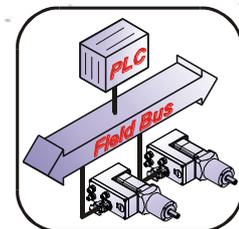
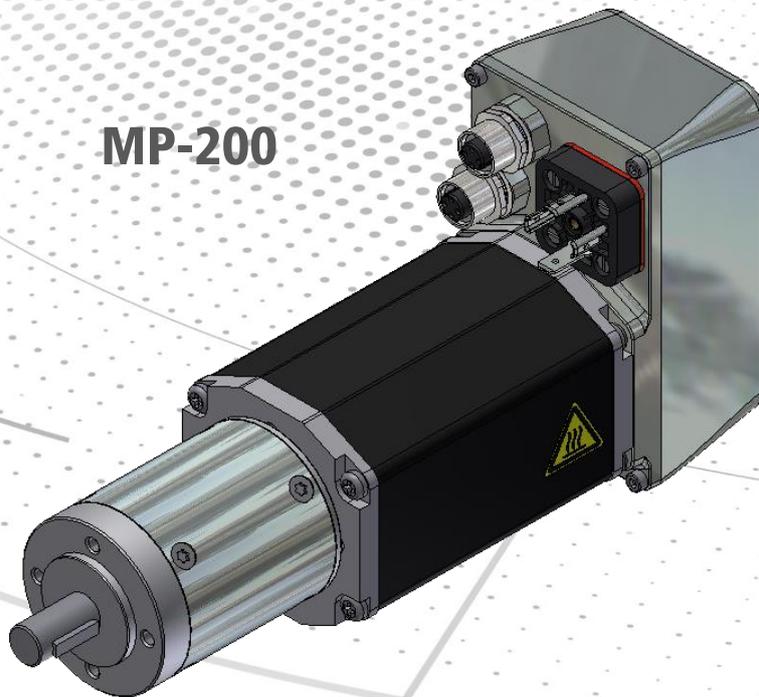


# Decentralized positioning drives

Download of the latest version of this manual: [TR-EMO-BA-GB-0022](#)

**MP-200**



- \_ Additional safety instructions
- \_ EtherCAT communication
- \_ Configuration / Parameterization
- \_ Troubleshooting / Diagnosis

**User Manual  
Interface**

---

## **TR-Electronic GmbH**

D-78647 Trossingen  
Eglishalde 6  
Tel.: (0049) 07425/228-0  
Fax: (0049) 07425/228-33  
E-mail: [info@tr-electronic.de](mailto:info@tr-electronic.de)  
<http://www.tr-electronic.de>

---

### **Copyright protection**

This Manual, including the illustrations contained therein, is subject to copyright protection. Use of this Manual by third parties in contravention of copyright regulations is not permitted. Reproduction, translation as well as electronic and photographic archiving and modification require the written content of the manufacturer. Violations shall be subject to claims for damages.

---

### **Subject to modifications**

The right to make any changes in the interest of technical progress is reserved.

---

### **Document information**

Release date / Rev. date:	09/29/2020
Document / Rev. no.:	TR-EMO-BA-GB-0022 v03
File name:	TR-EMO-BA-GB-0022-03.docx
Author:	MÜJ

---

### **Font styles**

*Italic* or **bold** font styles are used for the title of a document or are used for highlighting.

`Courier` font displays text, which is visible on the display or screen and software menu selections.

" < > " indicates keys on your computer keyboard (such as <RETURN>).

---

### **Brand names**

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

All other specified products, names and logos serve exclusively for information purposes and may be trademarks of their respective owners, without any special marking to indicate this.

---

# Contents

<b>Contents .....</b>	<b>3</b>
<b>List of tables .....</b>	<b>8</b>
<b>List of figures .....</b>	<b>9</b>
<b>List of formulas.....</b>	<b>9</b>
<b>Revision index .....</b>	<b>10</b>
<b>1 General information .....</b>	<b>11</b>
1.1 Target group .....	11
1.2 Applicability .....	11
1.3 References.....	12
1.4 Abbreviations used / Terminology .....	13
<b>2 Additional safety instructions.....</b>	<b>15</b>
2.1 Definition of symbols and instructions .....	15
2.2 Organizational measures.....	15
<b>3 Technical data.....</b>	<b>16</b>
3.1 Current firmware versions.....	16
3.2 Electrical characteristics .....	16
3.3 EtherCAT / DS-402.....	17
3.4 CE Conformity.....	18
<b>4 EtherCAT Information .....</b>	<b>19</b>
4.1 EtherCAT functional principle .....	19
4.2 Protocol.....	20
4.3 EtherCAT State Machine (ESM).....	21
4.3.1 Transition into the OPERATIONAL state.....	22
4.4 Device profile .....	23
4.4.1 CANopen over EtherCAT (CoE) .....	24
4.5 Object dictionary .....	25
4.6 Process and Service Data Objects.....	25
4.6.1 Compatibility with the CiA DS-301 communication profile.....	26
4.7 Transmission of SDO messages .....	27
4.8 PDO mapping .....	29
4.9 Further information .....	29
<b>5 DSP 402 drive profile.....</b>	<b>30</b>
5.1 DSP 402 state machine .....	31

5.1.1 Operating condition .....	32
5.1.2 State transitions .....	33
5.1.3 Change mode of operation .....	34
5.2 “Profile Position (PP)” operating mode .....	35
5.2.1 Control word.....	35
5.2.2 Status word .....	36
5.2.3 Perform positioning .....	37
5.2.3.1 Absolute / relative positioning.....	38
5.2.3.2 Transfer of new movement records.....	39
5.2.3.3 Termination of a positioning movement .....	39
5.2.3.4 Relevant parameters .....	39
5.3 “Profile Velocity (PV)” operating mode .....	40
5.3.1 Control word.....	40
5.3.2 Status word .....	41
5.3.3 Perform speed ramp .....	42
5.3.4 Relevant parameters.....	43
5.4 “Homing (HM)” operating mode .....	44
5.4.1 Control word.....	44
5.4.2 Status word .....	45
5.4.3 Perform position adjustment .....	46
5.4.4 Relevant parameters.....	46
5.5 “Cyclic Synchronous Position (CSP)” operating mode .....	47
5.5.1 Control word.....	47
5.5.2 Status word .....	48
5.5.3 Perform operating mode .....	49
5.5.4 Relevant parameters.....	49
5.6 “Cyclic Synchronous Velocity (CSV)” operating mode .....	50
5.6.1 Control word.....	50
5.6.2 Status word .....	51
5.6.3 Perform operating mode .....	52
5.6.4 Relevant parameters.....	52
5.7 “Cyclic Synchronous Torque (CST)” operating mode.....	53
5.7.1 Control word.....	53
5.7.2 Status word .....	54
5.7.3 Perform operating mode .....	55
5.7.4 Relevant parameters.....	55
5.8 Units.....	56
5.8.1 Object 0x608F: Position encoder resolution .....	56
5.8.2 Object 0x6090: Speed encoder resolution.....	56
5.8.3 Object 0x6093: Position factor .....	57
5.8.4 Object 0x6094: Speed factor .....	59
5.8.5 Object 0x6097: Acceleration factor .....	61
<b>6 Installation- / connection notes, pin assignments .....</b>	<b>66</b>
<b>7 Optional holding brake / ballast circuit .....</b>	<b>67</b>
<b>8 Commissioning.....</b>	<b>67</b>
8.1 Device description file .....	67

<b>9 Object directory .....</b>	<b>68</b>
9.1 Object types, data types .....	68
9.2 Communication-Specific Standard Objects CiA DS-301 .....	69
9.2.1 Object 0x1000: Device Type .....	70
9.2.2 Object 0x1001: Error Register .....	70
9.2.3 Object 0x1003: Pre-defined Error Field .....	71
9.2.4 Object 0x1008: Manufacturer's Device Name .....	72
9.2.5 Object 0x1009: Manufacturer's Hardware Version .....	72
9.2.6 Object 0x100A: Manufacturer's Software Version .....	72
9.2.7 Object 0x1010: Save Backup Parameters.....	73
9.2.8 Object 0x1011: Reset Backup Parameters.....	74
9.2.9 Object 0x1018: Identity .....	75
9.2.10 Structure of the mapping parameter .....	76
9.2.10.1 Objects 0x1600-0x1601: Receive PDO mapping.....	77
9.2.10.2 Objects 0x1A00-0x1A01: Transmit PDO mapping.....	78
9.2.11 Object 0x1C00: Sync Manager Communication Type.....	79
9.2.12 Object 0x1C12: SM2 PDO assignment .....	81
9.2.13 Object 0x1C13: SM3 PDO assignment .....	82
9.3 Manufacturer-specific objects .....	83
9.3.1 Object 0x2EF3: TrAbeData.MVI16 .....	83
9.3.2 Object 0x2F04: DS402M.MPU32 .....	85
9.3.3 Object 0x2F0E: DS402M.MPX .....	89
9.4 Objects of the DSP 402 device profile .....	91
9.4.1 Object 0x6007: Abort Connection Code .....	93
9.4.2 Object 0x603F: Error Code .....	93
9.4.3 Object 0x6040: Control word .....	93
9.4.4 Object 0x6041: Status word.....	94
9.4.5 Object 0x605A: Quick Stop Option Code .....	94
9.4.6 Object 0x605B: Shutdown Option Code .....	95
9.4.7 Object 0x605C: Disable Operation Option Code .....	95
9.4.8 Object 0x605D: Stop Option Code .....	96
9.4.9 Object 0x605E: Fault Reaction Option Code.....	96
9.4.10 Object 0x6060: Operation Mode .....	97
9.4.11 Object 0x6061: Operation Mode Display .....	97
9.4.12 Object 0x6062: Required Position .....	97
9.4.13 Object 0x6064: Actual Position.....	98
9.4.14 Object 0x6065: Following Error Window.....	98
9.4.15 Object 0x6066: Following Error Timeout .....	98
9.4.16 Object 0x6067: In-Position Window .....	99
9.4.17 Object 0x6068: In-Position Timeout.....	99
9.4.18 Object 0x6069: Sensor Velocity.....	99
9.4.19 Object 0x606B: Required Velocity .....	100
9.4.20 Object 0x606C: Actual Velocity .....	100
9.4.21 Object 0x606D: Velocity Window.....	100
9.4.22 Object 0x606E: Velocity Window Timeout.....	101
9.4.23 Object 0x606F: Velocity Threshold.....	101
9.4.24 Object 0x6070: Velocity Threshold Timeout.....	101
9.4.25 Object 0x6071: Target Torque .....	102
9.4.26 Object 0x6072: Torque Limit.....	102
9.4.27 Object 0x6073: Limit for Torque generating Current .....	102
9.4.28 Object 0x6074: Required Torque.....	103
9.4.29 Object 0x6075: Motor rated Current/mA.....	103

9.4.30 Object 0x6076: Motor rated Torque/mNm .....	103
9.4.31 Object 0x6077: Actual Torque .....	104
9.4.32 Object 0x6078: Actual Current.....	104
9.4.33 Object 0x6079: Actual DC Link Circuit Voltage/mV .....	104
9.4.34 Object 0x607A: Target Position .....	105
9.4.35 Object 0x607B: Position Range Limits .....	105
9.4.36 Object 0x607C: Home Offset.....	106
9.4.37 Object 0x607D: Software Position Limits.....	106
9.4.38 Object 0x607E: Polarity .....	107
9.4.39 Object 0x607F: Max Profile Velocity .....	107
9.4.40 Object 0x6080: Max Motor Speed .....	107
9.4.41 Object 0x6081: Profile Velocity.....	108
9.4.42 Object 0x6083: Profile Acceleration.....	108
9.4.43 Object 0x6084: Profile Deceleration .....	108
9.4.44 Object 0x6085: Quick Stop Deceleration.....	109
9.4.45 Object 0x608F: Position Encoder Resolution .....	109
9.4.46 Object 0x6090: Velocity Encoder Resolution.....	110
9.4.47 Object 0x6093: Position Factor.....	111
9.4.48 Object 0x6094: Velocity Encoder Factor .....	112
9.4.49 Object 0x6097: Acceleration Factor.....	113
9.4.50 Object 0x6098: Homing Method .....	113
9.4.51 Object 0x6099: Homing speeds.....	114
9.4.52 Object 0x609A: Homing Acceleration .....	114
9.4.53 Object 0x60B0: Position Offset.....	115
9.4.54 Object 0x60B1: Velocity Offset .....	115
9.4.55 Object 0x60B2: Torque Offset .....	115
9.4.56 Object 0x60C5: Max Acceleration .....	116
9.4.57 Object 0x60C6: Max Deceleration .....	116
9.4.58 Object 0x60F4: Following Error .....	116
9.4.59 Object 0x60FD: Digital Inputs .....	117
9.4.60 Object 0x60FE: Digital Outputs .....	117
9.4.61 Object 0x60FF: Target Velocity .....	118
9.4.62 Object 0x6402: Motor Type .....	118
9.4.63 Object 0x6502: Supported Drive Modes.....	118
9.4.64 Object 0x6508: Runtime in 0.1 hours .....	119
9.4.65 Object 0x6510: Measuring Range in revolutions .....	119
<b>10 Projecting with TwinCAT System Manager V2.10 .....</b>	<b>120</b>
10.1 Installation of the ESI file .....	120
10.2 Connection of the encoTRive device .....	120
10.3 Creation of a TwinCAT project for an encoTRive device .....	121
10.4 Changing data objects .....	122
10.5 Storing and resetting backup parameters.....	123
10.6 Stop conditions, Digital Inputs and Hardware Limit Switches .....	124
10.6.1 Turning on the drive .....	124
10.6.2 Stop conditions: Priorities, Behavior .....	124
10.6.3 Control of the Digital Inputs.....	127
10.6.4 Operation of the Digital Inputs as Limit Switches .....	128
10.7 Control word and Status word .....	131
10.7.1 General .....	131
10.7.2 Status bit 2 <sup>7</sup> - Warning .....	131

10.7.3 Status bit 2 <sup>9</sup> - Remote .....	131
10.7.4 Status bit 2 <sup>10</sup> - Target Reached .....	131
10.7.5 Status bit 2 <sup>11</sup> - Internal limit active .....	132
10.7.6 Status bit 2 <sup>12</sup> - Set-point-acknowledge, Zero Speed .....	132
10.7.7 Status bit 2 <sup>13</sup> - Following error .....	132
10.8 Profile Position Mode (operating mode 1) .....	133
10.9 Profile Velocity Mode (operating mode 3) .....	136
10.10 Cyclic synchronous modes (operating modes 8, 9, 10) .....	137
10.11 Backup-EEPROM, runtime .....	137
10.12 Position adjustment by Homing (operating mode 6) .....	138
10.13 Manual tuning of the position and speed regulation .....	141
10.14 Gear backlash cancellation (Opmodes -48=0xD0).....	143
<b>11 TwinCAT System Manager – Example program (Download) .....</b>	<b>144</b>
<b>12 Troubleshooting and diagnosis options .....</b>	<b>145</b>
12.1 SDO error codes .....	145
12.2 EMCY error information .....	146
12.2.1 EncoTRive RE - Emergencies .....	146
12.2.2 Error register, object 0x1001 .....	146
12.2.3 Error code, object 0x1003 (Bit 0-15) .....	147
12.2.3.1 General .....	147
<b>13 EncoTRive RE - Accessories, Test Setup .....</b>	<b>151</b>
13.1 Power supply cables with Solenoid Valve or M23 Connectors .....	151
13.2 Test parameters for "EncoTRive RELC" .....	151
<b>14 EtherCAT Conformance Test Tool 1.20.80.0 .....</b>	<b>152</b>

## List of tables

Table 1: References .....	12
Table 2: Abbreviations used / Terminology .....	14
Table 3: States, State Machine .....	21
Table 4: States of the status word .....	32
Table 5: Operating states, status word .....	32
Table 6: State transitions .....	34
Table 7: Change operating state .....	34
Table 8: Control word (object 0x6040), "Profile Position" operating mode .....	35
Table 9: Status word (object 0x6041), "Profile Position" operating mode .....	36
Table 10: Positioning parameter .....	39
Table 11: Control word (object 0x6040), "Profile velocity" operating mode .....	40
Table 12: Status word (object 0x6041), "Profile velocity" operating mode .....	41
Table 13: Speed ramp parameter .....	43
Table 14: Control word (object 0x6040), "Homing" operating mode .....	44
Table 15: Status word (object 0x6041), "Homing" operating mode .....	45
Table 16: Definition of the homing status bits 10, 12, 13 .....	45
Table 17: Control word (object 0x6040), "Cyclic synchronous position (CSP)" operating mode .....	47
Table 18: Status word (object 0x6041), "Cyclic synchronous position (CSP)" operating mode .....	48
Table 19: Control word (object 0x6040), "Cyclic synchronous velocity (CSV)" operating mode .....	50
Table 20: Status word (object 0x6041), "Cyclic synchronous velocity (CSV)" operating mode .....	51
Table 21: Control word (object 0x6040), "Cyclic synchronous torque (CST)" operating mode .....	53
Table 22: Status word (object 0x6041), "Cyclic synchronous torque (CST)" operating mode .....	54
Table 23: Object codes in encoTRive .....	68
Table 24: Attributes .....	68
Table 25: encoTRive data types .....	68
Table 26: Communication specific standard objects .....	69
Table 27: Profile specific objects .....	92
Table 28: Values for Quick Stop Option Code .....	94
Table 29: Values for Shutdown Option Code .....	95
Table 30: Values for Disable Operation Option Code .....	95
Table 31: Values for Halt Option Code .....	96
Table 32: Values for Fault Reaction Option Code .....	96
Table 33: Values for operating mode .....	97
Table 34: Direction reversal .....	107
Table 35: SDO Abort Codes .....	145
Table 36: EMCY Error register, object 0x1001 .....	146
Table 37: EMCY Error code, object 0x1003 .....	150

## List of figures

Figure 1: EtherCAT functional principle.....	19
Figure 2: Ethernet frame structure .....	20
Figure 3: EtherCAT state machine .....	21
Figure 4: CANopen over EtherCAT communication mechanism .....	23
Figure 5: CANopen organized in the ISO/OSI layer model.....	24
Figure 6: EtherCAT organized in the ISO/OSI layer model.....	24
Figure 7: Structure of the object dictionary.....	25
Figure 8: Comparison of PDO/SDO characteristics .....	25
Figure 9: DSP 402 state machine .....	31
Figure 10: Positioning ramp.....	37
Figure 11: Absolute positioning (top) and relative positioning (bottom) .....	38
Figure 12: Speed ramp.....	42
Figure 13: Cyclic synchronous position control function (CSP) .....	49
Figure 14: Cyclic synchronous velocity control function (CSV).....	52
Figure 15: Cyclic synchronous torque control function (CST).....	55

## List of formulas

Formula 1: Position encoder resolution.....	56
Formula 2: Default value, position encoder resolution .....	56
Formula 3: Speed encoder resolution .....	56
Formula 4: Default value, speed encoder resolution.....	56
Formula 5: Position factor.....	57
Formula 6: Gear ratio .....	57
Formula 7: Default value, position factor .....	57
Formula 8: Speed factor .....	59
Formula 9: Default value, speed factor .....	59
Formula 10: Acceleration factor .....	61
Formula 11: Default for acceleration factor .....	61

<b>Revision index</b>
-----------------------

<b>Revision</b>	<b>Date</b>	<b>Index</b>
First release	11/23/16	00
TwinCAT System Manager – Example program (Download)	02/06/17	01
<p>Added chapters:</p> <ul style="list-style-type: none"> <li>- 3.1, Current firmware versions</li> <li>- 9.3.3, Object 0x2F0E: DS402M.MPX</li> <li>- 10.6, Stop conditions, Digital Inputs and Hardware Limit Switches</li> <li>- 10.7, Control word and Status word</li> <li>- 12.2.1, EncoTRive RE - Emergencies</li> </ul> <p>Diverse corrections in the chapters:</p> <ul style="list-style-type: none"> <li>- 10.5, Storing and resetting backup parameters</li> <li>- 10.8, Profile Position Mode (operating mode 1)</li> <li>- 10.9, Profile Velocity Mode (operating mode 3)</li> </ul> <p>Reference to the EtherCAT Conformance Test Tool:</p> <ul style="list-style-type: none"> <li>- 10, Projecting with TwinCAT System Manager V2.10</li> </ul>	02/14/17	02
Adaptations, up to point of time: MAH 6-May-2019; in document "EncoTRiveReManualEng.rtf". The document is now fully integrated and no longer needs to be referenced.	09/29/20	03

## 1 General information

This Manual contains the following topics:

- Additional safety instructions
- EtherCAT communication
- Configuration / Parameterization
- Troubleshooting and diagnosis

As the documentation is arranged in a modular structure, this encoTRive EtherCAT User Manual is supplementary to other documentation, such as customer-specific user manuals, dimensional drawings, brochures etc.

The encoTRive EtherCAT User Manual may be included in the customer's specific delivery package or it may be requested separately.

### 1.1 Target group

This documentation is directed towards

- Commissioning, operating and maintenance personnel, who are instructed to carry out activities on the encoTRive RELC positioning drive.

The respective qualifications of the personnel are defined in the assembly/commissioning manual in the chapter entitled "Choice and qualifications of personnel; basic obligations".

### 1.2 Applicability

This encoTRive EtherCAT Manual applies exclusively for the following decentralized positioning drive types with **EtherCAT** interface:

- MP-XXX-ETC

The products are labeled with affixed nameplates and are components of a system.

The following documentation therefore also applies:

- operator's operating instructions specific to the system,
- this encoTRive EtherCAT manual,
- the assembly/installation instructions **TR-EMO-BA-DGB-0015**

### 1.3 References

1.	EN 50325-4	Industrial Communication Systems, based on ISO 11898 (CAN) for Controller Device Interfaces. Part 4: CANopen
2.	ETG.1300	ETG specification "Indicator and Labeling"
3.	CiA DS-301	CANopen communication profile based on CAL
4.	CiA DSP-402, P1 – 5	CANopen Profile for drives
5.	IEC/PAS 62407	Real-time Ethernet control automation technology (EtherCAT); International Electrotechnical Commission
6.	IEC 61158-1 - 6	Digital data communications for measurement and control - Fieldbus for use in industrial control systems - Protocols and Services, Type 12 = EtherCAT
7.	IEC 61784-2	Digital data communications for measurement and control - Additional profiles for ISO/IEC 8802-3 based communication networks in real-time applications, 12 = EtherCAT
8.	IEC 61800-7-200	Adjustable speed electrical power drives systems – Profile specifications
9.	IEC 61800-7-201	Adjustable speed electrical power drives systems – Profile type 1 specification
10.	ISO/IEC 8802-3	Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications
11.	ISO 15745-4 AMD 2	Industrial automation systems and integration - Open systems application integration framework - Part 4: Reference description for Ethernet-based control systems; Amendment 2: Profiles for Modbus TCP, EtherCAT and ETHERNET Powerlink
12.	IEEE 1588-2002	IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems

Table 1: References

## 1.4 Abbreviations used / Terminology

A	Ampere
ASCII	American Standard Code for Information Interchange
CAN	Controller Area Network
CCD	Command Code
CiA	CAN in Automation e.V.
COB	Communication Object
COB-ID	COB Identifier
CPU	Central Processing Unit
DS	Draft Standard
DSP	Draft Standard Proposal
EDS	Electronic Data Sheet
EMCY	Emergency
encoTRive	TR-specific term for the drive
HW	Hardware
inc	Increments
LSB	Least Significant Bit
mA	Milliampere
mm	Millimeter
mNm	Millinewton meter
mV	Millivolt
MSB	Most Significant Bit
Nm	Newtonmeter
NMT	Network Management
OV	Object directory
PC	Personal Computer
PDO	Process Data Object
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
PZD	Process data
ro	read only
RPDO	Receive PDO
rph	Revolutions per hour
rpm	Revolutions per minute
rps	Revolutions per second
RTR	Remote Transmission Request
rw	read/write
SDO	Service Data Object
sec	second
STW	Control word
STW.x	Bit x of the control word
PLC	Programmable Logic Controller

Continuation "Abbreviations used / Terminology"

SW	Software
SYNC	Synchronization
TPDO	Transmit PDO
V	Volt
XML	Extensible Markup Language
wo	write only
ZSW	Status word
ZSW.x	Bit x of the status word

**Table 2: Abbreviations used / Terminology**

---

## 2 Additional safety instructions

### 2.1 Definition of symbols and instructions



means that death or serious injury can occur if the required precautions are not met.

---



means that minor injuries can occur if the required precautions are not met.

---

---

**NOTICE**

means that damage to property can occur if the required precautions are not met.

---



indicates important information's or features and application tips for the product used.

---

### 2.2 Organizational measures

- This encoTRive EtherCAT Manual must be kept ready to hand at all times at the place of use of the encoTRive.
- Prior to commencing work, the personnel working with the encoTRive must have read and understood
  - the Assembly/Installation Instructions, particularly the chapter "**Basic Safety Instructions**",
  - and this encoTRive EtherCAT Manual, particularly the chapter "Additional safety instructions".

This particularly applies for personnel who are only deployed occasionally, e.g. in the parameterization of the encoTRive.

### 3 Technical data

#### 3.1 Current firmware versions

##### MP-060-ETC

**Device name**..... MP-060-ETC  
**Firmware file name** ..... M01907000  
**Motor type** ..... "Dunker BG65x25", 60W, 24V  
**CoE Manufacturer code** ..... 1289 = "TR Electronic GmbH"  
**CoE Device name** ..... "EncoTRive RELC"  
**CoE Product code** ..... 736055  
**CoE Revision number** ..... 0x10001000 = "V1.1"  
**CoE Hardware version** ..... "736055a"  
**CoE Software version** ..... "V1.1 7-May-2019"

##### MP-200-ETC

**Device name**..... MP-200-ETC  
**Firmware file name** ..... M01707000  
**Motor type** ..... "Schneider Electric BDM 754", 300W, 48V  
**CoE Manufacturer code** ..... 1289 = "TR Electronic GmbH"  
**CoE Device name** ..... EncoTRive RELC  
**CoE Product code** ..... 736055  
**CoE Revision number** ..... 0x10001000 = "V1.1"  
**CoE Hardware version** ..... 736055a  
**CoE Software version** ..... V1.1 7-May-2019

#### 3.2 Electrical characteristics

##### Logic supply

24 V version..... +10...+36 VDC  
 60 V version..... +10...+60 VDC  
 Polarity protection..... yes

**Current consumption, logic**..... 300 mA, 3 W

**Power supply** ..... 0...+54 VDC

Polarity protection..... yes

**Current consumption** ..... 4...20 A, 50...300 W; depending of the motor

**Drive**..... 3 phase field oriented control for brushless DC motors

**PWM and regulator cycle**..... 100 µs

**PWM output frequency** ..... 20 kHz due to symmetrical space vector modulation

**PWM power outputs**

3x motor phases ..... 60 V / 21.3 A peak; push-pull  
 1x holding brake ..... 60 V / 2 A; open-drain  
 1x ballast output ..... 60 V / 21.3A peak; open-drain

**CPU** ..... Hilscher NETX100 - 32 Bit ARM 926EJ-S, 200 MHz

**Ethernet** ..... 2 ports with integrated switch, 10/100 MBit with auto-negotiation and auto-crossover

**Digital I/O's**..... - 3 inputs 24 V, 44 kOhm  
 - common ground  
 - not isolated  
 - customization on request

**Analog I/O's**..... - 2 differential inputs  $\pm 10$  V, 10 kOhm,  
 - not isolated  
 - customization on request

### 3.3 EtherCAT / DS-402

**Supported EtherCAT cycle times**..... all down to 50  $\mu$ s

**I/O cycle frequency**.....  $\leq 10$  KHz, depending on the PDO configuration

**PDOs** ..... - 2x transmit with up to 200 bytes total  
 - 2x receive with up to 200 bytes total

**PDO-Mapping** ..... configurable with up to 80 entries each

**Sync manager for PDOs** ..... 1x transmit, 1x receive

**Sync manager assignment tables for PDOs**..... configurable with up to 2 entries each

**Full EtherCAT and DS402 conformance** ..... according to CTT 1.20.52 and 1.20.80

**FoE = Bootstrap mode for firmware updates** .... not supported

**DS-402 operating modes** ..... 1, 3, 6, 8, 9, 10

### 3.4 CE Conformity

#### **Low Voltage Directive 2014/35/EU:**

EN 61010-1:2011 - Overvoltage Category I,  
Power Supply only by SELV/PELV ( $\leq 60\text{Vdc}$ )

#### **EMC Directive 2014/35/EU:**

EN 61800-3:2014 - Emission Category C2: Immunity and emission limits according to the second (industrial) environment

->

EN 61000-4-2: Immunity against ESD (electrostatic discharges).

EN 61000-4-3: Immunity against radiated RF disturbances.

EN 61000-4-4: Immunity against electrical fast transient/burst immunity.

EN 61000-4-5: Immunity against surge.

EN 61000-4-6: Immunity against conducted RF disturbances.

CISPR 11, CISPR 16-1: Limits for conducted and radiated RF emissions.

#### **Tests that are not required by EN61800-3:**

EN 61000-4-29:

Immunity against voltage dips, short interruptions and voltage variations of DC powered devices: Omitted!

Reason: EN 61800-3 requires only tests according to EN 61000-4-11 that are applicable only for AC powered devices.

EN 61000-4-8: Immunity against power frequency magnetic fields: Omitted!

Reason: EN 61800-3 assumes that the magnetic interferences inside the device caused by the usual operation are much stronger than any disturbance that is suggested by the standard, so that this test is superfluous.

## 4 EtherCAT Information

EtherCAT (Ethernet for Control and Automation Technology) is a real-time Ethernet technology and is particularly suitable for communication between control systems and peripheral devices such as e.g. I/O systems, drives, sensors and actuators. EtherCAT was developed in 2003 by Beckhoff Automation GmbH and is available as an open standard. The "EtherCAT Technology Group" (ETG) user association was established for the further development of this technology.

EtherCAT is a publicly accessible specification, which was published by the IEC (IEC/Pas 62407) in 2005 and is part of ISO 15745-4. This part was integrated into the new editions of the international field bus standards IEC 61158 (Protocols and Services), IEC 61784-2 (Communication Profiles) and IEC 61800-7 (Drive Profiles and Communication).

### 4.1 EtherCAT functional principle

The EtherCAT technology overcomes the generally known limitations of other Ethernet solutions:

The Ethernet packet is no longer received in each slave first of all, then interpreted and the process data copied onward. The slave takes the data intended for it, while the frame passes through the device. Input data are likewise inserted into the frame as it passes through. The frames are only delayed by a few nano-seconds. The last slave in the segment sends the now completely processed frame back to the first slave, which returns the frame to the control as a response frame, so to speak. A logical ring structure thus results for the communication. As Fast-Ethernet works with Full Duplex, a physical ring structure also results.

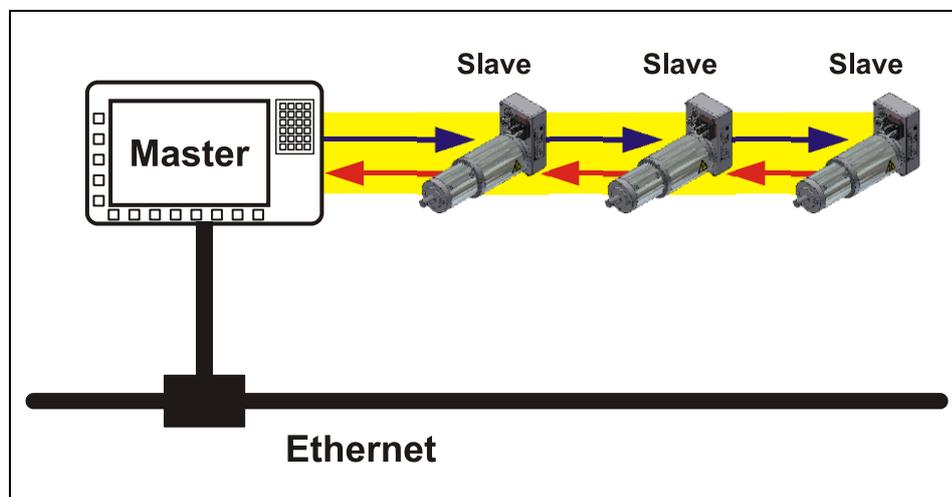


Figure 1: EtherCAT functional principle

## 4.2 Protocol

The EtherCAT protocol, optimized for process data, is transported directly in the Ethernet frame via a special Ether type. A complete transmission can consist of several sub-frames. The data sequence is independent of the physical sequence of the slaves in the network. The addressing can be freely selected:

Broadcast, Multicast and lateral communication between slaves are possible.

The protocol also supports acyclical parameter communication. The structure and meaning of the parameters is predetermined by the device profile **"CANopen Device Profile Drives and Motion Control CiA DSP-402"**.

UDP/IP datagrams are not supported. This means that the master and the EtherCAT slaves must be located in the same subnet. Communication across routers into other subnets is thus not possible.

EtherCAT exclusively uses standard frames in accordance with IEEE802.3 without shortening. EtherCAT frames can thus be sent by any Ethernet controllers (master), and standard tools (e.g. monitor) can be used.

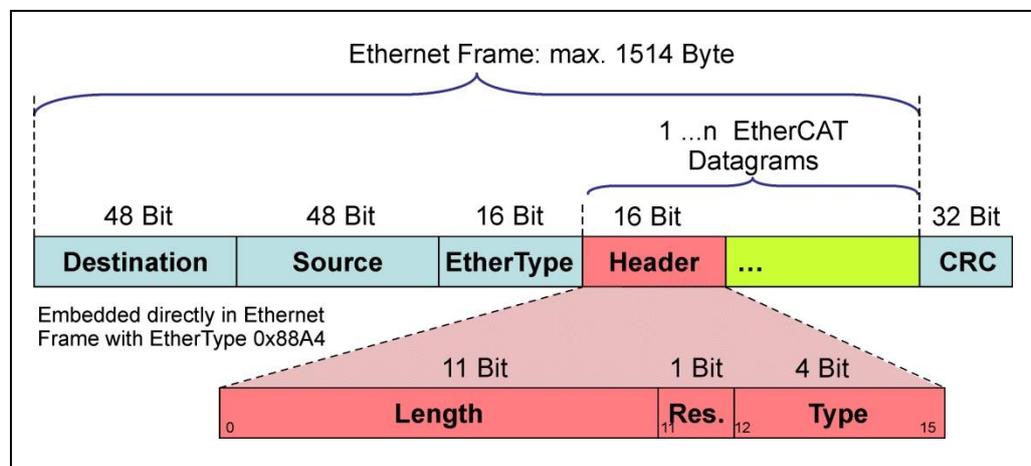


Figure 2: Ethernet frame structure

### 4.3 EtherCAT State Machine (ESM)

The Application Management contains the EtherCAT State Machine, which describes the states and state changes of the slave application. Apart from a few details, the ESM corresponds to the CANopen Network Management (NMT). In order to enable reliable starting behavior the "Safe Operational" state has been introduced in EtherCAT. In this state valid entries are transmitted, while the outputs remain in safe status.

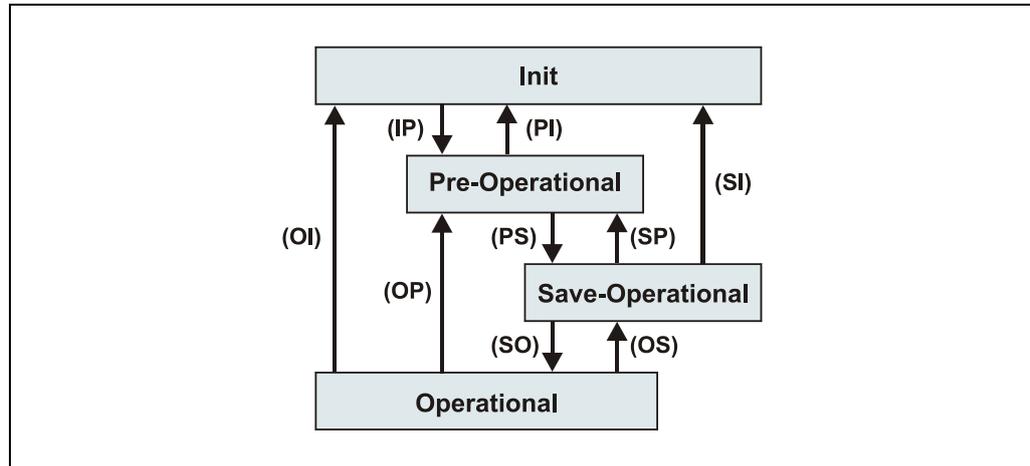


Figure 3: EtherCAT state machine

Status	Description
IP	Start Mailbox Communication
PI	Stop Mailbox Communication
PS	Start Input Update
SP	Stop Input Update
SO	Start Output Update
OS	Stop Output Update
OP	Stop Output Update, Stop Input Update
SI	Stop Input Update, Stop Mailbox Communication
OI	Stop Output Update, Stop Input Update, Stop Mailbox Communication

Table 3: States, State Machine

### 4.3.1 Transition into the OPERATIONAL state

#### **INIT**

The drive is initialized first of all and is then in INIT status. In this status, no direct communication is possible between master and drive via the application layer. The drive can be gradually transferred to OPERATIONAL status according to the state machine via the EtherCAT master:

#### **PRE-OPERATIONAL**

The "Start Mailbox Communication" command puts the drive into PRE-OPERATIONAL status. In this status only the mailbox is active first of all, and master and drive exchange application-specific initializations and parameters. In PRE-OPERATIONAL status only a parameterization via Service Data Objects is initially possible. However, it is possible to configure PDOs using SDOs.

#### **SAFE-OPERATIONAL**

The "Start Input Update" command puts the drive into SAFE-OPERATIONAL status. In this status the drive provides valid current input data, without changing the output data. The outputs are in safe status.

#### **OPERATIONAL**

The "Start Output Update" command puts the drive into OPERATIONAL status. In this status the drive provides valid input data and the master provides valid current output data. When the drive has detected the data received via the process data service, the status transition is confirmed by the drive. If activation of the output data was not possible, the drive remains in SAFE-OPERATIONAL status and outputs an error message.

## 4.4 Device profile

The device profile describes the application parameters and the functional behavior of the device, including the device class-specific state machine. With EtherCAT you do not develop individual device profiles for device classes. Instead, simple interfaces are provided for existing device profiles:

The encoTRive supports the **CANopen-over-EtherCAT** (CoE) mailbox protocol, and consequently the **"Device Profile Drives and Motion Control"**, CiA DS 402, known from CANopen.

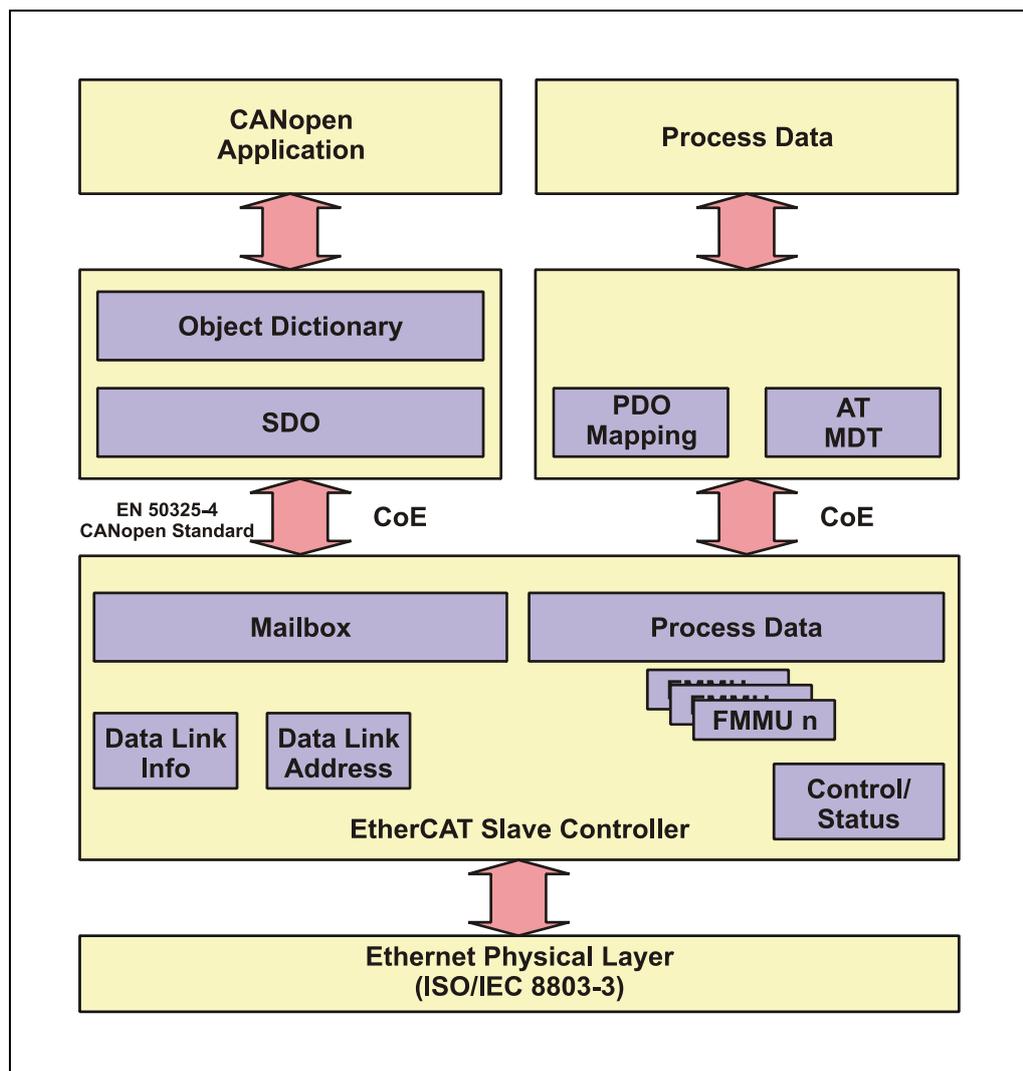


Figure 4: CANopen over EtherCAT communication mechanism

### 4.4.1 CANopen over EtherCAT (CoE)

EtherCAT can provide the same communication mechanisms as those known from <sup>1</sup>CANopen:

- Object dictionary
- PDO, Process Data Objects
- SDO, Service Data Objects
- NMT, Network Management

EtherCAT can thus be implemented on devices that were previously equipped with CANopen, with minimal expense. Extensive parts of the CANopen firmware can be re-used. The objects can be optionally extended.

Comparison of CANopen / EtherCAT in the ISO/OSI layer model

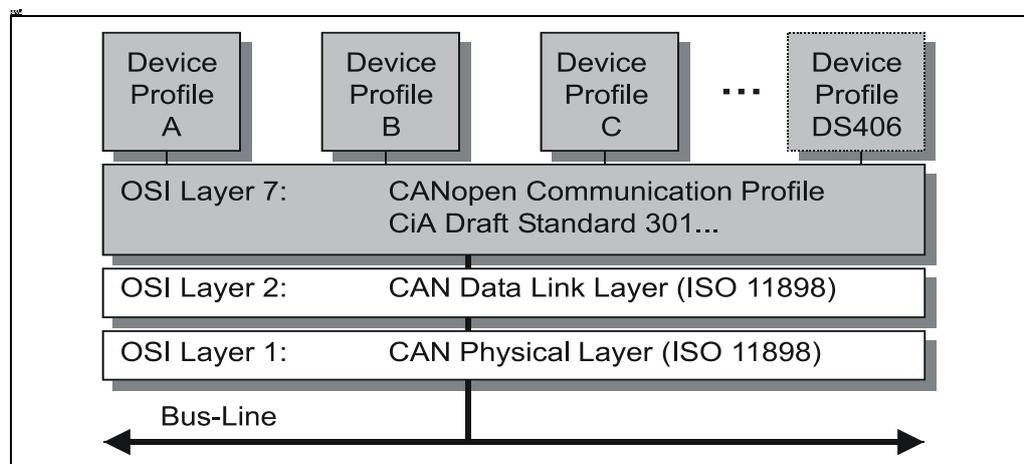


Figure 5: CANopen organized in the ISO/OSI layer model

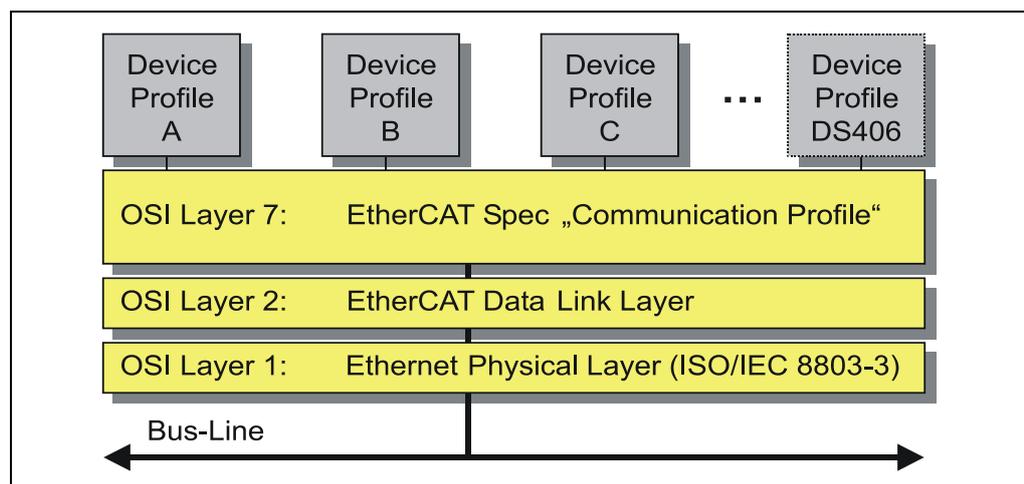


Figure 6: EtherCAT organized in the ISO/OSI layer model

<sup>1</sup> EN 50325-4: Industrial Communication Systems, based on ISO 11898 (CAN) for Controller Device Interfaces. Part 4: CANopen.

## 4.5 Object dictionary

The object dictionary structures the data of an EtherCAT device in a clear tabular arrangement. It contains all device parameters and all current process data, which are therefore also accessible via the SDO.

Index (hex)	Object
0x0000–0x0FFF	Data type definitions
0x1000–0x1FFF	CoE communication profile range (CiA DS-301)
0x2000–0x5FFF	Manufacturer-specific profile range
0x6000–0x9FFF	Device profile range (CiA DS-402)
0xA000–0xFFFF	Reserved

Figure 7: Structure of the object dictionary

## 4.6 Process and Service Data Objects

### Process Data Object (PDO)

Process Data Objects manage the process data exchange, e.g. the cyclical transmission of the control- and status word, position value and target position etc.

### Service Data Object (SDO)

Service Data Objects manage the parameter data exchange, e.g. the acyclical parameter setting of non-time-critical parameters such as software limit switches, direction of rotation etc.

The SDO provides an efficient communication mechanism for parameter data of any size. A service data channel for parameter communication is formed between the configuration master and the connected devices for this purpose. The device parameters can be written to or read from the device object dictionary with a unique frame handshake.

### Important features of SDO and PDO

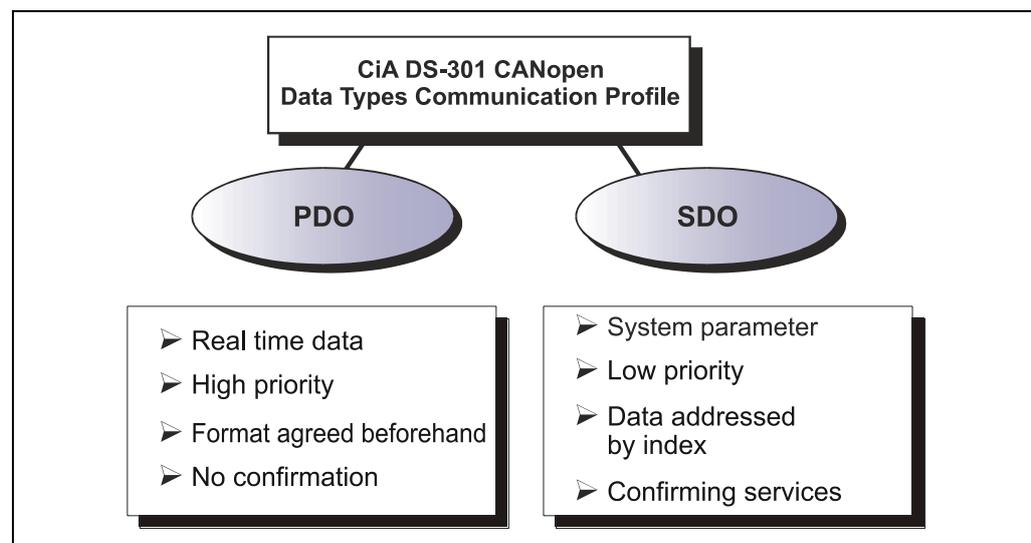


Figure 8: Comparison of PDO/SDO characteristics

#### 4.6.1 Compatibility with the CiA DS-301 communication profile

##### Supported services

- Initiate SDO Download
- Download SDO Segment
- Initiate SDO Upload
- Upload SDO Segment
- Abort SDO Transfer

## 4.7 Transmission of SDO messages

The entries of the object dictionary can be read or written with the SDO services. The SDO Transport Protocol allows the transmission of objects of any size. The EtherCAT SDO Protocol is equivalent to the CANopen SDO Protocol, in order to guarantee the re-use of existing protocol stacks.

The first byte of the first segment contains the necessary control information. The next three bytes of the first segment contain the index and sub-index of the object dictionary entries to be read or written. The last four bytes of the first segment are available for useful data. The second and following segments contain the control byte and useful data. The recipient confirms each segment or a block of segments, so that Peer-To-Peer communication (client/server) takes place.

In CAN-compatible mode the SDO protocol comprises 8 bytes, in order to correspond to the CAN data size. In extended mode the useful data are simply extended, without changing the protocol header. In this way the increased data volume of the EtherCAT mailbox is adapted to the SDO protocol, accelerating the transmission of large data volumes accordingly.

In addition, the mode "Complete Access" has been added which makes it possible to transmit the complete data of an index from the object dictionary in a single process. The data of all sub-indices are subsequently transmitted. The access is controlled via the "Initiate SDO Download / Upload" – write/read services, which also are supported by the drive. In the ESI file this is indicated with the mailbox attribute entry `CompleteAccess="1"`. If older EtherCAT master do not support this access, this entry must be set to "0".

Services with confirmation (Initiate SDO Upload, Initiate SDO Download, Download SDO Segment, and Upload SDO Segment) and services without confirmation (Abort SDO Transfer) are used for the execution of Segmented/Expedited transmission of Service Data Objects.

The so-called SDO Client (master) specifies in its "Request" the parameter, the access type (read/write) and the value if applicable. The so-called SDO Server (slave or drive) executes the write or read access and answers the request with a "Response". In the case of error, an abort code (Abort SDO Transfer) provides information on the cause of the error.



Normally the EtherCAT master provides appropriate mechanisms for the SDO transfer. Knowledge of the protocol structure and internal sequences is therefore not required.

This also applies to read out the complete CoE object directory. The drive supports the so-called "SDO Information Service". In online mode about this service it is possible to read out all supported objects in one read service.

---

### Write services, Client --> Server

- **Initiate SDO Download**  
The *Initiate SDO Download* service is used for an individual transmission of data, if the number of bytes can be accepted by the mailbox, or if a segmented transmission is to be started with more bytes.
- **Download SDO Segment**  
The *SDO Download Segment* service is used to transfer the additional data that could not be transferred with the *Initiate SDO Download* service. The master starts as many Download SDO Segment services as are required to transfer all data to the server.

### Read services, Server --> Client

- **Initiate SDO Upload**  
The *Initiate SDO Upload* service is used for an individual transmission of data, if the number of bytes can be accepted by the mailbox, or if a segmented transmission is to be started with more bytes. The server responds with the result of the upload request and the required data, in the event of successful execution.
- **Upload SDO Segment**  
The *SDO Upload Segment* service is used to transfer the additional data that could not be transferred with the *Initiate SDO Upload* service response. The server starts as many Upload SDO Segment services as are required to transfer all data from the server.

## 4.8 PDO mapping

PDO mapping refers to the mapping of application objects (real-time data, e.g. object 6064h "Actual Position" from the object dictionary into Process Data Objects, e.g. Object 1A00h (1st Transmit PDO).

The current mapping can be read via corresponding entries in the object dictionary, the so-called mapping tables. The number of mapped objects that are listed subsequently is found at the top of the mapping table (subindex 0). The tables are located in the object dictionary in index 0x1600 ff. for the RxPDOs and 0x1A00ff for the TxPDOs.

## 4.9 Further information

Further information on EtherCAT can be obtained on request from the ***EtherCAT Technology Group*** (ETG) at the following address:

---

ETG Headquarter  
Ostendstraße 196  
90482 Nuremberg  
Germany  
Phone: + 49 (0) 9 11 / 5 40 5620  
Fax: + 49 (0) 9 11 / 5 40 5629  
Email: [info@ethercat.org](mailto:info@ethercat.org)  
Internet: [www.ethercat.org](http://www.ethercat.org)

---

## 5 DSP 402 drive profile

The linguistic devices for controlling the drive are extensively independent of the manufacturer. For this reason, communication between the drive and the superimposed control system has been standardized in so-called **drive profiles**.

A **drive profile** specifies how an electrical drive is controlled via a field bus. It defines the behavior of the device and the method of accessing the drive data. The following sub areas in particular are controlled:

- Control and status monitoring
- Standardized parameterization
- Changing operating modes

### **encoTRive supports as EtherCAT slave the DSP 402 profile (CiA [2002b])**

The following information is typically exchanged between the control system and the drive:

The drive provides information on its current status (e.g. "*Drive running*") and possibly additional information such as the current position, current speed etc. In the other direction, the control system assigns positioning orders, for example, ("*Move at speed x to position y*"). Without the DSP 402 profile, each manufacturer would have to specify its own protocols for the transmission of commands and status messages, and there would consequently be many applications performing the same task in different ways.

## 5.1 DSP 402 state machine

If the drive is in **OPERATIONAL** status, drive-specific functions can be addressed. Figure 9 provides an overview of the states (Sx) defined in the DSP 402 drive profile and the transitions between these states (Tx). The current status is shown by the status word 0x6041. In Figure 9 the states are represented by rectangles. The state transitions are marked by arrows, which specify the condition for the corresponding transition. In most cases this is a command in the control word 0x6040. In an error situation, you go from each of the states into the state “S7: Fault reaction active” (Error reaction active).

S1-S8: Operation states (**S**tates)  
 T0-T16: State transitions (**T**ransitions)

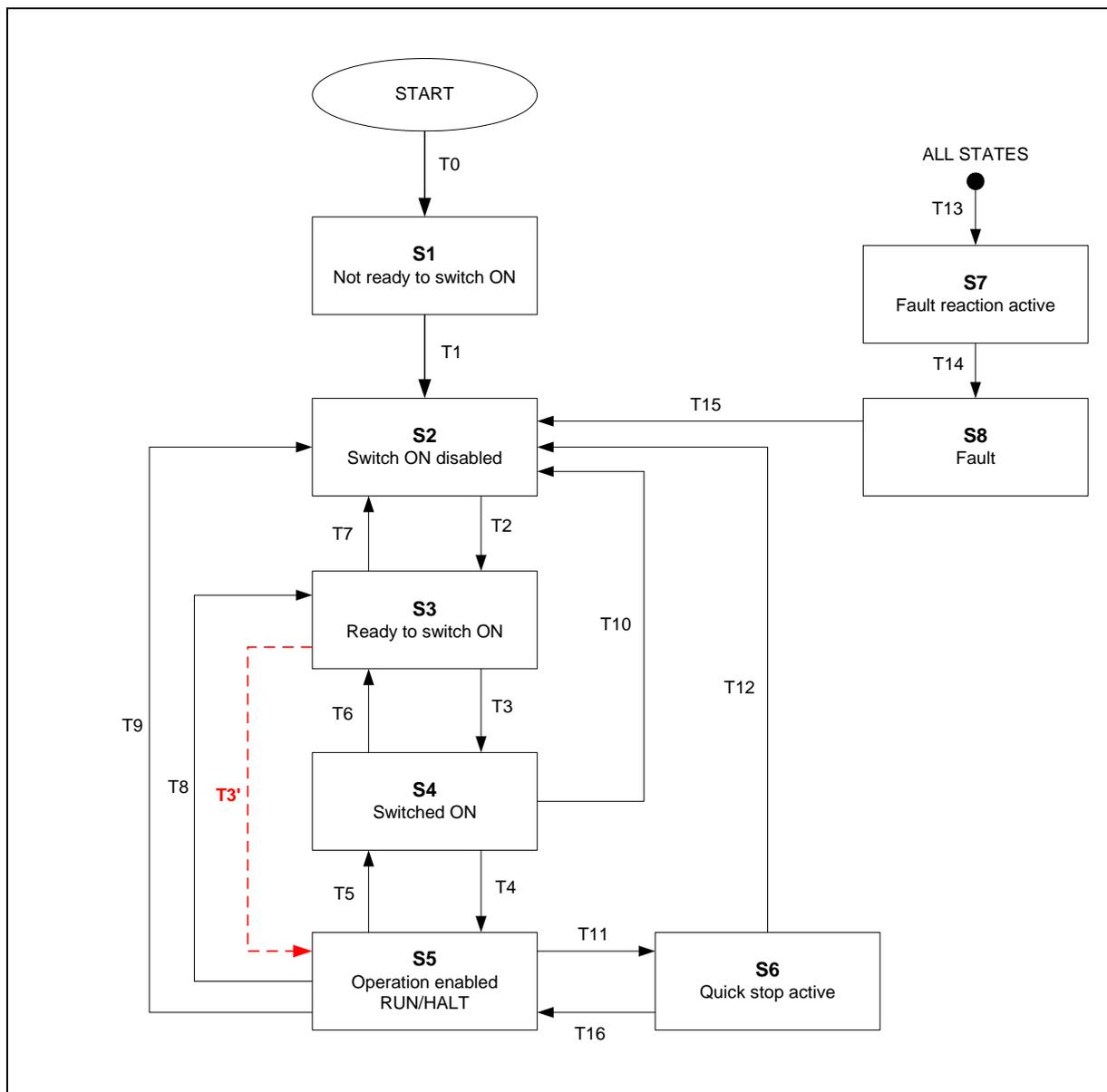


Figure 9: DSP 402 state machine

### 5.1.1 Operating condition

Status	Status word, 0x6041	Description
<b>S1:</b> Not ready to switch ON	xxxx xxxx x0xx 0000	The drive is in the initialization phase.
<b>S2:</b> Switch ON disabled	xxxx xxxx x1xx 0000	Drive initialization ended. The output stage cannot be switched on.
<b>S3:</b> Ready to switch ON	xxxx xxxx x01x 0001	The output stage can be switched on. Drive parameters can be changed.
<b>S4:</b> Switched ON	xxxx xxxx x01x 0011	Output stage switched on.
<b>S5:</b> Operation enabled	xxxx xxxx x01x 0111	The drive can be used in accordance with the set operating mode: see as from chapter "DSP 402 drive profile" page 30
<b>S6:</b> Quick stop active	xxxx xxxx x00x 0111	The drive performs a quick stop. If the Quick stop option code object 0x605A = 5, the drive then remains in this state.
<b>S7:</b> Fault reaction active	xxxx xxxx x0xx 1111	There is an error. The relevant troubleshooting is carried out.
<b>S8:</b> Fault	xxxx xxxx x0xx 1000	Troubleshooting is complete.

Table 4: States of the status word

x: not relevant bits

The operating condition is shown by means of appropriate bit pattern in the status word 0x6041. The bits in the status word are defined as follows:

Bit	Description
0	Ready to switch ON
1	Switched ON
2	Operation enabled
3	Fault
4	Voltage enabled
5	Quick stop
6	Switch ON disabled
7	Warning
8	Not used
9	Remote
10	Target reached
11	Internal limit active
12-13	Specific to the mode
14-15	Not used

Description of the mode of operation-specific bits, see following chapters:

- "Profile Position (PP)" operating mode, p 35
- "Profile Velocity (PV)" operating mode, p 40
- "Homing (HM)" operating mode, p 44
- "Cyclic Synchronous Position (CSP)" operating mode, p 47
- "Cyclic Synchronous Velocity (CSV)" operating mode, p 50
- "Cyclic Synchronous Torque (CST)" operating mode, p 53

Table 5: Operating states, status word

## 5.1.2 State transitions

The transition between these states occurs partially via internal events, partially via commands that are transferred in the control word (0x6040):

Transition	Description
T0: START -> NOT READY TO SWITCH ON	The drive performs an initialization.
T1: NOT READY TO SWITCH ON -> SWITCH ON DISABLED	Communication is activated.
T2: SWITCH ON DISABLED -> READY TO SWITCH ON	Drive receives the command "Shut down".
T3: READY TO SWITCH ON -> SWITCHED ON	The output stage is released.
T4: SWITCHED ON -> OPERATION ENABLE	Drive receives the command "Enable Operation".
T5: OPERATION ENABLE -> SWITCHED ON	Drive receives the command "Disable Operation".
T6: SWITCHED ON -> READY TO SWITCH ON	The drive unit is deactivated.
T7: READY TO SWITCH ON -> SWITCH ON DISABLED	Drive receives the command "Quick Stop"
T8: OPERATION ENABLE -> READY TO SWITCH ON	The drive unit is switched off. If it is in motion, it is stopped according to the <code>Shutdown Option Code (object 0x605B)</code> .
T9: OPERATION ENABLE -> SWITCH ON DISABLED	The drive unit is switched off. If it is in motion, it is stopped as quickly as possible and then deactivated.
T10: SWITCHED ON -> SWITCH ON DISABLED	The drive unit is switched off.
T11: OPERATION ENABLE -> QUICK STOP ACTIVE	The drive unit is stopped according to object <code>0x605A (Quick Stop Option Code)</code> .
T12: QUICK STOP ACTIVE -> SWITCH ON DISABLED	The transition is possible if the value $\neq 5$ is stored in object <code>0x605A (Quick Stop Option Code)</code> . The drive unit is then stopped as quickly as possible and then switched off.
T13: ALL STATES -> FAULT REACTION ACTIVE	The relevant troubleshooting is carried out.

Continuation: State transitions

T14: FAULT REACTION ACTIVE -> FAULT	Troubleshooting is complete.
T15: FAULT -> SWITCH ON DISABLED	The error was acknowledged with a rising edge in bit 7 of the status word.
T16: QUICK STOP ACTIVE -> OPERATION ENABLE	The drive receives the command "Enable Operation". The transition is possible if the value 5 or 6 is stored in object 0x605A (Quick Stop Option Code).

Table 6: State transitions

### 5.1.3 Change mode of operation

The operating condition is controlled via appropriate bit pattern in the control word 0x6040. The bits in the control word are defined as follows:

MSB				LSB			
15-9	8	7	6-4	3	2	1	0
reserved	Halt	<b>Fault Reset</b>	Mode specific	<b>Enable operation</b>	<b>Quick stop</b>	<b>Enable voltage</b>	<b>Switch ON</b>

Command	State transitions	Change of state to	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0
Shutdown	T2, T6, T8	S3: Ready to switch ON	X	X	1	1	0
Switch ON	T3	S4: Switched ON, <i>is not necessary urgently and can be skipped</i>	X	0	1	1	1
Disable Voltage	T7, T9, T10, T12	S2: Switch ON disabled	X	X	X	0	X
Quick Stop	T7, T10, T11	S2: Switch ON disabled S6: Quick stop active	X	X	0	1	X
Disable operation	T5	S4: Switched ON	X	0	1	1	1
Enable operation	T4, T16	S5: Operation enabled	X	1	1	1	1
Fault reset	T15	S2: Switch ON disabled	0->1	X	X	X	X

Table 7: Change operating state

X: not relevant bits

Description of the mode of operation-specific bits, see following chapters 5.2 up to 5.7.

## 5.2 “Profile Position (PP)” operating mode

### 5.2.1 Control word

Bit	CONTROL WORD		
0	Value 1: Switch ready for operation	These bits are used to control the state machine	
1	Value 1: Enable voltage		
2	Value 0: Decelerate with intermediate stop ramp		
3	Value 1: Execute operating mode		
4	New target position	0	Do not accept target position
		1	Accept target position
5	Accept parameter immediately	0	End current positioning, start next positioning
		1	Abort current positioning, start next positioning
6	Absolute / relative positioning	0	Target position is an absolute value
		1	Target position is a relative value
7	Transition 0->1: Reset error		
8	Stop	0	Execute positioning
		1	Stop axis
9-15	not used		

**Table 8: Control word (object 0x6040), “Profile Position” operating mode**

**5.2.2 Status word**

Bit	STATUS WORD		
0	Value 1: Ready to switch on		
1	Value 1: Ready for operation		
2	Value 1: Operating mode activated		
3	Value 1: Error present		
4	Value 1: Voltage switched on		
5	Value 0: Quick stop active		
6	Value 1: Not ready for operation		
7	Value 1: Warning present		
8	not used		
9	Manual operation	0	No parameter access via CAN bus
		1	Parameter access via CAN bus
10	Target reached	0	Stop = 0: Target position not reached
			Stop = 1 Axis decelerates
		1	Stop = 0: Target position reached
			Stop = 1 Axis speed = 0
11	Value 1: Exit working range Position value outside the preset range. Definition via object 0x607D.		
12	Acknowledgement of target position	0	Positioning values not accepted
		1	Positioning values accepted
13	Value 1: Following error A following error is indicated if the difference between position value and calculated ramp value is greater than the following error window preset in object 0x6065 for the minimum duration of the time interval specified in object 0x6066 (Following Error Timeout).		
14-15	not used		

**Table 9: Status word (object 0x6041), "Profile Position" operating mode**

### 5.2.3 Perform positioning

If the drive is in the "Ready for operation" state, positioning movements can be performed.

For this purpose the "Profile Position (PP)" operating mode must be set and the operating mode must then be executed with the control word:

- Object 0x6060 operating mode = 1

A positioning ramp is started by a falling edge (1->0) in STW.8. The positioning movement is carried out on a **ramp**, which is derived from the current settings for the speed, acceleration and deceleration:

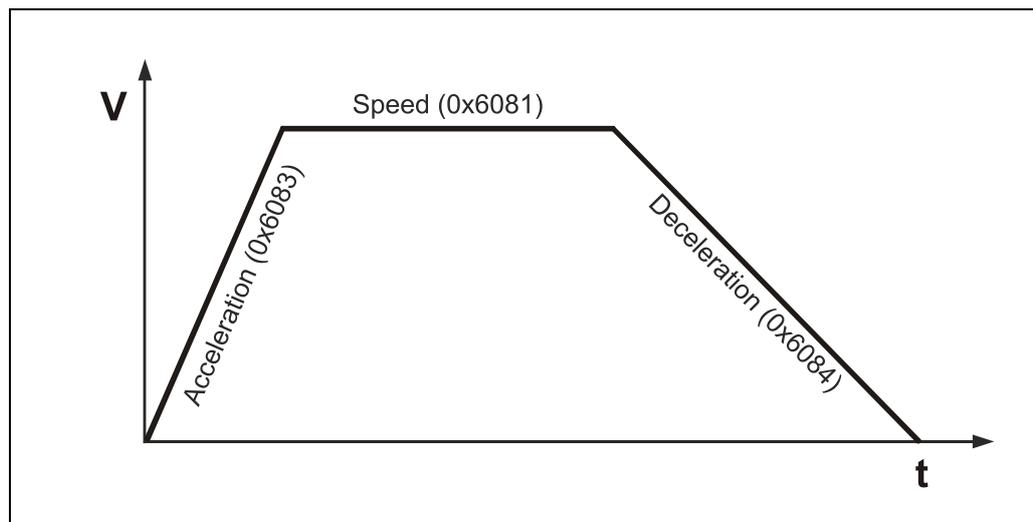


Figure 10: Positioning ramp

Depending on the distance between actual position at the start of positioning and target position, the required end speed (0x6081) is reached or not. If the travel is short, the phase with constant acceleration is directly followed by a phase with constant deceleration. There is no phase with constant speed in this case.

The conclusion of a positioning movement is indicated by ZSW.10. If this bit has the value 1, the positioning is complete. The applicable criterion for the end of a positioning movement is that the current position value lies within the position window (0x6067) around the target position for the duration of the position window time interval (0x6068).

### 5.2.3.1 Absolute / relative positioning

A positioning movement can be performed **absolutely** or **relatively**. This is differentiated by means of STW.6. If this bit is set, a relative positioning movement is performed, otherwise an absolute positioning movement is performed.

In the case of **absolute positioning**, the value for the target position (0x607A) is interpreted absolutely, i.e. the distance to be covered is the difference between actual and target position. In the case of relative positioning, the content of object 0x607A is interpreted as the distance to be covered.



Relative positionings must be within the travel range. The limits are defined by means of the objects 0x607B Position range limit and 0x607D Software position limit.

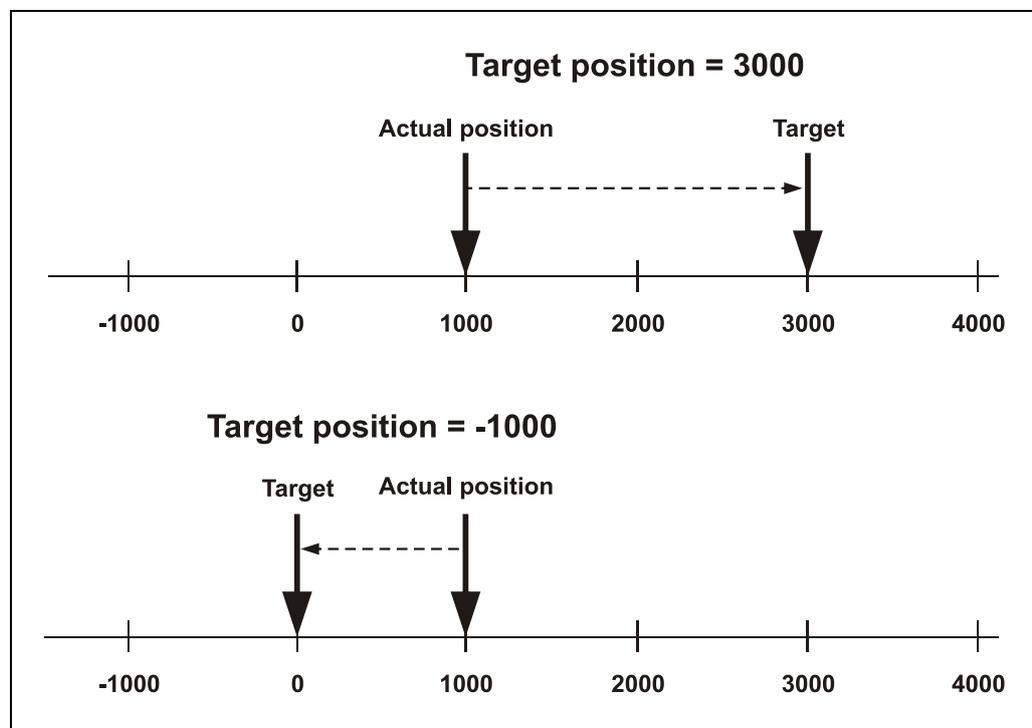


Figure 11: Absolute positioning (top) and relative positioning (bottom)

Figure 11 shows two examples:

In the top part of the figure, an absolute positioning movement is performed to the target position 3000. The drive changes its position until it reaches position 3000. A relative positioning movement is performed in the bottom part: Starting from the actual position 1000, a positioning movement by 1000 to the left (target position = -1000) is to be performed. This means that the positioning movement ends at the absolute position 0.

### 5.2.3.2 Transfer of new movement records

A new movement record (target position, speed, acceleration, deceleration) can be transferred to the drive while the previous positioning movement is still being performed. The movement record is activated with a positive edge in STW.4. Depending on STW.5 ("Accept parameter immediately"), the new movement record is acknowledged (ZSW.12 = 1) and executed immediately, or only after the current positioning movement is complete:

If STW.5 = 1, the new movement record is accepted "on-the-fly". The current positioning movement is not completed. If STW.5 = 0, the current positioning movement is performed first. The new positioning movement is then acknowledged (ZSW.12 = 1) and executed; whether or not the drive must stop depends on the current circumstances:

If the new target position requires travel in the other direction, or if the current speed is too high, the drive is stopped and a new ramp is started in the opposite direction. Otherwise, the current speed is compared with the required speed. If the travel is sufficient, the drive is accelerated or braked accordingly.

### 5.2.3.3 Termination of a positioning movement

If STW.8 ("Stop") is set during the movement, the current ramp is terminated and the drive stops. The deceleration at which this occurs can be specified in object 0x605D (Halt Option Code). Depending on the value of this parameter, deceleration occurs with the ramp for the quick stop or with the currently set deceleration. In order to then continue, STW.8 must be deleted and a new positioning movement started.

### 5.2.3.4 Relevant parameters

Index	Meaning
0x6040	Control word: Commands to the drive
0x6041	Status word: Return messages from the drive
0x6064	Current position
0x6067	Position window
0x6068	Position window time interval
0x607A	Target position
0x607D	Position range
0x607F	Maximum speed
0x6081	Speed (limited by object 0x607F)
0x6083	Acceleration (limited by object 0x60C5)
0x6084	Deceleration (limited by object 0x60C6)
0x60C5	Maximum acceleration
0x60C6	Maximum deceleration
0x605A	Quick Stop Option Code: Quick stop behavior
0x605D	Halt Option Code: Behavior in the event of termination of a ramp

**Table 10: Positioning parameter**

### 5.3 “Profile Velocity (PV)” operating mode

***There is a risk of physical injury and damage to property if the parameterized software limit switches in object 0x607D subindex 1 and 2 are exceeded!***

**⚠ WARNING**

**NOTICE**

- The parameterized software limit switches in object 0x607D, which refer to the position actual position value, are inoperative in "Profile Velocity" operating mode.

Range overruns can occur in rotary applications, for example, due to the integral position measuring system. Depending on the direction of rotation, this is expressed by a jump of the position value in object 0x6064:  
 Max --> Min / Min --> Max.

**The application must therefore not be dependent on the actual value of position!**

#### 5.3.1 Control word

Bit	CONTROL WORD		
0	Value1: Switch ready for operation	These bits are used to control the state machine	
1	Value1: Enable voltage		
2	Value0: Decelerate with intermediate stop ramp		
3	Value1: Execute operating mode		
4-6	not used		
7	Transition 0->1: Reset error		
8	Stop	0	Execute speed ramp
		1	Stop axis
9-15	not used		

**Table 11: Control word (object 0x6040), “Profile velocity” operating mode**

### 5.3.2 Status word

Bit	STATUS WORD		
0	Value 1: Ready to switch on		
1	Value 1: Ready for operation		
2	Value 1: Operating mode activated		
3	Value 1: Error present		
4	Value 1: Voltage switched on		
5	Value 0: Quick stop active		
6	Value 1: Not ready for operation		
7	Value 1: Warning present		
8	not used		
9	Manual operation	0	No parameter access via CAN bus
		1	Parameter access via CAN bus
10	Target reached	0	Stop = 0: Target speed not reached
			Stop = 1 Axis decelerates
		1	Stop = 0: Target speed reached
			Stop = 1 Axis speed = 0
11	Value 1: Exit working range Position value outside the preset range. Definition via object 0x607D.		
12	Speed	0	Speed ≠ 0
		1	Speed = 0
13	Max. following error	0	Max. following error not reached
		1	Max. following error reached
14-15	not used		

**Table 12: Status word (object 0x6041), "Profile velocity" operating mode**

### 5.3.3 Perform speed ramp

When the drive is in "Ready for operation" state, its movement can be speed-controlled.

For this purpose the "Profile Velocity (PV)" operating mode must be set and the operating mode must then be executed with the control word:

- Object 0x6060 operating mode = 3

The speed ramp is started by a falling edge (1->0) in STW.8.

Alternatively, the speed ramp can also be started as follows:

STW.1, STW.2: 0x06

--> STW.0: 0x07 is not necessary urgently and can be skipped

--> STW.3: 0x0F Execute operating mode, drive rotates

The **ramp** results from the current settings for the target speed, acceleration and deceleration:

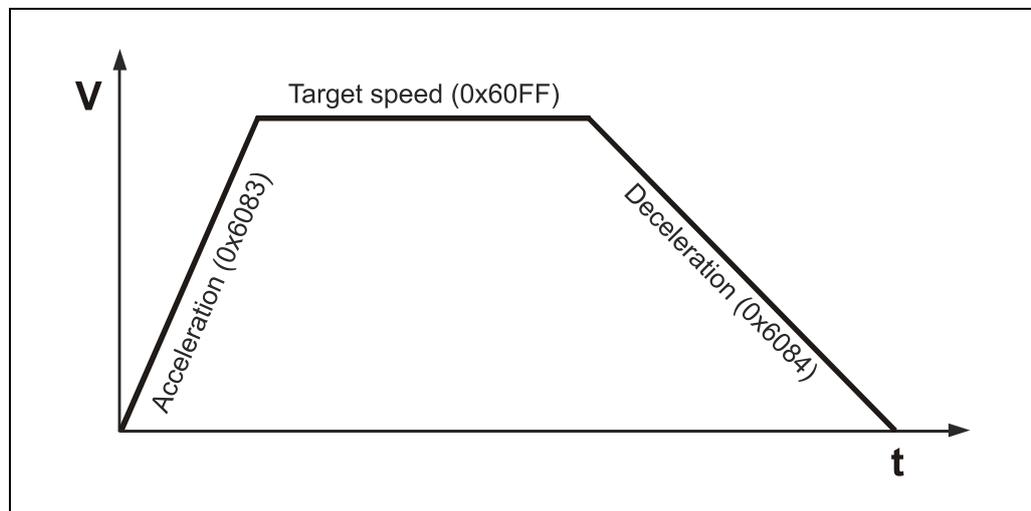


Figure 12: Speed ramp

The drive accelerates up to the preset target speed and retains this speed, until the drive is stopped manually.

The axis can be stopped using different methods:

- Set target speed in object 0x60FF to 0
- Stop axis with control word bit 8 = 0->1
- Execute NMT service "Stop Remote Node" command 0x02
- Reset STW.2: 0x07, end execution of operating mode

A change to the target speed during operation results in immediate execution of the ramp, which accelerates or decelerates the current speed value.

### 5.3.4 Relevant parameters

Index	Meaning
0x6040	Control word: Commands to the drive
0x6041	Status word: Return messages from the drive
0x6069	Measured speed
0x6071	Target torque
0x6072	Maximum torque
0x607E	Direction reversal
0x607F	Maximum speed
0x6080	Maximum motor speed
0x6083	Acceleration (limited by object 0x60C5)
0x6084	Deceleration (limited by object 0x60C6)
0x6085	Deceleration for quick stop
0x6094	Speed factor
0x606B	Nominal speed
0x606C	Actual speed
0x60FF	Target speed

**Table 13: Speed ramp parameter**

## 5.4 “Homing (HM)” operating mode

### 5.4.1 Control word

Bit	CONTROL WORD		
0	Value 1: Switch ready for operation	These bits are used to control the state machine	
1	Value 1: Enable voltage		
2	Value 0: Decelerate with intermediate stop ramp		
3	Value 1: Execute operating mode		
4	Homing operation start	0	Stop homing procedure
		1	Start or continue homing procedure
5-6	not used		
7	Transition 0->1: Reset error		
8	Halt	0	Enable bit 4
		1	Stop axis (0x605D)
9-15	not used		

Table 14: Control word (object 0x6040), “Homing” operating mode

## 5.4.2 Status word

Bit	STATUS WORD		
0	Value 1: Ready to switch on		
1	Value 1: Ready for operation		
2	Value 1: Operating mode activated		
3	Value 1: Error present		
4	Value 1: Voltage switched on		
5	Value 0: Quick stop active		
6	Value 1: Not ready for operation		
7	Value 1: Warning present		
8	not used		
9	Manual operation	0	No parameter access via CAN bus
		1	Parameter access via CAN bus
10	Target reached		
11	Value 1: Exit working range Position value outside the preset range. Definition via object 0x607D.		
12	Homing attained		
13	Homing error		
14-15	not used		

Table 15: Status word (object 0x6041), "Homing" operating mode

Bit 13	Bit 12	Bit 10	Definition
0	0	0	Homing procedure is in progress
0	0	1	Homing procedure is interrupted or not started
0	1	0	Homing is attained, but target is not reached
0	1	1	Homing procedure is completed successfully
1	0	0	F Homing error occurred, velocity is not 0
1	0	1	Homing error occurred, velocity is 0
1	1	x	not used

Table 16: Definition of the homing status bits 10, 12, 13

### 5.4.3 Perform position adjustment

Only the homing methods 17 to 30 and 35 are available, because the other methods require an index pulse from an incremental encoder, and the encoTRive devices are always equipped with absolute encoders.

If there are no limit switches, the homing method 35 must be used. Only for this method the drive may be turned off.

The drive must stand at a known position, which is the preset value. The scaling for the position measurement must be setup properly, so that the preset value is within the measuring range.

If the drive is in the "Ready for operation" state, the homing procedure can be started.

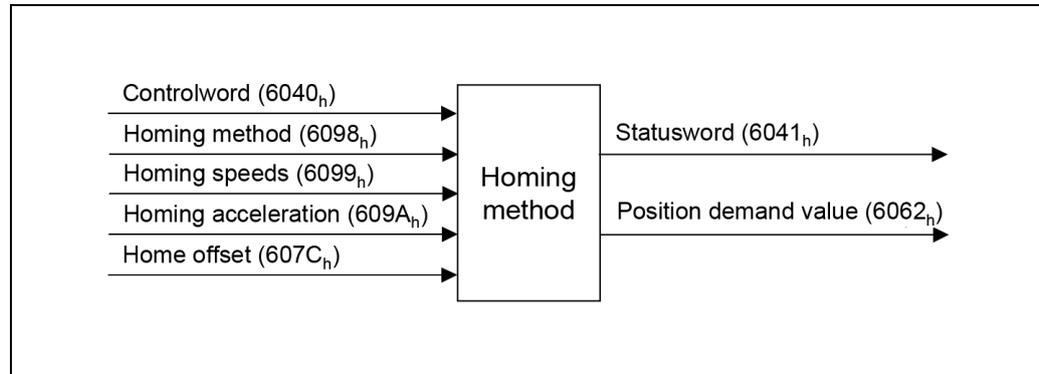
For this purpose the "Homing" operating mode must be set, the homing method must be specified and the operating mode must then be executed with the control word:

- Object 0x6060 operating mode = 6
- Object 0x6098 homing method = 35, as example

The homing procedure is started with a rising edge (0->1) in STW.4.

Also see chapter "Position adjustment by Homing (operating mode 6)" on page 138.

### 5.4.4 Relevant parameters



## 5.5 “Cyclic Synchronous Position (CSP)” operating mode

In this mode, the drive receives the position target at fixed time intervals (cycles) from the control. In this case, the drive no longer calculates ramps but only transfers the target values directly to the positioning control.

In this operating mode the limit switches on the digital inputs  $2^0$  and  $2^1$  are ignored.

The target position is transferred cyclically via PDO by means of the object 0x607A, to which the drive responds immediately. Bit 4 in the control word does not have to be set.

The target value is absolute and thus independent of how often it was sent per cycle.

### 5.5.1 Control word

Bit	CONTROL WORD		
0	Value 1: Switch ready for operation	These bits are used to control the state machine	
1	Value 1: Enable voltage		
2	Value 0: <b>Shutdown without stop ramp</b>		
3	Value 1: Execute operating mode		
4-6	not used		
7	Transition 0->1: Reset error		
8	Halt	0	<b>is not evaluated!</b>
		1	<b>is not evaluated!</b>
9-15	not used		

Table 17: Control word (object 0x6040), “Cyclic synchronous position (CSP)” operating mode

### 5.5.2 Status word

Bit	STATUS WORD		
0	Value 1: Ready to switch on		
1	Value 1: Ready for operation		
2	Value 1: Operating mode activated		
3	Value 1: Error present		
4	Value 1: Voltage switched on		
5	Value 0: Quick stop active		
6	Value 1: Not ready for operation		
7	Value 1: Warning present		
8	not used		
9	Manual operation	0	No parameter access via CAN bus
		1	Parameter access via CAN bus
10	not used		
11	Value 1: Exit working range Position value outside the preset range. Definition via object 0x607D.		
12	Drive follows the command value	0	Target position of object 0x607A ignored
		1	Target position of object 0x607A used as input to position control loop
13	Max. following errors	0	Max. following errors not reached
		1	Max. following errors reached
14-15	not used		

**Table 18: Status word (object 0x6041), "Cyclic synchronous position (CSP)" operating mode**

### 5.5.3 Perform operating mode

If the drive is in the "Ready for operation" state, positioning movements can be performed. Via the status word (object 0x6041, bit 12: Drive follows the command value) is displayed that the target values are transferred directly to the controller.

For this purpose the "Cyclic Synchronous Position (CSP)" operating mode must be set:

- Object 0x6060 operating mode = 8

To perform a positioning, object 0x607A must be specified with the target values. Also see chapter "Cyclic synchronous modes (operating modes 8, 9, 10)" on page 137.

### 5.5.4 Relevant parameters

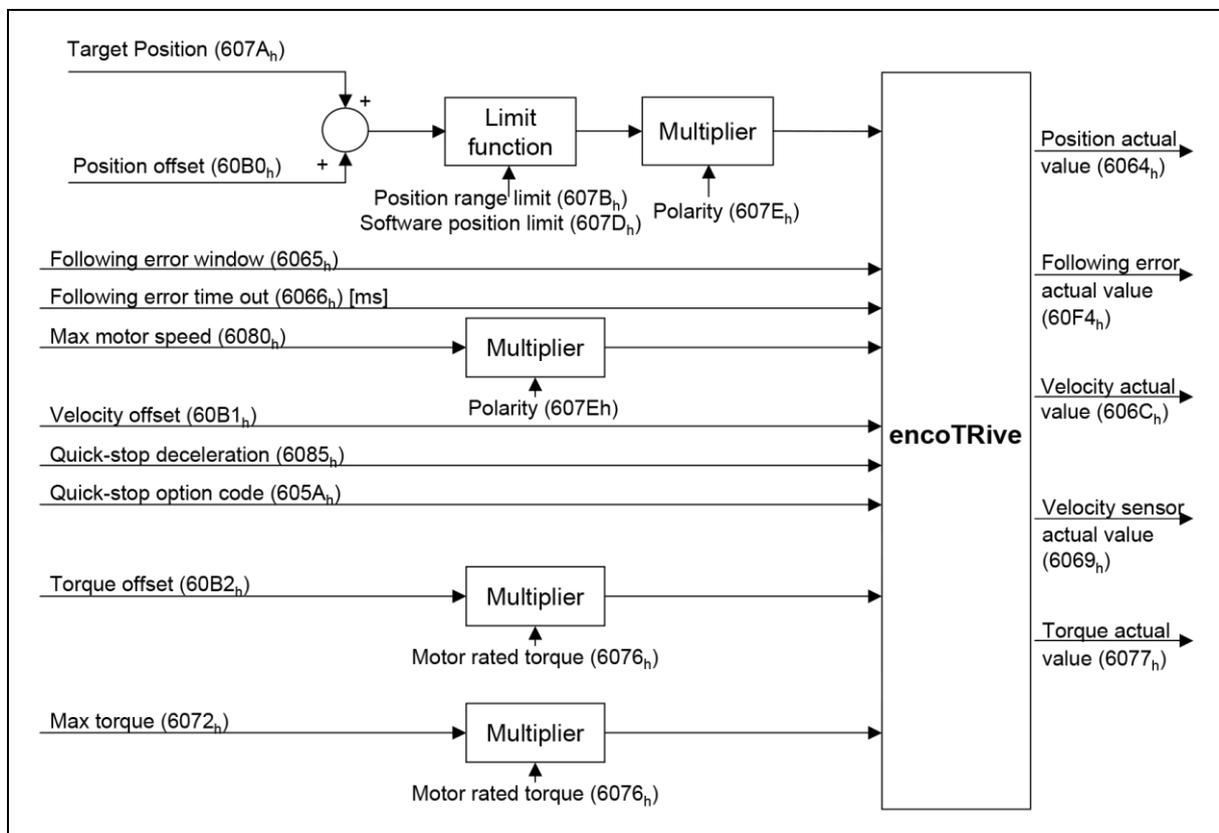


Figure 13: Cyclic synchronous position control function (CSP)

## 5.6 “Cyclic Synchronous Velocity (CSV)” operating mode

In this mode, the drive receives the velocity target at fixed time intervals (cycles) from the control. In this case, the drive no longer calculates ramps but only transfers the target values directly to the velocity controller.

In this operating mode the limit switches on the digital inputs 2<sup>0</sup> and 2<sup>1</sup> are ignored.

The target velocity is transferred cyclically via PDO by means of the object 0x60FF, to which the drive responds immediately. Bit 4 in the control word does not have to be set.

### 5.6.1 Control word

Bit	CONTROL WORD		
0	Value 1: Switch ready for operation	These bits are used to control the state machine	
1	Value 1: Enable voltage		
2	Value 0: <b>Shutdown without stop ramp</b>		
3	Value 1: Execute operating mode		
4-6	not used		
7	Transition 0->1: Reset error		
8	Halt	0	<b>is not evaluated!</b>
		1	<b>is not evaluated!</b>
9-15	not used		

**Table 19: Control word (object 0x6040), “Cyclic synchronous velocity (CSV)” operating mode**

## 5.6.2 Status word

Bit	STATUS WORD		
0	Value 1: Ready to switch on		
1	Value 1: Ready for operation		
2	Value 1: Operating mode activated		
3	Value 1: Error present		
4	Value 1: Voltage switched on		
5	Value 0: Quick stop active		
6	Value 1: Not ready for operation		
7	Value 1: Warning present		
8	not used		
9	Manual operation	0	No parameter access via CAN bus
		1	Parameter access via CAN bus
10	not used		
11	Value 1: Exit working range Position value outside the preset range. Definition via object 0x607D.		
12	Drive follows the command value	0	Target velocity of object 0x60FF ignored
		1	Target velocity of object 0x60FF used as input to velocity control loop
13-15	not used		

**Table 20: Status word (object 0x6041), “Cyclic synchronous velocity (CSV)” operating mode**

### 5.6.3 Perform operating mode

If the drive is in the "Ready for operation" state, it can be traveled velocity controlled. Via the status word (object 0x6041, bit 12: Drive follows the command value) is displayed that the target values are transferred directly to the controller.

For this purpose the "Cyclic Synchronous Velocity (CSV)" operating mode must be set:

- Object 0x6060 operating mode = 9

To perform a velocity controlled travel, object 0x60FF must be specified with the target values.

Also see chapter "Cyclic synchronous modes (operating modes 8, 9, 10)" on page 137.

### 5.6.4 Relevant parameters

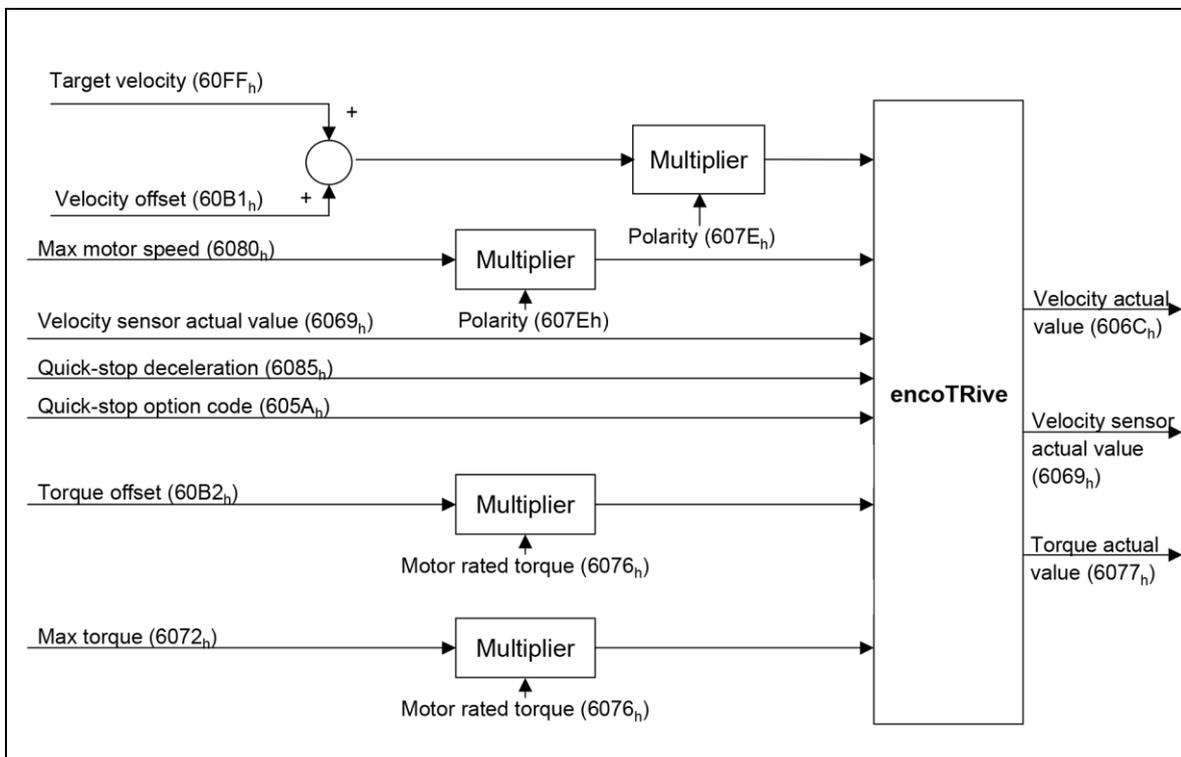


Figure 14: Cyclic synchronous velocity control function (CSV)

## 5.7 “Cyclic Synchronous Torque (CST)” operating mode

In this mode, the drive receives the torque target at fixed time intervals (cycles) from the control. In this case, the drive no longer calculates ramps but only transfers the target values directly to the torque or current controller.

In this operating mode the limit switches on the digital inputs 2<sup>0</sup> and 2<sup>1</sup> are ignored.

The target torque is transferred cyclically via PDO by means of the object 0x6071, to which the drive responds immediately. Bit 4 in the control word does not have to be set.

### 5.7.1 Control word

Bit	CONTROL WORD		
0	Value 1: Switch ready for operation	These bits are used to control the state machine	
1	Value 1: Enable voltage		
2	Value 0: <b>Shutdown without stop ramp</b>		
3	Value 1: Execute operating mode		
4-6	not used		
7	Transition 0->1: Reset error		
8	Halt	0	<b>is not evaluated!</b>
		1	<b>is not evaluated!</b>
9-15	not used		

**Table 21: Control word (object 0x6040), “Cyclic synchronous torque (CST)” operating mode**

### 5.7.2 Status word

Bit	STATUS WORD		
0	Value 1: Ready to switch on		
1	Value 1: Ready for operation		
2	Value 1: Operating mode activated		
3	Value 1: Error present		
4	Value 1: Voltage switched on		
5	Value 0: Quick stop active		
6	Value 1: Not ready for operation		
7	Value 1: Warning present		
8	not used		
9	Manual operation	0	No parameter access via CAN bus
		1	Parameter access via CAN bus
10	not used		
11	Value 1: Exit working range Position value outside the preset range. Definition via object 0x607D.		
12	Drive follows the command value	0	Target torque of object 0x6071 ignored
		1	Target torque of object 0x6071 used as input to torque control loop
13-15	not used		

Table 22: Status word (object 0x6041), "Cyclic synchronous torque (CST)" operating mode

### 5.7.3 Perform operating mode

If the drive is in the "Ready for operation" state, it can be traveled torque controlled. Via the status word (object 0x6041, bit 12: Drive follows the command value) is displayed that the target values are transferred directly to the controller.

For this purpose the "Cyclic Synchronous Torque (CST)" operating mode must be set:

- Object 0x6060 operating mode = 10

To perform a torque controlled travel, object 0x6071 must be specified with the target values.

Also see chapter "Cyclic synchronous modes (operating modes 8, 9, 10)" on page 137.

### 5.7.4 Relevant parameters

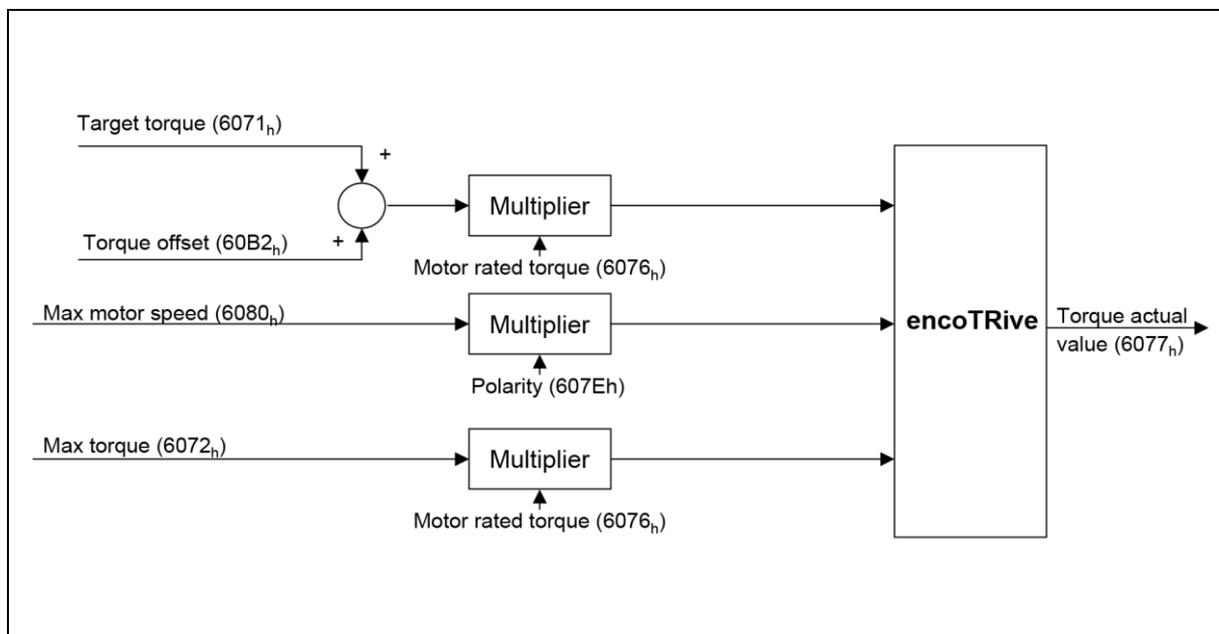


Figure 15: Cyclic synchronous torque control function (CST)

## 5.8 Units

For the conversion of different units, the DSP 402 profile provides the so-called **Factor group**. From this group, encoTRive implements the following partial quantity:

### 5.8.1 Object 0x608F: Position encoder resolution

This parameter defines the ratio between position increments and motor revolutions:

$$\text{Positionencoderresolution} = \frac{\text{Positionincrements}}{\text{Motor revolutions}}$$

Formula 1: Position encoder resolution

The fraction counter is defined in subindex 1, the denominator in subindex 2. The default value is

$$\text{Positionencoderresolution}_{\text{Default}} = \frac{4096}{1} = 4096 \left[ \frac{\text{Positionincrements}}{\text{Motor revolutions}} \right]$$

Formula 2: Default value, position encoder resolution

### 5.8.2 Object 0x6090: Speed encoder resolution

This parameter defines the ratio between speed increments per second and motor revolutions per second:

$$\text{Speedencoderresolution} = \frac{\frac{\text{Speed increments}}{\text{sec}}}{\frac{\text{Motor revolutions}}{\text{sec}}}$$

Formula 3: Speed encoder resolution

The fraction counter is defined by subindex 1, the denominator by subindex 2.

The default value is

$$\text{Speedencoderresolution}_{\text{Default}} = \frac{2^{31}}{5000} \left[ \frac{\text{Speedincrements}}{\text{Motor revolutions}} \right]$$

Formula 4: Default value, speed encoder resolution

### 5.8.3 Object 0x6093: Position factor

The position factor converts the desired position into increments, which the drive requires for the internal calculations. By multiplying with the position factor, position data can be converted into **user-defined position units**. Position units are length units, angles or increments.

$$\text{Position factor} = \frac{\text{Position encoder resolution} \bullet \text{Gear ratio}}{\text{Travel per gear revolution}}$$

Formula 5: Position factor

The position encoder resolution is defined by object 0x608F. The **gear ratio** denotes the gear reduction, given by

$$\text{Gear ratio} = \frac{\text{Motor revolutions}}{\text{Drive shaft revolutions}}$$

Formula 6: Gear ratio

The **travel per gear revolution** is given in position units.

The **counter** of the position factor is specified as subindex 1 of object 0x6093, the **denominator** as subindex 2.

The following applies:

$$\text{Position [position units]} \bullet \text{Position factor} = \text{Position [position increments]}$$

The unit of the position factor is position increments/position units

The position unit is therefore implicitly defined by the position factor.

The default value of object [0x6093] for the position factor is

$$\text{Position factor}_{\text{Default}} = \frac{1024}{1024}$$

Formula 7: Default value, position factor

This value, together with the encoder resolution from Formula 2, corresponds to a gear reduction of 1:1 and the position unit increments.

**Example 1:**

- Position unit = [mm]
- Pitch (travel per gear revolution) = 3 mm
- Gear ratio = 1
- Position encoder resolution = 1024 inc

The following values must be entered in the position factor object [**0x6093**] :

Subindex 1: 1024 / position encoder resolution x gear factor  
Subindex 2: 3 / pitch (travel per gear revolution)

**Example 2:**

- Position unit = [inc]
- Travel per gear revolution = 1024 inc
- Gear ratio = 1
- Position encoder resolution = 1024 inc

The following values must be entered in the position factor object [**0x6093**] :

Subindex 1: 1024 / position encoder resolution x gear factor  
Subindex 2: 1024 / travel per gear revolution

**Example 3:**

- Position unit = [degrees /10]
- Travel per gear revolution = 360 degrees = 3600/10 degrees
- Gear ratio = 1
- Position encoder resolution = 1024 inc

The following values must be entered in the position factor object [**0x6093**] :

Subindex 1: 1024 / position encoder resolution x gear factor  
Subindex 2: 3600 / travel per gear revolution ( 360 degrees )

### 5.8.4 Object 0x6094: Speed factor

The speed factor converts the desired speed into increments, which the drive requires for the internal calculations. By multiplying with the speed factor, speed data can be converted into user-defined **speed units**.

$$\text{Speed factor} = \frac{\text{Speed encoder resolution} \bullet \text{Gear reduction}}{\text{Speed units\_per} \frac{\text{Output revolution}}{\text{sec}}}$$

**Formula 8: Speed factor**

The **counter** of the speed factor is specified in subindex 1 of object 0x6094, the **denominator** as subindex 2.

The default value is:

$$\text{Speed factor}_{\text{Default}} = \frac{2^{31}}{300000}$$

**Formula 9: Default value, speed factor**

This value, together with the speed encoder resolution from Formula 4 and a gear reduction of 1:1, corresponds to the speed unit "rpm".

Object 0x606C (Actual Velocity) contains the current output speed in the desired unit. Default unit "rpm".

This means, for example, that if a value of 4350 is preset in object 0x6081 (Profile Velocity), an output speed of 4350 rpm will be reached.

Object 0x607F (Max. Profile Velocity) defines the maximum permitted speed.

The desired unit is now used in all speed-related objects, with the exception of object 0x6069 (Sensor Velocity), which specifies the current speed in speed increments.

**Example 1: Default**

- Speed unit = r/min (at the output)
- Output revolution = 1 r/ sec = 60 r/min
- Gear ratio = 1
- Default value for speed encoder resolution,  $2^{31} / 5000$ , see Formula 4

Calculation:  $(2^{31} / 5000) \times 1 / 60 = 2^{31} / [(5000 \times 60) / 1]$

The following values must be entered in the speed factor object [0x6094] :

Subindex 1:  $2^{31}$

Subindex 2:  $\frac{5000 \cdot 60}{1} = 300000$

**Example 2:**

- Speed unit = (degrees/10) /sec (at the output)
- Output revolution = 1 r/sec = 3600 (degrees /10<sup>1</sup>) / sec
- Gear ratio = 1
- Default value for speed encoder resolution,  $2^{31} / 5000$ , see Formula 4

Calculation:  $(2^{31} / 5000) \times 1 / 3600 = 2^{31} / [(5000 \times 3600) / 1]$

The following values must be entered in the speed factor object [0x6094] :

Subindex 1:  $2^{31}$

Subindex 2:  $\frac{5000 \cdot 3600}{1} = 18000000$

**Example 3:**

- Speed unit = (mm/100) /sec (at the output)
- Travel per gear revolution = 2 mm / object (0x6093)
- Output revolution = 1 r/sec = 2 (mm x 100<sup>1</sup>) /sec
- Gear ratio = 1
- Default value for speed encoder resolution,  $2^{31} / 5000$ , see Formula 4

Calculation:  $(2^{31} / 5000) \times 1 / 200 = 2^{31} / [(5000 \times 200) / 1]$

The following values must be entered in the speed factor object [0x6094] :

Subindex 1:  $2^{31}$

Subindex 2:  $\frac{5000 \cdot 200}{1} = 1000000$

<sup>1</sup> Resolution factor for higher resolution. Result: the possibility to exact value input.

### 5.8.5 Object 0x6097: Acceleration factor

The acceleration factor converts the acceleration into increment/s, which the drive requires for the internal calculations. By multiplying with the acceleration factor, acceleration data is converted into user-defined **acceleration units**.

The **counter** of the acceleration factor is specified as subindex 1 of object 0x6097, the **denominator** as subindex 2.

$$\text{Acceleration factor} = \frac{\text{Speed encoder resolution} \cdot \text{Gear reduction}}{\text{Acceleration units\_per} \frac{\text{Output revolution}}{\text{sec}^2}}$$

**Formula 10: Acceleration factor**

The default value is:

$$\text{Acceleration factor}_{\text{Default}} = \frac{2^{31}}{300000}$$

**Formula 11: Default for acceleration factor**

This value, together with the speed encoder resolution from Formula 4 and a gear reduction of 1:1, corresponds to the acceleration unit "(rpm) / sec".

**Example 1: Default**

- Acceleration unit = (r/min) /sec (at the output)
- Output revolution = 1 r/ sec = 60 (r/min) /sec
- Gear ratio = 1
- Default value for speed encoder resolution,  $2^{31} / 5000$ , see Formula 4

Calculation:  $(2^{31} / 5000) \times 1 / 60 = 2^{31} / [(5000 \times 60) / 1]$

The following values must be entered in the speed factor object [0x6097] :

Subindex 1:  $2^{31}$

Subindex 2:  $\frac{5000 \cdot 60}{1} = 300000$

**Example 2:**

- Acceleration unit = (degrees/10) /sec<sup>2</sup> (at the output)
- Output revolution = 1 r/sec = 3600 (degrees /10<sup>1</sup>) / sec<sup>2</sup>
- Gear ratio = 1
- Default value for speed encoder resolution,  $2^{31} / 5000$ , see Formula 4

Calculation:  $(2^{31} / 5000) \times 1 / 3600 = 2^{31} / [(5000 \times 3600) / 1]$

The following values must be entered in the speed factor object [0x6097] :

Subindex 1:  $2^{31}$

Subindex 2:  $\frac{5000 \cdot 3600}{1} = 18000000$

**Example 3:**

- Acceleration unit = (mm/100) /sec<sup>2</sup> (at the output)
- Travel per gear revolution = 2 mm / object (0x6093)
- Output revolution = <sup>1</sup> r/sec = 2 (mm x 100<sup>1</sup>) /sec
- Gear ratio = 1
- Default value for speed encoder resolution,  $2^{31} / 5000$ , see Formula 4

Calculation:  $(2^{31} / 5000) \times 1 / 200 = 2^{31} / [(5000 \times 200) / 1]$

The following values must be entered in the speed factor object [0x6097] :

Subindex 1:  $2^{31}$

Subindex 2:  $\frac{5000 \cdot 200}{1} = 1000000$

<sup>1</sup> Resolution factor for higher resolution. Result: the possibility to exact value input.

### Conversion into position increments

With the above factors it is possible to convert any user-defined units

- for travel,
- speed and acceleration into position increments,
- speed increments/sec and speed increments/sec<sup>2</sup>

The following factors can be used to express the latter units in position increments again:

$$\text{VelocityToPositionUnitFactor} = \frac{\text{Speed factor} \cdot \text{Positionencoder resolution}}{\text{Positionfactor} \cdot \text{Speed encoder resolution}}$$
$$\text{AccelerationToPositionUnitFactor} = \frac{\text{Acceleration factor} \cdot \text{Positionencoder resolution}}{\text{Positionfactor} \cdot \text{Speed encoder resolution}}$$

By multiplying with the **VelocityToPositionUnitFactor**, a user-defined speed in speed units is converted into a measuring system-related speed in position increments/sec. Similarly, user-defined acceleration data in acceleration units is converted into measuring system-related acceleration data in position increments/sec<sup>2</sup> by multiplying with the **AccelerationToPositionUnitFactor**.

**Examples:**

In the following examples, the default values for speed encoder resolution and position encoder resolution are used as the basis.

**1. Ramp calculation:**

- Gear reduction = 1:1
- Covered distance  $s = 700000$  degrees
- Acceleration from standstill = 150 rpm/sec
- Further acceleration from 150 rpm/sec to 2640 rpm, then constant
- Deceleration = 150 rpm/sec, until stationary

**How long does the ramp processing take, and which sections are covered ?**

The speed factor results from Formula 9.

The acceleration phase and the braking phase each last

$$t_{\text{acc}} = t_{\text{dec}} = 2640/150 = 17.60 \text{ sec.}$$

$$\begin{aligned} \text{VelocityToPositionUnitFactor} &= [(2^{31}/300000) * 2^{16}] / [(2^{16}/360) * (2^{31}/5000)] \\ &= 5000 * 360 / 300000 = 6, \end{aligned}$$

$$\text{AccelerationToPositionUnitFactor} = 6.$$

The speed of 2640 rpm is as follows in position units:

$$v_{\text{end}} = 2640 * 6 = 15840 \text{ degrees/sec.}$$

The acceleration and deceleration of 150 rpm/sec results in

$$a = b = 150 * 6 = 900 \text{ degrees/sec}^2,$$

During the acceleration phase, the distance covered is

$$s_a = 0.5 * a * t_{\text{acc}}^2 = 0.5 * 900 * 17.6^2 = 139392 \text{ degrees.}$$

The deceleration distance is likewise

$$s_b = t_{\text{dec}} * v_{\text{end}} - 0.5 * b * t_{\text{dec}}^2 = 17.6 * 15840 - 0.5 * 900 * 17.6^2 = 139392 \text{ degrees.}$$

Consequently, for travel at constant speed  $v_{\text{end}}$

$$s - (s_a + s_b) = 700000 - 2 * 139392 = 421216 \text{ degrees still remain.}$$

This phase therefore lasts  $421216/15840 = 26.6 \text{ sec.}$

In total, the ramp is (theoretically) traveled in  $2 * 17.6 + 26.6 = 61.8 \text{ sec.}$

## 2. Ramp calculation:

- Drive with slide, powered by a screw link actuator (translatory movement)
- Feed = 1 mm/spindle revolution
- Gear reduction = 40:1 = 40 motor revolutions per spindle revolution
- Slide movement  $s = 1000$  mm
- Acceleration from standstill = 20 rpm/sec
- Further acceleration from 20 rpm/sec to 240 rpm, then constant
- Deceleration = 60 rpm/sec, until stationary

Position factor =  $2^{16} \cdot 40$

1 mm feed =  $2^{16} \cdot 40$  position increments.

The following results for the speed factor:

$$\text{speed factor} = 40 \cdot 2^{31} / 300000.$$

This results in

$$\begin{aligned} \text{VelocityToPositionUnitFactor} &= [(40 \cdot 2^{31} / 300000) \cdot 2^{16}] / [(40 \cdot 2^{16}) \cdot (2^{31} / 5000)] \\ &= 5000 / 300000 = 1/60 \end{aligned}$$

$$\text{AccelerationToPositionUnitFactor} = 1/60$$

The acceleration phase lasts  $t_{\text{acc}} = 12$  sec, the braking phase  $t_{\text{dec}} = 4$  sec.

The following results for the acceleration:

$$a = 20 \cdot (1/60) = 1/3 \text{ mm/sec}^2,$$

and for the deceleration

$$b = 60 \cdot (1/60) = 1 \text{ mm/sec}^2,$$

The end speed of 240 rpm corresponds to

$$v_{\text{end}} = 240 \cdot (1/60) = 4 \text{ mm/sec.}$$

During the acceleration phase, the distance covered is

$$s_a = 0.5 \cdot a \cdot t_{\text{acc}}^2 = 0.5 \cdot (1/3) \cdot 12^2 = 24 \text{ mm}$$

The deceleration distance is likewise

$$s_b = t_{\text{dec}} \cdot v_{\text{end}} - 0.5 \cdot b \cdot t_{\text{dec}}^2 = 4 \cdot 4 - 0.5 \cdot 1 \cdot 4^2 = 8 \text{ mm.}$$

Therefore, for travel at constant speed  $v_{\text{end}}$

$$s - (s_a + s_b) = 1000 - 28 = 968 \text{ mm still remain.}$$

This phase therefore lasts  $968/4 = 242$  sec.

In total, the ramp is (theoretically) traveled in 258 sec.

## 6 Installation- / connection notes, pin assignments

EtherCAT supports linear, tree or star structures. The bus or linear structure used in the field buses is thus also available for Ethernet. This is particularly practical for system wiring, as a combination of line and stubs is possible.

For transmission according to the 100Base-TX Fast Ethernet standard, patch cables in category STP CAT5 must be used (2 x 2 shielded twisted pair copper wire cables).

Recommendation:

As Ethernet cables green readily assembled PROFINET cables with double shielding and D-coded M12x1 connectors are recommended.

The cables are designed for bit rates of up to 100 Mbit/s. The transmission speed is automatically detected by the drive and does not have to be set by means of a switch.

Addressing by switch is also not necessary; this is done automatically using the addressing options of the EtherCAT master.

The cable length between two subscribers may be max. 100 m, a total of 65535 subscribers are possible in the EtherCAT network.

In some cases the pin assignment of the power plug provides no separate logic supply, but instead a 10A open drain output for a ballast resistor that must be attached to the positive power supply.

The resistance must be chosen according to the power supply voltage:  
Ballast resistance = Power supply voltage / 10A

The pins for power + and – are reversed compared to a power cable for a traditional encoTRive. This prevents the destruction of the ballast output, if by accident a traditional cable is plugged in. Then the protection against reversed polarity blocks any supply current and nothing at all will happen.

Currently there is no plug for the digital and analog I/Os. Other pin assignments are available on request.

If the power line is not directly connected to a supply, but is instead turned on by a switch or a relay, for a short moment very high inrush currents of several 100 A may occur. Depending on the length and the positions of the cables, these inrush currents may create strong electromagnetic interferences.

Then it is strongly recommended, to use power cables with twisted pairs for + and – and a shield, that is connected to earth on both ends of the cable. Additionally the largest possible snap ferrite core should be attached on the power cable close to the encoTRive device.

Recommended parts: STAR-TEC Snap Ferrite 74271221 or 74271222 from Würth.

**Download pin assignment, version with “Valve Male Connector”:**

[www.tr-electronic.de/f/TR-EMO-TI-DGB-0109](http://www.tr-electronic.de/f/TR-EMO-TI-DGB-0109)

**Download pin assignment, version with “M23 Male Connector”:**

[www.tr-electronic.de/f/TR-EMO-TI-DGB-0112](http://www.tr-electronic.de/f/TR-EMO-TI-DGB-0112)



---

*In order to ensure safe, fault-free operation,*

- *ISO/IEC 11801, EN 50173 (European standard)*
- *ISO/IEC 8802-3*
- *and other pertinent standards and directives must be complied with!*

*In particular, the applicable EMC directive and the shielding and grounding directives must be observed!*

---

## 7 Optional holding brake / ballast circuit

If the motor is equipped with a holding brake, the brake output voltage 0x2F04:05 must be set according to its rating. Otherwise it may be overloaded or may not open.

If a ballast resistor is attached to the open drain ballast output, the limit of the ballast output voltage 0x2F04:06 must be set according to its rating. Otherwise the resistor may be overloaded or become ineffective.

The effective ballast output voltage is regulated by PWM. If the DC link voltage 0x6079 exceeds the ballast input voltage threshold 0x2F04:07, it increases linearly and reaches the rated value 0x2F04:05, as soon as the excess DC link voltage is equal to the full ballast input voltage 0x2F04:08.

## 8 Commissioning

### 8.1 Device description file

The XML file contains all information on the drive-specific parameters and the operating modes of the drive. The XML file is integrated by the EtherCAT network configuration tool, in order to enable correct configuration and commissioning of the drive.

**Download:**

- [www.tr-electronic.de/f/TR-EMO-ID-MUL-0018](http://www.tr-electronic.de/f/TR-EMO-ID-MUL-0018)

## 9 Object directory

### 9.1 Object types, data types

A parameter in the object directory can be a simple value, an array or a data structure. encoTRive uses the following types, which are distinguished by the **object code**:

Object code	Name	Meaning
7	VAR	Simple value, e.g. INTEGER8
8	ARRAY	Array from several elements of the same data type
9	RECORD	Data field, which is a combination of various simple data types

Table 23: Object codes in encoTRive

In the case of an ARRAY or RECORD parameter, the individual elements are accessed via the subindex. For simple values (VAR), the subindex is 0.

A parameter or an element of a parameter also has attributes which define the access to this parameter:

Attribute	Meaning
rw	read/write: Parameter can be read and written
ro	read only: Parameter can only be read
wo	write only: Parameter can only be written
const	Value is constant and read only.

Table 24: Attributes

encoTRive uses the following data types:

Coding	Data type	Length	Description
1	BOOLEAN	8 bit	Two possible values: 0 (false) or 1 (true)
2	INTEGER8	8 bit	Integer 8-bit value with sign. Range of values: -128 ... 127
3	INTEGER16	16 bit	Integer 16-bit value with sign. Range of values: -32768 ... 32767
4	INTEGER32	32 bit	Integer 32-bit value with sign. Range of values: $-2^{31} \dots 2^{31}-1$
5	UNSIGNED8	8 bit	Integer 8-bit value without sign. Range of values: 0...255
6	UNSIGNED16	16 bit	Integer 16-bit value without sign. Range of values: $0..2^{16}-1$ (0-65535)
7	UNSIGNED32	32 bit	Integer 32-bit value without sign. Range of values: $0..2^{32}-1$
9	Visible String	Variable	ASCII character string

Table 25: encoTRive data types

## 9.2 Communication-Specific Standard Objects CiA DS-301

The following table shows an overview of the supported indexes in the communication profile range (0x1000 – 0x1999):

Index (h)	Object	Name	Data type	Attr.	Page
1000	VAR	Device Type	Unsigned32	ro	70
1001	VAR	Error Register	Unsigned8	ro	70
1003	ARRAY	Pre-defined Error Field	Unsigned8	rw	71
1008	VAR	Manufacturer Device Name	String	const	72
1009	VAR	Manufacturer Hardware Version	String	const	72
100A	VAR	Manufacturer Software Version	String	const	72
1010	ARRAY	Save Backup Parameters	Unsigned32	rw	73
1011	ARRAY	Reset Backup Parameters	Unsigned32	rw	74
1018	RECORD	Identity	Unsigned32	ro	75
1600 <sup>1)</sup>	RECORD	RxPDO 1 Mapping	PDO Mapping	rw	77
1601 <sup>1)</sup>	RECORD	RxPDO 2 Mapping	PDO Mapping	rw	77
1A00 <sup>1)</sup>	RECORD	TxPDO 1 Mapping	PDO Mapping	rw	78
1A01 <sup>1)</sup>	RECORD	TxPDO 2 Mapping	PDO Mapping	rw	78
1C00	ARRAY	Sync Manager Communication Type	Unsigned8	ro	79
1C10	ARRAY	for information purposes	Unsigned16	ro	-
1C11	ARRAY	for information purposes	Unsigned16	ro	-
1C12	ARRAY	SM2 PDO Assignment	Unsigned16	rw	81
1C13	ARRAY	SM3 PDO Assignment	Unsigned16	rw	82

**Table 26: Communication specific standard objects**

<sup>1)</sup> Little-Endian-Format

### 9.2.1 Object 0x1000: Device Type

---

<b>Object name</b>	Device Type
<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	0x20192
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	The low-value 16 bits of the device type specify the device profile DSP 402 = 0x192 Value 2 in the higher value word denotes a servo drive

---

### 9.2.2 Object 0x1001: Error Register

---

<b>Object name</b>	Error Register
<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	The error register displays the error status of the device in bit code. A set bit indicates that a corresponding error is present. The exact cause of the error is indicated by object 0x1003. At the time of occurrence, an error is indicated by an EMCY message.

Bit	Meaning
0	General error
1	Current
2	Voltage
3	Temperature
4	Communication error (overflow, status error)
5	Device profile-specific error
6	reserved
7	Manufacturer-specific

---

### 9.2.3 Object 0x1003: Pre-defined Error Field

<b>Object name</b>	Pre-defined Error Field
<b>Object code</b>	ARRAY
<b>Description</b>	This parameter stores the last 8 occurring errors as maximum. The "newest" error is stored in subindex 1. As the subindex increases, the error entries become older. The error entries are of the type UNSIGNED32. The higher-value part (bits 16-31) is intended for device-specific information and is not used by the encoTRive. The meaning of bits 0-15 can be taken from Table 37 on page 150.
<b>Subindex 0: Number of error entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	8
<b>PDO mapping</b>	-
<b>Description</b>	Contains the number of stored errors. Writing the value 0 in this element means that the complete error list is deleted. Other values are not permitted and result in an abort message with the error code 0x0609 0030.
<b>Subindex 1: Error no. 1</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Last occurring error
<b>Subindex 2: Error no. 2</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Older error message (position 2)
...	
<b>Subindex 8: Error no. 8</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Oldest error message (position 8)

### 9.2.4 Object 0x1008: Manufacturer's Device Name

---

<b>Object name</b>	Manufacturer Device Name
<b>Object code</b>	VAR
<b>Data type</b>	Visible String
<b>Attribute</b>	Const
<b>Default</b>	"EncoTRive RELC"
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	The object contains the device name, which consists of the ASCII character string 'E', 'n', 'c', 'o', 'T', 'R', 'i', 'v', 'e', + device specific notation Transmission via "Upload SDO Segment Request Protocol".

---

### 9.2.5 Object 0x1009: Manufacturer's Hardware Version

---

<b>Object name</b>	Manufacturer Hardware Version
<b>Object code</b>	VAR
<b>Data type</b>	Visible String
<b>Attribute</b>	const
<b>Default</b>	" ", device specific
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	The object contains the hardware version of the device. This consists of an ASCII character string. Transmission via "Upload SDO Segment Request Protocol".

---

### 9.2.6 Object 0x100A: Manufacturer's Software Version

---

<b>Object name</b>	Manufacturer Software Version
<b>Object code</b>	VAR
<b>Data type</b>	Visible String
<b>Attribute</b>	const
<b>Default</b>	" ", device specific
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	The object contains the software version of the device. This consists of an ASCII character string. Transmission via "Upload SDO Segment Request Protocol".

---

## 9.2.7 Object 0x1010: Save Backup Parameters

<b>Object name</b>	Save Backup Parameters
<b>Object code</b>	ARRAY
<b>Description</b>	This object supports the saving of parameters in non-volatile memory (EEPROM). Changed parameters are accepted only if the storage command is executed! The storage command is performed if the signature "save" is written to subindex 1.
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	4
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Contains the largest array subindex
<b>Subindex 1: Store all parameters</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Stores all storable parameters. The parameters are only stored if the value <b>0x65766173</b> is written. This is the numeric value resulting from the "save" character string ('s' : 0x73, 'a': 0x61, 'v' : 0x76, 'e': 0x65).
<b>Subindex 3: Store application parameters (0x6xxx-objects)</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	see subindex 1
<b>Subindex 4: Store manufacturer-specific parameters (0x2xxx-objects)</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	see subindex 1

## 9.2.8 Object 0x1011: Reset Backup Parameters

<b>Object name</b>	Reset Backup Parameters
<b>Object code</b>	ARRAY
<b>Description</b>	This object supports the restoring of the default values of all writable parameters.
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	2
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Contains the largest array subindex
<b>Subindex 1: Factory settings for all parameters</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Configures all parameters, for which factory settings are available, with their factory settings. This only occurs if the value <b>0x64616F6C</b> is written. This is the numeric value resulting from the "load" character string ('l' : 0x6C, 'o' : 0x6F, 'a' : 0x61, 'd' : 0x64).
<b>Subindex 3: Factory settings for application parameters (0x6xxx-objects)</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	see subindex 1
<b>Subindex 4: Factory settings for manufacturer-specific parameters (0x2xxx-objects)</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	see subindex 1

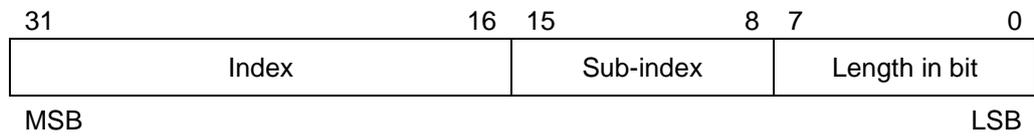
## 9.2.9 Object 0x1018: Identity

<b>Object name</b>	Identity
<b>Object code</b>	ARRAY
<b>Description</b>	This parameter contains general information on the encoTRive.
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	4
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Contains the largest array subindex
<b>Subindex 1: Manufacturer's identification</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	0x0509 (TR-Electronic)
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Contains the device vendor ID assigned by the ETG
<b>Subindex 2: Product Code</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	device specific
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Identifies the device version
<b>Subindex 3: Revision number</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	device specific
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Contains the revision number of the device, which defines the functionality and the individual versions.
<b>Subindex 4: Serial number</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	device specific
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Contains the serial number of the device

### 9.2.10 Structure of the mapping parameter

Sub-index 0 contains the number of valid object entries.

The following entries contain the information of the mapped application objects. The object describes the content of the PDO by their index, sub-index and length in bit:



Possible Receive-process data objects (0x1600, 0x1601)  
see ESI file

Possible Transmit-process data objects (0x1A00, 0x1A01)  
see ESI file

### 9.2.10.1 Objects 0x1600-0x1601: Receive PDO mapping

Corresponding to the ESI file by means of the receive-process-data-objects (0x1600 – 0x1601) the mapping-data can be transferred. The assignment of the objects is performed about “Object 0x1C12: SM2 PDO assignment”.

<b>Object name</b>	Receive PDO Mapping
<b>Object code</b>	ARRAY
<b>Description</b>	Each of these parameter configures the mapping for one Receive-PDO.
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	80
<b>PDO mapping</b>	-
<b>Description</b>	Number of mapped objects
<b>Subindex 1: First mapped object</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Indicates the index, sub-index and length in bit
- - -	
<b>Subindex 80: Eightieth mapped object</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Indicates the index, sub-index and length in bit

### 9.2.10.2 Objects 0x1A00-0x1A01: Transmit PDO mapping

Corresponding to the ESI file by means of the transmit-process-data-objects (0x1A00 – 0x1A01) the mapping-data can be transferred.

The assignment of the objects is performed about “Object 0x1C13: SM3 PDO assignment”.

---

<b>Object name</b>	Transmit PDO Mapping
<b>Object code</b>	ARRAY
<b>Description</b>	Each of these parameter configures the mapping for one Transmit-PDO.

---

<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	80
<b>PDO mapping</b>	-
<b>Description</b>	Number of mapped objects

---

<b>Subindex 1: First mapped object</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Indicates the index, sub-index and length in bit

---

- - -

---

<b>Subindex 80: Eightieth mapped object</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Indicates the index, sub-index and length in bit

---

### 9.2.11 Object 0x1C00: Sync Manager Communication Type

This object is used to define the number of communication channels used and the type of communication.

The following are supported:

- Mailbox sending and receive
- Process data input for the transmission of position values (Slave --> Master)
- Process data output for switching of the operating mode of the drive (Master --> Slave)

The inputs can only be read, the configuration of the communication channels occurs automatically when the EtherCAT master boots.

<b>Object name</b>	Sync Manager Communication Type
<b>Object code</b>	ARRAY
<b>Description</b>	Number of used communication channels and type of communication.
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	4
<b>Min</b>	0
<b>Max</b>	4
<b>PDO mapping</b>	-
<b>Description</b>	Number of used sync manager channels.
<b>Subindex 1: Communication Type Sync Manager 0</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	1: Receive mailbox (Master --> Slave)
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Communication Type Sync Manager 0
<b>Subindex 2: Communication Type Sync Manager 1</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	2: Send mailbox (Slave --> Master)
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Communication Type Sync Manager 1
<b>Subindex 3: Communication Type Sync Manager 2</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	3: Process data output (Master --> Slave)
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Communication Type Sync Manager 2

---

**Subindex 4: Communication Type Sync Manager 3**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	4: Process data input (Slave --> Master)
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Communication Type Sync Manager 3

---

## 9.2.12 Object 0x1C12: SM2 PDO assignment

The number and the respective object index of the assigned RxPDOs are defined by object 0x1C12. One of the Receive Process Data Objects listed under Sub-Index 1/2 can be assigned as process data output:

<b>Object name</b>	SM2 PDO Assignment
<b>Object code</b>	ARRAY
<b>Description</b>	PDO mapping object index of the assigned RxPDO
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	2
<b>PDO mapping</b>	-
<b>Description</b>	Number of assigned RxPDOs
<b>Subindex 1: Assigned RxPDO</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	PDO mapping object index of the assigned RxPDO. 0x1600: RxPDO 1 0x1601: RxPDO 2
<b>Subindex 2: Assigned RxPDO</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	PDO mapping object index of the assigned RxPDO. 0x1600: RxPDO 1 0x1601: RxPDO 2

### 9.2.13 Object 0x1C13: SM3 PDO assignment

The number and the respective object index of the assigned TxPDOs are defined by object 0x1C13. One of the Transmit Process Data Objects listed under Sub-Index 1/2 can be assigned as process data input:

<b>Object name</b>	SM3 PDO Assignment
<b>Object code</b>	ARRAY
<b>Description</b>	PDO mapping object index of the assigned TxPDO
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	2
<b>PDO mapping</b>	-
<b>Description</b>	Number of assigned TxPDOs
<b>Subindex 1: Assigned TxPDO</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	PDO mapping object index of the assigned TxPDO. 0x1A00: TxPDO 1 0x1A01: TxPDO 2
<b>Subindex 2: Assigned TxPDO</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	PDO mapping object index of the assigned TxPDO. 0x1A00: TxPDO 1 0x1A01: TxPDO 2

### 9.3 Manufacturer-specific objects

Index (h)	Object	Name	Data length	Attr.	Page
<b>Parameter</b>					
2EF3	RECORD	TrAbeData.MVI16	Integer16	ro	83
2F04	RECORD	DS402M.MPU32	Unsigned32	rw	85
2F0E	RECORD	DS402M.MPX	-	rw	89

#### 9.3.1 Object 0x2EF3: TrAbeData.MVI16

<b>Object code</b>	ARRAY
<b>Description</b>	For service purposes only
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	8
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Number of following entries
<b>Subindex 1: Current temperature 1°C</b>	
<b>Data type</b>	INTEGER16
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	-2 <sup>15</sup>
<b>Max</b>	+2 <sup>15</sup> -1
<b>PDO mapping</b>	yes
<b>Description</b>	-
<b>Subindex 2: Current temperature 2°C</b>	
<b>Data type</b>	INTEGER16
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	-2 <sup>15</sup>
<b>Max</b>	+2 <sup>15</sup> -1
<b>PDO mapping</b>	yes
<b>Description</b>	-
<b>Subindex 3: Current temperature 3°C</b>	
<b>Data type</b>	INTEGER16
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	-2 <sup>15</sup>
<b>Max</b>	+2 <sup>15</sup> -1
<b>PDO mapping</b>	yes
<b>Description</b>	-

**Subindex 4: Current temperature 4/°C**

<b>Data type</b>	INTEGER16
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	-

**Subindex 5: Analog input 1/mV**

<b>Data type</b>	INTEGER16
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	$-2^{15}$
<b>Max</b>	$+2^{15}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	-

**Subindex 6: Analog input 2/mV**

<b>Data type</b>	INTEGER16
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	$-2^{15}$
<b>Max</b>	$+2^{15}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	-

**Subindex 7: Analog input 3/mV**

<b>Data type</b>	INTEGER16
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	$-2^{15}$
<b>Max</b>	$+2^{15}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	-

**Subindex 8: Analog input 4/mV**

<b>Data type</b>	INTEGER16
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	$-2^{15}$
<b>Max</b>	$+2^{15}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	-

### 9.3.2 Object 0x2F04: DS402M.MPU32

<b>Object code</b>	ARRAY
<b>Description</b>	Masking of the digital inputs/outputs, ballast circuit
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	14
<b>PDO mapping</b>	-
<b>Description</b>	Number of following entries
<b>Subindex 1: Digital Input And Mask</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	disables individual bits of the input signal (0x60FD)
<b>Subindex 2: Digital Input Negation Mask</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	negates individual bits of the input signal (0x60FD)
<b>Subindex 3: Output And Mask</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	disables individual bits of the output signal (0x60FE)
<b>Subindex 4: Digital Output Negation Mask</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	negates individual bits of the output signal (0x60FE)

**Subindex 5: VBrake/mV**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	no
<b>Description</b>	in case of a motor with holding brake: Nominal value for the output voltage of the brake

**Subindex 6: VBallastOut max./mV**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	no
<b>Description</b>	in case of a ballast resistor at the open drain ballast output: Nominal value for the limit of the ballast output voltage

**Subindex 7: VBallastIn zero/mV**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	no
<b>Description</b>	

**Subindex 8: VBallastIn full/mV**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	no
<b>Description</b>	

---

---

**Subindex 9: Ls/μH (Stator inductivity)**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	-2 <sup>31</sup>
<b>Max</b>	+2 <sup>31</sup> -1
<b>PDO mapping</b>	no
<b>Description</b>	The proportional gains of the PI current controller are set by the DC voltage, divided according to stator inductivity, in order to maintain the necessary switch-off frequency of the closed control loop. Depending on the motor type, this inductivity can fall to 1/3 if the stator current is increased from 0 to the nominal value. In order to prevent gains that are too high and cause vibrations, the stator inductivity should be measured at maximum current.

**Subindex 10: RegPosrKS**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	-2 <sup>31</sup>
<b>Max</b>	+2 <sup>31</sup> -1
<b>PDO mapping</b>	no
<b>Description</b>	for internal purposes only

**Subindex 11: Max. duration of pos. ramp/Tc**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	-2 <sup>31</sup>
<b>Max</b>	+2 <sup>31</sup> -1
<b>PDO mapping</b>	no
<b>Description</b>	for internal purposes only

**Subindex 12: LsiscC30**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	-2 <sup>31</sup>
<b>Max</b>	+2 <sup>31</sup> -1
<b>PDO mapping</b>	no
<b>Description</b>	for internal purposes only

---

---

**Subindex 13: LsiscC31**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	no
<b>Description</b>	for internal purposes only

---

**Subindex 14: LsiscC32**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	no
<b>Description</b>	for internal purposes only

---

### 9.3.3 Object 0x2F0E: DS402M.MPX

<b>Object code</b>	ARRAY
<b>Description</b>	Configuration of the following error
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	4
<b>PDO mapping</b>	-
<b>Description</b>	Number of following entries
<b>Subindex 1: Following Error Options</b>	
<b>Data type</b>	INTEGER16
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	-2 <sup>15</sup>
<b>Max</b>	+2 <sup>15</sup> -1
<b>PDO mapping</b>	no
<b>Description</b>	<p>1 -&gt; A following error triggers a non-fatal fault, that reacts according to the option code 0x605E</p> <p>2 -&gt; A following error immediately turns off the drive.</p> <p>3 -&gt; A following error triggers a quick stop, that reacts according to the option code 0x605A.</p> <p>Hint: If this option code is 5 or higher, which means staying in quick stop, this state can be left without turning off the motor by toggling the control word bit 0x80 "Clear Fault" once or twice.</p> <p>0 or any other value -&gt; A following error only sets the status word bit and sends its emergency.</p> <p>Also see chapter "Stop conditions: Priorities, Behavior" on page 124.</p>
<b>Subindex 2: Q24 Profile Restart Window /rev.</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	0
<b>Max</b>	+2 <sup>32</sup> -1
<b>PDO mapping</b>	no
<b>Description</b>	<p>If the following error window is switched off, this limit for the following error is used in revolutions for restart of speed and positioning ramps. Q24 means 24 bits after the comma, therefore max. 256 revolutions.</p> <p>Function: Avoids sudden and uncontrolled movements of the drive after enabling of a blocked drive. Dependent on the adjusted mode of operation (0x6060 = 1 or 3), the target speed or target position is reached considering the velocity and acceleration limits.</p> <p>Also see chapter "Stop conditions: Priorities, Behavior" on page 124.</p>

---

**Subindex 3: Profile Restart Timeout/ms**

<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	0
<b>Max</b>	$+2^{16}-1$
<b>PDO mapping</b>	no
<b>Description</b>	In order to activate the ramp restart, the following error must at least exceed for this time the adjusted window in subindex two in this object.

---

**Subindex 4: Gear Backlash/Pos.units**

<b>Data type</b>	INTEGER32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	no
<b>Description</b>	Reserved

---

## 9.4 Objects of the DSP 402 device profile

Index (h)	Object	Name	Data length	Attr.	Page
<b>Parameter</b>					
^ 6007	VAR	Abort Connection Code	Integer16	rw	93
603F	VAR	Error Code	Unsigned16	ro	93
6040	VAR	Control word	Unsigned16	rw	93
6041	VAR	Status word	Unsigned16	ro	94
^ 605A	VAR	Quick Stop Option Code	Integer16	rw	94
^ 605B	VAR	Shutdown Option Code	Integer16	rw	95
^ 605C	VAR	Disable Operation Option Code	Integer16	rw	95
^ 605D	VAR	Stop Option Code	Integer16	rw	96
^ 605E	VAR	Fault Reaction Option Code	Integer16	rw	96
^ 6060	VAR	Operation Mode	Integer8	rw	97
6061	VAR	Operation Mode Display	Integer8	ro	97
6062	VAR	Required Position	Integer32	ro	97
6064	VAR	Actual Position	Integer32	ro	98
^ 6065	VAR	Following Error Window	Unsigned32	rw	98
^ 6066	VAR	Following Error Timeout	Unsigned16	rw	98
^ 6067	VAR	In-Position Window	Unsigned32	rw	99
^ 6068	VAR	In-Position Timeout	Unsigned16	rw	99
6069	VAR	Sensor Velocity	Integer32	ro	99
606B	VAR	Required Velocity	Integer32	ro	100
606C	VAR	Actual Velocity	Integer32	ro	100
^ 606D	VAR	Velocity Window	Unsigned16	rw	100
^ 606E	VAR	Velocity Window Timeout	Unsigned16	rw	101
^ 606F	VAR	Velocity Threshold	Unsigned16	rw	101
^ 6070	VAR	Velocity Threshold Timeout	Unsigned16	rw	101
6071	VAR	Target Torque	Integer16	rw	102
^ 6072	VAR	Torque Limit	Unsigned16	rw	102
^ 6073	VAR	Limit for Torque generating Current	Unsigned16	rw	102
6074	VAR	Required Torque	Integer16	ro	103
6075	VAR	Motor rated Current/mA	Unsigned32	rw	103
6076	VAR	Motor rated Torque/Nmm	Unsigned32	rw	103
6077	VAR	Actual Torque	Integer16	ro	104

^ : permanent storage must be performed about object 1010h

6078	VAR	Actual Current	Integer16	ro	104
6079	VAR	Actual DC Link Circuit voltage/mV	Unsigned32	ro	104
607A	VAR	Target Position	Integer32	rw	105
607B	ARRAY	Position Range Limit	Integer32	rw	105
607C	VAR	Home Offset	Integer32	rw	106
607D	ARRAY	Software Position Limits	Integer32	rw	106
^ 607E	VAR	Polarity	Unsigned8	rw	107
^ 607F	VAR	Max Profile Velocity	Unsigned32	rw	107
6080	VAR	Max Motor Speed	Unsigned32	rw	107
^ 6081	VAR	Profile Velocity	Unsigned32	rw	108
^ 6083	VAR	Profile Acceleration	Unsigned32	rw	108
^ 6084	VAR	Profile Deceleration	Unsigned32	rw	108
^ 6085	VAR	Quick Stop Deceleration	Unsigned32	rw	109
608F	ARRAY	Position Encoder Resolution	Unsigned32	rw	109
6090	ARRAY	Velocity Encoder Resolution	Unsigned32	ro	110
6093	ARRAY	Position Factor	Unsigned32	rw	111
6094	ARRAY	Velocity Encoder Factor	Unsigned32	rw	112
6097	ARRAY	Acceleration Factor	Unsigned32	rw	113
6098	VAR	Homing Method	Integer8	rw	113
6099	ARRAY	Homing speeds	Unsigned32	rw	114
^ 609A	VAR	Homing Acceleration	Unsigned32	rw	114
60B0	VAR	Position Offset	Integer32	rw	115
60B1	VAR	Velocity Offset	Integer32	rw	115
60B2	VAR	Torque Offset	Integer16	rw	115
^ 60C5	VAR	Max Acceleration	Unsigned32	rw	116
^ 60C6	VAR	Max Deceleration	Unsigned32	rw	116
60F4	VAR	Following Error	Integer32	ro	116
60FD	VAR	Digital Inputs	Unsigned32	ro	117
60FE	ARRAY	Digital Outputs	Unsigned32	rw	117
60FF	VAR	Target Velocity	Integer32	rw	118
6402	VAR	Motor Type	Unsigned16	ro	118
6502	VAR	Supported Drive Modes	Unsigned32	ro	118
6508	VAR	Runtime in 0.1 hours	Unsigned32	ro	119
6510	VAR	Measuring Range in revolutions	Unsigned40	ro	119

Table 27: Profile specific objects

^ : permanent storage must be performed about object 1010h

---

### 9.4.1 Object 0x6007: Abort Connection Code

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER16
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-32768
<b>Max</b>	32767
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter specifies the action to be taken by encoTRive when the network connection is interrupted. The following values are defined:  0: No action 1: Go into "Fault" status 2: Execute "Disable Voltage" command, see Table 6 page 34 3: Execute "Quick Stop" command, see Table 6 page 34

---

### 9.4.2 Object 0x603F: Error Code

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	65535
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter stores the last occurring error and corresponds to the low-value 16 bits of object 0x1003.

---

### 9.4.3 Object 0x6040: Control word

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	rww
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	65535
<b>PDO mapping</b>	yes
<b>Description</b>	Commands are transmitted to the drive in this object, see sections 5.1.3, 5.2.1 and 5.3.1.

---

#### 9.4.4 Object 0x6041: Status word

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	65535
<b>PDO mapping</b>	yes
<b>Description</b>	The drive transmits status information in this object, see sections 5.1.1, 5.2.2 and 5.3.2.

---

#### 9.4.5 Object 0x605A: Quick Stop Option Code

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER16
<b>Attribute</b>	rw
<b>Default</b>	2
<b>Min</b>	-32768
<b>Max</b>	32767
<b>PDO mapping</b>	yes
<b>Description</b>	The reaction after execution of a quick stop can be defined with this parameter.

The following values are supported by the encoTRive:

**Table 28: Values for Quick Stop Option Code**

<b>Value</b>	<b>Meaning</b>
0	Switches drive off. After coming to a stop, the drive goes into "Switch-on inhibit" status.
1	Decelerates drive to a stop with current deceleration. The drive then goes into "Switch-on inhibit" status.
2	Decelerates drive with deceleration for quick stop (0x6085). After coming to a stop, the drive goes into "Switch-on inhibit" status.
5	Decelerates with current deceleration. After coming to a stop the drive remains in "quick stop" status.
6	Decelerates with deceleration for quick stop (0x6085). After coming to a stop the drive remains in "Quick stop" status.

---

### 9.4.6 Object 0x605B: Shutdown Option Code

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER16
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-32768
<b>Max</b>	32767
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter is used to define the reaction of the drive during the state transition <i>Ready for operation</i> → <i>Ready to switch on</i> .

The following values are supported by the encoTRive:

**Table 29: Values for Shutdown Option Code**

Value	Meaning
0	Switches drive off.
1	Decelerates drive to a stop with current deceleration. Then switches drive off.

---

### 9.4.7 Object 0x605C: Disable Operation Option Code

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER16
<b>Attribute</b>	rw
<b>Default</b>	1
<b>Min</b>	-32768
<b>Max</b>	32767
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter is used to define the reaction of the drive during the state transition <i>Ready for operation</i> → <i>Switched on</i> .

The following values are supported by the encoTRive:

**Table 30: Values for Disable Operation Option Code**

Value	Meaning
0	Switches drive off.
1	Decelerates drive to a stop with current deceleration. Then switches drive off.

---

### 9.4.8 Object 0x605D: Stop Option Code

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER16
<b>Attribute</b>	rw
<b>Default</b>	1
<b>Min</b>	-32768
<b>Max</b>	32767
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter is used to define the reaction of the drive when bit 8 = "Stop motor" is active in the control word.

The following values are supported by the encoTRive:

**Table 31: Values for Halt Option Code**

Value	Meaning
0	Switches drive off.
1	Decelerates drive to a stop with current deceleration.
2	Decelerates drive with deceleration for Quick stop (0x6085).
3	Decelerates drive at current limit.
4	Decelerates drive at voltage limit.

---

### 9.4.9 Object 0x605E: Fault Reaction Option Code

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER16
<b>Attribute</b>	rw
<b>Default</b>	2
<b>Min</b>	-32768
<b>Max</b>	32767
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter is used to define the reaction of the drive in the event of a fault.

The following values are supported by the encoTRive:

**Table 32: Values for Fault Reaction Option Code**

Value	Meaning
0	Switches drive off.
1	Decelerates drive to a stop with current deceleration.
2	Decelerates drive with deceleration for Quick stop (0x6085).
3	Decelerates drive at current limit.
4	Decelerates drive at voltage limit.

---

#### 9.4.10 Object 0x6060: Operation Mode

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER8
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-128
<b>Max</b>	127
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter is used to define the current operating mode.

The following values are supported by the encoTRive:

**Table 33: Values for operating mode**

Value	Meaning
1	Profile Position
3	Profile Velocity
6	Homing (position adjustment)
8	Cyclic sync Position
9	Cyclic sync Velocity
10	Cyclic sync torque

---

#### 9.4.11 Object 0x6061: Operation Mode Display

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER8
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	-128
<b>Max</b>	127
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter is used to display the current operating mode (see Table 33, page 97).

---

#### 9.4.12 Object 0x6062: Required Position

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER32
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object, in position units, contains the position to be moved to, resulting from the control algorithm.

---

### 9.4.13 Object 0x6064: Actual Position

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER32
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object displays the actual position value.

---

### 9.4.14 Object 0x6065: Following Error Window

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	$2^{32}-1$ (tracking error monitoring switched off)
<b>Min</b>	0
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	<p>This object defines the interval for permissible actual position values around the nominal position value (0x6062). If the actual position value is outside this interval, a tracking error exists.</p> <p>A parameter value of <math>2^{32}-1</math> deactivates the tracking error monitoring.</p> <p>A tracking error can occur</p> <ul style="list-style-type: none"><li>• if the drive is blocked</li><li>• if the target speed cannot be reached</li><li>• if the control parameters are set unfavorably</li><li>• if the acceleration is too high</li></ul>

---

### 9.4.15 Object 0x6066: Following Error Timeout

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	rw
<b>Default</b>	100
<b>Min</b>	0
<b>Max</b>	$2^{16}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	The parameter value specifies the minimum time in 1 ms steps for which a tracking error must be present before it is displayed as such in the status word.

---

---

#### 9.4.16 Object 0x6067: In-Position Window

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	$2^{32}-1$ (position window monitoring switched off)
<b>Min</b>	0
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	This object defines a symmetrical interval around the target position (0x607A). If the actual position value is in this interval, the target position is considered to have been reached.

---

#### 9.4.17 Object 0x6068: In-Position Timeout

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	rw
<b>Default</b>	100
<b>Min</b>	0
<b>Max</b>	$2^{16}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	The parameter value specifies the minimum time in 1 ms steps for which the actual position value must be present in the position window, before "Target reached" is indicated in the status word.

---

#### 9.4.18 Object 0x6069: Sensor Velocity

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER32
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	The parameter contains the speed value measured on the speed encoder in speed increments.

---

**9.4.19 Object 0x606B: Required Velocity**

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER32
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object contains the target speed in speed units, which results from the control algorithm.

---

**9.4.20 Object 0x606C: Actual Velocity**

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER32
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object contains the current speed in speed units.

---

**9.4.21 Object 0x606D: Velocity Window**

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object monitors whether the required process velocity has been achieved after an eventual acceleration or deceleration (braking) phase. It is given in speed units.

---

---

#### 9.4.22 Object 0x606E: Velocity Window Timeout

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	The corresponding bit 10 “Target reached” is set in the status word when the difference between the “Target velocity” and the “Velocity actual value” is within the “Velocity window” longer than the “Velocity window timeout”. The value is given in multiples of milliseconds.

---

#### 9.4.23 Object 0x606F: Velocity Threshold

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	As soon as the “Velocity actual value” exceeds the “Velocity threshold” longer than the “Velocity threshold timeout” bit 12 velocity = 0 is reset in the status word. Below this threshold the bit is set and indicates that the axle is stationary. The value is given in speed units.

---

#### 9.4.24 Object 0x6070: Velocity Threshold Timeout

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	The value is given in multiples of milliseconds, description see object 606Fh

---

#### 9.4.25 Object 0x6071: Target Torque

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER16
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	10000
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter is the input value for the torque controller in profile torque mode and the value is given per thousand of rated torque (0x6076).

---

#### 9.4.26 Object 0x6072: Torque Limit

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	rw
<b>Default</b>	1250
<b>Min</b>	0
<b>Max</b>	10000
<b>PDO mapping</b>	yes
<b>Description</b>	This object represents the maximum permissible torque in the motor and is given per thousand of rated torque. The default value 1250 corresponds to 125 % of the rated torque (0x6076).

---

#### 9.4.27 Object 0x6073: Limit for Torque generating Current

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	rw
<b>Default</b>	1250
<b>Min</b>	0
<b>Max</b>	10000
<b>PDO mapping</b>	yes
<b>Description</b>	This object represents the maximum permissible torque generating current in the motor and is given per thousand of rated current (0x6075).

---

---

#### 9.4.28 Object 0x6074: Required Torque

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER16
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter is the output value of the torque limit function. The value is given per thousand of rated torque (0x6076).

---

#### 9.4.29 Object 0x6075: Motor rated Current/mA

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	7600
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object specifies the rated motor current in mA.

---

#### 9.4.30 Object 0x6076: Motor rated Torque/mNm

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	630
<b>Min</b>	0
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	This object contains the rated torque of the motor in mNm.

---

**9.4.31 Object 0x6077: Actual Torque**

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER16
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object contains the current torque of the motor. Unit: Thousandths of rated torque (0x6076)

---

**9.4.32 Object 0x6078: Actual Current**

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER16
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object contains the actual current value of the motor. Unit: Thousandths of rated current (0x6075)

---

**9.4.33 Object 0x6079: Actual DC Link Circuit Voltage/mV**

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object contains the current voltage of the DC voltage intermediate circuit in mV.

---

#### 9.4.34 Object 0x607A: Target Position

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	$-2^{31}$
<b>Max</b>	$2^{31}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	This object contains the target position in position units, see chapter "5.8 Units" page 56. Depending on the "absolute/relative" bit of the STW, the target position is interpreted as absolute or relative.

---

#### 9.4.35 Object 0x607B: Position Range Limits

---

<b>Object code</b>	ARRAY
<b>Description</b>	This parameter specifies the value range for the position value. When a position limit is reached or exceeded, the position value automatically jumps to the opposite end of the range. This behavior can be prevented with parameter 0x607D.

---

##### Subindex 0: Number of entries

<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	2
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Specifies the number of following entries

---

##### Subindex 1: Lower position limit

<b>Data type</b>	INTEGER32
<b>Attribute</b>	rw
<sup>2</sup> <b>Default</b>	$-2^{31}$
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	Minimum position value

---

##### Subindex 2: Upper position limit

<b>Data type</b>	INTEGER32
<b>Attribute</b>	rw
<sup>2</sup> <b>Default</b>	$2^{31}$
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	Maximum position value

---

<sup>2</sup> Depending on gear and pitch the value is different

### 9.4.36 Object 0x607C: Home Offset

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	The value is the difference between the zero position for the application and the machine home position (found during homing), it is measured in position units.

### 9.4.37 Object 0x607D: Software Position Limits

<b>Object code</b>	ARRAY
<b>Description</b>	This parameter contains the absolute limits for position values. The values are specified in position units relative to the machine zero point. In the case of a new target position, a check is made to see whether this is located in the software position range.
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	2
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Specifies the number of following entries
<b>Subindex 1: Lower position limit</b>	
<b>Data type</b>	INTEGER32
<b>Attribute</b>	rw
<sup>3</sup> <b>Default</b>	$-2^{31}$
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	Minimum position value
<b>Subindex 2: Upper position limit</b>	
<b>Data type</b>	INTEGER32
<b>Attribute</b>	rw
<sup>3</sup> <b>Default</b>	$2^{31}-1$
<b>Min</b>	$-2^{31}$
<b>Max</b>	$+2^{31}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	Maximum position value

<sup>3</sup> Depending on gear and pitch the value is different

### 9.4.38 Object 0x607E: Polarity

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	0
<b>Max</b>	255
<b>PDO mapping</b>	yes
<b>Description</b>	<p>This parameter defines whether the direction of position and/or speed is to be reversed.</p> <p>If a direction reversal is defined, the values of actual position value and nominal position or actual speed value and nominal speed are multiplied by -1.</p> <p>The parameter value is interpreted as follows:</p>

**Table 34: Direction reversal**

Bit 7	Bit 6	Bit 5-0
1 – Direction reversal at position	1 – Direction reversal at speed	reserved
0 – No direction reversal at position	0 – No direction reversal at speed	

---

### 9.4.39 Object 0x607F: Max Profile Velocity

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	8100
<b>Min</b>	0
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	<p>This parameter defines the maximum speed in speed units during a positioning movement.</p>

---

### 9.4.40 Object 0x6080: Max Motor Speed

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	9000
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	<p>This parameter specifies the maximum permitted speed of the motor shaft in rpm.</p>

---

#### 9.4.41 Object 0x6081: Profile Velocity

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	5000
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter specifies the speed in speed units, which is to be reached at the end of an acceleration ramp, see chapter 5.2 "Profile Position (PP)" operating mode page 35.

---

#### 9.4.42 Object 0x6083: Profile Acceleration

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	500
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter specifies in acceleration units the acceleration for the acceleration ramp, see chapter 5.2 "Profile Position (PP)" operating mode page 35.

---

#### 9.4.43 Object 0x6084: Profile Deceleration

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	1000
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter specifies in acceleration units the deceleration used to slow down during a positioning movement, see chapter 5.2 "Profile Position (PP)" operating mode page 35.

---

#### 9.4.44 Object 0x6085: Quick Stop Deceleration

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	2000
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter specifies in acceleration units the deceleration used to slow down to a quick stop, if the Quick Stop Option Code has a value of 2.

---

#### 9.4.45 Object 0x608F: Position Encoder Resolution

---

<b>Object code</b>	ARRAY
<b>Description</b>	This parameter defines the ratio of position increments to motor revolutions, see chapter "5.8 Units" page 56.

---

**Subindex 0: Number of entries**

<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	2
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Specifies the number of following entries

---

**Subindex 1: Position increments**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	4096
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Position increments

---

**Subindex 2: Motor revolutions**

<b>Data type</b>	INTEGER32
<b>Attribute</b>	ro
<b>Default</b>	1
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Motor revolutions

---

**9.4.46 Object 0x6090: Velocity Encoder Resolution**

---

<b>Object code</b>	ARRAY
<b>Description</b>	This parameter defines the ratio of speed increments per second to motor revolutions per second, see chapter "5.8 Units" page 56.

---

<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	2
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Specifies the number of following entries

---

<b>Subindex 1: Speed increments</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	$2^{31}$
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	-
<b>Description</b>	Speed increments per second

---

<b>Subindex 2: Motor revolutions</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	5000
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	-
<b>Description</b>	Motor revolutions per second

---

### 9.4.47 Object 0x6093: Position Factor

<b>Object code</b>	ARRAY
<b>Description</b>	Position increments are obtained from user-defined position units (e.g. degrees) by multiplying with the position factor. The position unit used by the user is implicitly defined by the position factor. If necessary gear and feed are also taken into account, see chapter "5.8 Units" page 56. The position factor is the ratio between position increments and position units, defined by Formula 5.
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	2
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Specifies the number of following entries
<b>Subindex 1: Position increments (counter)</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	1024
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	Position increments
<b>Subindex 2: Position increments (denominator)</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	1024
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	Position units

#### 9.4.48 Object 0x6094: Velocity Encoder Factor

<b>Object code</b>	ARRAY
<b>Description</b>	Speed increments are obtained from user-defined speed units (e.g. rpm) by multiplying with the speed factor. The speed unit used by the user is implicitly defined by the speed factor. If necessary gear and feed are also taken into account, see chapter "5.8 Units" page 56. The speed factor is the ratio between speed increments per second and speed units, defined by Formula 8.
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	2
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Specifies the number of following entries
<b>Subindex 1: Speed increments (counter)</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	$2^{31}$
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	Speed increments per second
<b>Subindex 2: Speed units (denominator)</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	300000
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	Speed units

#### 9.4.49 Object 0x6097: Acceleration Factor

<b>Object code</b>	ARRAY
<b>Description</b>	Speed increments per second are obtained from user-defined acceleration units (e.g. rpm/sec) by multiplying with the acceleration factor. The acceleration unit used by the user is implicitly defined by the acceleration factor. If necessary gear and feed are also taken into account, see chapter "5.8 Units" page 56. The acceleration factor is the ratio between speed increments per second and acceleration units, defined by Formula 10.
<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	2
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Specifies the number of following entries
<b>Subindex 1: Acceleration increments (counter)</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	$2^{31}$
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	Speed increments per second
<b>Subindex 2: Acceleration units (denominator)</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	300000
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	Acceleration units

#### 9.4.50 Object 0x6098: Homing Method

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER8
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	17
<b>Max</b>	35
<b>PDO mapping</b>	yes
<b>Description</b>	This object determines the method that will be used during homing. The methods 17 to 30 are supported = homing without index pulse and 35 = homing on basis of the current position (if no limit switches are available)

### 9.4.51 Object 0x6099: Homing speeds

---

<b>Object code</b>	ARRAY
<b>Description</b>	This parameter defines the speeds used during homing and is given in velocity units.

---

<b>Subindex 0: Number of entries</b>	
<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	2
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Number of entries

---

<b>Subindex 1: Homing velocity</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	Speed during search for switch

---

<b>Subindex 2: Homing velocity</b>	
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	Speed during search for zero

---

### 9.4.52 Object 0x609A: Homing Acceleration

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	-
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter establishes the acceleration to be used for all accelerations and decelerations for the homing modes and is given in acceleration units.

---

---

#### 9.4.53 Object 0x60B0: Position Offset

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object provides the offset of the target position and is given in user-defined position units.

---

#### 9.4.54 Object 0x60B1: Velocity Offset

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object provides the offset for the velocity value and is given in user-defined velocity units. In cyclic synchronous position mode, this object contains the input value for velocity feed forward. In cyclic synchronous velocity mode, it contains the commanded offset of the drive. The value is absolute.

---

#### 9.4.55 Object 0x60B2: Torque Offset

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER16
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object provides the offset for the torque value and is given in per thousand rated torque. In cyclic synchronous position mode and cyclic synchronous velocity mode, this object contains the input value for torque feed forward. In cyclic synchronous torque mode it contains the commanded additive torque of the drive, which is added to the target torque value. The value is absolute.

---

**9.4.56 Object 0x60C5: Max Acceleration**

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	100000
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter specifies the maximum permissible acceleration in user-defined acceleration units.

---

**9.4.57 Object 0x60C6: Max Deceleration**

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	100000
<b>Min</b>	1
<b>Max</b>	$2^{32}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter specifies the maximum permissible deceleration in user-defined acceleration units.

---

**9.4.58 Object 0x60F4: Following Error**

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER32
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	$-2^{31}$
<b>Max</b>	$2^{31}-1$
<b>PDO mapping</b>	yes
<b>Description</b>	This object provides the actual value of the following error and is given in user-defined position units.

---

### 9.4.59 Object 0x60FD: Digital Inputs

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	0
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter specifies the current status of the digital inputs.

encoTRive supports the following digital inputs:

Bit 0: lower position value reached (1 = yes, 0 = no)  
 Bit 1: upper position value reached (1 = yes, 0 = no)

---

### 9.4.60 Object 0x60FE: Digital Outputs

---

<b>Object code</b>	ARRAY
<b>Description</b>	This parameter enables control of the digital outputs provided by the encoTRive.

---

**Subindex 0: Number of entries**

<b>Data type</b>	UNSIGNED8
<b>Attribute</b>	ro
<b>Default</b>	2
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	-
<b>Description</b>	Specifies the number of following entries

---

**Subindex 1: Output bits**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0x0000 0000
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This element contains the defined output bits: Bit 0 = 0: stop brake closed

---

**Subindex 2: Bit mask**

<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	rw
<b>Default</b>	0x0000 0000
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	Release / block output: Bit 0 = 1: output 0 is being used

---

#### 9.4.61 Object 0x60FF: Target Velocity

---

<b>Object code</b>	VAR
<b>Data type</b>	INTEGER32
<b>Attribute</b>	rw
<b>Default</b>	0
<b>Min</b>	$-2^{31}$
<b>Max</b>	$2^{31} - 1$
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter specifies the target speed in user-defined speed units in the "Profile Velocity" operating mode.

---

#### 9.4.62 Object 0x6402: Motor Type

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED16
<b>Attribute</b>	ro
<b>Default</b>	3
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter specifies the type of motor used. In the case of encoTRive, this is a synchronous motor with permanent magnet = value 3

---

#### 9.4.63 Object 0x6502: Supported Drive Modes

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	0x0000 03A5
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	Yes
<b>Description</b>	This parameter specifies the operating modes supported by the drive in bit code. The default value 03A5h (0000 0011 1010 0101) means that encoTRive supports the "Profile Position" (bit 0), "Profile velocity" (bit 2), "Homing" (bit 5) "Cyclic synchronous position" (bit 7), "Cyclic synchronous velocity" (bit 8) and "Cyclic synchronous torque" (bit 9) operating modes.

---

---

#### 9.4.64 Object 0x6508: Runtime in 0.1 hours

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED32
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This parameter indicates the operation hours of the position encoder in 0.1 hours.

---

#### 9.4.65 Object 0x6510: Measuring Range in revolutions

---

<b>Object code</b>	VAR
<b>Data type</b>	UNSIGNED40
<b>Attribute</b>	ro
<b>Default</b>	-
<b>Min</b>	-
<b>Max</b>	-
<b>PDO mapping</b>	yes
<b>Description</b>	This object provides the distinguishable revolutions for high-resolution position encoders.

---

## 10 Projecting with TwinCAT System Manager V2.10

You can perform the following description with TwinCAT in the demo mode without a license.

General information about the use of EtherCAT drives:

*LED indicators:*

- ETG1300\_V1i1i0\_S\_D\_IndicatorLabelingSpecification.pdf

*Objects 1003, 1010 and 1111:*

- CiA\_Ds301\_v04020005cor3.pdf

*EtherCAT Drive-Profile DS-402:*

- CiA\_Dsp402\_2v03000004.pdf
- IEC61800-7-200\_22G\_184e\_FDIS\_ETG.pdf
- ETG.6010\_V1i0i0\_D\_D\_CiA402\_ImplDirective.pdf
- EtherCATDeviceDescription\_V0i3i0.pdf

As EtherCAT master for testing devices on notebook, desktop or industrial PCs with WINDOWS the "EtherCAT Conformance Test Tool" 1.20.80 is recommended. See also hints in chapter 14 on page 152.

As EtherCAT master for PLC projects on notebook, desktop or industrial PCs with WINDOWS "TwinCAT" 2.10 or 2.11 is recommended. The following descriptions refer to "TwinCAT\_2100\_1323\_32Bit".

### 10.1 Installation of the ESI file

Download ESI file:

[www.tr-electronic.de/f/TR-EMO-ID-MUL-0018](http://www.tr-electronic.de/f/TR-EMO-ID-MUL-0018)

It contains the "EtherCAT Slave Information": available SDOs, SDO attributes, PDO configuration.

Before starting the TwinCAT system manager, copy this file into the directory "IO\EtherCAT" of TwinCAT 2.10 or 2.11.

### 10.2 Connection of the encoTRive device

- Connect the drive by any EtherCAT line to the EtherCAT bus
- Connect the voltage supply and turn it on
- Now the green and the orange RUN LED are slowly blinking alternatingly every second

### 10.3 Creation of a TwinCAT project for an encoTRive device

- Start the PC with TwinCAT.
- Open the TwinCAT PLC control.
- Create a new PLC project from scratch.
- Open the library manager and add the EtherCAT library to the project. (you will need it later).
- Open the PLC source window. Set in the empty code window of the main task a single semicolon ';'. Now you can already compile the project. Save it.
- Open the TwinCAT System Manager. The currently installed ESI files are imported automatically from the TwinCAT directory IO.
- Set it to the expert mode.
- Start TwinCAT in the configuration mode.
- Create a new TwinCAT project from scratch.
- Assign the saved PLC project to this project.
- Attach an EtherCAT master to the bus.
- In the master settings select a compatible LAN adapter, install it and select it.
- Scan for boxes -> "EncoTRive RELC" should appear.
- Go to the tab "CoE". Now it should upload and display the whole CoE directory. The progress is indicated by a progress bar.
- Check the boxes for PDO assign and download in its tab "Process data".
- Change its PDOs as you need them, by inserting or deleting objects, or accept the default.
- Export all variables in the PDOs to the mapping.
- Create the mapping.
- Go to the PLC folder and export the mapping to the assigned PLC project.
- Go to the PLC folder, open the tab "Boot" and check all boxes.
- Start TwinCAT in the Run mode.
- Go to the PLC control. In global variables of the resources now the exported variables should appear.
- Recompile the project, go online and start it. Create it also as boot-project that TwinCAT loads as default.
- Go back to the System Manager. If it wants to reload the changed PLC project, say yes.
- Start TwinCAT again in the Run mode. Now the EncoTRive should go to operational: The green state LED should be constantly on.

## 10.4 Changing data objects

- To view the process data of a device, you must go to its tabs and pull up the splitter at the bottom. The device must be operational.
- To change any object that is assigned to the process data, you must right click on it in the process data display and force it. The device must be operational.
- To view the SDO data of a device, you must go to its tab "CoE". Depending on the window sizes it works better if push down the splitter of the process data. The device must be at least pre-operational.
- To change any SDO object, you must double click on it in the tab "CoE" and download it. The device must be at least pre-operational.

The hexadecimal index values and data of objects are written like this:

0x1234

-> Object Index 0x1234

0x1234:87

-> Object Index 0x1234, subindex 0x87

0x1234:87 = 54

-> The object at index 0x1234 and subindex 0x87 is set to 54.

0x1234:87 = 0x54

-> The object at index 0x1234 and subindex 0x87 is set to 0x54 = 84.

Some objects are write-protected, for example the ratings and the inductivity of the motor, and the serial number. If you try to change them by SDO downloads, you will get this SDO abort code:

0x08000022 - Data cannot be transferred or stored to the application because of the present device state.

If you really need to change a write-protected parameter, please contact the manufacturer of this device.

## 10.5 Storing and resetting backup parameters

All parameters, that can be stored non-volatile in the drive, have the attribute flag "backup" set in the ESI file.

Store all backup parameters:  
0x1010:01 = 0x65766173 = "save"

Store the application backup parameters 0x6000 - 0x9FFF:  
0x1010:03 = 0x65766173 = "save"

Store the manufacturer backup parameters 0x2000 - 0x5FFF:  
0x1010:04 = 0x65766173 = "save"

Load default values for all backup parameters (factory setting):  
0x1011:01 = 64616F6C = "load"

Load default values into the application backup parameters 0x6000 - 0x9FFF:  
0x1011:03 = 64616F6C = "load"

Load default values into the manufacturer backup parameters 0x2000 - 0x5FFF:  
0x1011:04 = 64616F6C = "load"

If these objects are read back by SDO, the result is 1, which indicates, that the function is available and must be triggered by the user, who must write the codes shown in the list above.

After sending a store or load command, the drive is busy for some time. While it updates the non-volatile memory, additional flags are set during reading back. During this time the logic power supply must not be turned off. Otherwise the new data may be lost!

0x1010.01 & 0x40000 -> "Pending non-volatile Save"

0x1010.01 & 0x80000 -> "Pending non-volatile Load"

Loading default values is only finished after rebooting the device, for example by turning the logic supply off and on. Loading immediately invalidates the checksums of the non-volatile stored parameters, but doesn't change the working memory. Loading the default values into it requires a reboot. Before that loading can be undone by another store operation, that recreates the invalidated checksums again.

## 10.6 Stop conditions, Digital Inputs and Hardware Limit Switches

### 10.6.1 Turning on the drive

Turning on the drive is only possible, if it has passed the self-test on startup successfully. Otherwise it will be stuck at the DS402 state "Not ready to switch on". Then the status bits  $2^6$  = "Switch-on disabled" and/or  $2^9$  = "Remote" are 0.

If a fault has occurred, it must be cleared before turning on the drive. If there are hardware failures or invalid parameters, that however must be fixed first. Otherwise the fault will immediately occur again.

"Control word" 0x6040 = 0x0000  
-> "Status word" 0x6041 = 0x0648 // switch on disabled, fault

"Control word" 0x6040 = 0x0080  
-> "Status word" 0x6041 = 0x0640 // switch on disabled, fault cleared

This sequence enables the operation and turns on the motor:

"Control word" 0x6040 = 0x0000  
-> "Status word" 0x6041 = 0x0640 // switch on disabled, fault

"Control word" 0x6040 = 0x0006  
-> "Status word" 0x6041 = 0x0621 // ready to switch on

"Control word" 0x6040 = 0x000F  
-> "Status word" 0x6041 = 0x0637 // operation enabled

### 10.6.2 Stop conditions: Priorities, Behavior

If the drive has been turned on and thus has reached the DS402 state "Operation enabled", stop conditions can be triggered by the following events. This list is sorted by the priorities, and the highest comes first:

- By a hardware failure, which causes a fatal fault. The drive goes to the DS402 state "Fault" and immediately switches off the motor. This state can be left only by a rising edge on the control word bit  $2^7$ .
- By clearing the control word bit  $2^1$  and turning the motor voltage off. The drive goes to the DS402 state "Switch-on Disabled" and immediately switches off the motor.
- By a non-fatal fault, that may be the result of hardware failures or of invalid parameters. The drive goes to the DS402 state "Fault Reaction Active", stops, then goes to the DS402 state "Fault" and switches off the motor. The reaction is selected by its option code 0x605E:

0 -> Immediately turn off the drive

1 -> Slow down at the "Profile Deceleration" 0x6084 before turning off the drive.

0xC001 -> Slow down at the "Homing Acceleration" 0x609A before turning off the drive.

2 or any other value -> Slow down at the "Quick stop Deceleration" 0x6085 before turning off the drive.

- By a loss of the connection to the fieldbus master. The reaction is selected by its option code 0x6007:

1 -> A loss of conn. triggers a non-fatal fault, that reacts according to the option code 0x605E.

2 -> A loss of conn. immediately turns off the drive.

3 -> A loss of conn. triggers a quick stop, that reacts according to the option code 0x605A.



If this option code is 5 or higher, which means staying in quick stop, this state can be left without turning off the motor by toggling the control word bit 0x80 "Clear Fault" once or twice.

0 or any other value -> A loss of conn. is ignored.

- By a following error. It may happen in any operation mode except the torque modes. The size of the "Following Error" 0x60F4 equals the "Required Position" 0x6062 minus the "Actual Position" 0x6064.

It occurs, if its absolute size exceeds the size of the "Following Error Window" for longer than the "Following Error Timeout".

It always restarts the last ramp, that was triggered by a changed target value, a command, a limit or homing switch, or an option code.

Otherwise if the drive is blocked temporarily and then released, it would shoot away ignoring all velocity and acceleration limits!

If the "Following Error Window" of DS402 is turned on by a value below 0xFFFFFFFF, size and timeout are set according to the objects 0x6065 and 0x6066, and a following error triggers the related emergency 0x8611.

In the "Profile Position Mode" or "Profile Velocity Mode" it also sets the bit 2<sup>13</sup> in the "Status word" 0x6041. This bit remains set until the current ramp is finished.

Otherwise size and timeout are switched to the manufacturer objects 0x2F0E:02 and 0x2F0E:03, and the following error becomes silent. It only restarts ramps, but doesn't send emergencies or sets its bit in the status word.

The default settings for these manufacturer objects are 1/8 revolution and 20 ms.

If the following error is not silent, the reaction is selected by the manufacturer object called "Following Error Option Code" 0x2F0E:01:

1 -> A foll.err. triggers a non-fatal fault, that reacts according to the option code 0x605E.

2 -> A foll.err. immediately turns off the drive.

3 -> A foll.err. triggers a quick stop, that reacts according to the option code 0x605A.



If this option code is 5 or higher, which means staying in quick stop, this state can be left without turning off the motor by toggling the control word bit 0x80 "Clear Fault" once or twice.

0 or any other value -> A foll.err. only sets the status word bit and sends its emergency.

While there is a following error, usually the current reaches the current limit, so that the status bit 2<sup>11</sup> is also set, at least temporarily.

Since the version from "V1.1 17-January-2017" the status bit "Following Error" is only reset, if the motor is turned on again, or if a fault is cleared, or if the option code is 0 = "Ignore" and no more following errors occurred during the timeout. It is not reset when the motor is turned off. Otherwise it would be impossible to find out, for which reason the drive went to "Fault" or turned itself off.

- By clearing the control word bit 2<sup>2</sup> and triggering a quick stop. The stop method is selected by the option code 0x605A:

5 -> Slow down at the "Profile Deceleration" 0x6084 and stay in quick stop.

6 -> Slow down at the "quick stop Deceleration" 0x6085 and stay in quick stop.

0xC005 -> Slow down at the "Homing Acceleration" 0x609A and stay in quick stop.

0 -> Immediately turn off the drive

1 -> Slow down at the "Profile Deceleration" 0x6084 before turning off the drive.

0xC001 -> Slow down at the "Homing Acceleration" 0x609A before turning off the drive.

2 or any other value -> Slow down at the "quick stop Deceleration" 0x6085 before turning off the drive.

When the drive is turned off, it goes to the state DS402 state "Switch-on Disabled". Before it can be switched back to "Operation enabled" by setting the control word bit 2<sup>2</sup> again.

- By clearing the control word bit 2<sup>2</sup> and triggering a shutdown. The drive goes to the DS402 state "Ready to switch on", stops and switches off the motor. The stop method is selected by the option code 0x605B:

0 -> Immediately turn off the drive

1 -> Slow down at the "Profile Deceleration" 0x6084 before turning off the drive.

0xC001 -> Slow down at the "Homing Acceleration" 0x609A before turning off the drive.

2 or any other value -> Slow down at the "Quick stop Deceleration" 0x6085 before turning off the drive.

- By clearing the control word bit  $2^0$  and disabling the operation. The drive goes to the DS402 state "Switched on", stops and switches off the motor. The stop method is selected by the option code 0x605C:

0 -> Immediately turn off the drive

1 -> Slow down at the "Profile Deceleration" 0x6084 before turning off the drive.

0xC001 -> Slow down at the "Homing Acceleration" 0x609A before turning off the drive.

2 or any other value -> Slow down at the "quick stop Deceleration" 0x6085 before turning off the drive.

- By setting the control word bit  $2^8$  and triggering a stop. The stop method is selected by the option code 0x605D:

1 -> Slow down at the "Profile Deceleration" 0x6084.

0xC001 -> Slow down at the "Homing Acceleration" 0x609A.

2 or any other value -> Slow down at the "Quick stop Deceleration" 0x6085.

- By the digital inputs  $2^0$  or  $2^1$ , that are connected to the hardware limit switches by wires or by SDOs or RPDOs, which are sent by the master. If the drive shall move away from the allowed range, it stops or stays stopped. Refer to the description below.

- By the target value 0 in the velocity modes, or by a position ramp in the profile position mode.

All acceleration values are clipped by the "Maximum Acceleration" 0x60C5.

All deceleration values are clipped by the "Maximum Deceleration" 0x60C6.

To detect a stop condition by bit 8 of the control word, read back the control word 0x6040.

To detect a stop condition by the digital input  $2^0$  and  $2^1$ , read back the object 0x60FD. All other stop conditions can be detected by reading the status word 0x6041.

All stop conditions are mutually exclusive, except the digital inputs  $2^0$  and  $2^1$ , that may happen at the same time as any other stop condition. In this case the drive compares the assigned decelerations and uses the highest value for the stop ramp. Thus it gets the highest priority.

### 10.6.3 Control of the Digital Inputs

By setting the digital input bits in the XOR-Mask 0x2F04:02 the polarity of the digital inputs can be reversed.

By clearing the digital input bits in the AND-Mask 0x2F04:01 the digital inputs can be blocked for testing, or for using digital inputs, that are transferred remotely from the master.

Since the software version "V1.1 20-September-2016" both masks are mappable, so that they can be written not only by SDOs, but also by RPDOs.

The resulting input signals, that have been transferred remotely from the master by SDOs or RPDOs, can be read back by 0x60FD, just like the real inputs:

```
0x2F04:01 = 0
```

```
// -> The "Digital Input AND Mask" blocks the actual digital input bits.
```

```
0x2F04:02 = <written remote digital inputs>
```

```
// -> The "Digital Input XOR Mask" negates single bits.
```

```
0x60FD = <resulting digital inputs> = <XOR mask> XOR (<AND mask> AND <actual digital inputs>)
```

```
// The evaluated "Digital Inputs" show the remotely transferred digital inputs just like real inputs.
```

#### 10.6.4 Operation of the Digital Inputs as Limit Switches

The digital input bits  $2^0$  and  $2^1$  work as lower = negative and upper = positive hardware limit switches in the "Profile Velocity Mode".

Since the software version "V1.1 20-September-2016" they also work in the "Profile Position Mode".

In other modes they have no effect, except in the "Homing Mode" according to the DS402 drive profile.

Effect of the hardware limit switches:

- If the input bit for the lower limit switch is set and the drive moves to descending positions, it stops.
- If the input bit for the upper limit switch is set and the drive moves to ascending positions, it stops.
- If both input bits are set, the drive can't move at all.

Thus it is still possible to return into the allowed range. Commands by the control word with target values, that would move the drive into the wrong direction, are ignored.

##### **Important hints:**

- Bit  $2^7$  of the object 0x607E reverses the polarity of the position.
- Bit  $2^6$  of the object 0x607E reverses the polarity of the velocity.
- The hardware limit switches always refer to the position. Therefore the lower switch must be located at a lower position than the upper one. If this is reversed, try switching the polarity flag of the position.
- The hardware limit switches and homing work properly with any combination of the polarity flags. However it is recommended to set the polarity flags for the position and the velocity equal. Otherwise the indicated values may be confusing: The velocity will be negative, if the position values are ascending, and the other way round.
- Since the software version "V1.1 7-December-2016" the limit switches use the "Homing Acceleration" 0x609A as deceleration, when they stop the drive, because the homing mode already links that parameter to the digital inputs. Before that version the limit switches used the stop deceleration, which is however not related to the digital inputs in any way.

- If hardware limit switches are used, it is recommended to prevent driving over the end of the measuring range by setting at least one of software position limits inside it. The measuring range of the position should be as large as possible with a symmetrical zero point, to avoid under- and overflows, and it must be a true divider of the measuring range of the drive encoder. Otherwise the position may jump by the remainder at some locations.

### **Example 1:**

4096 position units per revolution, measuring range = 65536 revolutions

*Resolution and measuring range of the drive encoder (read-only):*

"Position Increments per Motor Revolution" 0x608F:1/0x608F:2

.. =  $4096/1 = 2^{12}$

"Measuring range in revolutions" 0x6510

.. =  $65536 = 2^{16}$

*Required resolution 4096 "Position Units/Revolution"*

->

"Position Increments per Position Units" 0x6093:1/0x6093:2

.. =  $[0x608F:1]/4096$

.. =  $4096/4096$

->

Maximum position measuring range / "Position Units" =

.. =  $[0x6510] * ([0x608F:1]/[0x608F:2]) / ([0x6093:1]/[0x6093:2])$

.. =  $2^{(12+16)} = 2^{28} = 268435456$

.. = 0x10000000

*Required position measuring range  $2^{(12+16)}$  "Position Units" with a symmetrical zero*

->

"Min. Position Range Limit" 0x607B:1 =  $-2^{27} = -134217728$

.. = 0xF8000000

"Max. Position Range Limit" 0x607B:2 =  $+2^{27-1} = +134217727$

.. = 0x07FFFFFF

*Linear moving range at most 50% of the negative and positive measuring range:*

->

"Min. Software Position Limit" 0x607D:1  $\geq -2^{26} = -67108864$

.. = 0xFC000000

"Max. Software Position Limit" 0x607D:2  $\leq +2^{26-1} = +67108863$

.. = 0x03FFFFFF

*-"Max. Software Position Limit"  $\leq$  "Home Offset"  $\leq$  -"Min. Software Position Limit"*

->

"Home Offset" 0x607C  $\geq -[0x607D:1] = -2^{26-1} = -67108863$

and

"Home Offset" 0x607C  $\leq -[0x607D:2] = +2^{26} = +67108864$

**Example 2:**

360 position units per revolution, measuring range = 4096 revolutions

*Resolution and measuring range of the drive encoder (read-only):*

"Position Increments per Motor Revolution" 0x608F:1/0x608F:2

.. =  $4096/1 = 2^{12}$

"Measuring range in revolutions" 0x6510

.. =  $65536 = 2^{16}$

*Required resolution 360 "Position Units/Revolution"*

->

"Position Increments per Position Units" 0x6093:1/0x6093:2

.. =  $[0x608F:1]/360$

.. =  $4096/360$

->

Maximum position measuring range / "Position Units" =

.. =  $[0x6510] * ([0x608F:1]/[0x608F:2]) / ([0x6093:1]/[0x6093:2])$

.. =  $360 * 2^{12} = 1474560$

.. = 0x168000

*Required position measuring range  $2^{(12+16)}$  "Position Units" with a symmetrical zero*

->

"Min. Position Range Limit" 0x607B:1 =  $-360 * 2^{12}/2 = -737280$

.. = 0xFFFF4C000

"Max. Position Range Limit" 0x607B:2 =  $+360 * 2^{12}/2 - 1 = +737279$

.. = 0x000B3FFF

*Linear moving range at most 50% of the negative and positive measuring range:*

->

"Min. Software Position Limit" 0x607D:1  $\geq -360 * 2^{12}/4 = -368640$

.. = 0xFFFFA6000

"Max. Software Position Limit" 0x607D:2  $\leq +360 * 2^{12}/4 - 1 = +368639$

.. = 0x00059FFF

*-"Max. Software Position Limit"  $\leq$  "Home Offset"  $\leq$  -"Min. Software Position Limit"*

->

"Home Offset" 0x607C  $\geq -[0x607D:1] = -360 * 2^{12}/4 = -368640$

and

"Home Offset" 0x607C  $\leq -[0x607D:2] = +360 * 2^{12}/4 - 1 = +368639$

## 10.7 Control word and Status word

### 10.7.1 General

The "DS402 Control word" 0x6040 and the "DS402 Status word" 0x6041 are implemented according to the CANopen and EtherCAT drive profile DS402. Please refer to the official descriptions of this norm. However some details are not fixed, and the manufacturer have to decide on their own, if and how to implement them. Following chapters explain the details for the EncoTRive devices.



The basic bits always behave the same. Refer to the paragraph "Turning on the drive" in chapter 10.6 above as an example.

The behavior of other bits may depend on the operation mode.

---

### 10.7.2 Status bit 2<sup>7</sup> - Warning

This bit is set, while emergencies are pending. Refer to chapter 12.2 below for details about the emergencies and how to clear them.

### 10.7.3 Status bit 2<sup>9</sup> - Remote

This bit is set, as soon as the drive has passed the self-test on startup. While this bit is cleared, the drive can't be controlled by the fieldbus. This may be caused by a missing calibration or by a fatal hardware failure. In both cases the drive must be sent back to the manufacturer.

### 10.7.4 Status bit 2<sup>10</sup> - Target Reached

In the "Profile Position Mode" this bit is set, as soon as the position ramp is finished and the distance between the "Actual Position" 0x6064 and the "Target Position" 0x607A never exceeded the size of the "In-Position Window" 0x6067 during the "In-Position Timeout" 0x6068.

In the "Profile Velocity Mode" this bit is set, as soon as the velocity ramp is finished and the difference between the "Actual Velocity" 0x606C and the "Target Velocity" 0x60FF never exceeded the size of the "Velocity Window" 0x606D during the "Velocity Window Timeout" 0x606E.

### 10.7.5 Status bit 2<sup>11</sup> - Internal limit active

This bit is set in the "Profile Velocity Mode", if the absolute value of the "Target Velocity" 0x60FF is above the "Maximum Profile Velocity" 0x607F or the "Maximum Motor Speed" 0x6080.

This bit is set in the "Profile Position Mode", if a position ramp is started and the "Target Position" 0x607A is beyond the "Position Range" set by 0x607B, or the "Software Position Limits" set by 0x607D.

This bit is set in the "Profile Position Mode", if the "Profile Velocity" 0x6081 is above the "Maximum Profile Velocity" 0x607F or the "Maximum Motor Speed" 0x6080.

This bit is set, while the "Actual Torque generating Current" 0x6078 reaches the "Relative current Limit" 0x6073.



- The "Relative Torque Limit" 0x6072 always has the same value as the object 0x6073.
- The "Actual Torque" 0x6077 always has the same value as the object 0x6078.
- The values 0x6073 and 0x6078 both refer to the "Motor rated Current in mA" 0x6075.
- The values 0x6072 and 0x6077 both refer to the "Motor rated Torque in Nmm" 0x6076.

### 10.7.6 Status bit 2<sup>12</sup> - Set-point-acknowledge, Zero Speed

In the "Profile Position Mode" this bit is the set-point-acknowledge for the handshake that starts new position ramps.

If the drive accepts the handshake, it is set equal to the control word bit 2<sup>4</sup> "New set-point".

It is frozen,

- if the DS402 state is not "Operation enabled", but anything else like "Quick stop Active" or "Fault Reaction", or
- if the drive is stopped, or
- if a position ramp is already ongoing and the control word bit 2<sup>5</sup> "Change set point immediately" is cleared.

In the "Profile Velocity Mode" this bit is set, while the "Actual velocity" 0x606C is close to 0. The drive must move slower than the "Velocity Threshold" 0x606F during the "Velocity Threshold Timeout" 0x6070.

### 10.7.7 Status bit 2<sup>13</sup> - Following error

This bit is set in the "Profile Position Mode", if a following error occurs. For details refer to its description in the previous chapter 10.6.2 "Stop conditions: Priorities, Behavior".

The status bit "Following Error" is available in the "Profile Velocity Mode" too, instead of the "Maximum Slippage Error". That bit can only be applied to asynchronous motors, and the EncoTRive devices are always equipped with synchronous or PM motors.

PM means "Permanent magnet". PM motors are much smaller and lighter than asynchronous motors with the same output power, and can accelerate much more.

## 10.8 Profile Position Mode (operating mode 1)

At first the position factor 0x6093 should be set properly, that converts the position units into position encoder increments by multiplying. The count of increments per revolution is shown by the object 0x608F:

Position factor = (Position Encoder Resolution)\*(Required size of position units in revolutions)

Simple setup of the position factor:

Numerator = Count of position increments per revolution

Denominator = Count of position units per revolution

Also see chapter "Units" on page 56.

Then the user stroke must be set by choosing the proper position range limits 0x607B.

If it is no true divider of the encoder stroke, an emergency "ECY\_PARM\_EUSTROKDIVRMND" = 0x6356 is set.

Normally it is expected, that all user strokes have the same size, while the drive travels along the encoder stroke or encoder measuring range. But if there is a remainder, the last user stroke will have exactly the same size, so that it will be shorter than expected.

If a position ramp travels across the borders of this runt stroke, it will miss the target position according to the missing position units.

It is however possible to run the drive, if this warning was set. It can simply be cleared.

Then the parameters must be set, that describe the shape of position ramps, see chapter "Profile Position (PP)" operating mode on page 35.

If a special behavior on disabling the operation, shutdown, fault, stop, quick-stop and loss of connections is desired, the related option codes must also be set.

Now the drive can be turned on:

```
"Opmode" 0x6060 = 0x0001  
-> "OpmodeDisplay" 0x6061 = 0x0001
```

```
"Control word" 0x6040 = 0x0000  
-> "Status word" 0x6041 = 0x0640 // switch-on disabled
```

```
"Control word" 0x6040 = 0x0006  
-> "Status word" 0x6041 = 0x0621 // ready to switch on
```

```
"Control word" 0x6040 = 0x0007 // this step can be skipped  
-> "Status word" 0x6041 = 0x0623 // switched on
```

```
"Control word" 0x6040 = 0x000F  
-> "Status word" 0x6041 = 0x0637 // operation enabled
```

The drive should be turned on now. If it was rotating, it tries to stop.

Travelling to the "Target position" 0x607A is triggered by a rising edge (0->1) in STW.4 "New set-point":

```
"Control word" 0x6040 = 0x001F  
-> "Status word" 0x6041 = 0x1237
```

The drive travels to the location, that has been indicated by the "Target position" 0x607A, unless it is already travelling to another destination.

```
If the target position is reached  
-> "Status word" 0x6041 = 0x1637
```

```
"Control word" 0x6040 = 0x000F  
-> "Status word" 0x6041 = 0x637
```

```
"Control word" 0x6040 = 0x003F  
-> "Status word" 0x6041 = 0x1237
```

The drive immediately travels to the location, that has been indicated by the "Target position" 0x607A. If it is already travelling to another destination, it aborts that journey.

If the software position limits 0x607D are not tighter than the position range 0x607B, the drive always travels the shortest way, that may cross the start and end values. Then it will never travel further than half of the user stroke.

Otherwise it must avoid the start and end values, so that the travelled distance may be up to the full user stroke - 2.

If the destination is outside the range or position limits, the drive travels as far as possible, but it sets the status bit 11 "Internal limits active" in the status word 0x6041. If the torque reaches the torque limit, the drive also sets the status bit "Internal limits active".

---

The digital input bits  $2^0$  and  $2^1$  work as lower or negative and upper or positive limit switches. Refer to chapter 10.6 on page 124.

If the "In-Position Window" 0x6067 is active, the status bit "Target Reached" is not set immediately after finishing the ramp, but only if the position remains in this window for at least the "In-Position Time" 0x6068.

If the "Following Error Window" 0x6065 is active and the following error exceeds it for the following error time in 0x6066, the drive changes to the fault state, sets the status bit "Following error", and also sets the emergency "ECY\_POSCTRL\_FOLLOW" = 0x8611.



The fault reaction can be selected by its option code in 0x605E.

If the control bits "Quick-stop" or "Stop" are set, the drive stops according to the related option codes in 0x605A and 0x605D.

After clearing the fault state or recovering from quick-stop or stop, the drive does not automatically resume an aborted journey. It must be triggered again by another rising edge of the control bit "New set-point".

Sets of set-points are currently not supported.

---

## 10.9 Profile Velocity Mode (operating mode 3)

At first the velocity encoder factor 0x6094 should be set properly, that converts the speed units into velocity encoder increments by multiplying. The count of increments per revolution and second is shown by the object 0x608F:

Velocity Encoder Factor = (Velocity Encoder Resolution)\*(Required size of Speed units in revolutions/sec)

Simple setup of the velocity encoder factor for rpm as speed units:

1 rpm = 1/60 revolution/s

Numerator = Numerator of Vel. Enc. Resolution = 0x80000000

Denominator = Denominator of Vel. Enc. Resolution\*60 = 300000

Also see chapter "Units" on page 56.

The remaining set up and turning on of the drive works like the "Profile Position Mode", see chapter 10.8 on page 133.

Observe following exceptions:

The "Opmode" 0x6060 must be 3.

The "Profile Velocity" 0x6081 is ignored.

A changed "Target Velocity" 0x60FF is evaluated immediately. No control bits must be toggled. The "Target Position" 0x607A is ignored.

Instead of the "In-Position Window and Time" the "Velocity Window and Time" 0x606D and 0x606E are used. The "Velocity Window" can be disabled by setting it to 65535.

---

The digital input bits 2<sup>0</sup> and 2<sup>1</sup> work as lower or negative and upper or positive limit switches. Refer to chapter 10.6 on page 124.

The status bit 12 zero-speed is set according to the current drive speed, the "Velocity Threshold" and its timeout.

The "Velocity Threshold" can be disabled by setting it to 0.



The profile reserves the status bit 13 for the slippage error of asynchronous motors. For servo motors it is useless. For the convenience of users this bit is set again on following errors, just like in the "Profile Position Mode", see Status word on page 36.

Just as the operating mode also the target velocity is stored in the non-volatile memory, unlike any other target value. This is useful for the AUTO-SWITCH-ON function that turns the drive on without fieldbus, as soon as the power supply reaches the related voltage threshold.

The drive then can work for example as a stand-alone replacement of a spring.

---

## 10.10 Cyclic synchronous modes (operating modes 8, 9, 10)

These modes are implemented according to the Drive Profile CiA DSP-402.

The position offset 0x60B0 is evaluated only in the operating mode "Cyclic sync position mode (CSP)".

The velocity offset 0x60B1 is evaluated in the following operating modes:

- Cyclic sync position mode (CSP)
- Cyclic sync velocity mode (CSV)

The torque offset 0x60B2 is evaluated in all three operating modes:

- Cyclic sync position mode (CSP)
- Cyclic sync velocity mode (CSV)
- Cyclic sync torque mode (CST)

The target values and offsets are evaluated only on changes. Therefore it is possible in the CSP mode to send a target position and a velocity offset to travel smoothly between the position set-points. However if there are no new values and the velocity offset was not set to 0, the drive will never stop, regardless of the software position limits 0x607D.

The status bit 11 "Internal limits active" is set, if any set-point exceeds its limits, or if the torque reaches its limits.

---

The digital input bits 2<sup>0</sup> and 2<sup>1</sup> or limit switches are ignored. It is up to the responsibility of the bus master to consider them, when it calculates the target values.

Following errors are handled in the operating modes CSP, CSV and PP (Profile Position Mode). Also the status bit 13 "Following error" is set accordingly.



Decelerate with intermediate stop ramp is handled in the operating modes CSP, CSV, CST and PP and controlled by means of the control bit 2 "Quick Stop" in the control word 0x6040. The control bit 8 "Halt" is always ignored.

However currently all option codes for slowing down and stopping are ignored completely. Instead the drive is turned off immediately.

The "interpolation time period" 0x60C2 is not yet supported.

---

## 10.11 Backup-EEPROM, runtime

All data are stored both in the main EEPROM and in the backup EEPROM. If one of them fails, it is still possible to recover the data from the other one.

The total runtime of the device is stored every 3 minutes alternating in the main EEPROM and the backup EEPROM. It can be read from the object 0x6508.

Every store operation to an EEPROM triggers a complete verify of both of them. Therefore every 3 minutes the EEPROMS are completely verified, due to the updated runtime. If invalid blocks are found, the related emergencies are set.

## 10.12 Position adjustment by Homing (operating mode 6)

Since the firmware version 21-January-2016 only the standard position adjustment by the Homing operating mode is available.

For an easy readjustment after exchanging a damaged device, connect the available limit switches and the home switch to the digital inputs  $2^0$  to  $2^2$  and use a suitable homing method according to the "EtherCAT Drive Profile". Detailed explanations to the digital inputs see remark in chapter "Profile Position Mode (operating mode 1)", as from page 133.

Only the homing methods 17 to 30 and 35 are available, because the other methods require an index pulse from an incremental encoder, and the encoTRive devices are always equipped with absolute encoders.

If there are no limit switches, the homing method 35 must be used. Only for this method the drive may be turned off. The drive must stand at a known position, which is the preset value. The scaling for the position measurement must be setup properly, so that the preset value is within the measuring range.



The polarity flags for the position and the velocity must be set to their final values before starting the homing. If they are changed, the homing must be repeated!

---



In previous firmware versions a non-standard position adjustment was available too. It was sufficient to write the required value into the actual encoder position 0x6064. The zero position is immediately adjusted accordingly and stored in the non-volatile memory.

This method is disabled now for security reasons, because it is prone to accidental losses of the position adjustment, which might be a severe security risk.

---

### **Example 1:**

Sequence for the safe position adjustment by homing method 35:

"Operation mode" 0x6060 = 6 (Homing mode)

Wait until "Operation mode display" 0x6061 = 6

"Homing method" 0x6098 = 35

"Home offset" 0x607C = negative preset value (= negative known position)

"Control word" 0x6040 Bit 4 = 0.

Wait until "Status word" 0x6041 Bit 12 (Homing attained) is 0.

"Status word" 0x6041 Bit 13 (Homing Error) must be 0.

"Control word" 0x6040 Bit 4 = 1.

Wait until "Status word" 0x6041 Bit 12 (Homing attained) is 1.

->

The drive adjusts the current position to the negative home offset, toggles 0x6041 Bit 12 from 0 to 1, and stores the new position adjustment permanently.

"Status word" 0x6041 Bit 13 (Homing Error) must be 0.

**Example 2:**

Sequence for the safe position adjustment by the homing method 17

"Operation mode" 0x6060 = 6 (Homing mode)

Wait until "Operation mode display" 0x6061 = 6

"Homing method" 0x6098 = 17

"Home offset" 0x607C = negative preset value (= negative known position)

"Fast homing speed" 0x6099:1 = 1200 (rpm)

"Slow homing speed" 0x6099:2 = 60 (rpm)

"Homing acceleration" 0x609A = 6000 (rpm/sec)

"Control word" 0x6040 Bit 4 = 0.

Wait until "Status word" 0x6041 Bit 12 (Homing attained) is 0.

"Status word" 0x6041 Bit 13 (Homing Error) must be 0.

"Control word" 0x6040 Bit 4 = 1.

Wait until "Status word" 0x6041 Bit 12 (Homing attained) is 1.

->

The drive travels down at the fast speed 0x6099:1, until the lower hardware limit switch 0x60FD Bit 0 toggles from 0 to 1.

Then the drive travels up again at the slow speed 0x6099:2, until this bit toggles back from 1 to 0.

It adjusts this position to the negative home offset, toggles 0x6041 Bit 12 from 0 to 1, and stops, while the new position adjustment is stored permanently.

"Status word" 0x6041 Bit 13 (Homing Error) must be 0.

**Example 3:**

Shifting of the software position limits by the home offset

The home position is the position of the switch, which is used to finish the homing procedure by triggering the final position adjustment.

The home offset is the distance from the home switch to the zero position. If the home switch is below the zero position, the home offset is positive. Otherwise it is negative, unless the home switch is right at the zero position. Then the home offset is of course zero.

Because the software position limits always refer to the home position, they are shifted by subtracting the the home offset, before the drives clips the target position according to the corrected limits.

Because the drive will travel to the home switch during homing, it is wise to increase the measuring range, so that it includes this position too, plus a negative reserve.

It also must include the corrected min. software position limit, minus a negative reserve.

It also must include the corrected max. software position limit, plus a positive reserve.

**Example 4:**

Example with all position values in position units:

Measuring range = 4096

Negative position reserve = 200

"Home offset" 0x607C = +60

"Min. Software Position Limit" 0x607D:1 = -10

"Max. Software Position Limit" 0x607D:2 = +3100

->

"Min. Position Range Limit" 0x607B:1 = - home offset - negative reserve

.. = -260

"Max. Position Range Limit" 0x607B:2 = min. pos. range limit + measuring range - 1

.. = +3835

"Corrected Min. Software Position Limit" = min. sw. pos. limit - home offset

.. = -70

"Corrected Max. Software Position Limit" = max. sw. pos. limit - home offset

.. = +3040

Positive position reserve = max. pos. range limit - corrected max. sw. pos. limit

.. = 795

"Actual Position" 0x6064 at the zero position

.. = +0

"Actual Position" 0x6064 at the home position = -home offset

.. = -60

"Actual Position" 0x6064 at the min. sw. pos. limit = corrected min. sw. pos. limit =

.. = -70

"Actual Position" 0x6064 at the max. sw. pos. limit = corrected max. sw. pos. limit =

.. = +3040

## 10.13 Manual tuning of the position and speed regulation

### Required parameters:

"Main cycle frequency  $1/T_c=fc/Hz$ " 0x2EF4:0x1 // read-only, PWM or regulator cycle frequency.

"Write protect key" 0x2EF4:0x7 // 0x6C430000 unlocks writing regulator parameters.

"Q17 RegPosWi/Tc" 0x2F02:0x4 // Position integral part

"Q17 RegPosW1/Tc" 0x2F02:0x5 // Position proportional part

"Q17 RegPosW2/Tc" 0x2F02:0x6 // Position differential part

"Q15 RegPosW3/W4" 0x2F02:0x7 // Position forward control, must be 0 or 0x8000.

"RegPosrKS" 0x2F04:0xA // Global position amplification

Q17 and Q15 mean fixed point numbers with 17 and 15 bit after the decimal point. Therefore Q15 0x8000 is the same as the floating point value 1.

$1/T_c$  means the reciprocal value of the PWM or regulator cycle  $T_c$ . This is the main or PWM cycle frequency, usually 10000 Hz. In every cycle the regulators update their output signals, depending on the input signals. If the transfer functions of the regulators are shown as Bode gain plots, the regulator parameters  $W_i$ ,  $W_1$ ,  $W_2$ ,  $W_3$  and  $W_4$  are the corner frequencies. These are the joints of the curve segments, where the slope changes most.

### General hints:

For standard loads with a reasonable gear ratio the default parameters for the position and speed regulation should work good enough. Heavy loads with a low gear ratio or stick-slip friction may require some tuning.

If there is no brake resistor attached to the chopper output of the drive, an external brake chopper must be connected to the supply for the motor power. Otherwise the drive may not be able to brake according to the ramp parameters and overshooting is inevitable. Also an oscillating drive may cause a heavy recuperation and high overvoltages on the power supply, which may destroy electronic components.

### Recommended:

- External brake resistor 3 Ohm/150 Watt
- External brake chopper "miControl mcEr-L40"

Check if the drive is able to reach the target acceleration, speed and deceleration of the ramp settings. This can be done by activating the following error:

"FollowingErrorWindow" 0x6065 = 1/4 motor revolution

"FollowingErrorTimeout/ms" 0x6066 = 100

If following errors occur frequently, decrease the ramp settings accordingly or mount another gearbox. If the drive is physically unable to follow the reference ramp, lagging and overshooting is inevitable.

For calculating the physical limits of a drive, gearbox and load combination TR offers an Excel calculation sheet as a free download:

"EncoTRiveMotorComparisonPosLin.xls.7z"

Only change the parameters of the position regulation. Changing other manufacturer parameters may worsen the function of the drive a lot. If they are related to the current regulation, they may even destroy the drive. The write-protection exists for good reasons.

After each change of the regulator parameters the tuning should be checked. Either alternatingly stop and run the drive at the target speed, or if the position range is limited, travel between to points. Record the speed during the test and look for overshooting and wobbling. Currently the drive can not yet record the speed itself.

**A possible way to tune the position and speed regulation for heavy loads:**

Unlock the general write protection of the manufacturer parameters by writing 0x6C430000 to the "Write protect key" 0x2EF4:0x7.

Reduce the integral part to about 10% percent of the current value and set the forward control to 0.

Increase the proportional part by about 25% percent steps of the current value, until overshooting, humming or oscillating occurs. Then return to the previous value. Test the tuning by driving.

Increase the differential part by about 25% percent steps of the current value, until overshooting humming or oscillating occurs. Then return to the previous value. Test the tuning by driving.

Repeat changing the proportional and differential part several times, until there is no further improvement. Increase the integral part again, until overshooting or wobbling occurs, and return to the previous setting. Finally try to reactivate the forward control by setting it to 0x8000.

If the 16 bit range of the regulator parts is not sufficient, the global 32 bit amplification must be changed.

For calculating this value TR offers an Excel calculation sheet as a free download: "EncoTRive\_Global\_Regulator\_Amplifikation.xls.7z"

If you found better settings, save the changes for the next startup by writing 0x65766173 to "Save Manufacturer Parameters" 0x1010:0x4.

## 10.14 Gear backlash cancellation (Opmodes -48=0xD0)

Since the firmware version 6-May-2019 the new opmode -48= 0xD0 is available for gear backlash cancellation. It works exactly equal to the opmode 1, except that the target position is finally always approached in the same direction from a distance, which is set by the parameter "Gear Backlash".

The absolute value of this parameter should be somewhat larger than the actual backlash, so that the gear is pushed out of the backlash in any case. If this parameter is positive, the target position is always approached from above, otherwise always from below. If it is 0, this opmode works exactly the same as opmode 1.

In order to maintain the accuracy of the position, this final move is always executed on a drive command, even if the target position has not changed. In this case the drive moves away by the "Gear Backlash" value, stops and returns. If the drive is already approaching from the proper side, it nevertheless stops for a short moment in a distance according to the "Gear Backlash" value. Thus the final move is always performed with the same speed and acceleration.

The 2 points of the final move are indicated by the read-only parameters "Backlash target pos." 1 and 2. 2 is always equal to the actual target position. 1 is the target position plus the "Gear Backlash".

The "Backlash Trailing Pos." tries to estimate the output of a gearbox with a positive or negative backlash in the worst case. First it calculates the distance from the trailing position to the actual one. If its sign and the sign of the backlash are different, the trailing position is set equal to the actual position. Otherwise while the absolute value of the distance is less than the absolute value of the backlash, it just keeps its value. Otherwise it is set equal to the actual position minus the backlash.

By writing the "Gear Backlash" also the "Backlash Trailing Pos." is reset to the actual position.

### **Required parameters:**

"Gear Backlash/Pos.units" 0x2F0E:0x4 // must be somewhat larger than the actual backlash.

"Backlash target pos.1/Pos. units" 0x2E05:0x11 // read-only.

"Backlash target pos.2/Pos. units" 0x2E05:0x12 // read-only.

"Backlash trailing pos./Pos. units" 0x2E05:0x13 // read-only.

## 11 TwinCAT System Manager – Example program (Download)

TwinCAT 2.10 and 2.11 compatible, controls up to 30 drives simultaneously via EtherCAT.

Following functions are supported:

- Firmware update
- Automatic test
- Manuel travelling to target speed
- Manuel travelling to target speed, absolute and relative
- Example program for creation of own TwinCAT applications

Download:

[www.tr-electronic.de/f/zip/TR-EMO-SW-GB-0037](http://www.tr-electronic.de/f/zip/TR-EMO-SW-GB-0037)

## 12 Troubleshooting and diagnosis options

### 12.1 SDO error codes

In the case of an error (SDO Response CCD = 0x80), Abort SDO Transfer Request Protocol is transmitted instead of the response.

Code	Description
0x05 03 00 00	Toggle bit not alternated
0x05 04 00 00	SDO protocol timeout
0x05 04 00 01	Client/Server command invalid or unknown
0x05 04 00 05	Memory too small
0x06 01 00 00	Unsupported object access
0x06 01 00 01	Read access to an object that can only be written
0x06 01 00 02	Write access to an object that can only be read
0x06 02 00 00	Object not present in the object dictionary
0x06 04 00 41	The object cannot be mapped in the PDO
0x06 04 00 42	The quantity and length of the mapped objects exceed the PDO length
0x06 04 00 43	General parameter incompatibility
0x06 04 00 47	General incompatibility in the device
0x06 06 00 00	Access error due to a hardware error
0x06 07 00 10	Wrong data type, length of service parameters incorrect
0x06 07 00 12	Wrong data type, length of service parameters too large
0x06 07 00 13	Wrong data type, length of service parameters too small
0x06 09 00 11	Sub-index does not exist
0x06 09 00 30	Parameter value range exceeded, only during write access
0x06 09 00 31	Written parameter value too large
0x06 09 00 32	Written parameter value too small
0x06 09 00 36	Maximum value is smaller than minimum value
0x08 00 00 00	General error
0x08 00 00 20	Data cannot be transmitted or stored in the application
0x08 00 00 21	Data cannot be transmitted or stored in the application. Reason: local control
0x08 00 00 22	Data cannot be transmitted or stored in the application. Reason: current device status
0x08 00 00 23	Dynamic creation error in the object dictionary, or no object dictionary present

Table 35: SDO Abort Codes

## 12.2 EMCY error information

### 12.2.1 EncoTRive RE - Emergencies

The emergencies are stored in the "Emergency List" 0x1003. New emergencies are inserted at the begin. Thus they are sorted by their age: The 1st entry is the youngest or most recent one.

Empty entries contain the value 0. The first emergency is also shown in the "Error Code" 0x603F, which can be mapped to TPDOs.

The emergencies are cleared mostly by writing 0 to the count of emergencies 0x1003:00. However some emergencies are caused by invalid parameters and can only be cleared by setting correct values.

While emergencies are pending, the "Error Register" 0x1001 shows the stored categories or kinds of all emergencies, and in the "DS402 Status word" 0x6041 the bit 2<sup>7</sup> is set.

The current value of the error register is appended to an emergency message as the last byte.



- If the condition for an emergency occurs more than once, and it is already stored in the emergency list 0x1003, it is not sent again! Furthermore, it is not moved to the 1st place.  
If it is required to react soon on such a condition, this list should be cleared fast enough after receiving an emergency.
- If the emergency list is full, no more emergencies are added and sent!

### 12.2.2 Error register, object 0x1001

The error register specifies the cause of the error in bit code. Several errors can also be displayed simultaneously by a set bit.

The more detailed error cause can be found in bits 0 - 15 in object 0x1003, see following pages. At the time of occurrence, an error is indicated by an EMCY message.

Bit	Meaning
0	General error
1	Current
2	Voltage
3	Temperature
4	Communication error (overflow, status error)
5	Device profile-specific
6	Reserved, always 0
7	Manufacturer-specific

**Table 36: EMCY Error register, object 0x1001**

## 12.2.3 Error code, object 0x1003 (Bit 0-15)

### 12.2.3.1 General

The error list in object 0x1003 can be acknowledged (deleted) by means of bit 7 in the control word (object 0x6040), but only if the cause of the error was eliminated before. Some error codes (warnings) are also deleted automatically, for example EMCY messages which indicate undervoltage or overvoltage in the intermediate circuit. The reason for this is that the message can only be transmitted when the bus error has been eliminated.

The error code "0x0000" is transmitted for each EMCY message that is deleted. The result can be found in object 0x1003:

Error code	Meaning
0x0000	Error reset or no error
0x1000	Generic error
0x2000	Current – generic error
0x2100	Current, device input side
0x2200	Current, inside of the device
0x2300	Output current too high or sum of phase current $\neq 0$ : short circuit, overspeed
0x230A	Negative edge on power drive protection input A disabled PWM outputs 1-3
0x230B	Negative edge on power drive protection input B disabled PWM outputs 4-6
0x3000	Voltage – generic error
0x3100	Voltage, mains
0x3200	Voltage, inside of the device
0x3210	DC link overvoltage: wrong drive power supply or chopper can't eliminate whole braking current
0x3220	DC link overvoltage: too weak or missing drive power supply or lines too long
0x3230	Output load error: no load connected or all drivers destroyed
0x32FD 0x32FE 0x32FF	Offset voltages for the phase current inputs are too far away from the ADC midrange (hardware failure, missing resistors )
0x3300	Voltage, output
0x3311	Phase U, output overvoltage: phase U interrupted or phase U driver destroyed
0x3312	Phase V, output overvoltage: phase V interrupted or phase U driver destroyed
0x3313	Phase W, output overvoltage: phase W interrupted or phase U driver destroyed
0x4000	Temperature – generic error
0x4100	Temperature, ambient
0x4200	Temperature, inside of the device
0x4210	Shutdown due to high temperature of device (CPU)
0x4280	Shutdown due to low temperature of device (CPU) or sensor failure
0x4290	Warning due to high or low temperature of device (CPU)
0x4310	Shutdown due to high temperature of drive (power stages)
0x4380	Shutdown due to low temperature of drive (power stages) or sensor failure
0x4390	Warning due to high temperature of drive (power stages)

...

...

0x5000	Hardware – generic error
0x5110	Low logic supply voltage: power supply too weak or lines too long
0x5510	RAM error
0x5520	EPROM error (Flash)
0x5530	EEPROM access error
0x6000	Software – generic error
0x6010	Software Reset: CPU crash due to EMI or firmware bug
0x6100	Software, inside of the device
0x6200	Software, user
0x6300	Software, data set
0x6310	Loss of parameters: fatal checksum failure in EEPROM
0x6320	General parameter error: parameter from nonvolatile storage or SDO/PDO download out of range
0x6343	Warning: A position ramp could not be started, because it would last longer than the maximum ramp duration. Standard = 3600 sec.
0x6344	The calculation of the position ramp couldn't be finished in time, before the synchronization segment expired.
0x6345	The requested operating mode is not supported.
0x6346	The limit for the torque generating current (0x6073) had to be reduced, in order to avoid exceeding the ADC current measuring range.
0x6347	The limit for the rotor field compensating current had to be reduced, in order to avoid exceeding the ADC current measuring range.
0x6348	An acceleration or deceleration limit was too low. It was increased to 27.9 rpm internally in order to avoid overflows when calculating the duration of velocity ramps.
0x6349	DS-402 stroke = 0x607B.max – 0x607B.min+1 must have at least 2 position units.
0x6350	Position factor is too small: at most $2^{28}$ position units per revolution are allowed, so that center and borders of position units can be safely distinguished.
0x6351	Velocity encoder factor is too small: at most 1 speed unit/increment is allowed.
0x6352	Acceleration factor is too small: at most 1 acceleration unit/increment is allowed.
0x6353	Position factor or stroke is too small: user stroke must have at least 2 position units.
0x6354	Position factor or stroke is too large: the user stroke is larger than the encoder stroke: electronic gearbox required
0x6355	Position factor or stroke is too large: the user stroke exceeds $2^{32}$ revolutions.
0x6356	Warning: the user stroke is no true divider of the encoder stroke, but there is a remainder: Consecutive strokes may have different end-values.
0x6357	Warning: the user stroke is no true divider of the encoder stroke, but there is a remainder above the remainder threshold: Consecutive strokes may have different end-values.
0x6358	The overlapping range of the position range 0x607B and the position limits 0x607D must have at least 2 position units.
0x6359	The preset value is outside of the position range.

...

...

0x63FE	A ratio had to be corrected, because it was infinite or 0. Numerator and denominator may have both changed.
0x63FF	A range had to be corrected, because the upper limit was not larger than the lower one. The stroke equals 1 + upper limit - lower limit and must have at least 2 increments. Lower and upper limit may have both changed.
0x7000	Additional modules – generic error
0x7300	Main disk sensor failed: too much consecutive read errors -> fatal fault
0x731F	Gear disk sensors failed: too much consecutive read errors -> fault if not calibrating
0x7320	Position sensor failed: absolute and incremental position doesn't fit. Possible causes: missing calibration, broken gear -> fault if not calibrating If the offset of a satellite disk exceeds the adjustment tolerance, an emergency warning 0x7320+n is sent, where n is the satellite number. This means that the gear disk calibration should be repeated. But unless there is an emergency 0x7300 or 0x7320, the current position or velocity ramp can be finished safely.
0x732F	Warning: the gear disks have only been calibrated quickly. Then the function of the gear can't be guaranteed, because the full measuring range obviously has not been tested, and exceeding the tolerances is more probably due to the poor accuracy of the quick calibration.
0x7330	If not calibrating the gear disks: Bad or missing calibration of the gear disks. -> fault if not calibrating anything. During the gear disk calibration: AS5040/45 linearity error due to a magnetic field that is out of range: missing magnet, magnet too is close or too far away. -> fault Actually the emergencies 0x7330+n are sent, where n is the gear disk number from 1 to GEARDISKS_MAX. 0 is the main disk.
0x73F0	If not calibrating the gear disks: Bad or missing calibration of the gear disks. -> fault if not calibrating anything. During the gear disk calibration: AS5040/45 linearity error due to a magnetic field that is out of range: missing magnet, magnet too is close or too far away. -> fault Actually the emergencies 0x7330+n are sent, where n is the gear disk number from 1 to GEARDISKS_MAX. 0 is the main disk.
0x73E0	Warning: the gear disks have only been calibrated quickly. Then the function of the gear can't be guaranteed, because the full measuring range obviously has not been tested, and exceeding the tolerances is more probably due to the poor accuracy of the quick calibration.
0x8000	Monitoring – generic error
0x8100	Monitoring, communication – generic error
0x8110	Monitoring, CAN overrun (loss of messages)
0x8120	Monitoring, CAN switched to error passive
0x8130	Monitoring, Life-Guard error or Heartbeat error
0x8140	Monitoring, recovered from bus off
0x8150	Monitoring, collision of the CAN-ID
0x8200	Protocol error – generic error

...

...

0x8210	Protocol error, PDO not processed due to length error
0x8220	Protocol error, PDO length exceeded
0x8611	Position control - following error: motor blocked, motor load or required acceleration too high
0x9000	External error – generic error
0xF000	Additional functions – generic error
0xFF00	Device specific – generic error
0xFF80	Command can only be executed if the drive operation is enabled.
0xFF81	Command can only be executed if the drive is switched off.
0xFF82	Homing mode is not supported.

**Table 37: EMCY Error code, object 0x1003**

## 13 EncoTRive RE - Accessories, Test Setup

### 13.1 Power supply cables with Solenoid Valve or M23 Connectors

*Professional high quality shielded power supply cables*

These cables are meant for connecting SELV power supplies with 4 mm banana jacks or terminals to the drives. They are ready for use according to the pin assignments of the EncoTRive devices, including a green-yellow line for PE.

Due to the shielding and the crimped contacts they are suitable for operation with high loads, endurance tests and immunity and radiation tests according to the CE requirements.

TR-Material number, length and description:

P 0227-321	6m	7x2.5mm <sup>2</sup> , shielded, banana plugs 4 mm to M23 Speedcon 7-pin socket
P 0227-322	1m	4x2.5mm <sup>2</sup> , shielded, adapter cable M23 to solenoid valve 3+1 pin socket.
P 0227-323	2m	7x2.5mm <sup>2</sup> , shielded, banana plugs 4 mm to M23 Speedcon 7-pin socket
P 0227-324	6m	7x2.5mm <sup>2</sup> , shielded, crimped lugs to M23 Speedcon 7-pin socket.
P 0227-326	2m	7x2.5mm <sup>2</sup> , shielded, crimped lugs to M23 Speedcon 7-pin socket.

*Tinkered sloppy unshielded power supply cable with 3+1 pin solenoid valve female connector*

These cables are meant for connecting SELV power supplies with 4 mm banana jacks to the drives. They must be tinkered by the user himself and are only applicable for quick tests on the the workbench without heavy loads! Due to the missing cable shield the rig will exceed the limits for radiated HF noise according to "EN 61800-3 Category C2", "CISPR 11" and "CISPR 16-1"!

TR-Material number, length/count and description:

64200161	2m	Cable "Lapp Ölflex" 3x 1.5mm <sup>2</sup> , unshielded, no PE!.
620001287	1	Banana plug 4 mm black for the common ground inlet.
620001288	2	Banana plug 4 mm red for the logic and power supply inlet.
620001364	1	Solenoid Valve Female Connector with 3+1 pins.

### 13.2 Test parameters for "EncoTRive RELC"

All parameters have the default values, with the following exceptions:

Max. current = 400 per thousand of the rated current

Max. profile velocity = 4100 vel. units

Profile velocity = 1200 vel. units

Position factor = 0x1000/360 (1 pos.unit = 1 degree)

Position range = -1024\*360 .. +1024\*360-1 = -368640..+368639 = 0xFFFA6000..0x00059FFF

Quickstop option code = 5

## 14 EtherCAT Conformance Test Tool 1.20.80.0

The EtherCAT Conformance Test Tool or ECTT with the version 1.20.80.0 is a little bit buggy; so it needs some tricks for running the check with EncoTRive devices:

- The EncoTRive device must have a firmware, that is compiled with the define ECTT\_STRINGFIX. This clips the length of all EtherCAT string objects to 25 characters. Otherwise the upload of the Online Object Directory will fail.
- If the TwinCAT system is running, stop it with the symbol in the taskbar. Otherwise the ECTT can access the LAN port for the EtherCAT bus.
- Copy the ESI files for the EncoTRive devices into the ECTT subdirectory "DeviceDescriptions".
- The EncoTRive device, that shall be tested, is call DUT = Device under Test. Now it must be attached to the EtherCAT bus and powered up.
- Start the ECTT. Add the proper EtherCAT master device and select the proper LAN adapter. Scan the boxes. The DUT should appear below the master in the tree-view to the left. Select it, so that its tabs for communication and testing appear to the right.
- Select the tab of the device and below the tab "Startup". Enter the SDO commands for setting up the sync masters and PDOs. Otherwise the "TF-2300\_AL CoE SDO information test" will fail, because it can't switch the device to the states SAFE-OP and OP.

SDO startup commands required for EncoTRive RELC:

```
T = Transition: PS = PreOp to SafeOp
I = Index      (hexadecimal, 2 byte)
S = Subindex  (hexadecimal, 1 byte)
CA = Complete Access: f = false, t = true
Data          (hexadecimal, list of bytes - little endian)
```

```
T, I, S, CA, Data // Comment
PS, 0x1600, 0, t, 0200100040602000ff60 // set RPDO1
PS, 0x1601, 0, t, 0000 // set RPDO2
PS, 0x1A00, 0, t, 02001000416020006460 // set TPDO1
PS, 0x1A01, 0, t, 0000 // set TPDO2
PS, 0x1c12, 0, t, 01000016 // set SM2
PS, 0x1c13, 0, t, 0100001A // set SM3
```

Complete access means, that a whole array or record is written at one time, and not element by element.

Save the project file. If you load it later, you can repeat the tests without typing the startup commands again.

- 
- Select the tab of the device and below the tab "CoE Online".  
Select the dictionary type "Online (SDO info)", the object list type "All objects", and press "Update list".  
The dictionary is uploaded now and appears in the object list below, but the values are still empty.
  - Wait until the progress bar at the bottom of the window disappears, then press "Upload all". Now the values also appear in the object list.  
Without these values "TF-2300\_AL CoE SDO information test" will fail to assign the PDOs to the sync masters and the PDO mappings.
  - Select the "TF-2300\_AL CoE SDO information test" in the tree-view to the left. Select the tab "Test" to the right and press the button "Start".  
If you expand the tree-view, all green circles under this test should become green after a while.

All other test doesn't require special preparations or firmware versions. You may run them at any time in any sequence.