maxon motor

maxon motor control | EPOS3 EtherCAT Positioning Controllers

Communication Guide | Edition May 2016

EPOS3 EtherCAT

Positioning Controllers

Communication Guide

epos.maxonmotor.com

Document ID: rel6559

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 EPOS3 EtherCAT is considered as partly completed machinery according to EU directive 2006/42/EC, Article 2, Clause (g) and therefore **is 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 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

••page intentionally left blank••

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
«Abcd»	indicating a title or a name (such as of document, product, mode, etc.)
(n)	referring to an item (such as order number, list item, etc.)
➔	denotes “see”, “see also”, “take note of” or “go to”

Table 1-1 Notations used in this Document

1.4 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
EtherCAT®	© EtherCAT Technology Group, DE-Nuremberg

Table 1-2 Brand Names and Trademark Owners

1.5 Sources for additional Information

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

#	Reference
[1]	maxon motor: EPOS3 EtherCAT Communication Guide EPOS CD-ROM or www.maxonmotor.com
[2]	USB Implementers Forum: Universal Serial Bus Revision 2.0 Specification www.usb.org/developers/docs

Table 1-3 Sources for additional Information

1.6 Copyright

The present document – including all parts thereof – is protected by copyright. Any use (including reproduction, translation, microfilming and other means of electronic data processing) beyond the narrow restrictions of the copyright law without the prior approval of maxon motor ag, is not permitted and subject to persecution under the applicable law.

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

The present document provides you with the communication interfaces details on the EPOS3 EtherCAT Positioning Controllers. It contains descriptions of the USB interface.

maxon motor control’s EPOS3 EtherCAT is a small-sized, full digital, smart positioning control unit. Due to its flexible and high efficient power stage, the EPOS3 EtherCAT 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. It is specially designed to be commanded and controlled as a slave node in the network. In addition, the unit can be operated through any USB communication port.

Find the latest edition of the present document, as well as additional documentation and software to the EPOS3 EtherCAT Positioning Controllers also on the internet: ➔www.maxonmotor.com

2.1 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:

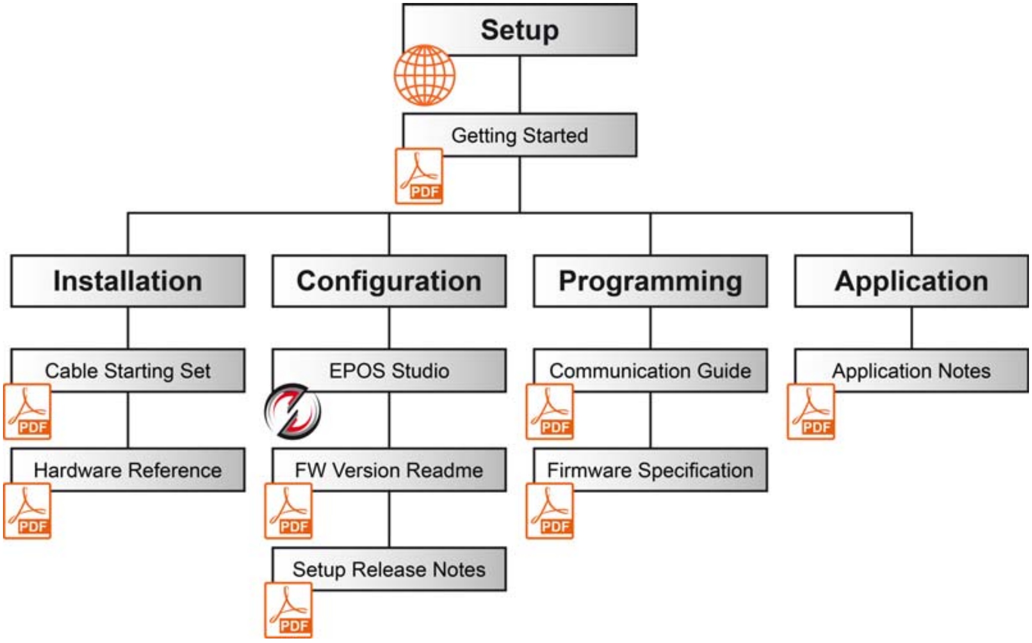

Figure 2-1 Documentation Structure

••page intentionally left blank••

3 EtherCAT

The EPOS3 EtherCAT positioning controllers’ implementation of EtherCAT follows the EtherCAT Technology Group (ETG) specifications.

Reference

You may access all relevant data and the free-for-download documentation (available in different languages) at EtherCAT’s WebSite (➔http://ethercat.org/). Navigate to downloads section and search for the document “EtherCAT Technology Introduction”.

The document “EtherCAT_Introduction_xxxx.pdf” will serve well as an introduction to EtherCAT and does include information on the technology, implementation, and possible applications.

In regard to EPOS3 EtherCAT’s firmware and hardware, please consult maxon’s comprehensive documentation set. Among others, you will find the following documents:

EPOS3 EtherCAT Firmware Specification

- Operating modes
- Communication and error handling
- Object dictionary
- etc.

EPOS3 EtherCAT Application Notes Collection

- Integration in a Master Beckhoff TwinCAT
- Device programming
- Configuration and tuning
- Data recording
- etc.

EPOS3 70/10 EtherCAT Hardware Reference

- Technical data
- Wiring diagrams and connection overview
- etc.

••page intentionally left blank••

4 EPOS3 Command Reference (USB)

4.1 Read Functions

4.1.1 Read Object Dictionary Entry (4 Data Bytes and less)

«ReadObject»

Read an object value from the Object Dictionary at the given Index and SubIndex.

Request Frame			
OpCode		0x10	
Len	USB	2	
		WORD Index	Index of Object
Parameters		(Low) BYTE SubIndex	SubIndex of Object
		(High) BYTE Dummy	0

Response Frame			
OpCode		0x00	
Len	USB	4	
Parameters		DWORD ErrorCode	➔“Error Code Definition” on page 6-21
		BYTE Data[4]	Data Bytes read

4.1.2 Read Object Dictionary Entry (5 Data Bytes and more)

«InitiateSegmentedRead»

Start reading an object value from the Object Dictionary at the given Index and SubIndex. Use the command “SegmentRead” to read the data.

Request Frame			
OpCode		0x12	
Len	USB	2	
		WORD Index	Index of Object
Parameters		(Low) BYTE SubIndex	SubIndex of Object
		(High) BYTE Dummy	0

Response Frame			
OpCode		0x00	
Len	USB	2	
Parameters		DWORD ErrorCode	➔“Error Code Definition” on page 6-21

«SegmentRead»

Read a data segment of the object initiated with the command “InitiateSegmentedRead”.

Request Frame					
OpCode		0x14			
Len	USB	1			
Parameters		(Low) BYTE ControlByte	not used toggle not used	[Bit 0…5] [Bit 6] [Bit 7]	– Toggle Bit –
		(High) BYTE Dummy	Byte without meaning		

Response Frame					
OpCode		0x00			
Len	USB	3…34			
Parameters		DWORD ErrorCode	➔“Error Code Definition” on page 6-21		
		(Low) BYTE ControlByte	length toggle more	[Bit 0…5] [Bit 6] [Bit 7]	Number of data bytes Toggle Bit More segments to read
		BYTE Data[0…63]	Data Bytes read		

4.2 Write Functions

4.2.1 Write Object Dictionary Entry (4 Data Bytes and less)

«WriteObject»

Write an object value to the Object Dictionary at the given Index and SubIndex.

Request Frame			
OpCode		0x11	
Len	USB	4	
Parameters		WORD Index	Index of Object
		(Low) BYTE SubIndex	SubIndex of Object
		(High) BYTE Dummy	0
		BYTE Data[4]	Data Bytes read

Response Frame			
OpCode		0x00	
Len	USB	2	
Data		DWORD ErrorCode	➔“Error Code Definition” on page 6-21

4.2.2 Write Object Dictionary Entry (5 Data Bytes and more)

«InitiateSegmentedWrite»

Start writing an object value to the Object Dictionary at the given Index and SubIndex. Use the command "SegmentWrite" to write the data.

Request Frame			
OpCode		0x13	
Len	USB	4	
Parameters		WORD Index	Index of Object
		(Low) BYTE SubIndex	SubIndex of Object
		(High) BYTE Dummy	0
		DWORD ObjectLength	Total number of bytes to write

Response Frame			
OpCode		0x00	
Len	USB	2	
Data		DWORD ErrorCode	➔"Error Code Definition" on page 6-21

«SegmentWrite»

Write a data segment to the object initiated with the command "InitiateSegmentedWrite".

Request Frame					
OpCode		0x15			
Len	USB	1…32			
Parameters		(Low) BYTE ControlByte	length toggle not used	[Bit 0…5] [Bit 6] [Bit 7]	Number of data bytes Toggle Bit –
		BYTE Data[0…63]	Data bytes to write		

Response Frame					
OpCode		0x00			
Len	USB	3			
Data		DWORD ErrorCode	➔"Error Code Definition" on page 6-21		
		(Low) BYTE ControlByte	length toggle not used	[Bit 0…5] [Bit 6] [Bit 7]	Number of data bytes Toggle Bit –
		(High) BATE Dummy	Byte without meaning		

••*page intentionally left blank*••

5 USB Communication

5.1 Data Link Layer

5.1.1 Flow Control

The EPOS3 EtherCAT Positioning Controllers always communicates as a slave. A frame is only sent as an answer to a request. All EPOS3 EtherCAT commands send an answer. The master always must start the communication by sending a packet structure.

Below described are the data flow while transmitting and receiving frames.

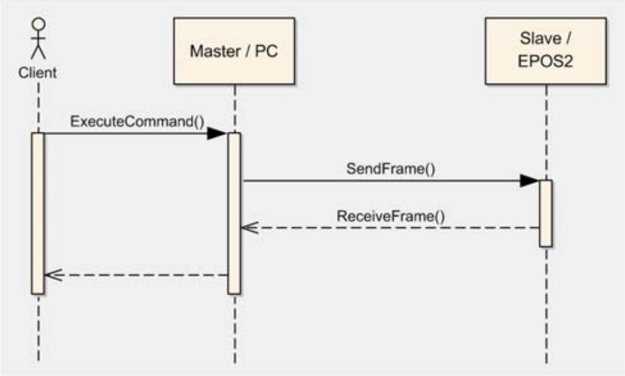

Figure 5-2 USB Communication – Command Sequence

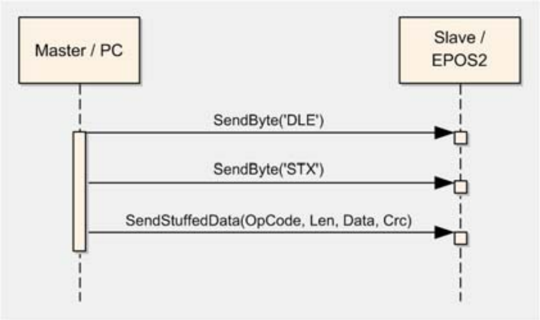

Figure 5-3 USB Communication – Sending a Data Frame to EPOS3

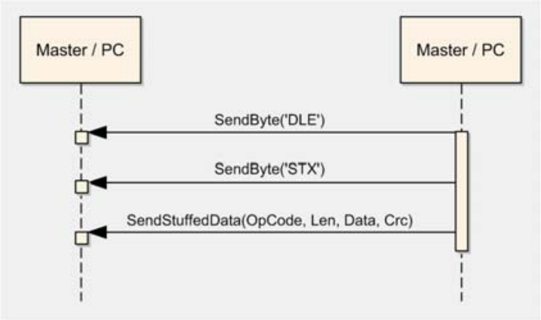

Figure 5-4 USB Communication – Receiving a Response Data Frame from EPOS3

5.1.2 Frame Structure

The data bytes are sequentially transmitted in frames. A frame composes of…

- a synchronization,
- a header,
- a variably long data field and
- a 16-bit long cyclic redundancy check (CRC) for verification of data integrity.

"DLE" (8-bit)	"STX" (8-bit)	OpCode (8-bit)	Len (8-bit)	Data[0] (16-bit)	…	Data[Len-1] (16-bit)	CRC (16-bit)
SYNC		HEADER		DATA			CRC

Figure 5-5 USB Communication – Frame Structure

SYNC The first two bytes are used for frame synchronization.

"DLE" Starting frame character "DLE" (Data Link Escape) = 0x90

"STX" Starting frame character "STX" (Start of Text) = 0x02

HEADER The header consists of 2 bytes. The first field determines the type of data frame to be sent or received. The next field contains the length of the data fields.

OpCode Operation command to be sent to the slave. For details on the command set ➔"EPOS3 Command Reference (USB)" on page 4-11.

Len represents the number of words (16-bit value) in the data fields [0…143].

DATA The data field contains the parameters of the message. The low byte of the word is transmitted first.

Data[i] The parameter word of the command. The low byte is transmitted first.

CRC The 16-bit CRC checksum using the algorithm CRC-CCITT.
The CRC calculation includes all bytes of the frame except the synchronization bytes, the data bytes must be calculated as a word.
First, you will need to shift to the data word's high byte.
This represents the opposite way as you transmit the data word.
For calculation, the 16-bit generator polynomial "$x^{16}+x^{12}+x^5+1$" is used.

CRC Checksum of the frame. The low byte is transmitted first.

Note
The CRC is calculated before stuffing the data. The elements "OpCode" to "Data[Len-1]" are included in CRC calculation. The synchronization elements "DLE" and "STX" are not included.

5.1.3 Error Control

5.1.3.1 Acknowledge

As a reaction to a bad OpCode or CRC value, the slave sends a frame containing the corresponding error code.

5.1.3.2 CRC Calculation

Packet M(x):	WORD DataArray[n]
Generator Polynom G(x):	10001000000100001 (= $x^{16}+x^{12}+x^{5}+x^{0}$)
DataArray[0]:	HighByte(Len) + LowByte(**OpCode**)
DataArray[1]:	**Data[0]**
DataArray[2]:	**Data[1]**
…	…
DataArray[n-1]:	**0x0000 (ZeroWord)**

```
WORD CalcFieldCRC(WORD* pDataArray, WORD numberOfWords)
{
   WORD shifter, c;
   WORD carry;
   WORD CRC = 0;

   //Calculate pDataArray Word by
   Word
   while(numberOfWords--)
   {
      shifter = 0x8000;             //Initialize BitX to Bit15
      c = *pDataArray++;            //Copy next DataWord to c
      do
      {
         carry = CRC & 0x8000;      //Check if Bit15 of CRC is set
         CRC <<= 1;                 //CRC = CRC * 2
         if(c & shifter) CRC++;     //CRC = CRC + 1, if BitX is set in c
         if(carry) CRC ^= 0x1021;   //CRC = CRC XOR G(x), if carry is true
         shifter >>= 1;             //Set BitX to next lower Bit, shifter = shifter/2
      } while(shifter);
   }
   return CRC
}
```

Figure 5-6 USB Communication – CRC Calculation

5.1.4 Character Stuffing

The sequence “DLE” and “STX” are reserved for frame start synchronization. If the character “DLE” appears at a position between “OpCode” and “CRC” and is not a starting character, the character must be doubled (character stuffing). Otherwise, the protocol begins to synchronize for a new frame. The character “STX” needs not to be doubled.

Examples:

Sending Data	0x21, **0x90**, 0x45
Stuffed Data	0x21, **0x90**, **0x90**, 0x45

Sending Data	0x21, **0x90, 0x02,** 0x45
Stuffed Data	0x21, **0x90**, **0x90**, **0x02**, 0x45

Sending Data	0x21, **0x90**, **0x90**, 0x45
Stuffed Data	0x21, **0x90**, **0x90**, **0x90**, **0x90**, 0x45

Important:
Character stuffing is used for all bytes in the frame except the starting characters.

5.1.5 Transmission Byte Order

The unit of data memory in EPOS3 EtherCAT is a word (16-bit value). To send and receive a word (16-bit) via the serial port, the low byte will be transmitted first.
Multiple byte data (word = 2 bytes, long words = 4 bytes) are transmitted starting with the less significant byte (LSB) first.

A word will be transmitted in following order: byte0 (LSB), byte1 (MSB).
A long word will be transmitted in following order: byte0 (LSB), byte1, byte2, byte3 (MSB).

5.1.6 Timeout Handling

The timeout is handled over a complete frame. Hence, the timeout is evaluated over the sent data frame, the command processing procedure and the response data frame. For each frame (frames, data processing), the timer is reset and timeout handling will recommence.

Object	Index	SubIndex	Default
USB Frame Timeout	0x2006	0x00	500 [ms]

Table 5-4 USB Communication – Timeout Handling

Note
To cover special requirements, the timeout may be changed by writing to the Object Dictionary!

5.1.7 Slave State Machine

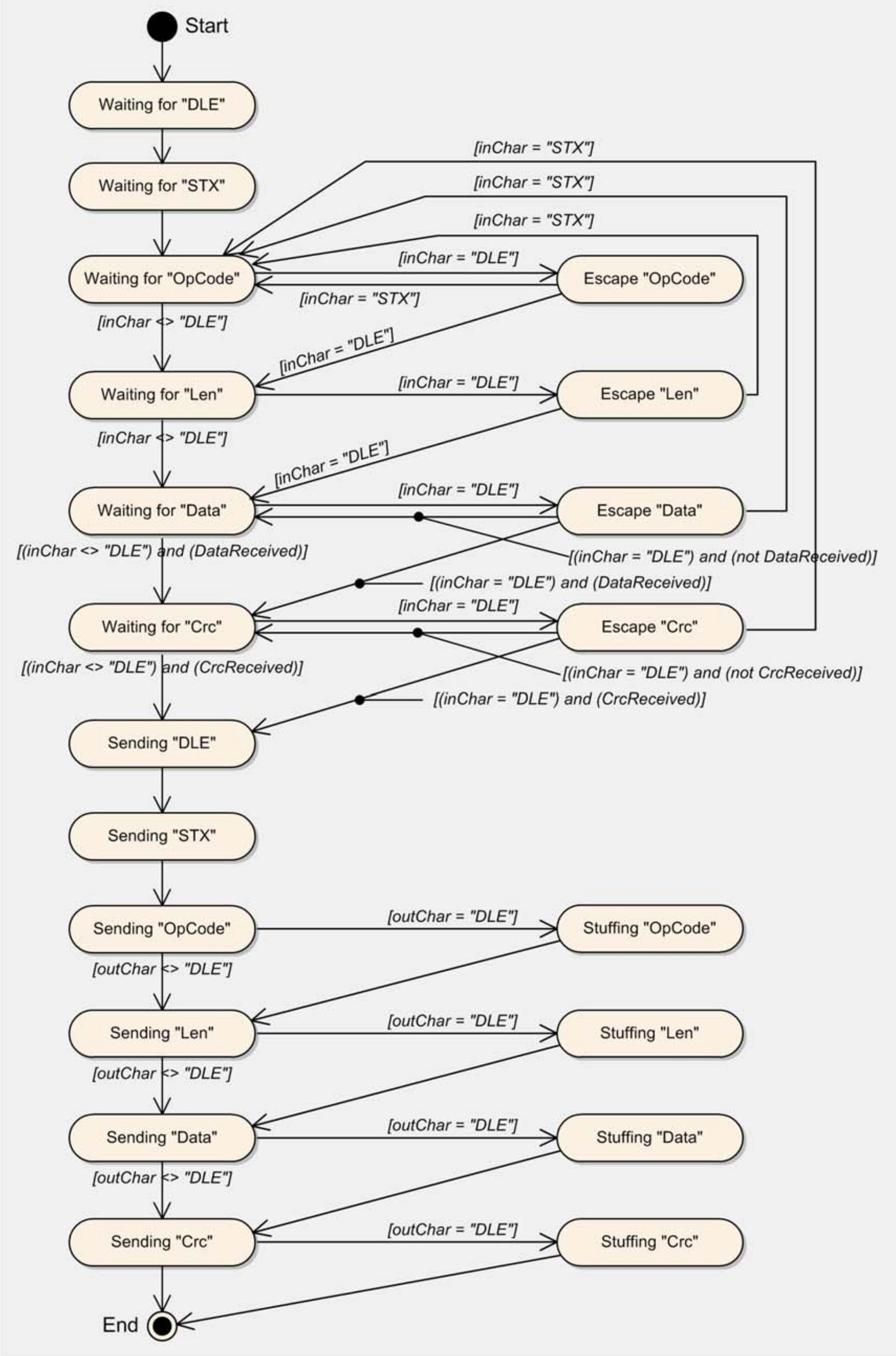

Figure 5-7 USB Communication – Slave State Machine

5.1.8 Example: Command Instruction

The following example shows composition and structure of the EPOS3 EtherCAT messages during transmission and reception via USB. The command sent to the EPOS3 EtherCAT is “ReadObject”, it may be used to read an object with 4 Bytes and less.

ReadObject “Home Position” (Index = 0x2081, SubIndex = 0x00)

“DLE”	“STX”	OpCode	Len	Data[0]	Data[1]	CRC
0x90	0x02	0x10	0x02	0x2081	0x0000	0xB43E

DLE	0x90	= Data Link Escape
STX	0x02	= Start of Text
OpCode	0x10	= ReadObject
Data[0]	0x2081	= 2 Words
LowByte data[1]	0x00	= SubIndex
HighByte data[1]	0x00	= Dummy

Table 5-5 ReadObject “Home Position”

Transmission Order: 0x90,0x02,0x10,0x02,0x81,0x20,0x00,0x00,0x3E,0xB4.

The EPOS3 EtherCAT will answer to the command “ReadObject” with an answer frame and the returned parameters in the data block as follows:

Answer to ReadObject “Home Position” (Index = 0x2081, SubIndex = 0x00)

“DLE”	“STX”	OpCode	Len	Data[0]	Data[1]	Data[2]	Data[3]	CRC
0x90	0x02	0x00	0x04	0x0000	0x0000	0x8090	0x0000	0x0834

DLE	0x90	= Data Link Escape
STX	0x02	= Start of Text
OpCode	0x00	= Answer
Len	0x04	= 4 Words
Data[0]	0x0000	= LowWord ErrorCode
Data[1]	0x0000	= HighWord ErrorCode
Data[2]	0x8090	= LowWord “HomePosition”
Data[3]	0x0000	= HighWord “HomePosition”

Table 5-6 Answer to ReadObject “Home Position”

Reception Order: 0x90,0x02,0x00,0x04,0x00,0x00,0x00,0x00,0x90,0x90,0x80,0x00,0x00,0x34,0x08

Note
Observe character stuffing methodology (➔“Character Stuffing” on page 5-18).

5.2 Physical Layer

Electrical Standard

maxon EPOS3 EtherCAT drives’ USB interface follows the «Universal Serial Bus Specification Revision 2.0». You may wish to download the file from the internet (for URL ➔“Sources for additional Information” on page 1-6), full details are described in chapter “7.3 Physical Layer”.

6 Error Code Definition

Refer to separate document ➔«EPOS3 EtherCAT Firmware Specification», chapter “Error Handling” for detailed information on error codes, device-specific errors, and error handling methodology.

••page intentionally left blank••

List of Figures

LIST OF TABLES

INDEX

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