CAN 300 / CAN 400 with Maxon Motor

CAN Communication Module for S7-300/S7-400
Application Example Maxon Motor with CANopen

Manual

Edition 1 / 2006-02-09

Systeme Helmholz **GmbH** ● Gewerbegebiet Ost 36 ● 91085 Weisendorf● Germany
Tel.: +49 9135 7380-0 ● Fax: +49 9135 7380-50 ● E-mail: info@helmholz.de ● Internet: www.helmholz.de

Systeme Helmholz GmbH

Gewerbegebiet Ost 36, 91085 Weisendorf, Germany

Note:

We have checked the content of this manual for conformity with the hardware and software described. Nevertheless, because deviations cannot be ruled out, we cannot accept any liability for complete conformity. The data in this manual have been checked regularly and any necessary corrections will be included in subsequent editions. We always welcome suggestions for improvement.

Contents

1 Overview 5

1.1 Application and function description 5

1.2 PLC configuration 6

1.3 Maxon motor configuration 7

2 Configuring the CAN 300 module 8

2.1 Setting the CAN bus baudrate 8

2.2 Setting the transmission mode (protocol) 8

2.3 Acceptance masks 9

2.4 Network management 9

2.5 Timer 10

3 Settings on the EPOS 11

3.1 Configuration of the PDOs used (mapping) 11

4 Programming in the PLC 12

4.1 Overview 12

4.1.1 Example FC 10 (cycle/SDO/PDO/network management) 12

4.1.2 Example FC 13 (writing an SDO list) 13

4.2 Content of the status word 14

5 EPOS24/5 parameter file 15

1 Overview

1.1 Application and function description

CANopen

This manual describes the application example (handling blocks) for controlling a Maxon motor with a S7-300 using a CAN 300 module. The application example is also applicable to the CAN 400 module for the S7-400. The functionality is transferrable.

It is for use as a supplement to the “CAN 300” or “CAN 400” manual. The information in these manuals is assumed to be known, in particular, the description of the CANopen protocol and the CANopen handling.

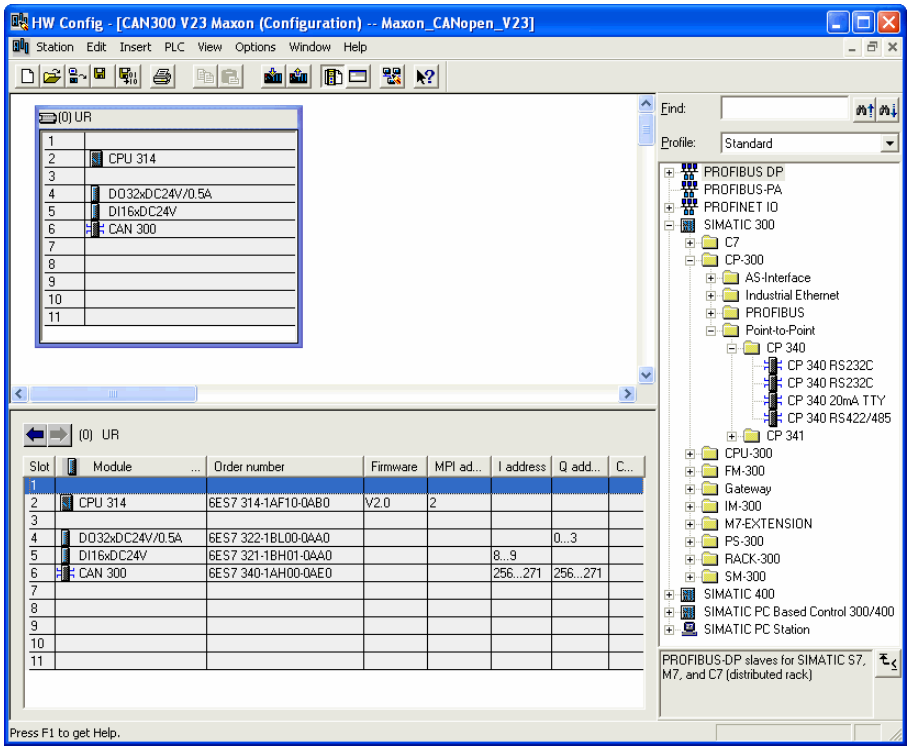
In addition to this, the CANopen profile description “DSP-402 Device Profile for Drives and Motion” and the CANopen description from Maxon “EPOS Firmware Specification”, especially Chapter “Object Dictionary” are very helpful.

Maxon Motor AG has also made an extensive range of handling blocks for use with the CAN 300 with which all functionality of the EPOS controller can be programmed. These “encapsulate” the CANopen Master handling blocks and provide all motor functions.

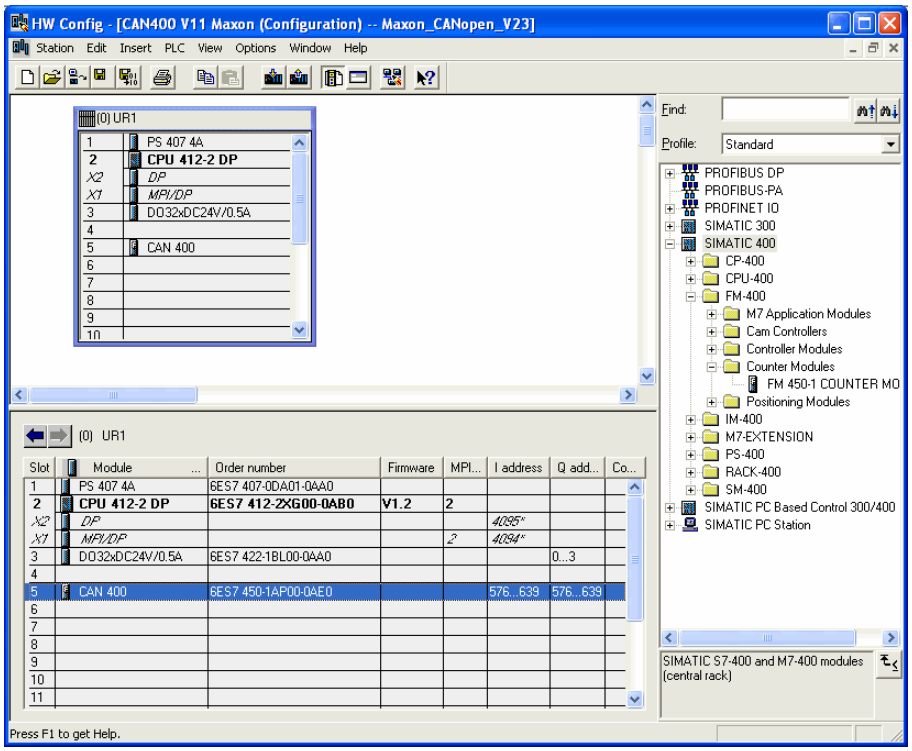
The example described here is more simply structured to show the direct use of PDOs, SDOs, and network management. The example can also be applied to other drive manufacturers that support the CANopen profile DS402 without any problem.

1.2 PLC configuration

An S7-300 CPU314 (6ES7-314-1AF10), a 32-bit digital output, a 16-bit digital input module, and a CAN 300 module are used.

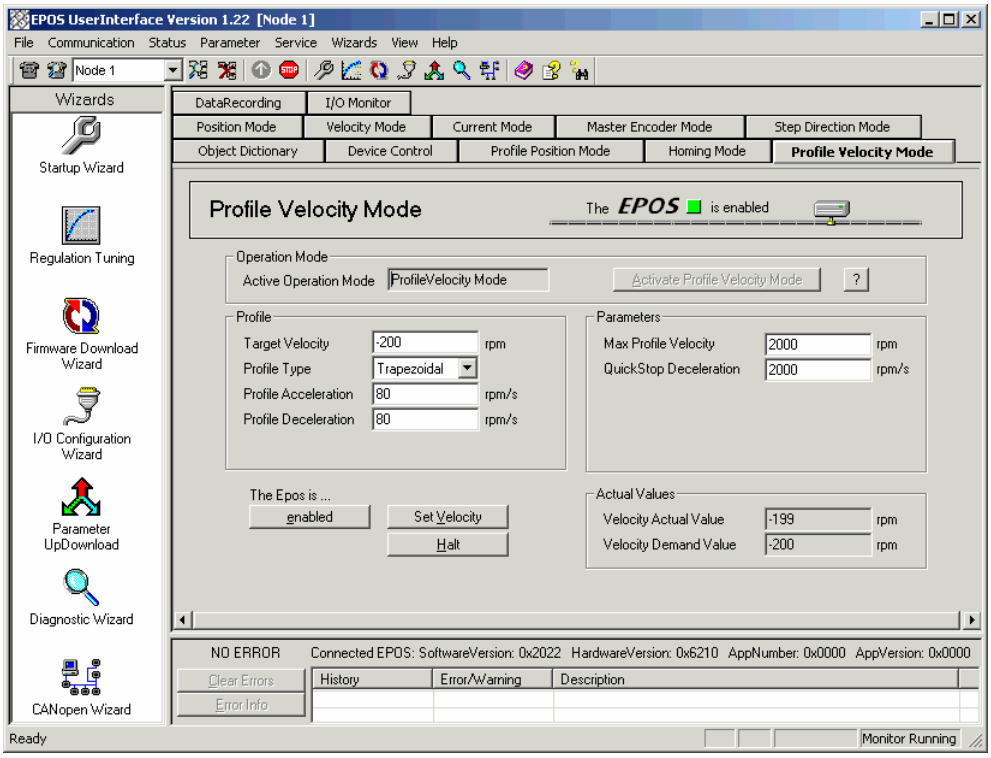


For the CAN 400, the hardware configuration looks like this:



1.3 Maxon motor configuration

The handling example was created with an EPOS 24/5 (firmware 2022). The motor is connected as node 1 and is used in velocity mode. The example program can be moved forward and backward between two final positions.



This handling example assumes that the EPOS system is ready for use (cabling, tuning, parameters of the motor used, etc.).

Please read carefully the Maxon Motor documentation on configuring and commissioning the EPOS and the motor.

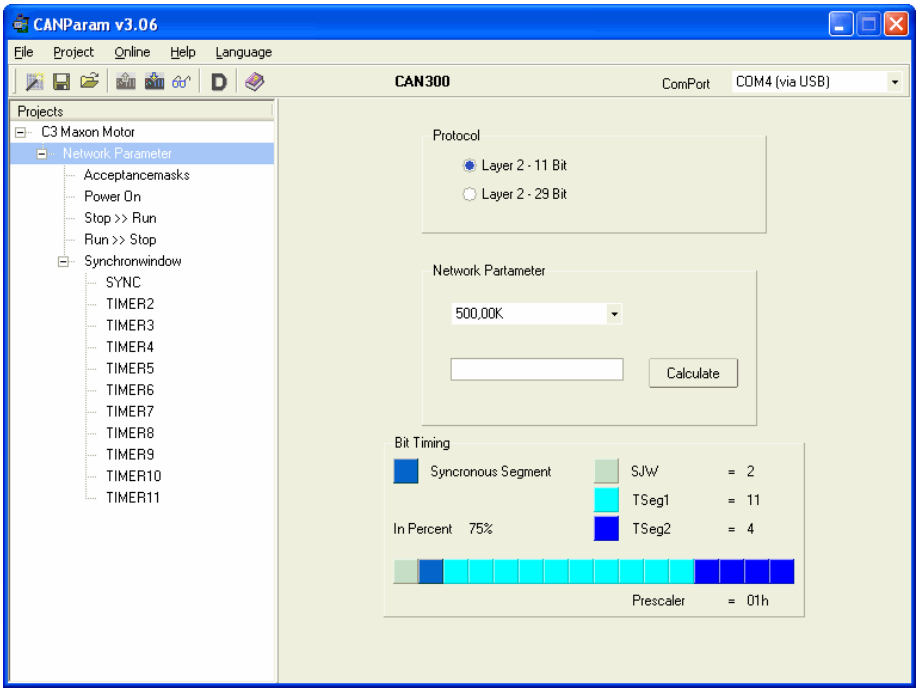
With the “Object Dictionary” of the EPOS UserInterfaces it is possible to monitor the CANopen SDO Objects directly.

2 Configuring the CAN 300 module

To be able to use the example program, the attached CAN project “C3 Maxon Motor.par” must be imported into the CAN 300 module. For the CAN 400 module, the “C4 Maxon Motor.par” project is attached.

2.1 Setting the CAN bus baudrate

The CAN bus baudrate must be set to match the setting for the EPOS.

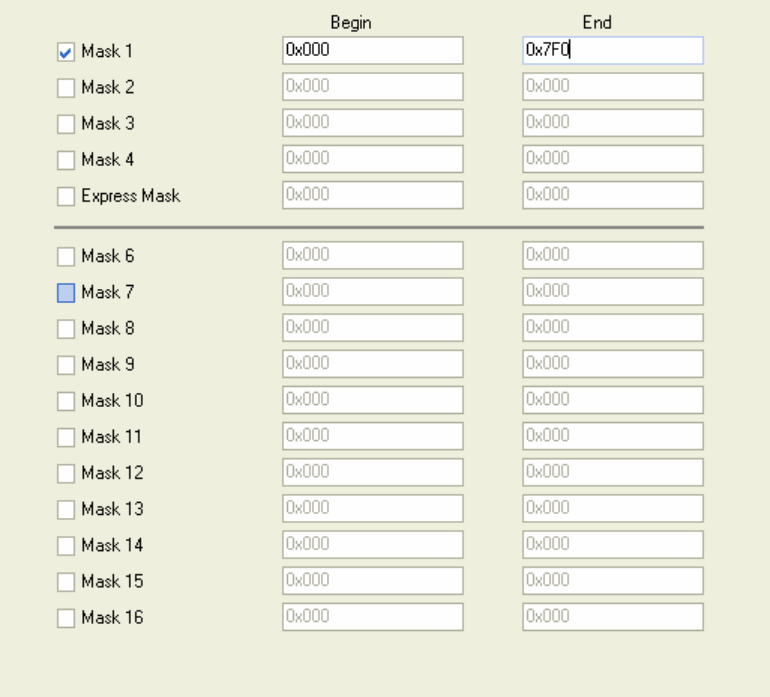


2.2 Setting the transmission mode (protocol)

The transmission mode must always be set to “Layer 2 – 11Bit” for all CANopen applications.

2.3 Acceptance masks

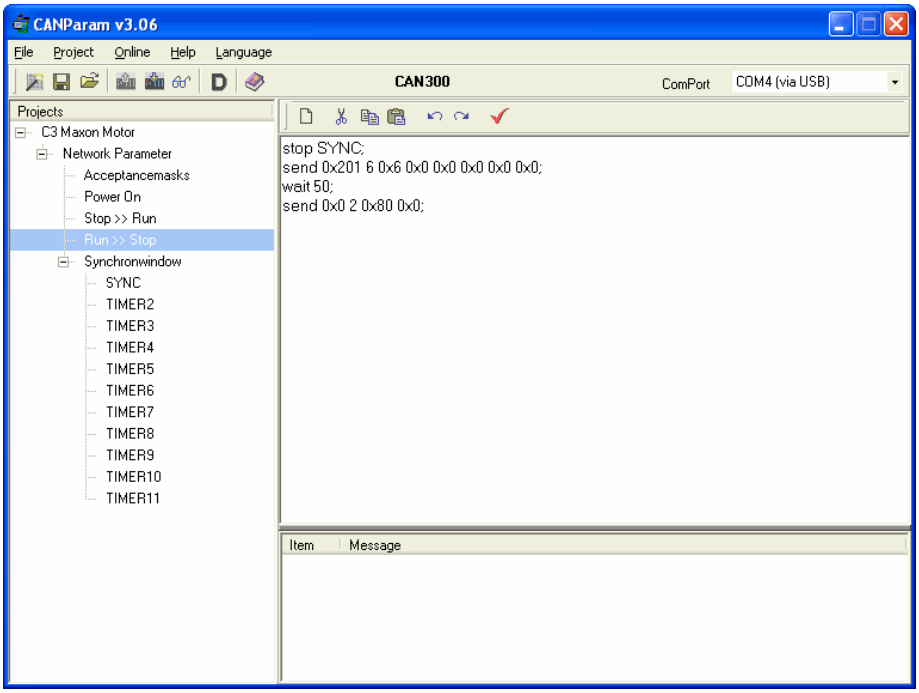
For CANopen Master applications, all CAN bus telegrams are normally always allowed through to the PLC.



2.4 Network management

In the example project, the scripts “Off->On” and “On->Off” are used.

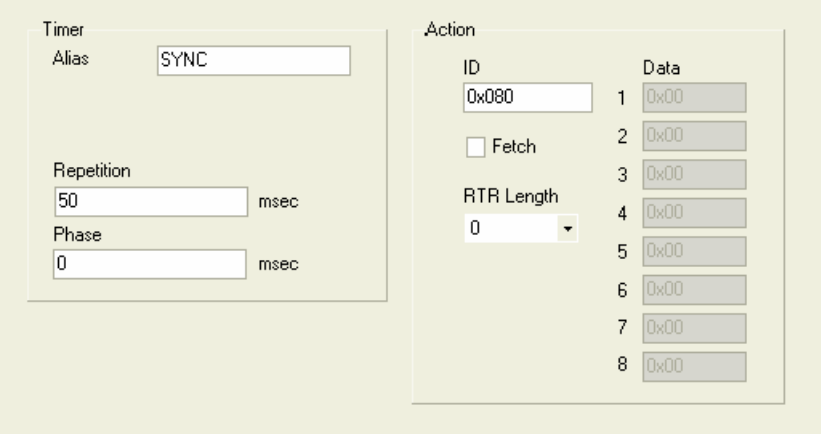
In the start-up script, the Maxon motor is put into a defined state and the SYNC timer is activated.



If the PLC is stopped, the “On->Off” script is executed. This script stops the SYNC timer, transmits a telegram to the PDO1 of the slave, with which the speed is set to 0, and then stops the CANopen bus (NMT Stop all Nodes).

2.5 Timer

The SYNC timer is started in the “Off->On” script.



3 Settings on the EPOS

The EPOS must be commissioned correctly according to the Maxon Motor instructions (cabling, tuning, parameters of the motor used, etc.). It may be necessary to perform a “fault reset” of the EPOS before starting the PLC with the EPOS UserInterface.

The parameters for communication (PDO Mapping) and the basic settings for the velocity profile are transmitted to the EPSO through the PLC (SDO transmission with FC13).

3.1 Configuration of the PDOs used (mapping)

The mapping of the PDOs in the EPOS is performed via objects 1600 ff. (RxPDOs) and 1A00 ff. (TxPDOs).

TxPDO1 (is transmitted by the EPOS): ID 181_{hex}

Byte 0-3: SDO 606C/0 Velocity Actual Value

Byte 4-7: SDO 6064/0 Position Actual Value

TxPDO2 (is transmitted by the EPOS): ID 281_{hex}

Byte 0+1: SDO 6041/0 Status Word

Byte 2: SDO 6061/0 Mode of Operation Display

RxPDO1 (is received by the EPOS): ID 201_{hex}

Byte 0+1: SDO 6040/0 Control Word

Byte 2-5: SDO 60FF/0 Target Velocity

RxPDO2 (is received by the EPOS): ID 301_{hex}

Byte 0+1: SDO 6040/0 Control Word

Byte 2: SDO 6060/0 Modes of Operation

The TxPDOs is only transmitted from the EPOS if the EPOS is in CANopen mode “operational”.

It is only possible to change the PDO mappings in CANopen mode “preoperational”.

4 Programming in the PLC

4.1 Overview

The example contains the handling blocks of the CANopen Master handling (FC40 – FC49) that are explained in detail in the “CAN 300” manual. The application example is structured for the CAN 400 in the same way. The explanations can be applied to it without any problem.

FC 10 and FC 13 are the blocks specially created for handling the Maxon motor and they call the blocks of the CANopen Master handling.

The functions of the handling blocks are triggered by the bits of input byte 8. Input byte 8 is copied to MB108 (in the OB1). The example can therefore also be used to control MB108 without an input or test module.

4.1.1 Example FC 10 (cycle/SDO/PDO/network management)

At the start of FC 10, a cycle block FC 49 is called to fetch telegrams received via the CAN bus or to execute transmit jobs.

The receive data of the PDO1 and PDO2 of Node 1 (“Velocity Actual Value”, “Position actual value”, “Status word”, “Mode of Operation”) are copied from the PDO-DBs into MD60, MD64, MW68, and MB70 to be able to access them as the block continues to run.

In the second network, complete initialization of the EPOS is performed, triggered by input bit 8.7. The initialization sequence is:

1. NMT set Preoperational: Stop CANopen node and get it ready for parameterization
2. Init SDOs: Call of FC13 to transfer all necessary SDOs (PDO mapping, velocity profile)
3. NMT Start All Nodes: Start all CANopen nodes; the PDOs are running as from this moment
4. Drive Shutdown & Set Mode of Operation: The motor is put into shutdown via PDO2 and the velocity mope is activated
5. Drive Enable: The motor is enabled via the PDO2

Network 3 contains the actual closed-loop control. This is activated via input byte 8.0. However, it only runs if the EPOS has been initialized first (IB 8.7).

Initialization (IB8.7) and closed-loop control (IB 8.0) must be initiated one after the other.

The PDO1 can be transmitted, triggered by the input byte 8.1 in network 4. PDO1 is transmitted with the “control word” and the velocity setting value from MD76 and the lower byte of the control word is read from IB9.

Network 4 contains an example of fetching and transmitting an SDO (FC 41) via input bits 8.2 and 8.3. The parameters passed have been routed to MW. These functions can be used for your own experiments.

In the last network, it is possible to activate Nodeguarding with input bit 8.6 via FC 47. DB47 contains a list with the node numbers 1+2+3 that are queried cyclically.

4.1.2 Example FC 13 (writing an SDO list)

A list of SDOs is written in FC13. The list of the SDOs to be written with its values is located in DB 13. This functionality is called in network 2 of FC10.

In FC 13, FC 41 „SDOWriteRead“ is called one after the other for each SDO in DB 13 and the result (error number or OK message) is also stored in DB 13.

For the change to the PDO mapping that is performed via FC 13, the EPOS must be in the “preoperational” state.

4.2 Content of the status word

The status word of the EPOS24/5 is transmitted cyclically to the S7 via the PDO2 and stored in MW68.

State	Statusword [binary]	Description
Start	x0xx xxx0 x000 0000	Bootup
Not Ready to Switch On	x0xx xxx1 x000 0000	The current offset will be measured The drive function is disabled
Switch On Disabled	x0xx xxx1 x100 0000	The drive initialization is complete The drive parameters may be changed The drive function is disabled
Ready to Switch On	x0xx xxx1 x010 0001	The drive parameters may be changed The drive function is disabled
Switched On	x0xx xxx1 x010 0011	The drive function is disabled
Refresh	x1xx xxx1 x010 0011	Refresh Power Stage
Operation Enable	x0xx xxx1 x011 0111	No faults have been detected The Drive function is enabled and power is applied to motor
Quick Stop Active	x0xx xxx1 x001 0111	The quick stop function is being executed The Drive function is enabled and power is applied to motor
Fault Reaction Active (disabled)	x0xx xxx1 x000 1111	A fault has occurred in the drive The drive function is disabled
Fault Reaction Active (enabled)	x0xx xxx1 x001 1111	A fault has occurred in the drive The quick stop function is being executed The Drive function is enabled and power is applied to motor
Fault	x0xx xxx1 x000 1000	A fault has occurred in the drive The drive parameters may be changed The drive function is disabled

The control word is transmitted from the S7 to the EPOS24/5 via the PDO1. This can be triggered via input byte 8.1 (network 4). Only the lower byte of the control word is relevant to normal applications.

Command	Lowbyte of Controlword [binary]
Shutdown	0xxx x110
Switch On	0xxx x111
Disable Voltage	0xxx xx0x
Quick Stop	0xxx x01x
Disable Operation	0xxx 0111
Enalbe Operation	0xxx 1111
Fault Reset	0xxx xxxx → 1xxx xxxx

You will find more detailed explanations on the status and control word in the “EPOS Firmware Specification” in Chapter 8.

5 EPOS24/5 parameter file

The highlighted objects are necessary for correct functioning of the handling example. These are correctly set by the FC 13.

ParameterFile Version 1.02 maxon motor ag, CH-6072 Sachseln

Object Filter All Objects
Version Software: 0x2022 Hardware: 0x6210
Application Number: 0x0000 Version: 0x0000

Index	Sub	Name	Type		ValueHex	ValueDec
0x1000	0x00	Device Type	UInt32	Const	0x00020192	131474
0x1001	0x00	Error Register	UInt8	RO	0x00	0
0x1018	0x01	Vendor ID	UInt32	RO	0x000000FB	251
0x1018	0x02	Product Code	UInt32	RO	0x62102022	1645223970
0x1003	0x00	Number of Errors	UInt8	RW	0x00	0
0x1003	0x01	Error History (1)	UInt32	RO	0x00000000	0
0x1003	0x02	Error History (2)	UInt32	RO	0x00000000	0
0x1003	0x03	Error History (3)	UInt32	RO	0x00000000	0
0x1003	0x04	Error History (4)	UInt32	RO	0x00000000	0
0x1003	0x05	Error History (5)	UInt32	RO	0x00000000	0
0x1005	0x00	COB-ID SYNC Message	UInt32	RW	0x00000080	128
0x1008	0x00	Device Name	String	Const	EPOS	
0x100C	0x00	Guard Time	UInt16	RW	0x0000	0
0x100D	0x00	Life Time Factor	UInt8	RW	0x00	0
0x1010	0x00	Largest SubIndex Supported	UInt8	RO	0x01	1
0x1010	0x01	Save All Parameters	UInt32	RW	0x00000001	1
0x1011	0x00	Largest SubIndex Supported	UInt8	RO	0x05	5
0x1011	0x01	Restore All Default Parameters	UInt32	RW	0x00000001	1
0x1011	0x05	Restore Default PDO COB-IDs	UInt32	RW	0x00000001	1
0x1014	0x00	COB-ID Emergency Object	UInt32	RO	0x00000081	129
0x1017	0x00	Producer Heartbeat Time	UInt16	RW	0x0000	0
0x1200	0x01	SDO COB-ID Client->Server (rx)	UInt32	RO	0x00000601	1537
0x1200	0x02	SDO COB-ID Server->Client (tx)	UInt32	RO	0x00000581	1409
0x1400	0x00	Largest SubIndex Supported	UInt8	RO	0x02	2
0x1400	0x01	COB-ID used by RxPDO 1	UInt32	RW	0x00000201	513
0x1400	0x02	Transmission Type	UInt8	RW	0xFF	255
0x1401	0x00	Largest SubIndex Supported	UInt8	RO	0x02	2
0x1401	0x01	COB-ID used by RxPDO 2	UInt32	RW	0x00000301	769
0x1401	0x02	Transmission Type	UInt8	RW	0xFF	255
0x1402	0x00	Largest SubIndex Supported	UInt8	RO	0x02	2
0x1402	0x01	COB-ID used by RxPDO 3	UInt32	RW	0x00000401	1025
0x1402	0x02	Transmission Type	UInt8	RW	0xFF	255
0x1403	0x00	Largest SubIndex Supported	UInt8	RO	0x02	2
0x1403	0x01	COB-ID used by RxPDO 4	UInt32	RW	0x00000501	1281
0x1403	0x02	Transmission Type	UInt8	RW	0xFF	255
0x1600	**0x00**	**Number of Mapped Appl. Objects in RxPDO 1**	**UInt8**	**RW**	**0x02**	**2**
0x1600	0x01	1st Mapped Object in RxPDO 1	UInt32	RW	0x60400010	1614807056
0x1600	**0x02**	**2nd Mapped Object in RxPDO 1**	**UInt32**	**RW**	**0x60FF0020**	**1627324448**
0x1600	0x03	3rd Mapped Object in RxPDO 1	UInt32	RW	0x00000000	0
0x1600	0x04	4th Mapped Object in RxPDO 1	UInt32	RW	0x00000000	0
0x1600	0x05	5th Mapped Object in RxPDO 1	UInt32	RW	0x00000000	0
0x1600	0x06	6th Mapped Object in RxPDO 1	UInt32	RW	0x00000000	0
0x1600	0x07	7th Mapped Object in RxPDO 1	UInt32	RW	0x00000000	0
0x1600	0x08	8th Mapped Object in RxPDO 1	UInt32	RW	0x00000000	0
0x1601	0x00	Number of Mapped Appl. Objects in RxPDO 2	UInt8	RW	0x02	2
0x1601	0x01	1st Mapped Object in RxPDO 2	UInt32	RW	0x60400010	1614807056
0x1601	0x02	2nd Mapped Object in RxPDO 2	UInt32	RW	0x60600008	1616904200
0x1601	0x03	3rd Mapped Object in RxPDO 2	UInt32	RW	0x00000000	0
0x1601	0x04	4th Mapped Object in RxPDO 2	UInt32	RW	0x00000000	0
0x1601	0x05	5th Mapped Object in RxPDO 2	UInt32	RW	0x00000000	0
0x1601	0x06	6th Mapped Object in RxPDO 2	UInt32	RW	0x00000000	0
0x1601	0x07	7th Mapped Object in RxPDO 2	UInt32	RW	0x00000000	0
0x1601	0x08	8th Mapped Object in RxPDO 2	UInt32	RW	0x00000000	0
0x1602	0x00	Number of Mapped Appl. Objects in RxPDO 3	UInt8	RW	0x02	2
0x1602	0x01	1st Mapped Object in RxPDO 3	UInt32	RW	0x60400010	1614807056
0x1602	0x02	2nd Mapped Object in RxPDO 3	UInt32	RW	0x607A0020	1618608160
0x1602	0x03	3rd Mapped Object in RxPDO 3	UInt32	RW	0x00000000	0
0x1602	0x04	4th Mapped Object in RxPDO 3	UInt32	RW	0x00000000	0
0x1602	0x05	5th Mapped Object in RxPDO 3	UInt32	RW	0x00000000	0
0x1602	0x06	6th Mapped Object in RxPDO 3	UInt32	RW	0x00000000	0
0x1602	0x07	7th Mapped Object in RxPDO 3	UInt32	RW	0x00000000	0
0x1602	0x08	8th Mapped Object in RxPDO 3	UInt32	RW	0x00000000	0
0x1603	0x00	Number of Mapped Appl. Objects in RxPDO 4	UInt8	RW	0x02	2
0x1603	0x01	1st Mapped Object in RxPDO 4	UInt32	RW	0x60400010	1614807056
0x1603	0x02	2nd Mapped Object in RxPDO 4	UInt32	RW	0x60FF0020	1627324448
0x1603	0x03	3rd Mapped Object in RxPDO 4	UInt32	RW	0x00000000	0
0x1603	0x04	4th Mapped Object in RxPDO 4	UInt32	RW	0x00000000	0
0x1603	0x05	5th Mapped Object in RxPDO 4	UInt32	RW	0x00000000	0
0x1603	0x06	6th Mapped Object in RxPDO 4	UInt32	RW	0x00000000	0
0x1603	0x07	7th Mapped Object in RxPDO 4	UInt32	RW	0x00000000	0
0x1603	0x08	8th Mapped Object in RxPDO 4	UInt32	RW	0x00000000	0
0x1800	0x00	Largest SubIndex Supported	UInt8	RO	0x03	3

0x1800	0x01	COB-ID used by TxPDO 1	UInt32	RW	0x00000181	385
0x1800	**0x02**	**Transmission Type**	**UInt8**	**RW**	**0x01**	**1**
0x1800	0x03	Inhibit Time	UInt16	RW	0xFFFF	65535
0x1801	0x00	Largest SubIndex Supported	UInt8	RO	0x03	3
0x1801	0x01	COB-ID used by TxPDO 2	UInt32	RW	0x00000281	641
0x1801	**0x02**	**Transmission Type**	**UInt8**	**RW**	**0x01**	**1**
0x1801	0x03	Inhibit Time	UInt16	RW	0xFFFF	65535
0x1802	0x00	Largest SubIndex Supported	UInt8	RO	0x03	3
0x1802	0x01	COB-ID used by TxPDO 3	UInt32	RW	0x00000381	897
0x1802	0x02	Transmission Type	UInt8	RW	0xFD	253
0x1802	0x03	Inhibit Time	UInt16	RW	0xFFFF	65535
0x1803	0x00	Largest SubIndex Supported	UInt8	RO	0x03	3
0x1803	0x01	COB-ID used by TxPDO 4	UInt32	RW	0x00000481	1153
0x1803	0x02	Transmission Type	UInt8	RW	0xFD	253
0x1803	0x03	Inhibit Time	UInt16	RW	0xFFFF	65535
0x1A00	**0x00**	**Number of Mapped Appl. Objects in TxPDO 1**	**UInt8**	**RW**	**0x02**	**2**
0x1A00	**0x01**	**1st Mapped Object in TxPDO 1**	**UInt32**	**RW**	**0x606C0020**	**1617690656**
0x1A00	**0x02**	**2nd Mapped Object in TxPDO 1**	**UInt32**	**RW**	**0x60640020**	**1617166368**
0x1A00	0x03	3rd Mapped Object in TxPDO 1	UInt32	RW	0x00000000	0
0x1A00	0x04	4th Mapped Object in TxPDO 1	UInt32	RW	0x00000000	0
0x1A00	0x05	5th Mapped Object in TxPDO 1	UInt32	RW	0x00000000	0
0x1A00	0x06	6th Mapped Object in TxPDO 1	UInt32	RW	0x00000000	0
0x1A00	0x07	7th Mapped Object in TxPDO 1	UInt32	RW	0x00000000	0
0x1A00	0x08	8th Mapped Object in TxPDO 1	UInt32	RW	0x00000000	0
0x1A01	0x00	Number of Mapped Appl. Objects in TxPDO 2	UInt8	RW	0x02	2
0x1A01	0x01	1st Mapped Object in TxPDO 2	UInt32	RW	0x60410010	1614872592
0x1A01	0x02	2nd Mapped Object in TxPDO 2	UInt32	RW	0x60610008	1616969736
0x1A01	0x03	3rd Mapped Object in TxPDO 2	UInt32	RW	0x00000000	0
0x1A01	0x04	4th Mapped Object in TxPDO 2	UInt32	RW	0x00000000	0
0x1A01	0x05	5th Mapped Object in TxPDO 2	UInt32	RW	0x00000000	0
0x1A01	0x06	6th Mapped Object in TxPDO 2	UInt32	RW	0x00000000	0
0x1A01	0x07	7th Mapped Object in TxPDO 2	UInt32	RW	0x00000000	0
0x1A01	0x08	8th Mapped Object in TxPDO 2	UInt32	RW	0x00000000	0
0x1A02	0x00	Number of Mapped Appl. Objects in TxPDO 3	UInt8	RW	0x02	2
0x1A02	0x01	1st Mapped Object in TxPDO 3	UInt32	RW	0x60410010	1614872592
0x1A02	0x02	2nd Mapped Object in TxPDO 3	UInt32	RW	0x60640020	1617166368
0x1A02	0x03	3rd Mapped Object in TxPDO 3	UInt32	RW	0x00000000	0
0x1A02	0x04	4th Mapped Object in TxPDO 3	UInt32	RW	0x00000000	0
0x1A02	0x05	5th Mapped Object in TxPDO 3	UInt32	RW	0x00000000	0
0x1A02	0x06	6th Mapped Object in TxPDO 3	UInt32	RW	0x00000000	0
0x1A02	0x07	7th Mapped Object in TxPDO 3	UInt32	RW	0x00000000	0
0x1A02	0x08	8th Mapped Object in TxPDO 3	UInt32	RW	0x00000000	0
0x1A03	0x00	Number of Mapped Appl. Objects in TxPDO 4	UInt8	RW	0x02	2
0x1A03	0x01	1st Mapped Object in TxPDO 4	UInt32	RW	0x60410010	1614872592
0x1A03	0x02	2nd Mapped Object in TxPDO 4	UInt32	RW	0x606C0020	1617690656
0x1A03	0x03	3rd Mapped Object in TxPDO 4	UInt32	RW	0x00000000	0
0x1A03	0x04	4th Mapped Object in TxPDO 4	UInt32	RW	0x00000000	0
0x1A03	0x05	5th Mapped Object in TxPDO 4	UInt32	RW	0x00000000	0
0x1A03	0x06	6th Mapped Object in TxPDO 4	UInt32	RW	0x00000000	0
0x1A03	0x07	7th Mapped Object in TxPDO 4	UInt32	RW	0x00000000	0
0x1A03	0x08	8th Mapped Object in TxPDO 4	UInt32	RW	0x00000000	0
0x6040	0x00	ControlWord	UInt16	RW	0x0000	0
0x6041	0x00	StatusWord	UInt16	RO	0x1740	5952
0x6060	0x00	Modes of Operation	Int8	RW	0x03	3
0x6061	0x00	Modes of Operation Display	Int8	RO	0x03	3
0x6062	0x00	Position Demand Value	Int32	RO	0x00004577	17783
0x6064	0x00	Position Actual Value	Int32	RO	0x00003CAB	15531
0x6065	0x00	Max Following Error	UInt32	RW	0x000007D0	2000
0x6067	0x00	Position Window	UInt32	RW	0xFFFFFFFF	4294967295
0x6068	0x00	Position Window Time	UInt16	RW	0x0000	0
0x6069	0x00	Velocity Sensor Actual Value	Int32	RO	0x00000000	0
0x606B	0x00	Velocity Demand Value	Int32	RO	0x00000000	0
0x606C	0x00	Velocity Actual Value	Int32	RO	0x00000000	0
0x6078	0x00	Current Actual Value	Int16	RO	0xFFE6	-26
0x607A	0x00	Target Position	Int32	RW	0x00000000	0
0x607C	0x00	Home Offset	Int32	RW	0x00000000	0
0x607D	0x01	Min Position Limit	Int32	RW	0x80000000	-2147483648
0x607D	0x02	Max Position Limit	Int32	RW	0x7FFFFFFF	2147483647
0x607F	**0x00**	**Max Profile Velocity**	**UInt32**	**RW**	**0x000007D0**	**2000**
0x6081	0x00	Profile Velocity	UInt32	RW	0x000003E8	1000
0x6083	**0x00**	**Profile Acceleration**	**UInt32**	**RW**	**0x00000050**	**80**
0x6084	**0x00**	**Profile Deceleration**	**UInt32**	**RW**	**0x00000050**	**80**
0x6085	**0x00**	**QuickStop Deceleration**	**UInt32**	**RW**	**0x000007D0**	**2000**
0x6086	0x00	Motion ProfileType	Int16	RW	0x0000	0
0x6089	0x00	Position Notation Index	Int8	RW	0x00	0
0x608A	0x00	Position Dimension Index	UInt8	RW	0xAC	172
0x608B	0x00	Velocity Notation Index	Int8	RW	0x00	0
0x608C	0x00	Velocity Dimension Index	UInt8	RW	0xA4	164
0x608D	0x00	Acceleration Notation Index	Int8	RW	0x00	0
0x608E	0x00	Acceleration Dimension Index	UInt8	RW	0xA4	164
0x6098	0x00	Homing Method	Int8	RW	0x07	7
0x6099	0x01	Speed for Switch Search	UInt32	RW	0x00000064	100
0x6099	0x02	Speed for Zero Search	UInt32	RW	0x0000000A	10
0x609A	0x00	Homing Acceleration	UInt32	RW	0x000003E8	1000
0x60F6	0x01	Current Regulator P-Gain	Int16	RW	0x0320	800
0x60F6	0x02	Current Regulator I-Gain	Int16	RW	0x00C8	200
0x60F9	0x01	Speed Regulator P-Gain	Int16	RW	0x03E8	1000
0x60F9	0x02	Speed Regulator I-Gain	Int16	RW	0x0064	100
0x60FB	0x01	Position Regulator P-Gain	Int16	RW	0x0096	150
0x60FB	0x02	Position Regulator I-Gain	Int16	RW	0x000A	10

0x60FB	0x03	Position Regulator D-Gain	Int16	RW	0x00C8	200
0x60FB	0x04	Velocity Feedforward Factor	UInt16	RW	0x0000	0
0x60FB	0x05	Acceleration Feedforward Factor	UInt16	RW	0x0000	0
0x60FF	0x00	TargetVelocity	Int32	RW	0xFFFFFF38	-200
0x6402	0x00	MotorType	UInt16	RW	0x000A	10
0x6410	0x01	Continuous Current Limit	UInt16	RW	0x1388	5000
0x6410	0x02	Output Current Limit	UInt16	RW	0x2710	10000
0x6410	0x03	PolePair Number	UInt8	RW	0x01	1
0x6410	0x04	Maximal Speed in CurrentMode	UInt16	RW	0x7530	30000
0x6410	0x05	Thermal Time Constant Winding	UInt16	RW	0x0028	40
0x6502	0x00	Supported Drive Modes	UInt32	Const	0x003F0025	4128805
0x2000	0x00	Node ID	UInt8	RW	0x01	1
0x2001	0x00	CAN Bitrate	UInt16	RW	0x0002	2
0x2002	0x00	RS232 Baudrate	UInt16	RW	0x0003	3
0x2003	0x01	Software Version	UInt16	RO	0x2022	8226
0x2003	0x02	Hardware Version	UInt16	RO	0x6210	25104
0x2003	0x03	Application Number	UInt16	RO	0x0000	0
0x2003	0x04	Application Version	UInt16	RO	0x0000	0
0x2003	0x05	Basesector Version	UInt16	RO	0x001D	29
0x2004	0x00	Serial Number	UInt64	Const	0x00000000	0
0x2005	0x00	RS232 Frame Timeout	UInt16	RW	0x01F4	500
0x2008	0x00	Miscellaneous Configuration	UInt16	RW	0x0000	0
0x200C	0x01	Custom persistent memory 1	UInt32	RW	0x00000000	0
0x200C	0x02	Custom persistent memory 2	UInt32	RW	0x00000000	0
0x200C	0x03	Custom persistent memory 3	UInt32	RW	0x00000000	0
0x200C	0x04	Custom persistent memory 4	UInt32	RW	0x00000000	0
0x2010	0x00	DataRecorder Control	UInt16	RW	0x0000	0
0x2011	0x00	DataRecorder Configuration	UInt16	RW	0x0003	3
0x2012	0x00	DataRecorder Sampling Period	UInt16	RW	0x000A	10
0x2013	0x00	DataRecorder Number of Preceding Samples	UInt16	RW	0x0000	0
0x2014	0x00	DataRecorder Number of Sampling Variables	UInt16	RW	0x0000	0
0x2015	0x01	DataRecorder Index of Variable 1	UInt16	RW	0x0000	0
0x2015	0x02	DataRecorder Index of Variable 2	UInt16	RW	0x0000	0
0x2015	0x03	DataRecorder Index of Variable 3	UInt16	RW	0x0000	0
0x2015	0x04	DataRecorder Index of Variable 4	UInt16	RW	0x0000	0
0x2016	0x01	DataRecorder SubIndex of Variable 1	UInt16	RW	0x0000	0
0x2016	0x02	DataRecorder SubIndex of Variable 2	UInt16	RW	0x0000	0
0x2016	0x03	DataRecorder SubIndex of Variable 3	UInt16	RW	0x0000	0
0x2016	0x04	DataRecorder SubIndex of Variable 4	UInt16	RW	0x0000	0
0x2017	0x00	DataRecorder Status	UInt16	RO	0x0000	0
0x2018	0x00	DataRecorder Max Number of Samples	UInt16	RO	0x0200	512
0x2019	0x00	DataRecorder Number of Recorded Samples	UInt16	RO	0x0000	0
0x201A	0x00	DataRecorder Vector Start Offset	UInt16	RO	0x0000	0
0x201B	0x00	DataRecorder Data Buffer	Domain	RO	Can't be displayed!	
0x2020	0x00	Encoder Counter	UInt16	RO	0xC355	50005
0x2021	0x00	Encoder Counter at Index Pulse	UInt16	RO	0xC1E2	49634
0x2022	0x00	Hallsensor Pattern	UInt16	RO	0x0003	3
0x2027	0x00	Current Actual Value Averaged	Int16	RO	0xFFE6	-26
0x2028	0x00	Velocity Actual Value Averaged	Int16	RO	0x0000	0
0x2030	0x00	CurrentMode Setting Value	Int16	RW	0x0000	0
0x2062	0x00	PositionMode Setting Value	Int32	RW	0x00000000	0
0x206B	0x00	VelocityMode Setting Value	Int32	RW	0x00000000	0
0x2070	0x01	Configuration of Digital Input1	UInt16	RW	0x000F	15
0x2070	0x02	Configuration of Digital Input2	UInt16	RW	0x000E	14
0x2070	0x03	Configuration of Digital Input3	UInt16	RW	0x000D	13
0x2070	0x04	Configuration of Digital Input4	UInt16	RW	0x0002	2
0x2070	0x05	Configuration of Digital Input5	UInt16	RW	0x0001	1
0x2070	0x06	Configuration of Digital Input6	UInt16	RW	0x0000	0
0x2071	0x01	DigIn Functionalities State	UInt16	RO	0x0000	0
0x2071	0x02	DigIn Functionalities Mask	UInt16	RW	0xE007	57351
0x2071	0x03	DigIn Functionalities Polarity	UInt16	RW	0x0000	0
0x2071	0x04	DigIn Functionalities Execution Mask	UInt16	RW	0x0008	8
0x2074	0x01	Position Marker Captured Position	Int32	RO	0x00000000	0
0x2074	0x02	Position Marker Edge Type	UInt8	RW	0x00	0
0x2074	0x03	Position Marker Mode	UInt8	RW	0x00	0
0x2074	0x04	Position Marker Counter	UInt16	RW	0x0000	0
0x2074	0x05	Position Marker History (1)	Int32	RO	0x00000000	0
0x2074	0x06	Position Marker History (2)	Int32	RO	0x00000000	0
0x2078	0x01	DigOutput State	UInt16	RW	0x0001	1
0x2078	0x02	DigOutput Mask	UInt16	RW	0x0000	0
0x2078	0x03	DigOutput Polarity	UInt16	RW	0x0000	0
0x2079	0x01	Configuration of Digital Output1	UInt16	RW	0x000F	15
0x2079	0x02	Configuration of Digital Output2	UInt16	RW	0x000E	14
0x2079	0x03	Configuration of Digital Output3	UInt16	RW	0x000D	13
0x2079	0x04	Configuration of Digital Output4	UInt16	RW	0x000C	12
0x207C	0x01	Analog Input 1	Int16	RO	0x0018	24
0x207C	0x02	Analog Input 2	Int16	RO	0x0018	24
0x2080	0x00	Current Threshold for Homing Mode	UInt16	RW	0x01F4	500
0x2081	0x00	Home Position	Int32	RW	0x00000000	0
0x20F4	0x00	Following Error Actual Value	Int16	RO	0x0000	0
0x2210	0x01	Encoder Pulse Number	UInt16	RW	0x01F4	500
0x2210	0x02	Position Sensor Type	UInt16	RW	0x0001	1
0x2210	0x04	Position Sensor Polarity	UInt16	RW	0x0000	0
0x2300	0x01	Digital Position Desired Value	Int32	RO	0x00000000	0
0x2300	0x02	Digital Position Scaling Numerator	UInt16	RW	0x0001	1
0x2300	0x03	Digital Position Scaling Denominator	UInt16	RW	0x0001	1
0x2300	0x04	Digital Position Polarity	UInt8	RW	0x00	0

Notes