDO FE not implemented 1: remote SDO FE implemented
6	0: remote EMCY FE not implemented 1: remote EMCY FE implemented
5	0: PDO forwarding not supported 1: PDO forwarding supported
4	0: No severe error detected 1: Severe error has occurred
3…0	reserved (0)

Table 6-36 Functional Elements and Error State – Structure

6.2.29 Program Data

DESCRIPTION

Used for program download.

Name	Program Data
Index	0x1F50
Number of entries	0x01

DESCRIPTION

Used to send an IEC-61131 program to the OpenPCS runtime system.

REMARKS

Access is not permitted as long as communication with OpenPCS programming system is established. Download is only permitted if the program had been cleared in advance.

Name	Program Number 1	
Index	0x1F50	
Subindex	0x01	
Type	DOMAIN	
Access	WO	
Default Value	–	
Value Range	–	–

6.2.30 Program Control

DESCRIPTION

Used to control a downloaded program.

Name	Program Control
Index	0x1F51
Number of entries	0x01

DESCRIPTION

Used to send an IEC-61131 program to the OpenPCS runtime system.

REMARKS

Write access is not permitted when a communication with OpenPCS programming system is established or no valid program is available on the device.

- A start or clear command is only permitted if the program status is stopped.
- A stop command is only permitted if the program status is started.

Name	Program Number 1	
Index	0x1F51	
Subindex	0x01	
Type	UNSIGNED8	
Access	RW	
Default Value	0x00	
Value Range	➔Table 6-37	–

Access	Value	Description
Write	0x00	Stop Program
	0x01	Start Program (Cold Start)
	0x02	Reset Program (Reset Device)
	0x03	Clear Program (Erase)
	0x81	Cold Start (manufacturer-specific)
	0x82	Warm Start (manufacturer-specific)
	0x83	Hot Start (manufacturer-specific)
Read	0x00	Program stopped
	0x01	Program started
	0x02	Program stopped
	0x03	No program available

Table 6-37 Program Control Code

6.2.31 Application Software Identification

DESCRIPTION

Used to verify the version of the program and to check if a re-download is necessary.

Name	Application Software Identification
Index	0x1F56
Number of entries	0x01

DESCRIPTION

Used to verify program number 1. Contains date and time of the program build. If no valid program is available, the object produces the value “0” (zero).

Name	Program Number 1	
Index	0x1F56	
Subindex	0x01	
Type	UNSIGNED32	
Access	RO	
Default Value	–	
Value Range	–	–

6.2.32 Flash Status Identification

DESCRIPTION

On start of download the Flash status indication shall be set to 0xXXXX00001. On successful completion of the Flashing process the Flash status shall be set to 0xXXXX0000. If an error occurs, bit 0 shall be set to 0 and bit 1 to 7 shall contain the error reason.

Name	Flash Status Identification
Index	0x1F57
Number of entries	0x01

DESCRIPTION

Used to check the flashing process.

Name	Program Number 1	
Index	0x1F57	
Subindex	0x01	
Type	UNSIGNED32	
Access	RO	
Default Value	0	
Value Range	➔Table 6-38	–

Bit	Value	Description
0	0	Status OK
	1	Download in progress
1…7	0	No error
	1	No valid program
	2	Data format unknown
	3	Data format error or data CRC error
	4	Flash not cleared before write
	5	Flash write error
	6	General address error
	7	Flash secured (write access not permitted)
	64	General internal error
	65…127	reserved (manufacturer-specific)
8…15	–	reserved (always “0”)
16	0	Initialization of program download inactive
	1	Initialization of program download active
17…31	–	manufacturer-specific

Table 6-38 Flash Status Identification – Structure

6.2.33 NMT Startup

DESCRIPTION

Configures the startup behavior of the device that performs the Nmt.

Name	Nmt Startup	
Index	0x1F80	
Subindex	0x00	
Type	UNSIGNED32	
Access	RW	
Default Value	0x00000003	
Value Range	–	–

Bit 31…8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
reserved 0x0000 00	reserved 0	Stop all nodes	Flying master	Reset all nodes	Start node	Self starting	Start all nodes	Nmt master

Table 6-39 Nmt Startup – Structure

Bit	Value	Description
Nmt master	0 1	not supported EPOS2 P is in Nmt master mode
Start all nodes	0 1	Nmt service “start remote node” for each Node-ID Nmt service “start remote node” with Node-ID = 0
Self starting	0 1	EPOS2 P switches into Nmt state Operational in bootup procedure EPOS2 P does not switch itself to Nmt state Operational automatically
Start node	0 1	EPOS2 P starts the slaves EPOS2 P does not start the slaves and the application may start them
Reset all nodes	0	Node reset on error of mandatory slave not implemented
Flying master	0	Nmt flying master not implemented
Stop all nodes	0	Node stop on error of mandatory slave not implemented

Table 6-40 Nmt Startup – Bit Definition

6.2.34 NMT Slave Assignment

DESCRIPTION

Part of the network list. It assigns slaves to the Nmt Master.

Name	NMT Slave Assignment	
Index	0x1F81	
Number of entries	InternalNet: SlaveNet:	1 127

Name	NMT Slave Assignment Node 1 to Number of Entries	
Index	0x1F81	
Subindex	1…number of entries	
Type	UNSIGNED32	
Access	RW	
Default Value	InternalNet: SlaveNet:	0x0000000D 0x00000000
Value Range	–	–

Bit 31…16	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
reserved 0x0000 00	reserved 0	Software update	Software version	Reset communi-cation	Manda-tory	Nmt boot slave	Restore	Nmt slave

Table 6-41 Nmt Startup – Structure

Bit		Description
Nmt slave	0 1	CANopen device in Nmt master mode or not available in network CANopen device in Nmt slave mode and available in network
Restore	0	CANopen device may be used without prior resetting
Nmt boot slave	0 1	Configuration and Nmt service ‘Start remote node’ not allowed Configuration and Nmt service ‘Start remote node’ allowed
Mandatory	0 1	CANopen device may be present prior to network start-up (optional) CANopen device shall be present prior to network start-up (mandatory)
Reset communication	0	no Reset Communication check implemented
Software version	0	no software verification implemented
Software update	0	no automatic software update implemented

Table 6-42 Nmt Startup – Bit Definition

6.2.35 Request NMT

DESCRIPTION

Represents the current Nmt State of the connected CANopen Slaves when reading the objects (heartbeat must be active) and make possible to send nmt commands to the connected CANopen Slaves when writing the Object.

Name	Request NMT	
Index	0x1F82	
Number of entries	InternalNet: SlaveNet:	2 128

Name	Request NMT Node 1 to Number of Entries	
Index	0x1F82	
Subindex	1…number of entries	
Type	UNSIGNED8	
Access	RW	
Default Value	0	
Value Range	–	–

Name	Request NMT all Nodes	
Index	0x1F82	
Subindex	1…number of entries	
Type	UNSIGNED8	
Access	WO	
Default Value	0	
Value Range	–	–

OnReadAccess	Value	OnWriteAccess
Nmt state unknown	0x00	–
Nmt state stopped	0x04	Request Nmt Stop Command
Nmt state Operational	0x05	Request Nmt Start Command
–	0x06	Request Nmt Reset Command
–	0x07	Request Nmt Reset Communication Command
Nmt state Pre-operational	0x7F	Request Enter Pre-operational Command

Table 6-43 Request Nmt – Commands and State Information

6.2.36 Device Type Identification

DESCRIPTION

Part of the network list, allowing to enter values for expected device types (object 0x1000, subindex 0x00 of the corresponding slave).

Name	Device Type Identification	
Index	0x1F84	
Number of entries	InternalNet: SlaveNet:	1 127

Name	Device Type Identification Node 1 to Number of Entries	
Index	0x1F84	
Subindex	1…number of entries	
Type	UNSIGNED32	
Access	RW	
Default Value	InternalNet: SlaveNet:	0x00010192 0x00000000
Value Range	–	–

6.2.37 Vendor Identification

DESCRIPTION

Part of the network list. It allows entering values for expected Vendor identification (object 0x1018 subindex 0x01 of the corresponding slave).

Name	Vendor Identification	
Index	0x1F85	
Number of entries	InternalNet: SlaveNet:	1 127

Name	Vendor Identification Node 1 to Number of Entries	
Index	0x1F85	
Subindex	1…number of entries	
Type	UNSIGNED32	
Access	RW	
Default Value	InternalNet: SlaveNet:	0x000000FB 0x00000000
Value Range	–	–

6.2.38 Product Code

DESCRIPTION

Part of the network list. It allows entering values for expected Product Code (object 0x1018 subindex 0x02 of the corresponding slave).

Name	Product Code	
Index	0x1F86	
Number of entries	InternalNet: SlaveNet:	1 127

Name	Product Code Node 1 to Number of Entries	
Index	0x1F86	
Subindex	1…number of entries	
Type	UNSIGNED32	
Access	RW	
Default Value	0	
Value Range	–	–

6.2.39 Revision Number

DESCRIPTION

Part of the network list. It allows entering values for expected Vendor identification (object 0x1018 subindex 0x03 of the corresponding slave).

Name	Revision Number	
Index	0x1F87	
Number of entries	InternalNet: SlaveNet:	1 127

Name	Revision Number Node 1 to Number of Entries	
Index	0x1F87	
Subindex	1…number of entries	
Type	UNSIGNED32	
Access	RW	
Default Value	0	
Value Range	–	–

6.2.40 Serial Number Node

DESCRIPTION

Part of the network list. It allows entering values for expected Serial Number (object 0x1018 subindex 0x04 of the corresponding slave).

Name	Serial Number	
Index	0x1F88	
Number of entries	InternalNet: SlaveNet:	1 127

Name	Serial Number Node 1 to Number of Entries	
Index	0x1F88	
Subindex	1…number of entries	
Type	UNSIGNED32	
Access	RW	
Default Value	0	
Value Range	–	–

6.2.41 Boot Time

DESCRIPTION

Defines the time that will elapse before the bootup procedure accesses the slaves, given in milliseconds (ms).

Name	Boot Time	
Index	0x1F89	
Subindex	0x00	
Type	UNSIGNED32	
Access	RW	
Default Value	500	
Value Range	200	30000

6.2.42 Node ID

DESCRIPTION

The identification of the CANopen node defined by hardware switches or by the Layer Setting Services (LSS).

REMARKS

Changes to this object will only have an effect after restart. Therefore, store all parameters and set DIP switch to “0” before restart.

Name	Node ID	
Index	MasterNet: InternalNet: SlaveNet:	0x2000 0x5000 0x5000
Subindex	0x00	
Type	UNSIGNED8	
Access	MasterNet: InternalNet: SlaveNet:	RW RO RW
Default Value	MasterNet: InternalNet: SlaveNet:	– 127 127
Value Range	1	127

6.2.43 CAN Bitrate

DESCRIPTION

Holds the desired bit rate of the CAN interface. It can differ from the actually configured bit rate if automatic bit rate detection is or was activated.

REMARKS

Changes to this object will only have an effect after restart. Therefore, store all parameters before restart.

Automatic bit rate detection is activated in the following cases:

- CAN Bitrate is set to “9” and saved (followed by a reset/power on).
- DIP switch “CAN automatic bit rate detection” is set (followed by a reset/power on).
- Bit rate was considered as wrong due to many error frames (after 3 seconds of no bus traffic).

RELATED OBJECTS

➔“CAN Bitrate Display” on page 6-88

Name	CAN Bitrate	
Index	MasterNet: InternalNet: SlaveNet:	0x2001 0x5001 0x5001
Subindex	0x00	
Type	UNSIGNED16	
Access	MasterNet: InternalNet: SlaveNet:	RW RO RW
Default Value	0	
Value Range	➔Table 6-44	–

Value	Bit Rate
0	1 Mbit/s
1	800 kbit/s
2	500 kbit/s
3	250 kbit/s
4	125 kbit/s
5	*reserved*
6	50 kbit/s
7	20 kbit/s
8	*reserved*
9	(auto detection)

Table 6-44 CAN Bit Rate Codes

6.2.44 RS232 Baudrate

DESCRIPTION

Sets the baud rate of the serial communication interface.

Changes to this object will only have an effect after restart. Therefore, store all parameters before restart.

Name	RS232 Baudrate	
Index	0x2002	
Subindex	0x00	
Type	UNSIGNED16	
Access	RW	
Default Value	5	
Value Range	➔Table 6-45	–

Value	Baud Rate
0	9.6 kBaud
1	14.4 kBaud
2	19.2 kBaud
3	38.4 kBaud
4	57.6 kBaud
5	115.2 kBaud

Table 6-45 RS232 Baud Rate Codes

6.2.45 EPOS2 P Version Numbers

DESCRIPTION

Contains the EPOS2 P's software version.

Name	EPOS2 P Version Numbers
Index	0x2003
Number of entries	0x04

Name	EPOS2 P Software Version	
Index	0x2003	
Subindex	0x01	
Type	UNSIGNED16	
Access	RO	
Default Value	–	
Value Range	–	–

DESCRIPTION

Contains the EPOS2 P's hardware version.

Name	EPOS2 P Hardware Version		
Index	0x2003		
Subindex	0x02		
Type	UNSIGNED16		
Access	RO		
Default Value	–		
Value Range	0x7310	EPOS2 P 24/5	–

DESCRIPTION

If the value is not zero, an application-specific firmware is installed on this EPOS2 P.

Name	EPOS2 P Application Number	
Index	0x2003	
Subindex	0x03	
Type	UNSIGNED16	
Access	RO	
Default Value	–	
Value Range	–	–

DESCRIPTION

Used as version number of the application or as internal revision number.

Name	EPOS2 P Application Version	
Index	0x2003	
Subindex	0x04	
Type	UNSIGNED16	
Access	RO	
Default Value	–	
Value Range	–	–

6.2.46 Module Version Numbers

DESCRIPTION

Contains the software version of the internal module.

Name	Module Version Numbers
Index	0x2004
Number of entries	0x04

Name	Module Software Version	
Index	0x2004	
Subindex	0x01	
Type	UNSIGNED16	
Access	RO	
Default Value	–	
Value Range	–	–

DESCRIPTION

Contains the hardware version of the internal module.

Name	Module Hardware Version	
Index	0x2004	
Subindex	0x02	
Type	UNSIGNED16	
Access	RO	
Default Value	–	
Value Range	–	–

DESCRIPTION

If not “0” (zero), an application-specific firmware is installed on the internal module.

Name	Module Application Number	
Index	0x2004	
Subindex	0x03	
Type	UNSIGNED16	
Access	RO	
Default Value	–	
Value Range	–	–

DESCRIPTION

The application version is used as version number of an application or as internal revision number.

Name	Module Application Version	
Index	0x2004	
Subindex	0x04	
Type	UNSIGNED16	
Access	RO	
Default Value	–	
Value Range	–	–

6.2.47 RS232 Frame Timeout

DESCRIPTION

Defines the timeout over a RS232 communication frame. It is scaled in milliseconds [ms].

Name	RS232 Frame Timeout	
Index	0x2005	
Subindex	0x00	
Type	UNSIGNED16	
Access	RW	
Default Value	500	
Value Range	–	–

6.2.48 USB Frame Timeout

DESCRIPTION

Defines the timeout over an USB communication frame. It is scaled in milliseconds [ms].

Name	USB Frame Timeout	
Index	0x2006	
Subindex	0x00	
Type	UNSIGNED16	
Access	RW	
Default Value	500	
Value Range	–	–

6.2.49 CAN SDO Frame Timeout

DESCRIPTION

Defines the timeout over a SDO communication frame. It is scaled in milliseconds [ms].

Name	CAN SDO Frame Timeout	
Index	0x2007	
Subindex	0x00	
Type	UNSIGNED16	
Access	RW	
Default Value	500	
Value Range	–	–

6.2.50 CAN Bitrate Display

DESCRIPTION

Represents the actually configured CAN bit rate. Its value can differ from the value of the object CAN Bitrate if automatic bit rate detection is or was active. In all other cases the value of these two objects are identical.

Name	CAN Bitrate Display	
Index	0x200A	
Subindex	0x00	
Type	UNSIGNED16	
Access	RO	
Default Value	–	
Value Range	➔Table “CAN Bit Rate Codes” on page 6-83	–

6.2.51 Serial Number

DESCRIPTION

The serial number of the EPOS2P can be read here. If “0” (zero), the serial number is unknown.

Name	Serial Number	
Index	0x200B	
Subindex	0x00	
Type	UNSIGNED64	
Access	RO	
Default Value	–	
Value Range	–	–

6.2.52 Startup Program Control

DESCRIPTION

Controls the start of a stored application program.

Name	Startup Program Control	
Index	0x2F51	
Subindex	0x00	
Type	UNSIGNED16	
Access	RW	
Default Value	–	
Value Range	➔Table 6-45	–

Value	Bootup State	Description
0	Stopped	no application execution
1	Coldstart	retain variables will be initialized
2	Warmstart	retain variables contain same value as before power down
3	Hotstart	

Table 6-46 Program Control Codes

6.2.53 Process Image

DESCRIPTION

Used to read or write several process variables of a Master Network in one block.

Name	Process Image
Index	0x2F70
Number of entries	0x03

DESCRIPTION

Used to configure a section of the process image for write access. The access on the defined range is given by ➔“Process Image Domain” on page 6-90.

REMARKS

Selection length set to 0x0000 represents access on all process input variables regardless of the selection start value.

Name	Process Image Range Inputs	
Index	0x2F70	
Subindex	0x01	
Type	UNSIGNED32	
Access	RW	
Default Value	0	
Value Range	➔Table 6-47	–

DESCRIPTION

Used to configure a section of the process image for read access. The access on the defined range is given by ➔“Process Image Domain” on page 6-90”.

REMARKS

Selection length set to 0x0000 represents access on all process input variables regardless of the selection start value.

Name	Process Image Range Outputs	
Index	0x2F70	
Subindex	0x02	
Type	UNSIGNED32	
Access	RW	
Default Value	0	
Value Range	➔Table 6-47	–

Bit	Range	Description
31…16	0x0000…0x00E0	Selection length
15…0	0x0000…0x00DF	Selection start

Table 6-47 Process Image Range (Inputs & Outputs) – Structure

DESCRIPTION

Provides access to the process variables as configured with the objects ➔“Process Image Range Inputs” on page 6-89 and ➔“Process Image Range Outputs” on page 6-89, respectively.

Name	Process Image Domain	
Index	0x2F70	
Subindex	0x03	
Type	DOMAIN	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.54 Axis Data

DESCRIPTION

The association between the logical axis number 0 to 31 (element “AxisNo” in AXIS_REF structure for the motion control function blocks) and the physical slave devices (Slave Number defined by CAN Port number and Node ID) is done here.

Name	Axis Data	
Index	0x5280…0x529F	
Number of entries	MasterNet: InternalNet: SlaveNet:	1 4 4

Name	Axis 0 to 31 Slave Number	
Index	0x5280…0x529F	
Subindex	0x01	
Type	UNSIGNED8	
Access	MasterNet: InternalNet: SlaveNet:	RW RO RO
Default Value	Index 0x5280 Index 0x5281 Index 0x5282 … Index 0x529F	0x0101 0x0201 0x0202 … 0x021F
Value Range	Port number: 0,0 1 2	Node ID: Axis unused Internal node 1 External nodes 1…127

Bit 15…8	Bit 7…0
Port number	Node ID

Table 6-48 Axis Data – Structure

DESCRIPTION

Internally used.

REMARKS

The EPOS2 P internal processing of the motion control function blocks needs the status of the connected axis. Therefore, the statusword of the associated axis should be transferred to this object by PDOs. The PDO map-pings are set correct for the axis 0 to 31 (EPOS [Internal] and external node 1 to 31) when the default parameter setting is used.

Name	Axis 0 to 31 Statusword	
Index	0x5280…0x529F	
Subindex	0x02	
Type	UNSIGNED16	
Access	RW	
Default Value	–	
Value Range	–	–

Description

The EPOS2 P internal processing of the motion control function blocks needs the modes of operation of the connected axis. Therefore, the modes of operation display of the associated axis should be transferred to this object by PDOs.

Name	Axis 0 to 31 Modes of Operation	
Index	0x5280…0x529F	
Subindex	0x04	
Type	UNSIGNED8	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.55 Process Input Integer 8

DESCRIPTION

Process input channel for 8-bit integer variables.

Name	Process Input Integer 8	
Index	0xA000	
Number of entries	MasterNet: InternalNet: SlaveNet:	16 4 64

Name	Process Input INT8 – 1 to Number of Entries	
Index	0xA000	
Subindex	1…number of entries	
Type	INTEGER8	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.56 Process Input Unsigned 8

DESCRIPTION

Process input channel for unsigned 8-bit integer variables.

Name	Process Input Unsigned 8	
Index	0xA040	
Number of entries	MasterNet: InternalNet: SlaveNet:	16 4 64

Name	Process Input UINT8 – 1 to Number of Entries	
Index	0xA040	
Subindex	1…number of entries	
Type	UNSIGNED8	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.57 Process Input Integer 16

DESCRIPTION

Process input channel for 16-bit integer variables.

Name	Process Input Integer 16	
Index	0xA0C0	
Number of entries	MasterNet: InternalNet: SlaveNet:	16 4 64

Name	Process Input INT16 – 1 to Number of Entries	
Index	0xA0C0	
Subindex	1…number of entries	
Type	INTEGER16	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.58 Process Input Unsigned 16

DESCRIPTION

Process input channel for unsigned 16-bit integer variables.

Name	Process Input Unsigned 16	
Index	0xA100	
Number of entries	MasterNet: InternalNet: SlaveNet:	16 4 64

Name	Process Input UINT16 – 1 to Number of Entries	
Index	0xA100	
Subindex	1…number of entries	
Type	UNSIGNED16	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.59 Process Input Integer 32

DESCRIPTION

Process input channel for 32-bit integer variables.

Name	Process Input Integer 32	
Index	0xA1C0	
Number of entries	MasterNet: InternalNet: SlaveNet:	16 4 64

Name	Process Input INT32 – 1 to Number of Entries	
Index	0xA1C0	
Subindex	1…number of entries	
Type	INTEGER32	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.60 Process Input Unsigned 32

DESCRIPTION

Process input channel for unsigned 32-bit integer variables.

Name	Process Input Unsigned 32	
Index	0xA200	
Number of entries	MasterNet: InternalNet: SlaveNet:	16 4 64

Name	Process Input UINT32 – 1 to Number of Entries	
Index	0xA200	
Subindex	1…number of entries	
Type	UNSIGNED32	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.61 Process Input Integer 64

DESCRIPTION

Process input channel for 64-bit integer variables.

Name	Process Input Integer 64	
Index	0xA400	
Number of entries	InternalNet: SlaveNet:	2 32

Name	Process Input INT64 – 1 to Number of Entries	
Index	0xA400	
Subindex	1…number of entries	
Type	INTEGER64	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.62 Process Input Unsigned 64

DESCRIPTION

Process input channel for unsigned 64-bit integer variables.

Name	Process Input Unsigned 64	
Index	0xA200	
Number of entries	InternalNet: SlaveNet:	2 32

Name	Process Input UINT64 – 1 to Number of Entries	
Index	0xA200	
Subindex	1…number of entries	
Type	UNSIGNED64	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.63 Process Output Integer 8

DESCRIPTION

Process output channel for 8-bit integer variables.

Name	Process Output Integer 8	
Index	0xA480	
Number of entries	MasterNet: InternalNet: SlaveNet:	16 4 64

Name	Process Output INT8 – 1 to Number of Entries	
Index	0xA480	
Subindex	1…number of entries	
Type	INTEGER8	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.64 Process Output Unsigned 8

DESCRIPTION

Process output channel for unsigned 8-bit integer variables.

Name	Process Output Unsigned 8	
Index	0xA4C0	
Number of entries	MasterNet: InternalNet: SlaveNet:	16 4 64

Name	Process Output UINT8 – 1 to Number of Entries	
Index	0xA4C0	
Subindex	1…number of entries	
Type	UNSIGNED8	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.65 Process Output Integer 16

DESCRIPTION

Process output channel for 16-bit integer variables.

Name	Process Output Integer 16	
Index	0xA540	
Number of entries	MasterNet: InternalNet: SlaveNet:	16 4 64

Name	Process Output INT16 – 1 to Number of Entries	
Index	0xA540	
Subindex	1…number of entries	
Type	INTEGER16	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.66 Process Output Unsigned 16

DESCRIPTION

Process output channel for unsigned 16-bit integer variables.

Name	Process Output Unsigned 16	
Index	0xA580	
Number of entries	MasterNet: InternalNet: SlaveNet:	16 4 64

Name	Process Output UINT16 – 1 to Number of Entries	
Index	0xA580	
Subindex	1…number of entries	
Type	UNSIGNED16	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.67 Process Output Integer 32

DESCRIPTION

Process output channel for 32-bit integer variables.

Name	Process Output Integer 32	
Index	0xA640	
Number of entries	MasterNet: InternalNet: SlaveNet:	16 4 64

Name	Process Output INT32 – 1 to Number of Entries	
Index	0xA640	
Subindex	1…number of entries	
Type	INTEGER32	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.68 Process Output Unsigned 32

DESCRIPTION

Process output channel for unsigned 32-bit integer variables.

Name	Process Output Unsigned 32	
Index	0xA680	
Number of entries	MasterNet: InternalNet: SlaveNet:	16 4 64

Name	Process Output UINT32 – 1 to Number of Entries	
Index	0xA680	
Subindex	1…number of entries	
Type	UNSIGNED32	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.69 Process Output Integer 64

DESCRIPTION

Process output channel for 64-bit integer variables.

Name	Process Output Integer 64	
Index	0xA880	
Number of entries	InternalNet: SlaveNet:	2 32

Name	Process Output INT64 – 1 to Number of Entries	
Index	0xA880	
Subindex	1…number of entries	
Type	INTEGER64	
Access	RW	
Default Value	–	
Value Range	–	–

6.2.70 Process Output Unsigned 64

DESCRIPTION

Process output channel for unsigned 64-bit integer variables.

Name	Process Output Unsigned 64	
Index	0xAC80	
Number of entries	InternalNet: SlaveNet:	2 32

Name	Process Output UINT64 – 1 to Number of Entries	
Index	0xAC80	
Subindex	1…number of entries	
Type	UNSIGNED64	
Access	RW	
Default Value	–	
Value Range	–	–

7 Firmware Version History

7.1 Version Overview

Date [d.m.y]	Version		Application		Description
	Software	Hardware	#	Version	
20.12.2013	0205h	7310h	0000h	0000h	Bug fixing
14.01.2013	0204h	7310h	0000h	0000h	Bug fixing
15.12.2011	0203h	7310h	0000h	0000h	Bug fixing
18.03.2011	0202h	7310h	0000h	0000h	New feature, bug fixing
03.09.2010	0201h	7310h	0000h	0000h	New feature, bug fixing
23.04.2010	0200h	7310h	0000h	0000h	Initial firmware release

Table 7-49 Version Overview

7.2 Software Version 0205h

0205h	Hardware	Firmware Filename
Binary Files	EPOS2 P 24/5	Epos_P_0205h_7310h_0000h_0000h.bin

Table 7-50 Software Version 0205h (1)

0205h	Description	
Changes	Bugfix	FB MC_SetOperationMode: Writing of negative values corrected.
	Bugfix	FB CAN_SetTxPdoEvent: Delayed “Done” output clearing fixed.
	Bugfix	STRING functions “delete”, “insert”, “mid”, “replace” reactivated.
	Bugfix	Obsolete axis internal process data at receiving Boot-Up message from slave fixed.

Table 7-51 Software Version 0205 (2)

7.3 Software Version 0204h

0204h	Hardware	Firmware Filename
Binary Files	EPOS2 P 24/5	Epos_P_0204h_7310h_0000h_0000h.bin

Table 7-52 Software Version 0204h (1)

0204h	Description	
Changes	Bugfix	Ladder program (LD) startup problem fixed.
	Bugfix	Instantiating of multiple tasks with the same interrupt event prevented.
	Bugfix	Scheduler improved for tasks of type “Interrupt” or “Timer”.

Table 7-53 Software Version 0204 (2)

7.4 Software Version 0203h

0203h	Hardware	Firmware Filename
Binary Files	EPOS2 P 24/5	Epos_P_0203h_7310h_0000h_0000h.bin

Table 7-54 Software Version 0203h (1)

0203h	Description	
Changes	Bugfix	EPOS2-internal bugfixes.

Table 7-55 Software Version 0203h (2)

7.5 Software Version 0202h

0202h	Hardware	Firmware Filename
Binary Files	EPOS2 P 24/5	Epos_P_0202h_7310h_0000h_0000h.bin

Table 7-56 Software Version 0202h (1)

0202h	Description	
Changes	Bugfix	EMCY message handling of CAN-M adapted: • NMT state stopped according to the spec. CiA 301. • Warnings will not be sent now.
	Bugfix	Network bootup improved: • Detection of slave with autobitrate. • Ensure program start due to a reset node command. • Sporadic unforced "CAN Passive Errors" eliminated.
	Function Blocks	FB "MC_WritePrameter": • ParameterNumber range extended with SaveAllParameters (1000).

Table 7-57 Software Version 0202h (2)

7.6 Software Version 0201h

0201h	Hardware	Firmware Filename
Binary Files	EPOS2 P 24/5	Epos_P_0201h_7310h_0000h_0000h.bin

Table 7-58 Software Version 0201h (1)

0201h	Description	
Changes	Bugfix	Handling of 8 byte variables for function block “CAN_SetTxPdoEvent” fixed.
	Bugfix	Default values of function blocks “CAN_Nmt”, “CAN_SdoRead”, “CAN_SdoWrite”, and “CAN_SetTxPdoEvent” corrected.
	Bugfix	Communication to OpenPCS improved.
	Bugfix	Clearing of OpenPCS error via «EPOS Studio» activated.
	Bugfix	Handling of slave devices with activated Automatic Bitrate Detection improved.
	Bugfix	Doubled axis definitions check added.
New Features	CDA	Control Data Analyzer implemented

Table 7-59 Software Version 0201h (2)

7.7 Software Version 0200h

0200h	Hardware	Firmware Filename
Binary Files	EPOS2 P 24/5	Epos_P_0200h_7310h_0000h_0000h.bin

Table 7-60 Software Version 0200h (1)

0200h	Description	
Changes	none	Initial release
New Features	none	Initial release

Table 7-61 Software Version 0200h (2)

LIST OF FIGURES

••page intentionally left blank••

LIST OF TABLES

INDEX

M

N

O

P

R

S

T

U

V

maxon motor ag
Brünigstrasse 220
P.O.Box 263
CH-6072 Sachseln
Switzerland

Phone +41 41 666 15 00
Fax +41 41 666 16 50

www.maxonmotor.com