



# **CANopen Reference Manual**

## **Stepper Motor Positioning Controls and Plug & Drive Motors**

NANOTEC ELECTRONIC GmbH & Co. KG  
Gewerbestraße 11  
D-85652 Landsham near Munich, Germany

Tel. +49 (0)89-900 686-0  
Fax +49 (0)89-900 686-50  
[info@nanotec.de](mailto:info@nanotec.de)

---

## Editorial

© 2010

**Nanotec<sup>®</sup> Electronic GmbH & Co. KG**

Gewerbestraße 11

D-85652 Landsham / Pliening, Germany

Tel.: +49 (0)89-900 686-0

Fax: +49 (0)89-900 686-50

Internet: [www.nanotec.de](http://www.nanotec.de)

All rights reserved!

MS-Windows 2000/XP/Vista are registered trademarks of Microsoft Corporation.

### Version/Change overview

Version	Date	Changes
2.2	08/20/2010	

## About this manual

### Contents

This document contains a brief overview of the most important functions of the Nanotec stepper motor positioning controls and Plug & Drive motors as well as their use via CANopen.



### Important information

Nanotec<sup>®</sup> reserves the right to make technical alterations and further develop hardware and software in the interests of its customers to improve the function of this product without prior notice.

For criticisms, proposals and suggestions for improvement, please contact the above address or send an email to: [info@nanotec.com](mailto:info@nanotec.com)

### Additional manuals

Please also note the following manuals from Nanotec:

<b>NanoCAN User Manual</b>	Configuration of the CAN communication for CANopen-capable controllers with the NanoCAN software	
<b>Programming manual</b>	Controller programming <ul style="list-style-type: none"> <li>• Command reference</li> <li>• NanoJ</li> <li>• COM interface</li> </ul>	
<b>Technical manuals</b>	Connection and commissioning of stepper motor controls or Plug & Drive motors	

The manuals are available for download at [www.nanotec.de](http://www.nanotec.de).

## Contents

<b>1</b>	<b>Commissioning</b> .....	<b>8</b>
1.1	Configuration of CAN communication.....	8
1.1.1	Rotary switch for CAN node ID .....	8
1.1.2	Configuration with the CANopenDemo software .....	9
1.1.3	Configuration via CAN objects .....	11
1.2	Motor configuration .....	12
<b>2</b>	<b>Getting started in CANopen mode</b> .....	<b>14</b>
2.1	Overview .....	14
2.2	Starting CANopen node.....	14
2.3	Requesting CANopen node status.....	15
2.4	Switching on the power drive .....	15
2.5	Selecting the operation mode .....	17
2.6	Starting a travel.....	17
2.7	Specifying a new end position (0x12345) .....	18
<b>3</b>	<b>Control and status word</b> .....	<b>19</b>
3.1	Introduction .....	19
3.2	Control word (SDO 0x6040).....	19
3.3	Status word (SDO 0x6041) .....	20
<b>4</b>	<b>Profile Position Mode (PP)</b> .....	<b>22</b>
4.1	Introduction .....	22
4.2	Control word (SDO 0x6040).....	22
4.3	Status word (SDO 0x6041) .....	23
4.4	Quick Stop Option Code (SDO 0x605A).....	24
4.5	Position Demand Value (SDO 0x6062) .....	24
4.6	Target Position (SDO 0x607A) .....	24
4.7	Home Offset (SDO 0x607C) .....	24
4.8	Profile Velocity (SDO 0x6081) .....	25
4.9	End Velocity (SDO 0x6082) .....	25
4.10	Profile Acceleration (SDO 0x6083).....	25
4.11	Profile Deceleration (SDO 0x6084).....	25
4.12	Quick Stop Deceleration (SDO 0x6085) .....	26
4.13	Motion profile Type (0x6086) .....	26
<b>5</b>	<b>Homing Mode</b> .....	<b>27</b>
5.1	Control word (0x6040).....	27
5.2	Status word (0x6041).....	27
5.3	Homing Method (0x6098) .....	28
5.4	Homing Speeds (0x6099) .....	30
5.5	Homing Acceleration (0x609A) .....	31

<b>6</b>	<b>Velocity Mode</b> .....	<b>32</b>
6.1	Control word (0x6040).....	32
6.2	Status word (0x6041).....	32
6.3	VL target velocity (0x6042) .....	33
6.4	VL velocity demand (0x6043) .....	33
6.5	VL velocity actual value (0x6044) .....	33
6.6	VL velocity min max amount (0x6046).....	33
6.7	VL velocity acceleration (0x6048).....	34
6.8	VL velocity deceleration (0x6049).....	34
6.9	VL velocity quick stop (0x604A).....	34
6.10	VL dimension factor (0x604C) .....	35
<b>7</b>	<b>Interpolated Position Mode</b> .....	<b>36</b>
7.1	Introduction .....	36
7.2	Chronological progress .....	36
7.3	Control word (SDO 0x6040).....	37
7.4	Status word (SDO 0x6041) .....	38
7.5	Interpolation Sub-Mode Select (SDO 0x60C0).....	38
7.6	Interpolation Data Record (SDO 0x60C1) .....	38
7.7	Interpolation Time Period (0x60C2).....	39
7.8	Interpolation Data Configuration (0x60C4) .....	39
7.9	Recommendations for PDO Mapping .....	41
7.10	Information about commissioning .....	42
<b>8</b>	<b>SDOs that are not mode-specific</b> .....	<b>43</b>
8.1	Digital inputs (SDO 0x60FD).....	43
8.2	Digital outputs (SDO 0x60FE Subindex 1) .....	43
8.3	Digital outputs bitmask (SDO 0x60FE Subindex 2).....	44
8.4	Polarity (0x607E).....	44
8.5	Producer Heartbeat Time (0x1017) .....	44
8.6	DEVICE TYPE (0x1000) .....	45
8.7	ERROR REGISTER (0x1001) .....	45
8.8	COB_ID_EMERGENCY_MESSAGE (0x1014) .....	46
8.9	IDENTITY_OBJECT_VENDOR_ID (0x1018) .....	46
8.10	RPDO1_COMMUNICATION_PARAMETER (0x1400).....	47
8.11	RPDO2_COMMUNICATION_PARAMETER (0x1401).....	47
8.12	RPDO3_COMMUNICATION_PARAMETER (0x1402).....	48
8.13	RPDO4_COMMUNICATION_PARAMETER (0x1403) #.....	49
8.14	RPDO1_MAPPING (0x1600).....	49
8.15	RPDO2_MAPPING (0x1601).....	50
8.16	RPDO3_MAPPING (0x1602).....	50
8.17	RPDO4_MAPPING (0x1603).....	51

8.18	TPDO1_COMMUNICATION_PARAMETER (0x1800).....	51
8.19	TPDO2_COMMUNICATION_PARAMETER (0x1801).....	52
8.20	TPDO3_COMMUNICATION_PARAMETER (0x1802).....	53
8.21	TPDO4_COMMUNICATION_PARAMETER (0x1803).....	55
8.22	TPDO1_MAPPING (0x1A00).....	56
8.23	TPDO2_MAPPING (0x1A01).....	56
8.24	TPDO3_MAPPING (0x1A02).....	57
8.25	TPDO4_MAPPING (0x1A03).....	57
8.26	ERROR_CODE (0x603F).....	58
8.27	VL_DIMENSION_FACTOR (0x604C).....	58
8.28	MODES_OF_OPERATION (0x6060).....	58
8.29	MODES_OF_OPERATION_DISPLAY (0x6061).....	59
8.30	POSITION_ACTUAL_INTERNAL_VALUE (0x6063).....	59
8.31	POSITION_ACTUAL_VALUE (0x6064).....	60
8.32	FOLLOWING_ERROR_WINDOW (0x6065).....	60
8.33	FOLLOWING_ERROR_TIME_OUT (0x6066).....	60
8.34	POSITION_WINDOW (0x6067).....	60
8.35	POSITION_WINDOW_TIME (0x6068).....	61
8.36	POS_RANGE_LIMIT (0x607B).....	61
8.37	SW_POS_LIMIT (0x607D).....	61
8.38	POSITION_ENCODER_RESOLUTION (0x608F).....	62
8.39	GEAR_RATIO (0x6091).....	62
8.40	FEED_CONSTANT (0x6092).....	63
8.41	MAX_ACCEL (0x60C5).....	63
8.42	MAX_DECEL (0x60C6).....	64
8.43	SUPPORTED_DRIVE_MODES (0x6502).....	64
8.44	Boolean Dummy Object (0x0001).....	64
8.45	Integer8 Dummy Object (0x0002).....	65
8.46	Integer16 Dummy Object (0x0003).....	65
8.47	Integer32 Dummy Object (0x0004).....	65
8.48	Unsigned8 Dummy Object (0x0005).....	65
8.49	Unsigned16 Dummy Object (0x0006).....	66
8.50	Unsigned32 Dummy Object (0x0007).....	66
<b>9</b>	<b>Analog Input .....</b>	<b>67</b>
9.1	Overview .....	67
9.2	Analog Input (0x6401).....	67
9.3	Analog Global Interrupt Enable (0x6423).....	68
9.4	Analog Input Interrupt Upper Limit (0x6424).....	68
9.5	Analog Input Interrupt Lower Limit (0x6425).....	69
9.6	Analog Input Interrupt Delta (0x6426).....	69

9.7	Analog Input Interrupt negative Delta (0x6427) .....	70
9.8	Analog Input Interrupt positive Delta (0x6427) .....	70
<b>10</b>	<b>Manufacturer-specific CAN Objects.....</b>	<b>72</b>
10.1	STEP_MODE (0x2000).....	72
10.2	ENABLE_CL (0x2001).....	72
10.3	CL_CONFIGURATION (0x2002).....	73
10.4	A/D Converter (0x2003).....	76
10.5	Current control (0x2004).....	76
10.6	CAN enable and baud rate (0x2005).....	78
10.7	Motor Pole Pairs (0x2006).....	78
10.8	Brake Wait Time (0x2007).....	79
10.9	Milliseconds Input Debounce Time (0x2008).....	79
10.10	Node ID (0x2009).....	80
10.11	CL is enabled (0x200A).....	80
10.12	CL POSCNT Offset (0x200B).....	80
10.13	CL load angle curve (0x200C).....	81
10.14	Encoder rotation direction change (0x200D).....	82
10.15	DSPdrive current controller parameter (0x200E).....	82
10.16	Speed mode controller type (0x200F).....	83
10.17	External reference run IO (0x2010).....	84
10.18	COB ID SYNC message (0x1005).....	84
10.19	Hardware Version (0x1009).....	85
10.20	Software Version (0x100A).....	85
10.21	Guard Time (0x100C).....	86
10.22	Life Time Factor (0x100D).....	86
10.23	Store Parameters (0x1010).....	86
10.24	Restore Parameters (0x1011).....	87
10.25	Restore Parameters (0x608D).....	87
10.26	Restore Parameters (0x608E).....	87
<b>11</b>	<b>Process Data Objects (PDO).....</b>	<b>88</b>
11.1	Purpose of the PDFOs.....	88
11.2	PDO Mapping.....	88
11.3	Dummy Objects.....	90
<b>12</b>	<b>Commissioning via RS485.....</b>	<b>91</b>
<b>13</b>	<b>Appendix: Abbreviations used.....</b>	<b>92</b>
<b>14</b>	<b>Appendix: Possible error messages.....</b>	<b>93</b>

# 1 Commissioning

## 1.1 Configuration of CAN communication

### 1.1.1 Rotary switch for CAN node ID

#### Function

In the case of controllers with a rotary switch, how the controller determines its baud rate and node ID can be selected with the rotary switch. A hexadecimal number is set with the rotary switches that is made up of the 1st digit (of the left rotary switch) and the 16th digit (of the right rotary switch).

The node ID of a CANopen node can be set to between 1 and 127. The controller reads it from the rotary switches or from the EEPROM when the voltage is applied.

For more details on writing the node ID and the baud rate via CAN objects, see section 1.1.3.

#### Settings

The dependency of the node ID and the baud rate on the value set on the rotary switch are listed in the following table.

<b>Rotary switch value (decimal, SMCI47-S)</b>	<b>0</b>	<b>1-127</b>	<b>128</b>	<b>129-255</b>
<b>Rotary switch value (hex, SMCI47-S)</b>	<b>0x00</b>	<b>0x01-0x7F</b>	<b>0x80</b>	<b>0x81-0xFF</b>
<b>Rotary switch value (decimal, PD4-N)</b>	<b>0</b>	<b>1-7</b>	<b>8</b>	<b>9-15</b>
<b>Rotary switch value (hex, PD4-N)</b>	<b>0x0</b>	<b>0x1-0x7</b>	<b>0x8</b>	<b>0x9-0xF</b>
Node ID of rotary switch value		X		X-128 or X-8
Node ID from EEPROM	X		X	
Baud rate fixed to 1Mbaud	X	X		
Baud rate from EEPROM			X	X

#### Example

If the left-hand rotary switch is set to 1 and the right-hand rotary switch is set to 2, this results in the number  $16 \cdot 2 + 1 = 33$ .

Here the baud rate is set to 1 Mbaud and the node ID is 33.

#### PD4-N

The PD4-N has only one rotary switch. The node ID can, therefore, only be set in the range from 1 to 7. Other node IDs can only be set via SDO.

#### SMCI12

The SMCI12 has no rotary switch due to the confined space conditions. The node ID and baud rate can only be set via SDO.



## 1.1.2 Configuration with the CANopenDemo software

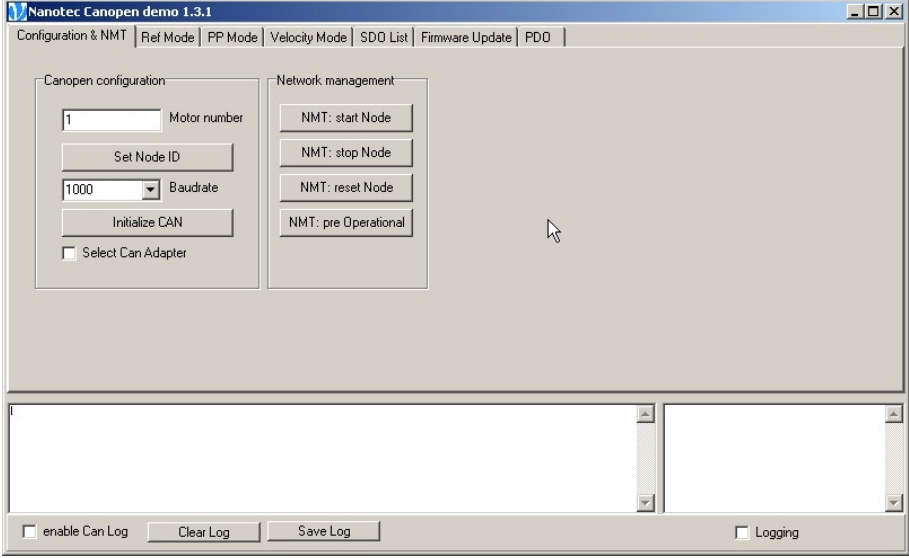
### CANopenDemo

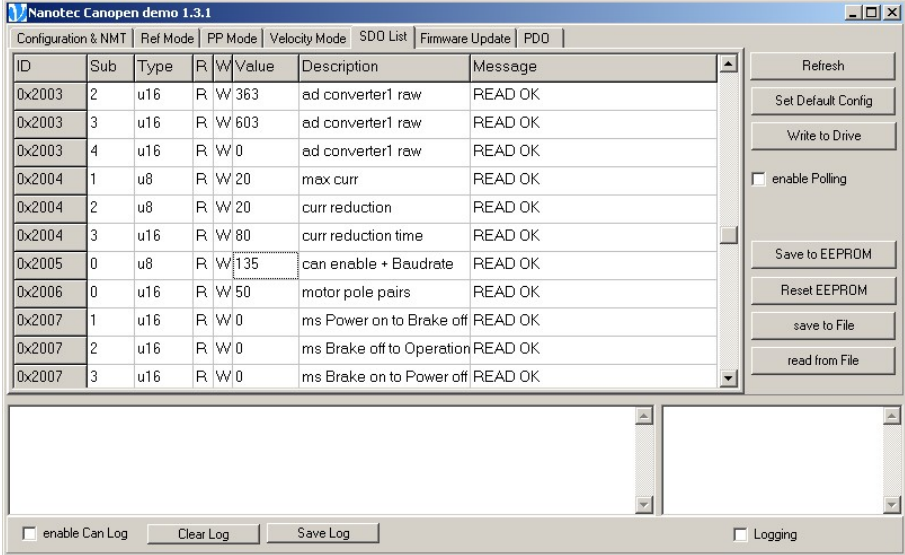
With the aid of the CANopenDemo software from Nanotec, the CAN communication and communication parameters of the stepper motor positioning controls can be set up in a simple way on a PC with CAN interface .

### Procedure

The motor controller is connected to a PC with CAN interface via CAN.

Proceed as follows when putting a motor controller into operation with CANopenDemo:

Step	Action
1	Start CANopenDemo.
2	<p>On the (&lt;Configuration &amp; NMT&gt;) tab, enter the set node ID on the motor controller ("Motor number" field). The baud rate is set to 1Mbaud as standard.</p> 
3	Press the <Initialize CAN> button to accept the node ID and to initialize the CAN device (e.g. PCI board).

Step	Action																
4	<p>Press the &lt;Reset EEPROM&gt; button on the &lt;SDO List&gt; tab to write the default values to the EEPROM.</p> <p>Press the &lt;Refresh&gt; button to display these values in the table.</p> 																
5	<p>Set the baud rate of the CAN interface.</p> <p>The possible baud rates and the corresponding value of the SDO are listed in the following table.</p> <table border="1" data-bbox="518 1093 1422 1451"> <thead> <tr> <th>Baud rate</th> <th>Value SDO 0x2005, Sub 0x0</th> </tr> </thead> <tbody> <tr> <td>10 kbaud</td> <td>Not possible</td> </tr> <tr> <td>20 kbaud</td> <td>130</td> </tr> <tr> <td>50 kbaud</td> <td>131</td> </tr> <tr> <td>125 kbaud</td> <td>132</td> </tr> <tr> <td>250 kbaud</td> <td>133</td> </tr> <tr> <td>500 kbaud</td> <td>134</td> </tr> <tr> <td>1000 kbaud (default setting)</td> <td>135</td> </tr> </tbody> </table> <p>In the above mentioned example (value of the rotary switch is 33) the baud rate is set permanently to 1MBaud. Therefore, the value 135 must be entered in SDO 0x2005.</p>	Baud rate	Value SDO 0x2005, Sub 0x0	10 kbaud	Not possible	20 kbaud	130	50 kbaud	131	125 kbaud	132	250 kbaud	133	500 kbaud	134	1000 kbaud (default setting)	135
Baud rate	Value SDO 0x2005, Sub 0x0																
10 kbaud	Not possible																
20 kbaud	130																
50 kbaud	131																
125 kbaud	132																
250 kbaud	133																
500 kbaud	134																
1000 kbaud (default setting)	135																
6	<p>Press the &lt;Write to Drive&gt; button in order to adopt the entered value in the control.</p> <p>Press the &lt;Save to EEPROM&gt; button to save the entered value in the EEPROM. Only by saving it in the EEPROM is the value still present even after the controller is switched on again.</p>																
7	<p>Optional: Set the rotary switch of the controller (if present) to a value greater than 128 (0x80) if a baud rate other than 1MBaud is to be active.</p>																
8	<p>Press the &lt;NMT: reset Node&gt; button on the &lt;Configuration &amp; NMT&gt; tab to restart the controller.</p>																
9	<p>Adjust the baud rate and node ID in NanoCAN to the values configured in the controller.</p>																

### 1.1.3 Configuration via CAN objects

#### Procedure

Proceed as follows to put a motor controller into operation using CAN objects:

Step	Action														
1	<p>Set the value 1 (left switch to 0, right switch to 1) with the two rotary switches.</p> <p>This sets the baud rate permanently to 1MBaud and the node ID to 1. Invalid values for the baud rate in the EEPROM are thus ignored.</p>														
2	<p>Make default settings of the firmware. To do so, write the string "load" in the "restore all default parameters" SDO (0x1011 Subindex 1).</p> <table border="1"> <thead> <tr> <th>COB ID</th> <th>Data bytes</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>601</td> <td>23 11 10 01 6C 6F 61 64</td> <td>Load defaults</td> </tr> <tr> <td>581</td> <td>60 11 10 01 00 00 00 00</td> <td>Response: OK (takes up to 5 seconds)</td> </tr> </tbody> </table> <p>All settings are reset to the presets defined in the EDS, with the exception of the object for setting the baud rate (see Step 4).</p>	COB ID	Data bytes	Description	601	23 11 10 01 6C 6F 61 64	Load defaults	581	60 11 10 01 00 00 00 00	Response: OK (takes up to 5 seconds)					
COB ID	Data bytes	Description													
601	23 11 10 01 6C 6F 61 64	Load defaults													
581	60 11 10 01 00 00 00 00	Response: OK (takes up to 5 seconds)													
3	<p>Set the node ID with rotary switches.</p> <p>The node ID of a CANopen node can be set to between 1 and 127. This node ID is read by the controller from the rotary switches when the voltage is applied.</p>														
4	<p>If the set value of the rotary switch results in a number between 1 and 127, the CAN baud rate is set to 1MBaud.</p> <p>To set a different baud rate: Set the required baud rate via a CANboard with the aid of a service data object (SDO).</p> <p>The object to be written has the SDO ID 0x2005 subindex 0x0, data type unsigned8.</p> <p>The following table provides information about the baud rates that can be set:</p> <table border="1"> <thead> <tr> <th>Baud rate</th> <th>Value for SDO 0x2005 Sub 0x0, type unsigned8</th> </tr> </thead> <tbody> <tr> <td>20 kbaud</td> <td>130</td> </tr> <tr> <td>50 kbaud</td> <td>131</td> </tr> <tr> <td>125 kbaud</td> <td>132</td> </tr> <tr> <td>250 kbaud</td> <td>133</td> </tr> <tr> <td>500 kbaud</td> <td>134</td> </tr> <tr> <td>1000 kbaud</td> <td>135</td> </tr> </tbody> </table>	Baud rate	Value for SDO 0x2005 Sub 0x0, type unsigned8	20 kbaud	130	50 kbaud	131	125 kbaud	132	250 kbaud	133	500 kbaud	134	1000 kbaud	135
Baud rate	Value for SDO 0x2005 Sub 0x0, type unsigned8														
20 kbaud	130														
50 kbaud	131														
125 kbaud	132														
250 kbaud	133														
500 kbaud	134														
1000 kbaud	135														
5	<p>Write the value 0x65766173 (unsigned32) into the SDO 0x1010 subindex 0x1 in order to save the value in the internal EEPROM of the controller.</p> <p>It takes approx. 5 seconds until the firmware has transferred all settings into the EEPROM and has sent the SDO response (see Step 2).</p>														
6	<p>Disconnect the controller from the power supply.</p>														
7	<p>Set the rotary switch to the value Node ID + 128 so that the saved value is adopted as the baud rate.</p> <p>If values other than the values listed in the table are set for the baud rate, the controller does respond via the CAN interface but does not accept any travel instructions.</p>														

Step	Action									
8	<p>Save the settings in the EEPROM. To do so, write the value (u32)0x65766173 into the object 0x1010 Subindex 1 (the value corresponds to the string "save").</p> <table border="1"> <thead> <tr> <th>COB ID</th> <th>Data bytes</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>601</td> <td>23 10 10 01 73 61 76 65</td> <td>Save all settings</td> </tr> <tr> <td>581</td> <td>60 10 10 01 00 00 00 00</td> <td>Response: OK</td> </tr> </tbody> </table>	COB ID	Data bytes	Description	601	23 10 10 01 73 61 76 65	Save all settings	581	60 10 10 01 00 00 00 00	Response: OK
COB ID	Data bytes	Description								
601	23 10 10 01 73 61 76 65	Save all settings								
581	60 10 10 01 00 00 00 00	Response: OK								

## 1.2 Motor configuration

### General information

Before commissioning the motor controller, the motor parameters in the controller must be adapted to the respective connected motor if they differ from the default settings.

These settings include:

- Phase and quiescent current of the motor (0x2004)
- Number of pole pairs of the motor (0x2006)
- Encoder resolution (0x608F)
- Gear factor (0x6091)

### Phase and quiescent current

The object 0x2004 Subindex 1 is used to set the phase current as a percentage of the maximum current. For the SMC147-S, a value of 7.5A (rms value) corresponds to 100%. The default setting is 20%.

The quiescent current as well as the phase current is set in percent of the maximum current with Subindex 2. This current is applied to the motor winding when the motor is at a standstill. The default setting is 20%.

### Number of pole pairs

The number of pole pairs of the stepper motor determines its step angle. The formula for the conversion is:

$$\text{Step angle} = 360^\circ / (4 * \text{number of pole pairs})$$

This means that a motor with a step angle of 1.8° has 50 pole pairs (default setting in the SMC147-S) and a 0.9° motor has 100 pole pairs.

The number of pole pairs must be correctly entered in the object 0x2006 Subindex 0.

### Encoder resolution

If an encoder is used, its resolution must be known to the controller. The resolution is specified in increments per rotation. Because of the quadrature principle, an encoder with 500 marks per rotation, for example, has four times the number of increments per rotation: 2000.

This value must be entered in the object 0x608F Subindex 1. The default value is 2000.

## Gear factor

When using gearing, its translation or reduction ratio can be specified in the “Gear Ratio” object (0x6091). This ratio is given as a fraction: The number of motor revolutions is in Subindex 1 and the number of revolutions of the gearing on the drive side is in Subindex 2. The default setting is 1/1.

## Conversion of position, speed and acceleration

If the motor parameters are correctly set, the user does not need to adjust the ramps, travel distances and speeds when changing the step mode or when changing from open loop to closed loop mode.

In PP, HM, and IP mode, the values for speed, acceleration and position refer to the “Feed Constant” object (feed constant, SDO 0x6092).

The numerator is saved in Subindex 1 of this object and the denominator of the feed constant is saved in Subindex 2. The default value is 2000/1. For the user this means that, independent of the step mode, a path of 2000 as target position (SDO 0x607A) always corresponds to one revolution of the driving shaft.

For speed, this applies accordingly to a second and for the acceleration, this applies correspondingly to a second squared.

In VL mode the “VL Dimension Factor” object (SDO 0x604C) is used accordingly. In contrast to the other supported modes, however, the speeds or acceleration refer here to a minute or a minute squared.

## Maximum speeds

Maximum speeds and accelerations (PP and HM mode) can be found in the following table.

Motor type	1.8°	0.9°		
Pole pairs (0x2006)	50	100		
Max. rpm	4800	2400		
Step mode	Feed (0x6092 sub1) (steps per revolution)	Feed (0x6092 sub1) (steps per revolution)	Max. profile velocity (0x6081)	Max. accel. (0x6083, 0x6084)
Full step	200	400	16000	1600000
Half step	400	800	32000	3200000
Quarter step	800	1600	64000	6400000
Fifth step	1000	2000	80000	8000000
Eighth step	1600	3200	128000	12800000
Tenth step	2000	4000	160000	16000000
1/16 step	3200	6400	256000	25600000
1/32 step	6400	12800	512000	51200000
1/64 step	12800	25600	1024000	102400000

## 2 Getting started in CANopen mode

### 2.1 Overview

After the supply voltage is applied, the controller is not automatically ready for operation immediately.

The following steps have to be carried out first:

- Start CANopen node
- Request CANopen node status
- Switch on the power drive
- Select operation mode
- Starting a travel
- Specify new end position

The individual steps are described in detail in the following sections.

### 2.2 Starting CANopen node

#### “Operational” status

In order to use the features and functions of the controller, each time the controller is switched on it must be put into operational status.

This occurs by sending a network management message with the COB ID 0x0 and the 2-byte long contents: <Command> and <node ID>.

#### Input in IXXAT MiniMon

The full input in IXXAT MiniMon is: “0 1 22”.

- 0: COB ID for NMT message
- 1: Start node
- 22: CANopen node ID (here 0x22 or 34)

#### Commands

The commands are:

- 0x01: Start node (switches to operational, status 0x05)
- 0x02: Stop node (switches to stopped, status 0x04)
- 0x80: Change to pre-operational (status 0x7F, status after application of operating voltage)
- 0x81: Restart of the firmware, reset of all CANopen settings to the last values stored in the EEPROM
- 0x82: Restart of the firmware, reset of all CANopen settings to the last values stored in the EEPROM

## 2.3 Requesting CANopen node status

### Status requests

The status can be requested with a Remote Transmission Request (RTR) of COB ID 0x700 + node ID.

A motor with the node ID 34(dec) sends its network status to COB ID 0x700 + 34 = 0x722.

In order to receive this message, a Remote Transmission Request (RTR) must be sent for this COB ID.

It is also possible to have the motor send this message cyclically (see SDO 0x1017: Dynamic Heartbeat Time).

### Possible status

The following states are available:

- Pre-operational status (status after applying the operating voltage, after restart and reset): **0x7F**  
In this state SDOs can be requested and written, but no PDOs can be read or written.
- Stopped status: **0x04**  
In this mode neither SDOs nor PDOs can be requested.
- Operational status: **0x05**  
In this mode both SDOs and PDOs can be read and written

## 2.4 Switching on the power drive

### Control word

The power drive is switched on via the control word. This can be reached under the service data object (SDO) 0x6040.

### Requesting the status word

After sending each command, it is recommended to check whether the intended status has been reached as status transitions are prevented (e.g. by a undervoltage error) or can be delayed (e.g. by the delay time of the mechanical brake or through the execution time of internal transitions) by requesting the status word.

### Input in IXXAT MiniMon

For activation, several transitions of the control words are necessary. The transitions are made as an input in IXXAT MiniMon for CANopen node ID 34:

A multiple transition during switch-on (e.g. from “Switch on Disabled” to “Operation Enabled”) is not possible. Only a jump from “Ready to Switch On” to “Operational” is possible.

COB ID	Data bytes	Description
622	2B 40 60 00 00 00	Switch On Disabled (basic state)
5A2	60 40 60 00 00 00 00 00	Response: OK
622	40 41 60 00	Request of the status word
5A2	4B 41 60 00 60 02 00 00	Response: Switch on Disabled

COB ID	Data bytes	Description
622	2B 40 60 00 06 00	Ready to Switch On
5A2	60 40 60 00 00 00 00 00	Response: OK
622	40 41 60 00	Request of the status word
5A2	4B 41 60 00 21 02 00 00	Response: Ready to Switch On

COB ID	Data bytes	Description
622	2B 40 60 00 07 00	Switch On (power drive on)
5A2	60 40 60 00 00 00 00 00	Response: OK
622	40 41 60 00	Request of the status word
5A2	4B 41 60 00 33 02 00 00	Response: Switch On, Voltage Enabled

COB ID	Data bytes	Description
622	2B 40 60 00 0F 00	Operation Enabled
5A2	60 40 60 00 00 00 00 00	Response: OK
622	40 41 60 00	Request of the status word
5A2	4B 41 60 00 37 02 00 00	Response: Operation Enabled, Voltage Enabled

### Explanation of the data bytes

Explanation of the data to be entered in hex:

- 622: The COB ID for sending SDOs to the CANopen node 34(dec)
- 2B: Command for writing a SDO with 2 data bytes
- 40: Low-order byte of the SDO ID 0x6040
- 60: High-order byte of the SDO ID 0x6040
- 00: Subindex 0x00
- 00, 06, 07, 0f: Least significant byte of the control word

It is important to note that the data in CANopen are always transmitted in Intel notation, i.e. the low-order bytes first.

### SDO addressing via CAN

Service data objects (SDO) are used to configure the respective CAN nodes. Because there are many different possible CANopen nodes (motors, sensors, etc.), a maximum of  $2^{16} = 65536$  SDOs are provided in the CANopen standard.

The number of SDOs exceeds the number of COB IDs in CAN  $2^{11} = 2048$ . Nevertheless, to be able to address the SDOs via CAN, the SDO ID is transferred as part of the data section of a COB.

In addition, 256 sub-indices are still possible for each SDO. If more than only Subindex 0 is occupied in a SDO, the number of available subindices of the respective SDOs is noted in Subindex 0.



## 2.5 Selecting the operation mode

### Provisions

Changes of the mode can take place in the “Operation Enabled” status.

It is important to ensure that the motor does not move when commanding a mode change.

### Example

The selection of a mode is shown using the example of PP mode (Profile Position or Positioning Mode):

COB ID	Data bytes	Description
622	2F 60 60 00 01	Mode: Profile Position (PP)
5A2	60 60 60 00 00 00 00 00	Response: OK

## 2.6 Starting a travel

### Step 1

Set the “new setpoint” bit in the control word to start a travel:

COB ID	Data bytes	Description
622	2b 40 60 00 1F 00	Start of a movement
5A2	60 40 60 00 00 00 00 00	Response: OK

### Step 2

Reset the bit to enable subsequent travel movements:

COB ID	Data bytes	Description
622	2b 40 60 00 0F 00	Reset of the start bit
5A2	60 40 60 00 00 00 00 00	Response: OK

The motor should now have moved a little. The motor no longer reacts to any further toggling of the bit as it has reached its end position.

## 2.7 Specifying a new end position (0x12345)

### Step 1

COB ID	Data bytes	Description
622	23 7A 60 00 45 23 01 00	End position to 0x12345
5A2	60 7A 60 00 00 00 00 00	Response: OK

### Step 2

COB ID	Data bytes	Description
622	2b 40 60 00 1F 00	Start of a movement
5A2	60 40 60 00 00 00 00 00	Response: OK

### Step 3

COB ID	Data bytes	Description
622	2b 40 60 00 0F 00	Reset of the start bit
5A2	60 40 60 00 00 00 00 00	Response: OK

## 3 Control and status word

### 3.1 Introduction

The control and status word are Service Data Objects (SDO) and are used to control and request the motor status. They are made up of individual bits.

The control and status word are mapped to Process Data Objects (PDO) as standard.

In the respective available modes (positioning, speed and reference mode) some bits always have the same meaning whereas others are specific to the respective mode.

Only those bits that have the same meaning in all modes are described.

### 3.2 Control word (SDO 0x6040)

#### Object description

<b>Designation</b>	Control word
<b>SDO ID</b>	0x6040
<b>Type</b>	u16
<b>Value range</b>	Bits

This object is available in the default setting as a PDO with the COB ID 0x200 + node ID. An RTR is not configured.

#### Bits 0, 1 and 3

Bit 0: Switch On

Bit 1: Enable Voltage

Bit 3: Enable Operation

The motor commands the state from Switch On Disabled to Operation Enabled with bits 0, 1 and 3 (see also section 2).

The states are:

- Switch On Disabled
- Ready to Switch On
- Switch On
- Operation Enabled

From the Switch On state, the holding brake is released and the electric field of the motor is active. A movement of the motor is only possible in the Operation Enabled state.

Additional states are:

- Quick Stop Active
- Fault Reaction Active
- Fault

The states from Switch On Disabled to Operation Enabled must be run through in the specified order. This occurs through the consecutive following setting of bits 0, 1 and 3. All three bits are set at the end of the switch-on process.

## Bit 2

Quick Stop (inverted: 0 means Quick Stop Active)

Bit 2 must always be set to “1” unless a Quick Stop is required. If this bit is set to “0”, the motor carries out a Quick Stop. During the Quick Stop, the motor is in the state “Quick Stop Active”. After the Quick Stop, the motor automatically goes into the state “Switch On Disabled”.

## Bit 4 to 15

Bits 4 to 6: Mode-specific.

Bit 7: Fault Reset.

If an error occurs, the firmware is in the Fault state after the error reaction. To set the firmware back to “Switch On Disabled”, this bit must carry out a transition from “0” to “1” (a duration of “1” is not sufficient here).

Bit 8: Stop (mode-specific).

Bit 9: Mode-specific.

Bit 10: Reserved.

Bits 11 to 15: Manufacturer-specific.

## 3.3 Status word (SDO 0x6041)

### Object description

<b>Designation</b>	Status word
<b>SDO ID</b>	0x6041
<b>Type</b>	u16
<b>Value range</b>	Bits

This object is available as a PDO with the COB ID 0x180 + node ID in the default setting. An RTR is not configured. Each time the status word changes, this object is automatically sent as a PDO.

### Bits

Bit 0: Ready to switch on

Bit 1: Switched on

Bit 2: Operation enabled: The selected operating mode is active and accepts commands (e.g. Profile Position Mode)

Bit 3: Fault: Set in case of a fault

Bit 4: Voltage enabled: Bit is set when the motor is provided with current

Bit 5: Quick stop

Bit 6: Switch on disabled

Bit 7: Warning

Bit 8: PLL sync complete: This bit is set as soon as the synchronisation with the SYNC object is complete.

Bit 9: Remote

Bit 10: Target reached: This bit is set when the motor has reached its target (Profile Position Mode)

Bit 11: Internal limit active: This bit is set when the desired values exceed the maximum limits.

Bits 12, 13: Mode-specific

Bits 14, 15: Manufacturer-specific (not used)

## 4 Profile Position Mode (PP)

### 4.1 Introduction

#### Function

The Profile Position Mode is used to reach positions relative to the last target position or absolute to the last reference position with a predefined speed and ramp.

#### Carrying out travel orders

This mode also supports the sending of new travel orders to the controller although the current travel order is not yet complete.

Depending on the setting in the control word, the travel order is only carried out after the current order (no option) or the new travel order interrupts the current travel order ("Change Set Immediately"). It is also possible that the current travel order is still carried out to the target position, but the transition to the new record is carried out as "flying" i.e. without reducing the speed (Change on Setpoint).

#### Position controller

If the position controller (closed loop) is active, the objects 0x6067 and 0x6068 are used in order to decide when the travel is regarded as complete. In addition, the SDO 0x6065 and 0x6066 are evaluated in order to display the following error in the status word.

If the position controller is not active (open loop), a position error correction after a travel can be carried out if an encoder is present. To do so, the settings in SDO 0x2004 Subindices 5 and 6 are used. If the error is greater than the tolerance set there, a correction travel is started.

### 4.2 Control word (SDO 0x6040)

#### Object description

<b>Designation</b>	Control word
<b>SDO ID</b>	0x6040
<b>Type</b>	u16
<b>Value range</b>	Bits

This object is available in the default setting as a PDO with the COB ID 0x200 + node ID. An RTR is not configured.

The motor is switched on and travel commands can be carried out with this object.

#### Bits

Bits 0, 1 and 3 are used for starting up the power drive. These are described in the General section.

Bit 2 is used to trigger an emergency stop. If it is set to "0", the motor carries out a quick stop with the ramp set in SDO 0x6085. The motor then goes into "Switch On Disabled" (see general description of the control word)

Bit 4 starts a travel order. This is carried out on a transition of "0" to "1".

Bit 5: If this bit is set to "1", a travel order triggered by bit 4 is immediately carried out. If it is "0", the travel order just being carried out is completed and only then is the next

travel order started. If no travel order is being carried out at the moment of starting, the next travel order is started immediately.

Bit 6: If “0”, the target position (SDO 0x607A) is absolute and if “1”, the target position is relative to the current position.

Bit 7: Error Reset (see general description of the control word)

Bit 8: Not used in this mode.

Bit 9: If this bit is set and the new travel command should only be carried out after the end of the current travel command, the speed is only changed when the first target position is reached. This means that braking is not performed before the first destination is reached as the motor should not stop at this position.

Bits 10 to 15: Reserved, to be set to 0.

## 4.3 Status word (SDO 0x6041)

### Object description

<b>Designation</b>	Status word
<b>SDO ID</b>	0x6041
<b>Type</b>	u16
<b>Value range</b>	Bits

This object is available as a PDO with the COB ID 0x180 + node ID in the default setting. An RTR is not configured. Each time the status word changes, this object is automatically sent as a PDO.

### Bits

Bits 0 to 9, 11, 14, 15: See section 3 “Control and status word”.

Bit 10: Target reached: This bit is set to 1 when the last target has been reached (motor at a standstill)

Bit 11: Internal limit active: This bit is set when one of the nominal values exceeds internal range limits. (e.g. default speed is greater than the maximum possible speed)

Bit 12: Set-point acknowledge: This bit confirms receipt of a new target point. It is set and reset synchronously to the “New set-point” bit in the control word.

An exception is if a new travel is started when another travel has not yet been completed and the next travel should only be carried out after the end of the first travel. In this case, the bit is only reset when the command has been accepted and the controller is ready to carry out new travel commands. If a new travel order is sent, the latest travel order is overwritten although this bit is still set.

Bit 13: Following error: This bit is set in Closed-Loop Mode if the following error is greater than the set limits (see chapter 8.32 and 8.33)

## 4.4 Quick Stop Option Code (SDO 0x605A)

### Object description

<b>Designation</b>	Quick Stop Option Code
<b>SDO ID</b>	0x605A
<b>Type</b>	s16, ro
<b>Value range</b>	3

Specifies the behavior upon a Quick Stop. Braking is currently only supported with maximum current and subsequent change to “Switch On Disabled”.

## 4.5 Position Demand Value (SDO 0x6062)

### Object description

<b>Designation</b>	Position Demand Value
<b>SDO ID</b>	0x6062
<b>Type</b>	s32 ro
<b>Value range</b>	s32

Specifies the current demanded position.

## 4.6 Target Position (SDO 0x607A)

### Object description

<b>Designation</b>	Target Position
<b>SDO ID</b>	0x607A
<b>Type</b>	s32 rw
<b>Value range</b>	-100000000 to 100000000

Specifies the target position.

Depending on the command of the control word, the end position is interpreted as relative to the current position or absolute to the reference position.

The direction can be reversed with the object 0x607E (polarity).

## 4.7 Home Offset (SDO 0x607C)

### Object description

<b>Designation</b>	Home Offset
<b>SDO ID</b>	0x607C
<b>Type</b>	s32 rw
<b>Value range</b>	s32

Specifies the difference between the zero position the application and the reference point the machine.



## 4.8 Profile Velocity (SDO 0x6081)

### Object description

<b>Designation</b>	Profile Velocity
<b>SDO ID</b>	0x6081
<b>Type</b>	u32 rw
<b>Value range</b>	Depending on the Feed Constant

Specifies the maximum traveling speed in steps per second.

## 4.9 End Velocity (SDO 0x6082)

### Object description

<b>Designation</b>	End Velocity
<b>SDO ID</b>	0x6082
<b>Type</b>	u32 rw
<b>Value range</b>	Depending on the Feed Constant

Specifies the minimum traveling speed for a trapezoidal ramp in steps per second.

## 4.10 Profile Acceleration (SDO 0x6083)

### Object description

<b>Designation</b>	Profile Acceleration
<b>SDO ID</b>	0x6083
<b>Type</b>	u32
<b>Value range</b>	1 to 100000

Specifies the acceleration ramp in steps/s<sup>2</sup>.

## 4.11 Profile Deceleration (SDO 0x6084)

### Object description

<b>Designation</b>	Profile Deceleration
<b>SDO ID</b>	0x6084
<b>Type</b>	u32
<b>Value range</b>	1 to 100000

Specifies the braking ramp in steps/s<sup>2</sup>.

## 4.12 Quick Stop Deceleration (SDO 0x6085)

### Object description

<b>Designation</b>	Quick Stop Deceleration
<b>SDO ID</b>	0x6085
<b>Type</b>	u32
<b>Value range</b>	1 to 100000

Specifies the emergency stop braking ramp in steps/s<sup>2</sup>.

## 4.13 Motion profile Type (0x6086)

### Object description

<b>Designation</b>	Motion profile type
<b>SDO ID</b>	0x6086
<b>Type</b>	s16
<b>Value range</b>	0-1

Specifies the ramp type.

Currently only a Sin2 (value=1) and a linear/trapezoidal ramp is supported (value = 0).

## 5 Homing Mode

### 5.1 Control word (0x6040)

#### Object description

<b>Designation</b>	Control word
<b>SDO ID</b>	0x6040
<b>Type</b>	u16
<b>Value range</b>	Bits

This object is available in the default setting as a PDO with the COB ID 0x200 + node ID. An RTR is not configured.

The motor is switched on and travel commands can be started with this object.

#### Bits

Bits 0, 1 and 3 are used for starting up the power drive. See section 3 “Control and status word”.

Bit 2 is used to trigger an emergency stop. If it is set to “0”, the motor carries out a quick stop with the ramp set in SDO 0x6085. The motor then goes into “Switch On Disabled” (see section 3 “Control and status word”).

Bit 4 starts the referencing. This is carried out until either the reference position has been reached or bit 4 is set to “0” again.

Bits 5 to 6: Reserved.

Bit 7: See section 3 “Control and status word”.

Bit 8: If this bit is set, the motor stops.

Bits 9 to 15: See section 3 “Control and status word”.

### 5.2 Status word (0x6041)

#### Object description

<b>Designation</b>	Status word
<b>SDO ID</b>	0x6041
<b>Type</b>	u16
<b>Value range</b>	Bits

This object is available as a PDO with the COB ID 0x180 + node ID in the default setting. An RTR is not configured. Each time the status word changes, this object is automatically sent as a PDO.

#### Bits

Bits 0 to 9, 11, 14, 15: See section 3 “Control and status word”.

Bit 10: Target reached: Set to “1” when the motor is at a standstill.

Bit 11: Unused.

Bit 12: Homing attained: Set to “1” when the reference position is reached.

Bit 13: Set to “1” when an error has occurred.

## 5.3 Homing Method (0x6098)

### Object description

<b>Designation</b>	Homing Method
<b>SDO ID</b>	0x6098
<b>Type</b>	u8
<b>Value range</b>	19, 20, 21, 22, 33, 34, 35, -2 to -7

This object selects the Homing Mode.

Modes 21 and 22 are only available from firmware 15-12-2008 (SMCI47-S).

#### Mode 19: External reference travel – switch as normally closed

- Search of the switch
- Motor rotates in a clockwise direction
- Speed from object 0x6099\_1 (Search for switch)
- As long as input 6 is high
- As soon as input 6 becomes low (switch reached) the direction is reversed
- Motor rotates in a clockwise direction
- Speed from object 0x6099\_2 (Search for zero)
- Until input 6 is high again (switch free again)
- Motor stops

#### Mode 20: External reference travel – switch as normally open

- Search of the switch
- Motor rotates in a clockwise direction
- Speed from object 0x6099\_1 (Search for switch)
- As long as input 6 is low
- When the switch is reached (input 6 high), the direction is reversed
- Motor rotates in a clockwise direction
- Speed from object 0x6099\_2 (Search for zero)
- Until input 6 becomes low again
- Motor stops

#### Mode 21: External reference travel – switch as normally closed

- Search of the switch
- Motor rotates in a clockwise direction
- Speed from object 0x6099\_1 (Search for switch)
- As long as input 6 is high
- When the switch is reached (input 6 low), the direction is reversed
- Motor rotates in a clockwise direction
- Speed from object 0x6099\_2 (Search for zero)
- Until input 6 becomes high again
- Motor stops

#### Mode 22: External reference travel – switch as normally open

- Search of the switch

- Motor rotates in a clockwise direction
- Speed from object 0x6099\_1 (Search for switch)
- As long as input 6 is low
- When the switch is reached (input 6 high), the direction is reversed
- Motor rotates in a clockwise direction
- Speed from object 0x6099\_2 (Search for zero)
- Until input 6 becomes low again
- Motor stops

**Mode 33: Internal reference run**

- Search for the index mark of the internal encoder
- Motor rotates in a clockwise direction
- Speed from object 0x6099\_2 (Search for zero)
- Until index mark is reached
- When the index mark is reached, the direction is reversed
- Motor rotates in a clockwise direction
- Motor shuts down as of the index mark
- Motor stops

**Mode 34: Internal reference run**

- Search for the index mark of the internal encoder
- Motor rotates in a clockwise direction
- Speed from object 0x6099\_2 (Search for zero)
- Until the index mark is reached
- When the index mark is reached, the direction is reversed
- Motor rotates in a clockwise direction
- Motor shuts down as of the index mark
- Motor stops

**Mode 35: Position reset**

- Sets the current position to Home Offset without the shaft moving

**Mode -2: Reference run set to blocking**

- Mode only functions with an encoder (OL and CL)
- First run:  
Motor rotates in a clockwise direction with speed from object 0x6099\_1 (Search for switch) until the shaft is blocked.  
The objects “Following Error Window” and “Following Error Timeout” are evaluated here.
- Motor moves backwards by one electrical revolution
- Second run:  
Motor rotates in a clockwise direction with speed from object 0x6099\_2 (Search for zero) until the shaft is blocked.  
The objects “Following Error Window” and “Following Error Timeout” are evaluated here.
- Motor moves backwards by one electrical revolution
- Motor moves to precisely the blocked position of the second run and sets the position to “Home Offset”

**Mode -3: Reference run to blocking**

- Like mode -2, only counterclockwise.

**Mode -4: Reference run to external IO node**

- Like Mode 19, only instead of input 6 an external IO node is used as the limit switch. (see also SDO 0x2010)

**Mode -5: Reference run to external IO node**

- Like Mode 20, only instead of input 6 an external IO node is used as the limit switch. (see also SDO 0x2010)

**Mode -6: Reference run to external IO node**

- Like Mode 21, only instead of input 6 an external IO node is used as the limit switch. (see also SDO 0x2010)

**Mode -7: Reference run to external IO node**

- Like Mode 22, only instead of input 6 an external IO node is used as the limit switch. (see also SDO 0x2010)

## 5.4 Homing Speeds (0x6099)

**Object description**

<b>Designation</b>	Homing Speeds
<b>SDO ID</b>	0x6099
<b>Type</b>	2x u32
<b>Value range</b>	1 to 25000

Specifies the speeds for the Homing mode in steps/s.

The speed for the search of the switch is specified in Subindex 1.

The (lower) speed for the search for the reference position is specified in Subindex 2.

**Notes**

The speed in Subindex 2 is also the starting speed for starting the acceleration ramp. If this is set too high, the motor loses steps or does not rotate at all. An excessive setting also leads to the index marking being overlooked. This speed should, therefore, be less than 1000 steps per second.

In addition, the speed in Subindex 1 must be greater than the speed in Subindex 2.

See also Homing Method (SDO 0x6098, section 5.3)

## 5.5 Homing Acceleration (0x609A)

### Object description

<b>Designation</b>	Homing Acceleration
<b>SDO ID</b>	0x609A
<b>Type</b>	u32
<b>Value range</b>	1 to 100000

Specifies the acceleration ramp for the Homing Mode in steps/s<sup>2</sup>. The ramp is only used when starting off. When the switch is reached, the unit is automatically switched to the lower speed and is stopped as soon as it reaches the limit position.

## 6 Velocity Mode

### 6.1 Control word (0x6040)

#### Object description

<b>Designation</b>	Control word
<b>SDO ID</b>	0x6040
<b>Type</b>	u16
<b>Value range</b>	Bits

This object is available as a static PDO with COB ID 0x200 + node ID. An RTR is not possible.

The motor is switched on and travel commands can be started with this object.

#### Bits

Bits 0, 1 and 3 are used for starting up the power drive. These are described in the General section.

Bit 2 is used to trigger an emergency stop. If it is set to "0", the motor carries out a quick stop with the ramp set in SDO 0x604A. The motor then goes into "Switch On Disabled" (see general description of the control word)

Bit 8: Halt: If this bit is set to "0", the motor accelerates to the target speed with the set ramp.

If it is set to "1", the motor brakes and comes to a stop.

### 6.2 Status word (0x6041)

#### Object description

<b>Designation</b>	Status word
<b>SDO ID</b>	0x6041
<b>Type</b>	u16
<b>Value range</b>	Bits

This object is available as a static PDO with COB ID 0x180 + node ID. An RTR is possible. Each time the status word changes, this object is automatically sent as a PDO.

#### Bit 11

Internal Limit Active: This bit is set when a nominal value exceeds or undercuts internal limits (e.g. target speed (SDO 0x6082) greater than 25000 steps/s).



## 6.3 VL target velocity (0x6042)

### Object description

<b>Designation</b>	VL target velocity
<b>SDO ID</b>	0x6042
<b>Type</b>	s16
<b>Value range</b>	-25000 to 25000

Specifies the target speed in steps/s.

Values greater than 0 stand for right-hand rotation, values less than 0 stand for left-hand rotation.

The direction can be reversed with the “Polarity” object.

## 6.4 VL velocity demand (0x6043)

### Object description

<b>Designation</b>	VL velocity demand
<b>SDO ID</b>	0x6043
<b>Type</b>	s16
<b>Value range</b>	-25000 to 25000

Specifies the current target speed in steps/s.

This object is read only.

## 6.5 VL velocity actual value (0x6044)

### Object description

<b>Designation</b>	VL velocity actual value
<b>SDO ID</b>	0x6044
<b>Type</b>	s16
<b>Value range</b>	-25000 to 25000

Specifies the current actual speed in steps/s.

This object is read only.

## 6.6 VL velocity min max amount (0x6046)

### Object description

<b>Designation</b>	VL velocity min max amount
<b>SDO ID</b>	0x6046
<b>Type</b>	2x u32
<b>Value range</b>	1 to 25000

The minimum speed and maximum speed in steps/s can be set with this object.

Subindex 1 contains the minimum speed.

Subindex 2 contains the maximum speed.

#### Notes

If the magnitude of a target speed (SDO 0x6042) is less than the minimum speed, the minimum speed applies. If the target speed is 0, the motor stops.

A target speed greater than the maximum speed sets the speed to the maximum speed and sets bit 11 (internal limit active) in the status word (SDO 0x6041).

## 6.7 VL velocity acceleration (0x6048)

#### Object description

<b>Designation</b>	VL velocity acceleration
<b>SDO ID</b>	0x6048
<b>Type</b>	u32, u16
<b>Value range</b>	u32, u16, both not equal 0

Sets the acceleration ramp in VL mode.

The acceleration is specified as a fraction:  
Speed change per time change.

Subindex 1 contains the speed change in steps/s (u32).

Subindex 2 contains the time change in s (u16).

#### Note

Neither the numerator nor the denominator must be set to 0.

## 6.8 VL velocity deceleration (0x6049)

#### Object description

<b>Designation</b>	VL velocity deceleration
<b>SDO ID</b>	0x6049
<b>Type</b>	u32, u16
<b>Value range</b>	u32, u16, both not equal 0

Sets the braking ramp in VL mode.

## 6.9 VL velocity quick stop (0x604A)

#### Object description

<b>Designation</b>	VL velocity quick stop
<b>SDO ID</b>	0x604A
<b>Type</b>	u32, u16
<b>Value range</b>	u32, u16, both not equal 0

Sets the braking ramp for the quick stop in VL mode.

The acceleration is specified as a fraction:  
Speed change per time change.

Subindex 1 contains the speed change in steps/s (u32).

Subindex 2 contains the time change in s (u16).

**Note**

Neither the numerator nor the denominator must be set to 0.

## 6.10 VL dimension factor (0x604C)

**Object description**

<b>Designation</b>	VL dimension factor
<b>SDO ID</b>	0x604C
<b>Type</b>	2x s32
<b>Value range</b>	s32, s32, both greater than 0

This object is a fraction which is used to convert the setpoint value presettings of the user in Velocity Mode into rpm.

(In VL mode the motor controller indicates the internal rotational speed in rpm.)

The numerator is in Subindex 1 here and the denominator of this fraction is in Subindex 2.

**Objects**

The conversion is applied to the following objects:

- 0x6042 (VL Target Velocity)
- 0x6043 (VL Velocity Demand)
- 0x6044 (VL Velocity Actual Value)
- 0x6046 (VL Velocity min/max Amount)
- 0x6048 (VL Velocity Acceleration)
- 0x6049 (VL Velocity Deceleration)
- 0x604A (VL Velocity Quick Stop)

**Conversion**

The conversion is carried out according to the following principle:

$$\text{User unit} * \text{conversion factor} = \text{unit in rpm}$$

**Example**

At a value of 2000, the user-specific unit should lead to one revolution per second (default in the firmware).

The conversion factor is then:  
60/2000 (SDO 0x606C Sub1/Sub2)

The firmware then carries out the conversion as follows:  
2000 \* 60/2000 = 60rpm (= one revolution per second)

For values that are updated by the firmware (SDO 0x6043 and 0x6044), the conversion runs accordingly so that the user can read the value from the SDO in the user-specific unit.

# 7 Interpolated Position Mode

## 7.1 Introduction

### Firmware

This mode is available as of firmware version 06-04-2009.

### Function

The Interpolated Position Mode is used to synchronise several axes. For this, a higher-order controller accepts the ramp or path calculation and transfers the respective demanded position at which the axis should find itself at a specific time to the SMC147-S. The SMC147-S interpolates between these position sampling points.

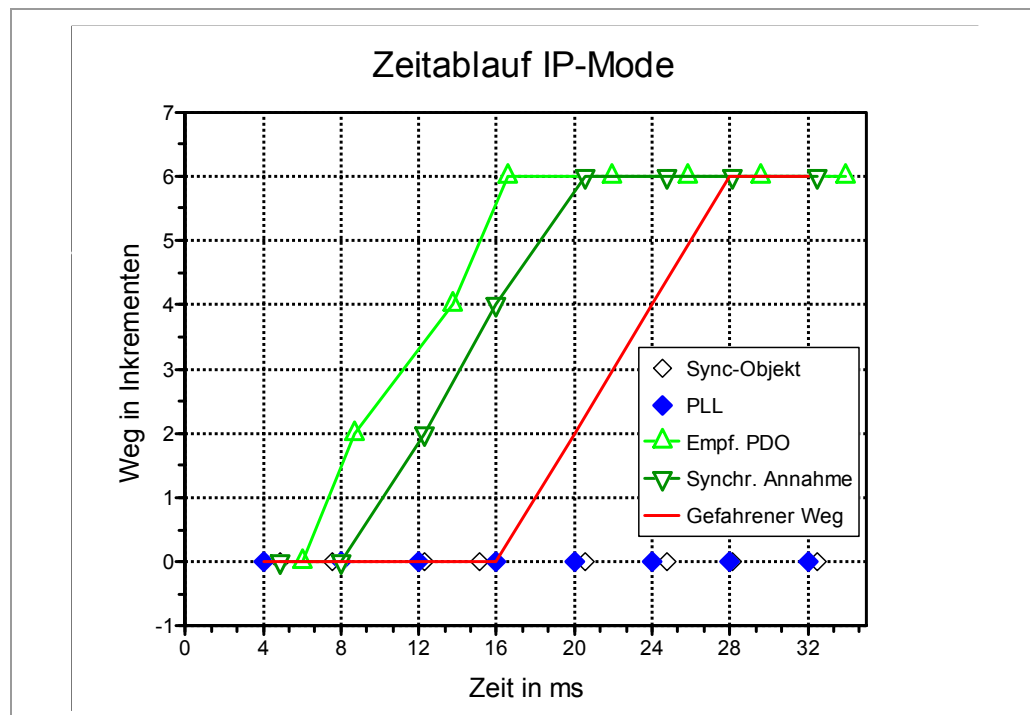
### Synchronisation to the SYNC object

For Interpolated Position Mode, it is necessary that the SMC147-S synchronised to the SYNC object. This SYNC object must be sent by the higher-order controller at regular time intervals. The synchronisation is carried out as soon as the SMC147-S is switched to NMT "Operational" mode (see section 2.2).

## 7.2 Chronological progress

### Display

The following figure shows the time progression in IP Mode. The time is plotted on the x-axis and the position on the y-axis.



## Explanations of the legend

### Sync object

The transparent diamond shapes, plotted along the x-axis, mark the points in time at which the SMC147-S receives a Sync object from the higher-order controller.

The time intervals of the sync objects are not constant in this case. The deviation of the points in time that results when the Sync object is received at equal time intervals is called jitter. Jitter is caused by a high CAN bus load, for instance.

### PLL

The SMC147-S compensates the jitter of the Sync object by synchronising it to the Sync objects with the aid of a so-called "Phase Locked Loop" (PLL). The resulting points in time (solid diamond shapes on the x-axis) are almost free of jitter.

### Recd. PDO

The triangles that point upwards indicate the points in time at which a new demanded position is received. The height of the triangles on the y-axis indicates the value of the demanded position. It is clear that the resulting curve between  $t=4\text{ms}$  and  $t=20\text{ms}$  is not a straight line although the position is always increased by two increments over the previous value. This occurs because of the time-related jitter that is also associated with the objects for the demanded position.

### Synchr. acceptance

At the points in time which are marked with the downward pointing triangles, the SMC147-S takes the PDO with the demanded position into its internal buffer. This point in time is precisely the time at which the sync object is received. From this point in time, it would also be possible to read this value via the SDO protocol. This curve also is not a straight line due to the jitter of the received Sync objects.

### Traveled path

The continuous line indicates the route traveled in Interpolated Position Mode. Because the interpolation time points are derived from the PLL, the jitter of the sync object has no influence on the shape of the curve.

The time delay between the demanded position and the traveled path is necessary because the starting and demanded position of the current interval must already be known at the beginning of the interpolation period (here  $2 \cdot 4\text{ms}$ ) of the SMC147-S in order to be able to calculate the speed this interval.

## 7.3 Control word (SDO 0x6040)

### Object description

<b>Designation</b>	Control word
<b>SDO ID</b>	0x6040
<b>Type</b>	u16
<b>Value range</b>	Bits

The power drive can be started up and the IP mode activated with this object.

### Bits

Bits 0, 1 and 3 are used for starting up the power drive. See section 3 "Control and status word".

Bit 2 is used to trigger an emergency stop. If it is set to "0", the motor stops abruptly. The motor then goes into "Switch On Disabled" (see section 3 "Control and status word").

Bit 4 activates the IP mode.

## 7.4 Status word (SDO 0x6041)

### Object description

<b>Designation</b>	Status word
<b>SDO ID</b>	0x6041
<b>Type</b>	u16
<b>Value range</b>	Bits

Bit12: IP mode active: This bit is set if the IP mode is active.

Remaining bits: See section 3 “Control and status word”.

## 7.5 Interpolation Sub-Mode Select (SDO 0x60C0)

### Object description

<b>Designation</b>	Interpolation Sub-Mode Select
<b>SDO ID</b>	0x60C0
<b>Type</b>	s16
<b>Value range</b>	0

The interpolation mode is selected with this object. Only a linear interpolation (value 0) is supported. Other values are not allowed and will be ignored.

## 7.6 Interpolation Data Record (SDO 0x60C1)

### Object description

<b>Designation</b>	Interpolation Data Record
<b>SDO ID</b>	0x60C1
<b>Type</b>	Array
<b>Number of entries</b>	1

This object contains the next target position to be moved to.

**Subindex 1**

<b>Designation</b>	1st Setpoint
<b>SDO ID</b>	0x60C1
<b>Subindex</b>	1
<b>Type</b>	s32
<b>Value range</b>	s32

The next target position is to be written in this subindex. When using the Interpolated Position Mode, it is recommended to map this object in a synchronous RPDO.

## 7.7 Interpolation Time Period (0x60C2)

**Object description**

In the SMCI47-S, these objects are present for reasons of compatibility. Set values are ignored. In the SMCI47-S the interpolation time is derived from the time intervals of the Synchronous object.

<b>Designation</b>	Interpolation Time Period
<b>SDO ID</b>	0x60C2
<b>Type</b>	Array
<b>Number of entries</b>	2

**Subindex 1**

<b>Designation</b>	Interpolation time period value
<b>SDO ID</b>	0x60C2
<b>Subindex</b>	1
<b>Type</b>	u8
<b>Value range</b>	u8

**Subindex 2**

<b>Designation</b>	Interpolation time index
<b>SDO ID</b>	0x60C2
<b>Subindex</b>	2
<b>Type</b>	s8
<b>Value range</b>	s8

## 7.8 Interpolation Data Configuration (0x60C4)

**Object description**

This object is used as a placeholder for future modes. Its values currently have no influence on the behavior of the Interpolated Position Mode.

<b>Designation</b>	Interpolation Data Configuration
<b>SDO ID</b>	0x60C4
<b>Type</b>	Array
<b>Number of entries</b>	6

#### Subindex 1

<b>Designation</b>	Maximum Buffer Size
<b>SDO ID</b>	0x60C4
<b>Subindex</b>	1
<b>Type</b>	u32
<b>Value range</b>	1

Maximum length of the buffer

#### Subindex 2

<b>Designation</b>	Actual Buffer Size
<b>SDO ID</b>	0x60C4
<b>Subindex</b>	2
<b>Type</b>	u32
<b>Value range</b>	1

Actual length of the buffer

#### Subindex 3

<b>Designation</b>	Buffer Organisation
<b>SDO ID</b>	0x60C4
<b>Subindex</b>	3
<b>Type</b>	u8
<b>Value range</b>	0

Selection between ring or FIFO buffer.

#### Subindex 4

<b>Designation</b>	Buffer Position
<b>SDO ID</b>	0x60C4
<b>Subindex</b>	4
<b>Type</b>	u16
<b>Value range</b>	1

Current buffer position. (always 1 in the SMC147-S; not requested or updated)



**Subindex 5**

<b>Designation</b>	Size of data record
<b>SDO ID</b>	0x60C4
<b>Subindex</b>	5
<b>Type</b>	u8
<b>Value range</b>	4

Has a meaning if, for example, a spline interpolation is used. Several values are necessary per interpolation section for this case. This object specifies the size of a record.

**Subindex 6**

<b>Designation</b>	Buffer Clear
<b>SDO ID</b>	0x60C4
<b>Subindex</b>	6
<b>Type</b>	u8
<b>Value range</b>	0

The buffer could be deleted with this object. Since the buffer of the SMCI47-S has a length of 1 only, deleting does not make any sense.

## 7.9 Recommendations for PDO Mapping

**Control word (RPDO1)**

The PDO mapping of the control word (RPDO1) could be changed in such a way that, in addition, the “Modes of Operation” object (SDO 0x6060) is also received with this PDO of the SMCI47-S. The Transmission Type can be left as 255 (asynchronous). However, it is also possible to readily configure this object as a synchronous PDO (Transmission Type 0 to 240). Once the IP mode is in operation, it is not necessary to continually update this object as that would cause unnecessary CAN bus load.

**Status words (TPDO1)**

The PDO mapping of the status word (TPDO1) could be changed in such a way that, in addition, the “Modes of Operation Display” object (SDO 0x6061) is also sent with this PDO of the SMCI47-S. The Transmission Type can be left as 255 (asynchronous). However, it is also possible to readily configure this object as a synchronous PDO (Transmission Type 0). It is then sent synchronously to the Sync object if the status word or “Modes of Operation Display” changes which does not occur during normal operation of the IP Mode. If logging of the position is desired, the “Position actual Value” object (SDO 0x6064) can also be attached to this TPDO as well. Then it is imperative, however, for this PDO to be configured as synchronous and cyclical (Transmission Type 1 to 240). It would then also be conceivable to set the Transmission Type to values greater than 1 so the current position is not sent in every Sync object. The user must decide what is sensible for him here.

**Interpolation Data Record (SDO 0x60C1)**

The main object for the IP mode is the “Interpolation Data Record” (SDO 0x60C1). This object has no standard mapping. It would be sensible to use mapping to RPDO2, Transmission Type 0 (cyclically synchronous). The higher-order controller must send the next demanded position in this PDO after each Sync object. The demanded position is then adopted in the next Sync object (see Figure 3). When using several controllers, it makes sense to pack the demanded position for two controllers in one PDO and to mask the target position of the other controller by an s32 dummy object

(SDO 0x0004) in the respective controller. Thus the necessary CAN bandwidth can be reduced and the saved bandwidth can be used for a higher interpolation frequency.

## 7.10 Information about commissioning

If other modes have been used before the IP mode is used, it is very probable that the current position of the rotor is not zero. If the IP mode is then selected, the user must ensure that the demanded position in the “Interpolation Data Record” object (SDO 0x60C1) corresponds to the actual position before interpolation is started with bit 4 of the control word.

If this is not done, a movement to the zero position can occur within an interpolation period (the time interval between two Sync objects) when switching on the interpolation. In Open Loop Mode this is expressed by jerking of the shaft or short whistling (steps are lost here). In Closed Loop Mode the controller uses maximum current until the target position is reached.

To prevent this, the “Position Demand value” object (SDO 0x6062) must be adopted for Open Loop Mode and the “Position actual value” object (SDO 0x6064) must be adopted for Closed Loop Mode as the starting position for the Interpolation Data Record (SDO 0x60C1).

## 8 SDOs that are not mode-specific

### 8.1 Digital inputs (SDO 0x60FD)

#### Object description

<b>Designation</b>	Digital Inputs
<b>SDO ID</b>	0x60FD
<b>Type</b>	u32
<b>Value range</b>	32-bit bitmask

The digital inputs of the motor can be read with this SDO.

This SDO is read only.

#### Bits

- Bits 0 to 1: Unassigned ("0")
- Bit 2: Input 6 (reference switch)
- Bits 3 to 15: Unassigned ("0")
- Bits 16 to 21: Input 1 to Input 6
- Bits 22 to 31: Unassigned ("0").

### 8.2 Digital outputs (SDO 0x60FE Subindex 1)

#### Object description

<b>Designation</b>	Digital Outputs
<b>SDO ID</b>	0x60FE
<b>Subindex</b>	1
<b>Type</b>	u32
<b>Value range</b>	32-bit bitmask

The digital outputs of the motor can be written with this SDO.

The writing of the outputs via this SDO is only possible if they are also released with the output mask (SDO 0x60FE Subindex 2) for the user. Otherwise, the outputs specify the motor status (see also manual).

#### Bits

- Bits 0 to 15 Unassigned ("0")
- Bits 16 to 17: Output 1 to Output 2
- Bits 18 to 31: Unassigned ("0").

## 8.3 Digital outputs bitmask (SDO 0x60FE Subindex 2)

### Object description

<b>Designation</b>	Bitmask
<b>SDO ID</b>	0x60FE
<b>Subindex</b>	2
<b>Type</b>	u32
<b>Value range</b>	32-bit bitmask

The digital outputs of the motor can be reserved for the user with this SDO.

A “0” means that the respective output is reserved for the user and can thus be changed with SDO 0x60FE Subindex 1. A “1” in the mask means that the status of the firmware has an effect on the outputs. These cannot then be affected by SDO 0x60FE Subindex 1.

### Bits

Bits 0 to 15 Unassigned (“0”)

Bits 16 to 17: Mask for Output 1 to Output 2

Bits 18 to 31: Unassigned (“0”)

If unassigned bits have a “1” written to them, the entire mask is discarded.

## 8.4 Polarity (0x607E)

### Object description

<b>Designation</b>	Polarity
<b>SDO ID</b>	0x607E
<b>Type</b>	u8
<b>Value range</b>	8-bit bitmask, bits 6 and 7 can be used, bits 0 to 5 are reserved (= 0)

### Bits

Bit 6: Reverses the direction of rotation in Velocity Mode.

Bit 7: Reverses the direction of rotation in Profile Position Mode.

(“1” means reversal is activated, “0” means direction of rotation as described in the respective mode).

## 8.5 Producer Heartbeat Time (0x1017)

### Object description

<b>Designation</b>	Producer Heartbeat Time
<b>SDO ID</b>	0x1017
<b>Type</b>	u16
<b>Value range</b>	u16

Sets the Heartbeat Time in ms (milliseconds).

Default setting is 0 (deactivated).

If the value is set to 0, the Heartbeat is switched off.

With values that are not equal to 0, the CANopen network status is sent to the COB ID 0x700 + node ID in the set interval.

**Attention:**

By setting shorter intervals, the CAN bus can be so heavily loaded that other communication is hardly still possible.

### Requesting the network status

If the Heartbeat is switched off, the network status can be requested by sending a Remote Transmission Request to the COB ID 0x700 + node ID.

The network status is a u8 and can have the following values:

- 0x7F: Pre-Operational (status after application of the operating voltage)
- 0x04: Node Disabled
- 0x05: Node Operational.

## 8.6 DEVICE TYPE (0x1000)

### Object description

<b>Designation</b>	Device Type
<b>SDO ID</b>	0x1000
<b>Type</b>	u16, ro
<b>Value range</b>	u16

Describes the control type. Here: 0x00008 (stepper motor)

## 8.7 ERROR REGISTER (0x1001)

### Object description

<b>Designation</b>	Error Register
<b>SDO ID</b>	0x1001
<b>Type</b>	u8
<b>Value range</b>	u8, ro

Error register: In the event of error the corresponding error bit is set. Currently only bit 0 (generic error) is supported. All other bits are always 0.

Bit 0: generic error

Bit 1: current (0)

Bit 2: voltage (0)

Bit 3: temperature (0)

Bit 4: communication error (overrun, error state)

Bit 5: device profile specific (0)

Bit 6: reserved (0)

Bit 7: manufacturer specific (0)

## 8.8 COB\_ID\_EMERGENCY\_MESSAGE (0x1014)

### Object description

<b>Designation</b>	Emergency Message
<b>SDO ID</b>	0x1014
<b>Type</b>	u32, ro
<b>Value range</b>	u32

Default Value 80h + node ID

### Bits

Bit 31 (MSB): 0 = EMCY exists, 1 = EMCY does not exist

Bit 30: 0 = reserved (always 0)

Bit 29: 0 = 11-bit ID (CAN 2.0A), 1 = 29-bit ID (CAN 2.0B)

Bits 28-11: 0 = if bit 29=0; Bits 28-11 of the 29-bit COB ID if bit 29=1:

Bit 10-0 (LSB): Bits 10-0 of the COB ID

## 8.9 IDENTITY\_OBJECT\_VENDOR\_ID (0x1018)

### Object description

<b>Designation</b>	Identity Object
<b>SDO ID</b>	0x1018
<b>Type</b>	record

### Subindex 0h: Number of entries

<b>Type</b>	u8
<b>Value range</b>	1 ... 4, ro

### Subindex 1h: Vendor ID

<b>Type</b>	u32, ro
-------------	---------

### Subindex 2h: Product code

<b>Type</b>	u32, ro
-------------	---------

### Subindex 3h: Revision number

<b>Type</b>	u32, ro
-------------	---------

### Subindex 4h: Serial number

<b>Type</b>	u32, ro
-------------	---------

## 8.10 RPDO1\_COMMUNICATION\_PARAMETER (0x1400)

### Object description

<b>Designation</b>	Receive PDO1 Communication Parameter
<b>SDO ID</b>	0x1400
<b>Type</b>	record
<b>Number of entries</b>	2

Contains the communication parameters for the RPDO 1.

### Subindex 01h: COB ID

<b>Type</b>	u32, rw
<b>Default value</b>	0x200+Node ID The default value does not take effect by changing the node ID, but only by resetting the EEPROM (see section 1.1.3).

#### Bits

Bit 31: 1 = PDO deactivated

Bit 30: 1 = RTR not allowed

Bit 29: 1 = 29-bit ID, 0 = 11-bit ID

Bits 28-11: Bits 28-11 of the 29-bit ID, otherwise 0

Bits 10-0: Bits 10-0 of the COB ID

### Subindex 02h: Transmission Type

<b>Type</b>	u8, rw
<b>Default value</b>	255

#### Values

0 to 240: Update of the object after receiving the next Sync object.

255: Immediate update of the object.

## 8.11 RPDO2\_COMMUNICATION\_PARAMETER (0x1401)

### Object description

<b>Designation</b>	Receive PDO2 Communication Parameter
<b>SDO ID</b>	0x1401
<b>Type</b>	record
<b>Number of entries</b>	2

Contains the communication parameters for RPDO 2.

### Subindex 01h: COB ID

<b>Type</b>	u32, rw
<b>Default value</b>	0x300 + Node ID The default value does not take effect by changing the node ID, but only by resetting the EEPROM (see section 1.1.3).

#### Bits

Bit 31: 1 = PDO deactivated

Bit 30: 1 = RTR not allowed

Bit 29: 1 = 29-bit ID, 0 = 11-bit ID

Bits 28-11: Bits 28-11 of the 29-bit ID, otherwise 0

Bits 10-0: Bits 10-0 of the COB ID

#### Subindex 02h: Transmission Type

<b>Type</b>	u8, rw
<b>Default value</b>	255

#### Values

0 to 240: Update of the object after receiving the next Sync object.

255: Immediate update of the object.

## 8.12 RPDO3\_COMMUNICATION\_PARAMETER (0x1402)

#### Object description

<b>Designation</b>	Receive PDO3 Communication Parameter
<b>SDO ID</b>	0x1403
<b>Type</b>	record
<b>Number of entries</b>	2

Contains the communication parameters for the RPDO 3.

#### Subindex 01h: COB ID

<b>Type</b>	u32, rw
<b>Default value</b>	0x400 + Node ID The default value does not take effect by changing the node ID, but only by resetting the EEPROM (see section 1.1.3).

#### Bits

Bit 31: 1 = PDO deactivated

Bit 30: 1 = RTR not allowed

Bit 29: 1 = 29-bit ID, 0 = 11-bit ID

Bits 28-11: Bits 28-11 of the 29-bit ID, otherwise 0

Bits 10-0: Bits 10-0 of the COB ID

#### Subindex 02h: Transmission Type

<b>Type</b>	u8, rw
<b>Default value</b>	255

#### Values

0 to 240: Update of the object after receiving the next Sync object.

255: Immediate update of the object.



## 8.13 RPDO4\_COMMUNICATION\_PARAMETER (0x1403) #

### Object description

<b>Designation</b>	Receive PDO4 Communication Parameter
<b>SDO ID</b>	0x1403
<b>Type</b>	record
<b>Number of entries</b>	2

Contains the communication parameters for the RPDO 4.

### Subindex 01h: COB ID

<b>Type</b>	u32, rw
<b>Default value</b>	0x500 + Node ID The default value does not take effect by changing the node ID, but only by resetting the EEPROM (see section 1.1.3).

#### Bits

Bit 31: 1 = PDO deactivated

Bit 30: 1 = RTR not allowed

Bit 29: 1 = 29-bit ID, 0 = 11-bit ID

Bits 28-11: Bits 28-11 of the 29-bit ID, otherwise 0

Bits 10-0: Bits 10-0 of the COB ID

### Subindex 02h: Transmission Type

<b>Type</b>	u8, rw
<b>Default value</b>	255

#### Values

0 to 240: Update of the object after receiving the next Sync object.

255: Immediate update of the object.

## 8.14 RPDO1\_MAPPING (0x1600)

### Object description

<b>Designation</b>	Receive PDO Mapping Parameter
<b>SDO ID</b>	0x1600
<b>Type</b>	record
<b>Number of mapped objects</b>	0-8

Describes which object is mapped to the COB ID defined in the object 0x1400.

### Subindex 1-8: Mapped objects

<b>Designation</b>	Receive PDO1 Mapping Parameter
<b>Type</b>	u32, rw

Describes one mapped object at a time.

#### Bits

Bits 31-16: index  
Bits 15-8: sub-index  
Bits 7-0: object length in bits

## 8.15 RPDO2\_MAPPING (0x1601)

### Object description

<b>Designation</b>	Receive PDO2 Mapping Parameter
<b>SDO ID</b>	0x1601
<b>Type</b>	record
<b>Number of mapped objects</b>	0-8

Describes which object is mapped to the COB ID defined in the object 0x1401.

### Subindex 1-8: Mapped objects

<b>Designation</b>	Receive PDO Mapping Parameter
<b>Type</b>	u32, rw

Describes one mapped object at a time.

#### Bits

Bits 31-16: index  
Bits 15-8: sub-index  
Bits 7-0: object length in bits

## 8.16 RPDO3\_MAPPING (0x1602)

### Object description

<b>Designation</b>	Receive PDO3 Mapping Parameter
<b>SDO ID</b>	0x1602
<b>Type</b>	record
<b>Number of mapped objects</b>	0-8

Describes which object is mapped to the COB ID defined in the object 0x1402.

### Subindex 1-8: Mapped objects

<b>Designation</b>	Receive PDO Mapping Parameter
<b>Type</b>	u32, rw

Describes one mapped object at a time.

#### Bits

Bits 31-16: index  
Bits 15-8: sub-index  
Bits 7-0: object length in bits

## 8.17 RPDO4\_MAPPING (0x1603)

### Object description

<b>Designation</b>	Receive PDO4 Mapping Parameter
<b>SDO ID</b>	0x1603
<b>Type</b>	record
<b>Number of mapped objects</b>	0-8

Describes which object is mapped to the COB ID defined in the object 0x1403.

### Subindex 1-8: Mapped objects

<b>Designation</b>	Receive PDO Mapping Parameter
<b>Type</b>	u32, rw

Describes one mapped object at a time.

#### Bits

Bits 31-16: index

Bits 15-8: sub-index

Bits 7-0: object length in bits

## 8.18 TPDO1\_COMMUNICATION\_PARAMETER (0x1800)

### Object description

<b>Designation</b>	Transmit PDO1 Communication Parameter
<b>SDO ID</b>	0x1800
<b>Type</b>	record
<b>Number of entries</b>	5

Contains the communication parameters for the TPDO 1.

### Subindex 01h: COB ID

<b>Type</b>	u32, rw
<b>Default value</b>	0x180 + Node ID The default value does not take effect by changing the node ID, but only by resetting the EEPROM (see section 1.1.3).

#### Bits

Bit 31: 1 = PDO deactivated

Bit 30: 1 = RTR not allowed

Bit 29: 1 = 29-bit ID, 0 = 11-bit ID

Bits 28-11: Bits 28-11 of the 29-bit ID, otherwise 0

Bits 10-0: Bits 10-0 of the COB ID

### Subindex 02h: Transmission Type

<b>Type</b>	u8, rw
<b>Default value</b>	255

#### Values

0: Sending of the object after the next Sync object if it has changed its value.

1 to 240: Sending of the object after the set number of Sync objects (1 to 240), independently of whether the object has changed.

255: Sending the object after a change, independent of the Sync object.

#### Subindex 03h: inhibit time

<b>Type</b>	u16, rw
<b>Default value</b>	1000

When Transmission Type 255 (asynchronous transmission on a change) is used, this value indicates the minimum time between the transmission of two consecutive objects in 100µs steps. For example, this can prevent the current position which changes continuously during travel from blocking the CAN bus.

#### Subindex 04h: reserved

<b>Type</b>	u8, rw
<b>Default value</b>	0

No meaning

#### Subindex 05h: event timer

<b>Type</b>	u16, rw
<b>Default value</b>	0

For Transmission Type 255 (asynchronous transmission upon a change), this value indicates the maximum time between two transmitted objects of the same type. This setting can be used to cyclically send objects that rarely change. A value of "0" in this setting deactivates this behavior (default).

## 8.19 TPDO2\_COMMUNICATION\_PARAMETER (0x1801)

### Object description

<b>Designation</b>	Transmit PDO2 Communication Parameter
<b>SDO ID</b>	0x1801
<b>Type</b>	record
<b>Number of entries</b>	5

Contains the communication parameters for the TPDO 2.

#### Subindex 01h: COB ID

<b>Type</b>	u32, rw
<b>Default value</b>	0x280 + Node ID The default value does not take effect by changing the node ID, but only by resetting the EEPROM (see section 1.1.3).

#### Bits

Bit 31: 1 = PDO deactivated

Bit 30: 1 = RTR not allowed

Bit 29: 1 = 29-bit ID, 0 = 11-bit ID

Bits 28-11: Bits 28-11 of the 29-bit ID, otherwise 0

Bits 10-0: Bits 10-0 of the COB ID

#### Subindex 02h: Transmission Type

<b>Type</b>	u8, rw
<b>Default value</b>	255

##### Values

0: Sending of the object after the next Sync object if it has changed its value.

1 to 240: Sending of the object after the set number of Sync objects (1 to 240), independently of whether the object has changed.

255: Sending the object after a change, independent of the Sync object.

#### Subindex 03h: inhibit time

<b>Type</b>	u16, rw
<b>Default value</b>	1000

When Transmission Type 255 (asynchronous transmission on a change) is used, this value indicates the minimum time between the transmission of two consecutive objects in 100µs steps. For example, this can prevent the current position which changes continuously during travel from blocking the CAN bus.

#### Subindex 04h: reserved

<b>Type</b>	u8, rw
<b>Default value</b>	0

No meaning

#### Subindex 05h: event timer

<b>Type</b>	u16, rw
<b>Default value</b>	0

For Transmission Type 255 (asynchronous transmission upon a change), this value indicates the maximum time between two transmitted objects of the same type. This setting can be used to cyclically send objects that rarely change. A value of "0" in this setting deactivates this behavior (default).

## 8.20 TPDO3\_COMMUNICATION\_PARAMETER (0x1802)

### Object description

<b>Designation</b>	Transmit PDO3 Communication Parameter
<b>SDO ID</b>	0x1802
<b>Type</b>	record
<b>Number of entries</b>	5

Contains the communication parameters for the TPDO 3.

**Subindex 01h: COB ID**

<b>Type</b>	u32, rw
<b>Default value</b>	0x380 + Node ID The default value does not take effect by changing the node ID, but only by resetting the EEPROM (see section 1.1.3).

**Bits**

Bit 31: 1 = PDO deactivated

Bit 30: 1 = RTR not allowed

Bit 29: 1 = 29-bit ID, 0 = 11-bit ID

Bits 28-11: Bits 28-11 of the 29-bit ID, otherwise 0

Bits 10-0: Bits 10-0 of the COB ID

**Subindex 02h: Transmission Type**

<b>Type</b>	u8, rw
<b>Default value</b>	255

**Values**

0: Sending of the object after the next Sync object if it has changed its value.

1 to 240: Sending of the object after the set number of Sync objects (1 to 240), independently of whether the object has changed.

255: Sending the object after a change, independent of the Sync object.

**Subindex 03h: inhibit time**

<b>Type</b>	u16, rw
<b>Default value</b>	1000

When Transmission Type 255 (asynchronous transmission on a change) is used, this value indicates the minimum time between the transmission of two consecutive objects in 100µs steps. For example, this can prevent the current position which changes continuously during travel from blocking the CAN bus.

**Subindex 04h: reserved**

<b>Type</b>	u8, rw
<b>Default value</b>	0

No meaning.

**Subindex 05h: event timer**

<b>Type</b>	u16, rw
<b>Default value</b>	0

For Transmission Type 255 (asynchronous transmission upon a change), this value indicates the maximum time between two transmitted objects of the same type. This setting can be used to cyclically send objects that rarely change. A value of "0" in this setting deactivates this behavior (default).

## 8.21 TPDO4\_COMMUNICATION\_PARAMETER (0x1803)

### Object description

<b>Designation</b>	Transmit PDO4 Communication Parameter
<b>SDO ID</b>	0x1803
<b>Type</b>	record
<b>Number of entries</b>	5

Contains the communication parameters for the TPDO 4.

### Subindex 01h: COB ID

<b>Type</b>	u32, rw
<b>Default value</b>	0x480 + Node ID The default value does not take effect on changing the node ID, but only by resetting the EEPROM (see section 1.1.3).

#### Bits

Bit 31: 1 = PDO deactivated

Bit 30: 1 = RTR not allowed

Bit 29: 1 = 29-bit ID, 0 = 11-bit ID

Bits 28-11: Bits 28-11 of the 29-bit ID, otherwise 0

Bits 10-0: Bits 10-0 of the COB ID

### Subindex 02h: Transmission Type

<b>Type</b>	u8, rw
<b>Default value</b>	255

#### Values

0: Sending of the object after the next Sync object if it has changed its value.

1 to 240: Sending of the object after the set number of Sync objects (1 to 240), independently of whether the object has changed.

255: Sending the object after a change, independent of the Sync object.

### Subindex 03h: inhibit time

<b>Type</b>	u16, rw
<b>Default value</b>	1000

When Transmission Type 255 (asynchronous transmission on a change) is used, this value indicates the minimum time between the transmission of two consecutive objects in 100µs steps. For example, this can prevent the current position which changes continuously during travel from blocking the CAN bus.

### Subindex 04h: reserved

<b>Type</b>	u8, rw
<b>Default value</b>	0

No meaning

### Subindex 05h: event timer

<b>Type</b>	u16, rw
<b>Default value</b>	0

For Transmission Type 255 (asynchronous transmission upon a change), this value indicates the maximum time between two transmitted objects of the same type. This setting can be used to cyclically send objects that rarely change. A value of “0” in this setting deactivates this behavior (default).

## 8.22 TPDO1\_MAPPING (0x1A00)

### Object description

<b>Designation</b>	Transmit PDO1 Mapping Parameter
<b>SDO ID</b>	0x1A00
<b>Type</b>	record
<b>Number of mapped objects</b>	0-8

Describes which object is mapped to the COB ID defined in the object 0x1800.

### Subindex 1-8: Mapped objects

<b>Designation</b>	Transmit PDO Mapping Parameter
<b>Type</b>	u32, rw

Describes one mapped object at a time.

#### Bits

31-16: index

15-8: sub-index

7-0: object length in bits

## 8.23 TPDO2\_MAPPING (0x1A01)

### Object description

<b>Designation</b>	Transmit PDO2 Mapping Parameter
<b>SDO ID</b>	0x1A01
<b>Type</b>	record
<b>Number of mapped objects</b>	0-8

Describes which object is mapped to the COB ID defined in the object 0x1801.

### Subindex 1-8: Mapped objects

<b>Designation</b>	Transmit PDO Mapping Parameter
<b>Type</b>	u32, rw

Describes one mapped object at a time.

#### Bits

31-16: index

15-8: sub-index



7-0: object length in bits

## 8.24 TPDO3\_MAPPING (0x1A02)

### Object description

<b>Designation</b>	Transmit PDO2 Mapping Parameter
<b>SDO ID</b>	0x1A02
<b>Type</b>	record
<b>Number of mapped objects</b>	0-8

Describes which object is mapped to the COB ID defined in the object 0x1802.

### Subindex 1-8: Mapped objects

<b>Designation</b>	Transmit PDO Mapping Parameter
<b>Type</b>	u32, rw

Describes one mapped object at a time.

#### Bits

31-16: index

15-8: sub-index

7-0: object length in bits

## 8.25 TPDO4\_MAPPING (0x1A03)

### Object description

<b>Designation</b>	Transmit PDO4 Mapping Parameter
<b>SDO ID</b>	0x1A03
<b>Type</b>	record
<b>Number of mapped objects</b>	0-8

Describes which object is mapped to the COB ID defined in the object 0x1803.

### Subindex 1-8: Mapped objects

<b>Designation</b>	Transmit PDO Mapping Parameter
<b>Type</b>	u32, rw

Describes one mapped object at a time.

#### Bits

31-16: index

15-8: sub-index

7-0: object length in bits

## 8.26 ERROR\_CODE (0x603F)

### Object description

<b>Designation</b>	Error code
<b>SDO ID</b>	0x603F
<b>Type</b>	u16, ro
<b>Value range</b>	u16

Contains the last error.

## 8.27 VL\_DIMENSION\_FACTOR (0x604C)

### Object description

<b>Designation</b>	vl dimension factor
<b>SDO ID</b>	0x604C
<b>Type</b>	record
<b>Number of entries</b>	2

Conversion factor for the user-specific speed (e.g. 1/s, increments/s, m/s,...) in revolutions per minute.

Speed [rpm] = speed [user-specific unit]/conversion factor [rpm/user-specific unit].

### Subindex 01h: vl dimension factor numerator

<b>Type</b>	u32, rw
<b>Default value</b>	1

### Subindex 02h: vl dimension factor denominator

<b>Type</b>	u32, rw
<b>Default value</b>	1

## 8.28 MODES\_OF\_OPERATION (0x6060)

### Object description

<b>Designation</b>	Modes of operation
<b>SDO ID</b>	0x6060
<b>Type</b>	s8, rw
<b>Value range / Default</b>	-128 - 10 / 0

Contains the current operating mode.

**Values**

Mode	Description
-128 to -1	Manufacturer-specific operation modes
-1	Closed-loop test run
0	No mode change/no mode assigned
+1	Profile Position Mode
+2	Velocity Mode
+3	Profile Velocity Mode
+4	Torque Profile Mode
+5	Reserved
+6	Homing Mode
+7	Interpolated Position Mode
+8	Cyclic Sync Position Mode
+9	Cyclic Sync Velocity Mode
+10	Cyclic Sync Torque Mode
+11 to +127	Reserved

## 8.29 MODES\_OF\_OPERATION\_DISPLAY (0x6061)

**Object description**

<b>Designation</b>	Modes of operation display
<b>SDO ID</b>	6061
<b>Type</b>	s8, ro
<b>Value range</b>	s8

Contains the current operating mode. -> see MODES\_OF\_OPERATION (SDO 0x6060)

## 8.30 POSITION\_ACTUAL\_INTERNAL\_VALUE (0x6063)

<b>Designation</b>	Position actual internal value
<b>SDO ID</b>	0x6063
<b>Type</b>	s32, ro
<b>Value range</b>	s32

Contains the current encoder position.

## 8.31 POSITION\_ACTUAL\_VALUE (0x6064)

### Object description

<b>Designation</b>	Position actual value
<b>SDO ID</b>	0x6064
<b>Type</b>	s32, ro
<b>Value range</b>	s32

Contains the current actual position (encoder position converted acc. to Feed Constant and Gear Ratio)

## 8.32 FOLLOWING\_ERROR\_WINDOW (0x6065)

### Object description

<b>Designation</b>	Following error window
<b>SDO ID</b>	0x6065
<b>Type</b>	U32, rw
<b>Value range</b>	u32

Specifies the maximum following error symmetrically to the demanded position. If the actual position deviates too greatly from the demanded position, a following error is issued.

## 8.33 FOLLOWING\_ERROR\_TIME\_OUT (0x6066)

### Object description

<b>Designation</b>	Following error timeout
<b>SDO ID</b>	0x6066
<b>Type</b>	u16, rw
<b>Value range</b>	u16

Time in milliseconds until too large a following error leads to an error message.

## 8.34 POSITION\_WINDOW (0x6067)

### Object description

<b>Designation</b>	Position window
<b>SDO ID</b>	0x6067
<b>Type</b>	u32, rw
<b>Value range</b>	u32

Specifies a symmetrical range relative to the target position within which the target is considered to be reached.

## 8.35 POSITION\_WINDOW\_TIME (0x6068)

### Object description

<b>Designation</b>	Position window time
<b>SDO ID</b>	0x6068
<b>Type</b>	u16, rw
<b>Value range</b>	u16

For this time period, the actual position must be within the position window so that the target position is considered to be reached.

## 8.36 POS\_RANGE\_LIMIT (0x607B)

### Object description

<b>Designation</b>	Position range limit
<b>SDO ID</b>	0x607B
<b>Type</b>	record
<b>Number of entries</b>	2

Contains the minimum and maximum position. If this range is exceeded or undercut, an overflow occurs. To prevent this overflow, see also SW\_POS\_LIMIT (SDO 0x607D).

### Subindex 01h: Min position range limit

<b>Type</b>	s32, rw
<b>Default value</b>	-100000000

### Subindex 02h: Max position range limit

<b>Type</b>	s32, rw
<b>Default value</b>	100000000

## 8.37 SW\_POS\_LIMIT (0x607D)

<b>Designation</b>	Software position limit
<b>SDO ID</b>	0x607D
<b>Type</b>	record
<b>Number of entries</b>	2

The target position must lie within the limits set here. Before the check, the home offset (SDO 0x607C) is deducted each time:

corrected min position limit = min position limit - home offset

corrected max position limit = max position limit - home offset.

**Subindex 01h: Min position limit**

<b>Type</b>	s32, rw
<b>Default value</b>	-100000000

**Subindex 02h: Max position limit**

<b>Type</b>	s32, rw
<b>Default value</b>	100000000

## 8.38 POSITION\_ENCODER\_RESOLUTION (0x608F)

**Object description**

<b>Designation</b>	Position encoder resolution
<b>SDO ID</b>	0x608F
<b>Type</b>	record
<b>Number of entries</b>	2

Encoder increments per revolution:

position encoder resolution = encoder increments / motor revolutions.

Note: Encoders with 768, 800, 1600, 2000, 2048, 4000, 4096, 8000 and 8192 increments per revolution are currently supported.

**Subindex 01h: encoder increments**

<b>Type</b>	u32, rw
<b>Default value</b>	2000

**Subindex 02h: motor revolutions**

<b>Type</b>	u32, rw
<b>Default value</b>	1

## 8.39 GEAR\_RATIO (0x6091)

**Object description**

<b>Designation</b>	Gear ratio
<b>SDO ID</b>	0x6091
<b>Type</b>	record
<b>Number of entries</b>	2

Number of motor revolutions per revolution of the driving axis:

gear ratio = motor shaft revolutions / driving shaft revolutions.

**Subindex 01h: Motor revolutions**

<b>Type</b>	u32, rw
<b>Default value</b>	1

**Subindex 02h: Shaft revolutions**

<b>Type</b>	u32, rw
<b>Default value</b>	1

## 8.40 FEED\_CONSTANT (0x6092)

**Object description**

<b>Designation</b>	Feed constant
<b>SDO ID</b>	0x6092
<b>Type</b>	record
<b>Number of entries</b>	2

Specifies the feed per revolution for a linear drive:

Feed constant = feed / revolutions.

**Subindex 01h: Feed**

<b>Type</b>	u32, rw
<b>Default value</b>	2000

**Subindex 02h: Shaft revolutions**

<b>Type</b>	u32, rw
<b>Default value</b>	1

## 8.41 MAX\_ACCEL (0x60C5)

**Object description**

<b>Designation</b>	Max acceleration
<b>SDO ID</b>	0x60C5
<b>Type</b>	u32, rw
<b>Value range</b>	u32

Contains the maximum admissible acceleration ramp. → see also MAX\_DECEL(SDO 0x60C6).

## 8.42 MAX\_DECEL (0x60C6)

### Object description

<b>Designation</b>	Max deceleration
<b>SDO ID</b>	0x60C6
<b>Type</b>	u32, rw
<b>Value range</b>	u32

Contains the maximum braking ramp. → see also MAX\_ACCEL (0x60C5).

## 8.43 SUPPORTED\_DRIVE\_MODES (0x6502)

### Object description

<b>Designation</b>	Supported drive modes
<b>SDO ID</b>	0x6502
<b>Type</b>	record, ro
<b>Value range</b>	u32

The object describes the supported drive modes. Only PP, HM, VL and IP are currently supported in CANopen operation.

### Bits

Bit	Meaning	supports
31-16	not used (manufacturer-specific)	0
15-20	reserved	0
9	CST	0
8	CSV	0
7	CSP	0
6	IP (Interpolated Position Mode)	1 (as of firmware version 06042009)
5	HM (Homing Mode)	1
4	R	0
3	TQ	0
2	PV	0
1	VL (Velocity Mode)	1
0	PP (Profile Position Mode)	1

## 8.44 Boolean Dummy Object (0x0001)

### Object description

<b>Designation</b>	Boolean
<b>SDO ID</b>	0x0001
<b>Type</b>	Bit, rw
<b>Value range</b>	Bit

Intended as a dummy object for PDO mapping.



Cannot be mapped as PDO.

## 8.45 Integer8 Dummy Object (0x0002)

### Object description

<b>Designation</b>	Signed Integer 8
<b>SDO ID</b>	0x0002
<b>Type</b>	Integer 8, rw
<b>Value range</b>	-128 to +128

Can be used as a dummy object for RxPDO.

## 8.46 Integer16 Dummy Object (0x0003)

### Object description

<b>Designation</b>	Signed Integer 16
<b>SDO ID</b>	0x0003
<b>Type</b>	Integer 16, rw
<b>Value range</b>	-32768 to +32767

Can be used as a dummy object for RxPDO.

## 8.47 Integer32 Dummy Object (0x0004)

### Object description

<b>Designation</b>	Signed Integer 32
<b>SDO ID</b>	0x0004
<b>Type</b>	Integer 32, rw
<b>Value range</b>	-2147483648 to 2147483647

Can be used as a dummy object for RxPDO.

## 8.48 Unsigned8 Dummy Object (0x0005)

### Object description

<b>Designation</b>	Signed Integer 8
<b>SDO ID</b>	0x0005
<b>Type</b>	Integer 8, rw
<b>Value range</b>	0 to 255

Can be used as a dummy object for RxPDO.

## 8.49 Unsigned16 Dummy Object (0x0006)

### Object description

<b>Designation</b>	Signed Integer 16
<b>SDO ID</b>	0x0006
<b>Type</b>	Integer 16, rw
<b>Value range</b>	0 to 65535

Can be used as a dummy object for RxPDO.

## 8.50 Unsigned32 Dummy Object (0x0007)

### Object description

<b>Designation</b>	Signed Integer 32
<b>SDO ID</b>	0x0007
<b>Type</b>	Integer 32, rw
<b>Value range</b>	0 to 4294967295

Can be used as a dummy object for RxPDO.

## 9 Analog Input

### 9.1 Overview

#### Generic

The SMCI47-S controller has an analog input that supports the trigger conditions as they are described in CiA CANopen 401.

#### Resolution

The input has a resolution of 10 bits at a conversion rate of 1kHz. The full resolution refers to an input voltage range of -10V to +10V.

#### Values

The output values are of the s16 type where only positive values are used. The corresponding SDO (0x6401,1) is continuously updated. The trigger conditions for correspondingly mapped TPDOs can be set in the objects 0x6423 to 0x6428. Scaling or offset calculation is not supported.

<b>Designation</b>	Read analogue input 16-bit
<b>SDO ID</b>	0x6401 Subindex 1
<b>Type</b>	s16, ro
<b>Value range</b>	0x0000 to 0x03FF
Contains the last converted value of the analog input. This object can be mapped as a PDO. Special trigger conditions are supported for the PDO	

### 9.2 Analog Input (0x6401)

#### Object description

This object contains the converted value of the analog input and can be mapped as TxPDO.

<b>Designation</b>	Analog input
<b>SDO ID</b>	0x6401
<b>Type</b>	Array
<b>Number of entries</b>	1

<b>Designation</b>	Number of analog inputs
<b>Subindex</b>	0
<b>Type</b>	u8, ro
<b>Value range</b>	1

<b>Designation</b>	Analog Input 1 conversion value
<b>Subindex</b>	1
<b>Type</b>	u16, ro
<b>Value range</b>	0x0 to 0x3FF (10-bit)

## 9.3 Analog Global Interrupt Enable (0x6423)

### Object description

Only when this object has been set to 1 can a change of the analog value trigger the sending of the corresponding PDO.

<b>Designation</b>	Analog Global Interrupt Enable
<b>SDO ID</b>	0x6423
<b>Type</b>	U8, rw
<b>Value range</b>	0 or 1
<b>Default</b>	0

## 9.4 Analog Input Interrupt Upper Limit (0x6424)

### Object description

If the analog value changes and is greater than the limit set in this object, a PDO is sent.

If the upper limit is less than the lower limit, the range between the upper and lower limit is active.

If the upper limit is greater than the lower limit, the ranges between 0 and lower limit and between upper limit and infinity are active.

<b>Designation</b>	Analog Input Interupt Upper Limit
<b>SDO ID</b>	0x6424
<b>Type</b>	Array
<b>Number of entries</b>	1

<b>Designation</b>	Number of analog inputs
<b>Subindex</b>	0
<b>Type</b>	u8, ro
<b>Value range</b>	1

<b>Designation</b>	Analog 1 upper limit
<b>Subindex</b>	1
<b>Type</b>	s32, rw
<b>Value range</b>	0x0 to 0x3FF (10-bit)

## 9.5 Analog Input Interrupt Lower Limit (0x6425)

### Object description

If the analog value changes and is less than the limit set in this object, a PDO is sent.

If the upper limit is less than the lower limit, the range between the upper and lower limit is active.

If the upper limit is greater than the lower limit, the ranges between 0 and lower limit and between upper limit and infinity are active.

<b>Designation</b>	Analog Input Interrupt Lower Limit
<b>SDO ID</b>	0x6425
<b>Type</b>	Array
<b>Number of entries</b>	1

<b>Designation</b>	Number of analog inputs
<b>Subindex</b>	0
<b>Type</b>	u8, ro
<b>Value range</b>	1

<b>Designation</b>	Analog 1 lower limit
<b>Subindex</b>	1
<b>Type</b>	s32, rw
<b>Value range</b>	0x0 to 0x3FF (10-bit)

## 9.6 Analog Input Interrupt Delta (0x6426)

### Object description

If the analog value changes and the difference to the last sent value is greater than the value set in this object, a PDO is sent. In addition, the limits from the upper and lower limits are evaluated.

<b>Designation</b>	Analog Input Delta
<b>SDO ID</b>	0x6426
<b>Type</b>	Array
<b>Number of entries</b>	1

<b>Designation</b>	Number of analog inputs
<b>Subindex</b>	0
<b>Type</b>	u8, ro
<b>Value range</b>	1

<b>Designation</b>	Change limit Analog 1
<b>Subindex</b>	1
<b>Type</b>	u32, rw
<b>Value range</b>	0x0 to 0x3FF (10-bit)

## 9.7 Analog Input Interrupt negative Delta (0x6427)

### Object description

If the analog value changes to a negative value and the difference to the last sent value is greater than the value set in this object, a PDO is sent. In addition, the limits from the upper and lower limits are evaluated. If this value is less than the analog input delta, it is not evaluated.

<b>Designation</b>	Analog Input negative Delta
<b>SDO ID</b>	0x6427
<b>Type</b>	Array
<b>Number of entries</b>	1

<b>Designation</b>	Number of analog inputs
<b>Subindex</b>	0
<b>Type</b>	u8, ro
<b>Value range</b>	1

<b>Designation</b>	Negative change limit Analog 1
<b>Subindex</b>	1
<b>Type</b>	u32, rw
<b>Value range</b>	0x0 to 0x3FF (10-bit)

## 9.8 Analog Input Interrupt positive Delta (0x6427)

### Object description

If the analog value changes to a positive value and the difference to the last sent value is greater than the value set in this object, a PDO is sent. In addition, the limits from the upper and lower limits are evaluated. If this value is less than the analog input delta, it is not evaluated.

<b>Designation</b>	Analog Input negative Delta
<b>SDO ID</b>	0x6427
<b>Type</b>	Array
<b>Number of entries</b>	1

<b>Designation</b>	Number of analog inputs
<b>Subindex</b>	0
<b>Type</b>	u8, ro
<b>Value range</b>	1

<b>Designation</b>	Positive change limit Analog 1
<b>Subindex</b>	1
<b>Type</b>	u32, rw
<b>Value range</b>	0x0 to 0x3FF (10-bit)

## 10 Manufacturer-specific CAN Objects

### 10.1 STEP\_MODE (0x2000)

#### Object description

<b>Designation</b>	Step mode
<b>SDO ID</b>	0x2000
<b>Type</b>	u8, rw
<b>Value range/Default value</b>	1,2,4,5,8,10,16,32,64,255

This object is obsolete. The step mode that can be set here results from the objects "Feed Constant" and "Gear Ratio". A smooth movement is given through microstep interpolation which is always active.

### 10.2 ENABLE\_CL (0x2001)

#### Object description

<b>Designation</b>	Enable closed loop
<b>SDO ID</b>	0x2001
<b>Type</b>	u8, rw
<b>Value range</b>	0.1

Activates/deactivates the closed loop control.

#### Prerequisites

Writing to this object is only possible if:

- CL referencing has been carried out (Homing Method -1 in the Homing Mode)
- The motor is in the status "Switch On Disabled"
- Step mode 10 is selected (object 0x2000)

If these conditions are not fulfilled, the firmware responds with the error code "Not in present device state" (error code 0x08000022).

Before using Closed Loop mode for the first time, the controller must be adapted to the motor/encoder combination. This is carried out using the rotor position test run (Modes of Operation (SDO 0x6060) = -1). Otherwise, there is a high probability that the controller will rotate with maximum current in the wrong direction.



## 10.3 CL\_CONFIGURATION (0x2002)

### Object description

<b>Designation</b>	Closed loop Configuration
<b>SDO ID</b>	0x2002
<b>Type</b>	record
<b>Number of entries</b>	24

Contains the control parameters for the Closed Loop Mode.

### Subindex 01h KP\_V\_Z

<b>Type</b>	u16, rw
<b>Default value</b>	2

Numerator of the proportional component of the speed controller.

### Subindex 02h: KP\_V\_N

<b>Type</b>	u16, rw
<b>Default value</b>	0

Denominator of the proportional component of the speed controller as a power of 2.

### Subindex 03h KI\_V\_Z

<b>Type</b>	u16, rw
<b>Default value</b>	1

Numerator of the integral component of the speed controller.

### Subindex 04h: KI\_V\_N

<b>Type</b>	u16, rw
<b>Default value</b>	6

Denominator of the integral component of the speed controller as a power of 2.

### Subindex 05h KD\_V\_Z

<b>Type</b>	u16, rw
<b>Default value</b>	0

Numerator of the differential component of the speed controller.

### Subindex 06h: KD\_V\_N

<b>Type</b>	u16, rw
<b>Default value</b>	0

Denominator of the differential component of the speed controller as a power of 2.

### Subindex 07h KP\_S\_Z

<b>Type</b>	u16, rw
<b>Default value</b>	100

Numerator of the proportional component of the position controller.

**Subindex 08h: KP\_S\_N**

<b>Type</b>	u16, rw
<b>Default value</b>	0

Denominator of the proportional component of the position controller as a power of 2.

**Subindex 09h KI\_S\_Z**

<b>Type</b>	u16, rw
<b>Default value</b>	2

Numerator of the integral component of the position controller.

**Subindex 0Ah: KI\_S\_N**

<b>Type</b>	u16, rw
<b>Default value</b>	0

Denominator of the integral component of the position controller as a power of 2.

**Subindex 0Bh KD\_S\_Z**

<b>Type</b>	u16, rw
<b>Default value</b>	300

Numerator of the differential component of the position controller.

**Subindex 0Ch: KD\_S\_N**

<b>Type</b>	u16, rw
<b>Default value</b>	0

Denominator of the differential component of the position controller as a power of 2.

**Subindex 0Dh KP\_CSV\_Z**

<b>Type</b>	u16, rw
<b>Default value</b>	50

Numerator of the proportional component of the speed cascade controller

**Subindex 0Eh: KP\_CSV\_N**

<b>Type</b>	u16, rw
<b>Default value</b>	0

Denominator of the proportional component of the speed cascade controller

**Subindex 0Fh KI\_CSV\_Z**

<b>Type</b>	u16, rw
<b>Default value</b>	2

Numerator of the integral component of the speed cascade controller

**Subindex 10h: KI\_CSV\_N**

<b>Type</b>	u16, rw
<b>Default value</b>	0

Denominator of the integral component of the speed cascade controller

**Subindex 11h KD\_CSV\_Z**

<b>Type</b>	u16, rw
<b>Default value</b>	0

Numerator of the differential component of the speed cascade controller

**Subindex 12h: KD\_CSV\_N**

<b>Type</b>	u16, rw
<b>Default value</b>	0

Denominator of the differential component of the speed cascade controller

**Subindex 13h KP\_CSS\_Z**

<b>Type</b>	u16, rw
<b>Default value</b>	2

Numerator of the proportional component of the position cascade controller

**Subindex 14h: KP\_CSS\_N**

<b>Type</b>	u16, rw
<b>Default value</b>	0

Denominator of the proportional component of the position cascade controller

**Subindex 15h KI\_CSS\_Z**

<b>Type</b>	u16, rw
<b>Default value</b>	1

Numerator of the integral component of the position cascade controller

**Subindex 16h: KI\_CSS\_N**

<b>Type</b>	u16, rw
<b>Default value</b>	6

Denominator of the integral component of the position cascade controller

**Subindex 17h KD\_CSS\_Z**

<b>Type</b>	u16, rw
<b>Default value</b>	0

Numerator of the differential component of the position cascade controller

**Subindex 18h: KD\_CSS\_N**

<b>Type</b>	u16, rw
<b>Default value</b>	0

Numerator of the differential component of the position cascade controller

## 10.4 A/D Converter (0x2003)

### Object description

<b>Designation</b>	Actual value of the A/D converter
<b>SDO ID</b>	0x2003
<b>Type</b>	record
<b>Number of entries</b>	4

Contains the actual values of the A/D converter (Raw Values)

### Subindex 01h A/D 1

<b>Type</b>	u16, ro
<b>Default value</b>	-

### Subindex 02h: A/D 2

<b>Type</b>	u16, ro
<b>Default value</b>	-

### Subindex 03h A/D 3

<b>Type</b>	u16, ro
<b>Default value</b>	-

### Subindex 04h: A/D 4

<b>Type</b>	u16, ro
<b>Default value</b>	-

## 10.5 Current control (0x2004)

### Object description

<b>Designation</b>	Current limiter
<b>SDO ID</b>	0x2004
<b>Type</b>	record
<b>Number of entries</b>	2

Contains the set current values.

### Subindex 01h: drive current

<b>Type</b>	u8, rw
<b>Default value</b>	20

Current that is used for normal travel. Data in %.

### Subindex 02h: current reduction

<b>Type</b>	u8, rw
<b>Default value</b>	20

Current that is used for the reduction. Data in %.

**Subindex 03h: current reduction time**

<b>Type</b>	u16, rw
<b>Default value</b>	80

Time in milliseconds from a standstill of the motor in Open Loop mode until the current is reduced.

**Subindex 04h: current for block reference run**

<b>Type</b>	u8, rw
<b>Default value</b>	0

Current for the block reference run in percent.  
 If the value is 0, the phase current is used.

**Subindex 05h: Swing out time**

<b>Type</b>	u16, rw
<b>Default value</b>	80

Time in milliseconds between a standstill of the axis and position reset after a block reference run and between standstill of the axis and a ready message in PP mode

**Subindex 06h: Position tolerance**

<b>Type</b>	u32, rw
<b>Default value</b>	0xFFFFFFFF

Tolerance after a travel in PP mode (Open Loop).

**Subindex 07h: Motor Type**

<b>Type</b>	u8, rw
<b>Default value</b>	0

Motor type:

- 0: Stepper motor with and without quadrature encoder
- 1: BLDC with Hall sensor and without quadrature encoder
- 2: BLDC with Hall sensor and with quadrature encoder

**Subindex 08h: Load angle sampling spacing**

<b>Type</b>	u16, rw
<b>Default value</b>	4096

Sampling point spacing of the load angle curve in SDO ox200C.  
 A value of 8192 corresponds to 1000 rpm.

**Subindex 09h: Cascade controller switch-on speed**

<b>Type</b>	u32, rw
<b>Default value</b>	327680

The cascade controller is activated as of this speed.  
 A value of 8192 corresponds to 1000 rpm.

#### Subindex 0Bh: Cascade controller switch-off speed

Type	u32, rw
Default value	512

The cascade controller is deactivated as of this speed.  
A value of 8192 corresponds to 1000 rpm.

#### Subindex 0Ch: Cascade controller switch-on speed

Type	u8, ro
Default value	0

Specifies whether the cascade controller is active.

#### Subindex 0Dh: Hall sensor mode

Type	u32, rw
Default value	2371605 (0x243015)

Used with BLDC motors to adjust the Hall sensor sections to the respective motor.  
Here the least significant 6 digits in hexadecimal notation represent one quadrant of the Hall sensors.

## 10.6 CAN enable and baud rate (0x2005)

### Object description

The CAN baud rate is adjusted with this object. For a more detailed description, see section 1.1.3.

Designation	CAN enable and baud rate
SDO ID	0x2005
Type	u8
Value range	130,131,132,133,134,135
Default value	135 (is not reset by an EEPROM Reset)

## 10.7 Motor Pole Pairs (0x2006)

### Object description

Designation	Number of pole pairs of the motor
SDO ID	0x2006
Type	u16
Value range	50 and 100
Default value	50

Specifies the number of pole pairs of the motor. After changing this parameter, the controller must be provided with power again. For use in Closed Loop, the motor must be readjusted with the aid of the rotor position test run.

Pool pair count	Step angle
50	1.8°
100	0.9°

## 10.8 Brake Wait Time (0x2007)

### Object description

<b>Designation</b>	Wait time for mech. brake
<b>SDO ID</b>	0x2007
<b>Type</b>	record
<b>Number of entries</b>	3

The SMCI47-S has an output for connection of a spring-loaded, electromagnetically releasable brake. Delay times for this brake can be set with this SDO.

While a braking wait time is carried out, no commands for changing the mode are executed.

### Subindex 01h: Milliseconds Power on to Brake off

<b>Type</b>	u16, rw
<b>Default value</b>	0
Specifies the wait time between switch-on of the power drive and release of the brake in milliseconds.	

### Subindex 02h: Milliseconds Brake off to Operational

<b>Type</b>	u16, rw
<b>Default value</b>	0
Specifies the wait time between releasing the brake and reaching the "Operational" state in milliseconds. This setting prevents the motor from rotating due to a command while the brake has not yet fully released.	

### Subindex 03h: Milliseconds Brake on to Power off

<b>Type</b>	u16, rw
<b>Default value</b>	0
Specifies the wait time between arresting the brake and switching off the power drive in milliseconds. This setting prevents the motor from moving due to a mechanical torque when the brake has not yet fully arrested.	

## 10.9 Milliseconds Input Debounce Time (0x2008)

### Object description

Used to debounce the digital inputs of the controller. After a flank on an input, no further flanks are processed within the time set in this object in milliseconds. A flank is only detected again after expiry of the debounce time. A debounce time of an input that is running has no effect on the detection of flanks on the other inputs.

<b>Designation</b>	Debounce time
<b>SDO ID</b>	0x2008
<b>Type</b>	u8, rw
<b>Value range</b>	0 to 255
<b>Default value</b>	20

## 10.10 Node ID (0x2009)

### Object description

This object is used to adjust the node ID if the node ID is set to 0 with the hex switches. If a node ID is set with the hex switches, this object can be described and saved in the EEPROM, but contains the value of the hex switches again after a restart.

<b>Designation</b>	Node ID
<b>SDO ID</b>	0x2009
<b>Type</b>	u8, rw
<b>Value range</b>	1 to 127
<b>Default value</b>	1 (is not reset by an EEPROM Reset)

## 10.11 CL is enabled (0x200A)

### Object description

It is possible to determine whether the Closed Loop Mode is active with the aid of this object. If the object returns the value 1, Closed Loop Mode is active.

<b>Designation</b>	CL is enabled
<b>SDO ID</b>	0x200A
<b>Type</b>	u8, ro
<b>Value range</b>	0 and 1
<b>Default value</b>	0

## 10.12 CL POSCNT Offset (0x200B)

### Object description

Specifies the distance between encoder index and rotor alignment. A value of 65536 corresponds to an electrical angle of 360° or 0°. This value is set by the CL test run and is different for each individual motor.



<b>Designation</b>	CL POSCNT Offset
<b>SDO ID</b>	0x200B
<b>Type</b>	u16, rw
<b>Value range</b>	0 to 65535
<b>Default value</b>	0

## 10.13 CL load angle curve (0x200C)

### Object description

Specifies the speed-dependent load angle. A value of 65536 corresponds to an electrical angle of 360° or 0°. This value is set by the CL test run and is identical for motors of the same series. Each of the subindices of 1 to 7 contains a load angle for a speed range. The speed interval between the individual values can be set in SDO 2004 Subindex 8.

<b>Designation</b>	CL load angle curve
<b>SDO ID</b>	0x200C
<b>Type</b>	Array
<b>Number of entries</b>	10

### Subindex 00h Number of entries

<b>Type</b>	u8, ro
<b>Default value</b>	10

### Subindex 01h: Load angle at a standstill

<b>Type</b>	u16, rw
<b>Default value</b>	16384

### Subindex 02h Load angle at speed = load angle sampling spacing \* 1

<b>Type</b>	u16, rw
<b>Default value</b>	17000

### Subindex 03h Load angle at speed = load angle sampling spacing \* 2

<b>Type</b>	u16, rw
<b>Default value</b>	17500

### Subindex 04h Load angle at speed = load angle sampling spacing \* 3

<b>Type</b>	u16, rw
<b>Default value</b>	17750

### Subindex 05h Load angle at speed = load angle sampling spacing \* 4

<b>Type</b>	u16, rw
<b>Default value</b>	18000

**Subindex 06h Load angle at speed = load angle sampling spacing \* 5**

<b>Type</b>	u16, rw
<b>Default value</b>	18000

**Subindex 07h Load angle at speed = load angle sampling spacing \* 6**

<b>Type</b>	u16, rw
<b>Default value</b>	18000

**Subindex 08h is not used**

<b>Type</b>	u16, rw
<b>Default value</b>	18000

**Subindex 09h is not used**

<b>Type</b>	u16, rw
<b>Default value</b>	18000

**Subindex 0Ah is not used**

<b>Type</b>	u16, rw
<b>Default value</b>	18000

## 10.14 Encoder rotation direction change (0x200D)

**Object description**

If this object is set 1, the direction of the quadrature encoder is reversed.

<b>Designation</b>	Reverse Encoder Direction
<b>SDO ID</b>	0x200D
<b>Type</b>	u8, rw
<b>Value range</b>	0 or 1
<b>Default value</b>	0

## 10.15 DSPdrive current controller parameter (0x200E)

**Object description**

For controllers with DSPdrive (PD4-N, SMC112) the current controller parameters can be changed.

<b>Designation</b>	DSPdrive current controller parameters
<b>SDO ID</b>	0x200E
<b>Type</b>	Array
<b>Number of entries</b>	6

**Subindex 00h Number of entries**

<b>Type</b>	u8, ro
<b>Default value</b>	6

**Subindex 01h: KP low (not used)**

<b>Type</b>	u16, rw
<b>Default value</b>	1

**Subindex 02h KP high**

<b>Type</b>	u16, rw
<b>Default value</b>	10

KP value at a standstill.

**Subindex 03h KP scale**

<b>Type</b>	u16, rw
<b>Default value</b>	58

KP value is increased in proportion to the speed.

**Subindex 04h KI low (not used)**

<b>Type</b>	u16, rw
<b>Default value</b>	1

**Subindex 05h KI high**

<b>Type</b>	u16, rw
<b>Default value</b>	10

KI value at a standstill

**Subindex 06h KI scale**

<b>Type</b>	u16, rw
<b>Default value</b>	200

KI value is increased in proportion to the speed.

## 10.16 Speed mode controller type (0x200F)

**Object description**

If this object is set to 1, the position controller is used in VL mode instead of the speed controller. This can be an advantage at very low speeds.

<b>Designation</b>	Controller type speed mode
<b>SDO ID</b>	0x200F
<b>Type</b>	u8, rw
<b>Value range</b>	0 or 1
<b>Default value</b>	0

## 10.17 External reference run IO (0x2010)

### Object description

An external IO node can also be used as a reference switch instead of input 6 of the controller.

<b>Designation</b>	External reference run IO
<b>SDO ID</b>	0x2010
<b>Type</b>	Array
<b>Number of entries</b>	2

### Subindex 00h Number of entries

<b>Type</b>	u8, ro
<b>Default value</b>	2

### Subindex 01h: Bit number

<b>Type</b>	u8, rw
<b>Default value</b>	0

Selects the bit in Subindex 2 that is to be interpreted as a reference switch.

### Subindex 02h Reference switch

<b>Type</b>	u8, rw
<b>Default value</b>	-

This object is used to be mapped as RxFDO. A bit from this object indicates the state of the limit switch to which the controller reacts in the corresponding reference run.

## 10.18 COB ID SYNC message (0x1005)

### Object description

<b>Designation</b>	COB ID SYNC message
<b>SDO ID</b>	0x1005
<b>Type</b>	u32, rw
<b>Value range</b>	0 to 0x7FF (see below)
<b>Default value</b>	0x80

The COB ID of the Sync object is set with this object.

### Bits

The individual bits have the following meaning:

Bit 31: No meaning

Bit 30: Device creates a Sync message (always 0)

Bit 29 uses 29-bit COB ID (always 0)

Bit 28-11: Extended COB ID (always 0)

Bit 10-0: COB ID

## 10.19 Hardware Version (0x1009)

### Object description

<b>Designation</b>	Hardware Version
<b>SDO ID</b>	0x1009
<b>Type</b>	String (ro)
<b>Value range</b>	-
<b>Default value</b>	-

This object contains the hardware version as a character string.

The length of the character string appears in Subindex 0 of this SDO. The individual characters are contained as of Subindex 1. The character string is not terminated by a zero string.

## 10.20 Software Version (0x100A)

### Object description

<b>Designation</b>	Software Version
<b>SDO ID</b>	0x1009
<b>Type</b>	String (ro)
<b>Value range</b>	-
<b>Default value</b>	-

This object contains the software version as character string.

The length of the character string appears in Subindex 0 of this SDO. The individual characters are contained as of Subindex 1. The character string is not terminated by a zero string.

## 10.21 Guard Time (0x100C)

### Object description

<b>Designation</b>	Guard Time
<b>SDO ID</b>	0x100C
<b>Type</b>	u16, rw
<b>Value range</b>	0 to 65535
<b>Default value</b>	0

The Guard Time set in ms. It specifies the time intervals at which a request of the NMT status (RTR of COB ID 0x700 + node ID) is expected.

The Node Guard function can be deactivated with the value 0.

## 10.22 Life Time Factor (0x100D)

### Object description

<b>Designation</b>	Life Time Factor
<b>SDO ID</b>	0x100D
<b>Type</b>	u8, rw
<b>Value range</b>	0 to 255
<b>Default value</b>	0

The Life Time Factor specifies after how many many missing NMT status requests (RTR of 0x700+ node ID) an error is generated.

The time resulting from the objects “Guard Time” and “Life Time Factor” must not exceed one minute.

## 10.23 Store Parameters (0x1010)

### Object description

<b>Designation</b>	Store Parameters
<b>SDO ID</b>	0x1010
<b>Type</b>	u32, rw
<b>Value range</b>	0x65766173
<b>Default value</b>	-

If 0x65766173 is written in this object, the entire object directory is written into the EEPROM of the controller. Thus, settings are retained after a restart of the controller. However, a switch-on (Power State machine) is still necessary.

## 10.24 Restore Parameters (0x1011)

### Object description

<b>Designation</b>	Restore Parameters
<b>SDO ID</b>	0x1011
<b>Type</b>	u32, rw
<b>Value range</b>	0x64616F6C
<b>Default value</b>	-

If 0x64616F6C is written in this object, the entire object directory is reset to the default values. The default values are also immediately transferred into the EEPROM.

## 10.25 Restore Parameters (0x608D)

### Object description

<b>Designation</b>	Acceleration notation index
<b>SDO ID</b>	0x608D
<b>Type</b>	u8, rw
<b>Value range</b>	u8
<b>Default value</b>	0

Reserved.

## 10.26 Restore Parameters (0x608E)

### Object description

<b>Designation</b>	Acceleration dimension index
<b>SDO ID</b>	0x608E
<b>Type</b>	s8, rw
<b>Value range</b>	s8
<b>Default value</b>	0

Reserved.

# 11 Process Data Objects (PDO)

## 11.1 Purpose of the PDFOs

Process Data Objects (PDOs) are used for the transfer of objects that need to be updated frequently during operation of the controller. For example, this is useful for the “Position Actual Value” object.

The advantages of PDOs (compared with SDOs) are the higher and adjustable priority, the low overhead and additional functions such as automatic sending upon a change or cyclical sending.

The higher priority and the low overhead of the PDOs result because the corresponding objects from the object directory are allocated to a CAN object with a certain COB ID without use of the SDO protocol. These allocations are set during the PDO mapping process.

PDOs can be differentiated into receive PDOs and transmit PDOs (RPDO, TPDO). RPDOs are received by the SMCI47-S and the received data are used in the set objects. TPDOs are transmitted by the SMCI47-S in certain (adjustable) situations.

## 11.2 PDO Mapping

### General procedure

The mapping of RPDOs/TPDOs is carried out in several steps:

Step	Action
1	Change to the “Pre-Operational” mode.
2	Deactivate the PDO to be changed and reset.
3	Write changed mapping and activate PDO again.
4	Bring the controller into “Operational” mode again in order to use the PDO features and functions.

### Example: RPDO Mapping

The changing of an RPDO mapping is described below by way of example. The changing of a TPDO mapping is carried out accordingly. The objects that are necessary for adjusting the other PDOs are described from section 8.10.

#### Pre-Operational Mode

In order for the objects to be mapped, it is necessary to switch into Pre-Operational Mode (0x80) first. For more details, see chapter 2.2.

COB ID	Data bytes	Description
0	80 01	Node 1 after Pre-Operational
701	7F	Response Node 1: Pre-Operational

#### Deactivating PDO

To deactivate a PDO, the most significant bit (MSB) in the object “Receive/Transmit PDO Communication Parameter : COB ID” must be set.

→ e.g. object 0x1400 Subindex 1 set to (u32)0x80000000 (deactivates RPDO 1).



COB ID	Data bytes	Description
601	23 00 14 01 00 00 00 80	Write COB ID
581	60 00 14 01 00 00 00 00	Response: OK

#### Set CAN object Identifier (COB ID)

A COB-ID must be assigned for the actual mapping. It is important to note that the COB ID is only assigned to a PDO. The smaller the COB-ID, the higher the priority on the CAN bus.

→ e.g. object 0x1400 Subindex 1 set to (u32)0x80000201.

COB ID	Data bytes	Description
601	23 00 14 01 01 02 00 80	Write COB ID
581	60 00 14 01 00 00 00 00	Response: OK

#### Set Transmission Type

The Transmission Type is specified in SDO 0x1400 Subindex 2. If the Transmission Type is set to 255, the objects mapped to the PDO are immediately updated on reception of the PDO. If the Transmission Type is set to a value of between 0 and 240, the objects mapped in the PDO are updated when receiving the Sync object.

-> e.g. object 0x1400 Subindex 2 set to (u8)255.

COB ID	Data bytes	Description
601	2F 00 14 02 FF 00 00 00	Write Transmission Type
581	60 00 14 02 00 00 00 00	Response: OK

At this point the Subindices 3 and 5 can then also be written with TPDO. These contain the Inhibit Time and the Event Time.

#### Deactivating mapping

The number of mapped objects reset to 0.

→ e.g. object 0x1600 Subindex 0 set to (u8)0.

COB ID	Data bytes	Description
601	2F 00 16 00 00 00 00 00	Write the number of mappings to 0
581	60 00 16 00 00 00 00 00	Response: OK

#### Changing the mapping

The SDO 0x1600 Subindices 1 to 8 defines which objects are mapped from the object directory to this PDO. Each subindex can be written with a u32 that contains the index and subindex of the object to be mapped. Bits 31 to 16 contain the index and bits 15 to 8 contain the subindex. Bits 7 to 0 specify the length of the object to be mapped in bits. Bits 7 to 0 must not be set, these are adopted from the firmware.

In the following example the control word (0x6040 sub 0) is mapped to RPDO 1:

→ e.g. object 0x1600 Subindex 1 set to (u32)60400000.

COB ID	Data bytes	Description
601	23 00 16 01 00 00 40 60	Write Mapping 1
581	60 00 16 01 00 00 00 00	Response: OK

If several objects are mapped to a PDO, it is important to note that a PDO has a maximum length of 8 bytes. If the maximum length of a PDO is exceeded, the firmware registers an error during mapping.

#### Activating mapping

For activation of the mapping, it is necessary to write the number of objects to be mapped to SDO 0x1600 Subindex 0.

→ e.g. object 0x1600 Subindex 0 set to (u8)1.

COB ID	Data bytes	Description
601	2F 00 16 00 01 00 00 00	Write number of mappings to 1
581	60 00 16 00 01 00 00 00	Response: OK

### Activating PDO

The mapping must be activated as the last step. To do so, the MSB in the SDO 0x1400 Subindex 0 must be deleted.

→ e.g. object 0x1400 Subindex 1 set to (u32)0x201.

COB ID	Data bytes	Description
601	2F 00 14 01 01 02 00 00	Write COB ID
581	60 00 14 01 00 00 00 00	Response: OK

### Activation of the node

As the PDOs only function in “Operational” mode, the mode must be changed to this mode after the end of PDO mapping per network management. See section 2.2.

COB ID	Data bytes	Description
0	01 01	Node after Operational
701	5	Response Node 1: Operational

### Save the settings in the EEPROM

So that the changed PDO mapping is not lost when the power supply is switched off, all settings can be saved in the EEPROM from which these are loaded on each start-up.

To do so, the value (u32)0x65766173 must be written to the object 0x1010 Subindex 1 (the value corresponds to the string “save”).

COB ID	Data bytes	Description
601	23 10 10 01 73 61 76 65	Save all settings
581	60 10 10 01 00 00 00 00	Response: OK

## 11.3 Dummy Objects

Dummy objects (SDO 0x0002 to 0x0007) can be used to hide the parts of an RPDO that are determined for another controller from their own controller.

To do this, only during PDO mapping at the required point must a corresponding dummy SDO be mapped instead of a user data SDO. A dummy object can be used repeatedly. Data that are written to a dummy object have no other effect on the controller, except that the dummy object is written.

## 12 Commissioning via RS485

In order to operate the controller via the RS485 interface (if present), the corresponding firmware must be installed. Operation of the controller in RS485 mode is not possible with the CANopen firmware.

## 13 Appendix: Abbreviations used

Abbreviation	Meaning
PDO	Process Data Object
TPDO	Transmit Process Data Object
RPDO	Receive Process Data Object
SDO	Service Data Object
SYNC	Synchronisation Object
EMCY	Emergency Object
ro	read only
rw	read write
PP	Profile Position Mode
VL	Velocity Mode
HM	Homing Mode (reference mode)
IP	Interpolated Position Mode
COB ID	CAN object ID

## 14 Appendix: Possible error messages

Error code	Description
0503 0000h	Toggle bit not alternated
0504 0000h	SDO protocol timed out.
0504 0001h	Client/server command specifier not valid or unknown.
0504 0002h	Invalid block size (block mode only).
0504 0003h	Invalid sequence number (block mode only).
0504 0004h	CRC error (block mode only).
0504 0005h	Out of memory
0601 0000h	Unsupported access to an object
0601 0001h	Attempt to read a write only object
0601 0002h	Attempt to write a read only object
0602 0000h	Object does not exist in the object dictionary
0604 0041h	Object cannot be mapped to the PDO
0604 0042h	The number and length of the objects to be mapped would exceed
0604 0043h	General parameter incompatibility reason
0604 0047h	General internal incompatibility in the device
0606 0000h	Access failed due to an hardware error
0607 0010h	Data type does not match, length of service parameter does not match
0607 0012h	Data type does not match, length of service parameter too high
0607 0013h	Data type does not match, length of service parameter too low
0609 0011h	Sub-index does not exist
0609 0030h	Value range of parameter exceeded (only for write access)
0609 0031h	Value of parameter written too high
0609 0032h	Value of parameter written too low
0609 0036h	Maximum value is less than minimum value
0800 0000h	general error
0800 0020h	Data cannot be transferred or stored to the application
0800 0021h	Data cannot be transferred or stored to the application because of
0800 0022h	Data cannot be transferred or stored to the application because of the
0800 0023h	Object dictionary dynamic generation fails or no object dictionary is