

CAN 300 / CAN 400 Application Example DS402 Maxon

CAN Communication Module for S7-300/S7-400
Application Example for CANopen DS402 with Maxon Motors

Manual

Edition 2 / 25.04.2007



All rights are reserved, including those of translation, reprinting, and reproduction of this manual, or parts thereof. No part of this manual may be reproduced, processed, copied, or transmitted in any way whatsoever (photocopy, microfilm, or other method) without the express written permission of Systeme Helmholtz GmbH, not even for use as training material, or using electronic systems. All rights reserved in the case of a patent grant or registration of a utility model or design.

Copyright © 2006, 2007 by

Systeme Helmholtz 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.

Step® and SIMATIC® are registered trademarks of SIEMENS AG

Contents

1	Overview	5
1.1	Application and function description	5
1.2	PLC configuration	6
1.3	Maxon motor configuration	7
1.4	Configuration of the PDOs used (mapping)	8
2	Configuring the CAN module	9
2.1	Setting the CAN bus baudrate	9
2.2	Setting the transmission mode (protocol)	9
2.3	Acceptance masks	10
2.4	Network management	10
2.5	Timer	11
3	Programming in the PLC	12
3.1	Overview	12
3.2	Init Drive (FB 37)	12
3.3	Homing (FB 30, FB 31)	13
3.4	Profile position (FB 32, FB 33)	14
3.5	Profile velocity (FB 34, FB 35)	15
3.6	Fault reset (FB 36)	16
3.7	Example FC 1 (Maxon_Test)	16
3.8	Use with several drives	17
3.9	Content of the status word	18



1 Overview

1.1 Application and function description

This manual describes the application example (handling blocks) for controlling a Maxon motor with a S7-300 using a CAN 300 module according to the CANopen profile DS402.

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.

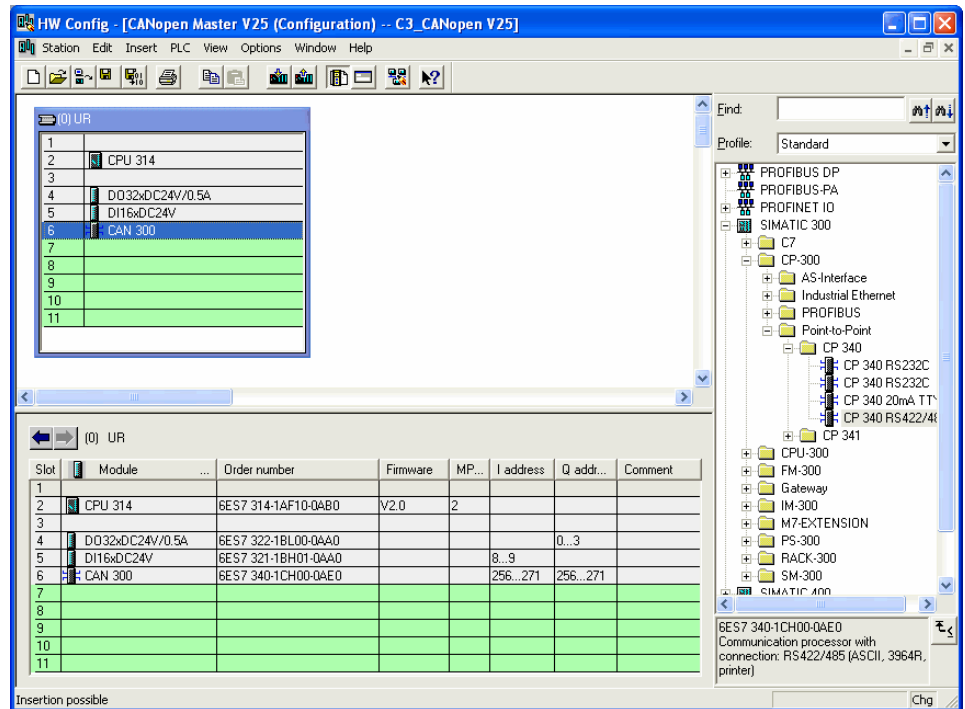
The handling blocks described here (FBs) provide the following basic functions:

- Basic initialization of a drive
- Profile velocity mode
- Profile position mode
- Homing mode
- Fault reset

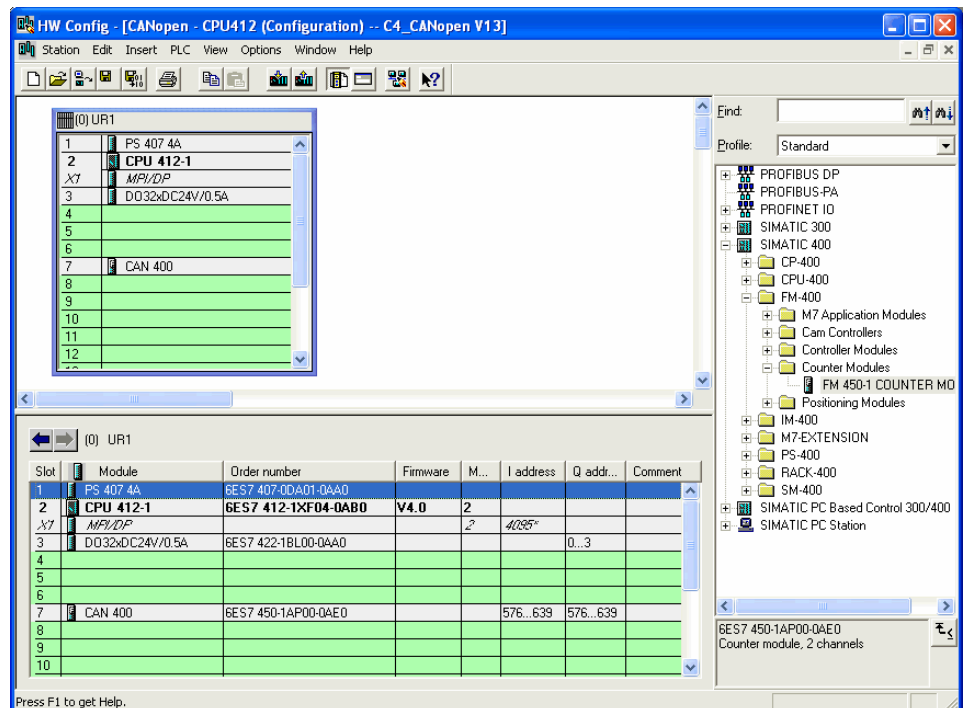
The handling blocks are functionally based on the Maxon instruction “EPOS Application Note Device Programming”.

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

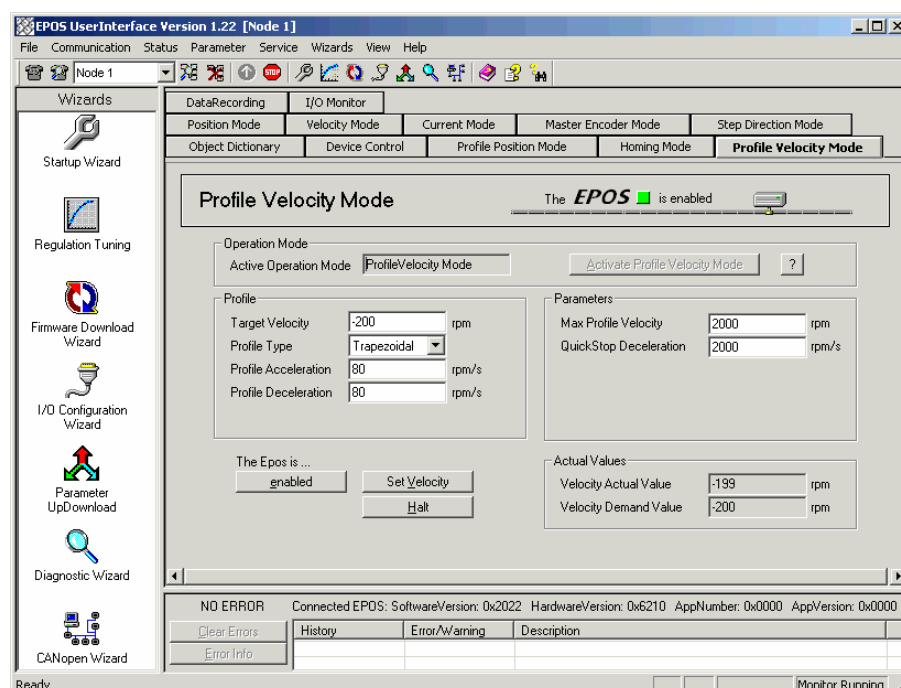
The handling example was created with an EPOS 24/5. The motor is connected as Node 1.

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.

We recommend testing the basic motor functions with the EPOS software:



1.4 Configuration of the PDOs used (mapping)

The handling blocks are based on the standard PDO mapping of the Maxon motor.

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 0-1: SDO 6041/0 Status Word

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

TxPDO3 (is transmitted by the EPOS): ID 381_{hex}

Byte 0+1: SDO 6041/0 Status Word

Byte 2-5: SDO 6064/0 Position actual value

TxPDO4 (is transmitted by the EPOS): ID 481_{hex}

Byte 0+1: SDO 6041/0 Status Word

Byte 2-5: SDO 606C/0 Velocity actual value

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

Byte 0+1: SDO 6040/0 Control Word

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

RxPDO3 (is received by the EPOS): ID 401_{hex}

Byte 0+1: SDO 6040/0 Control Word

Byte 2-5: SDO 607A/0 Target position

RxPDO4 (is received by the EPOS): ID 501_{hex}

Byte 0+1: SDO 6040/0 Control Word

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

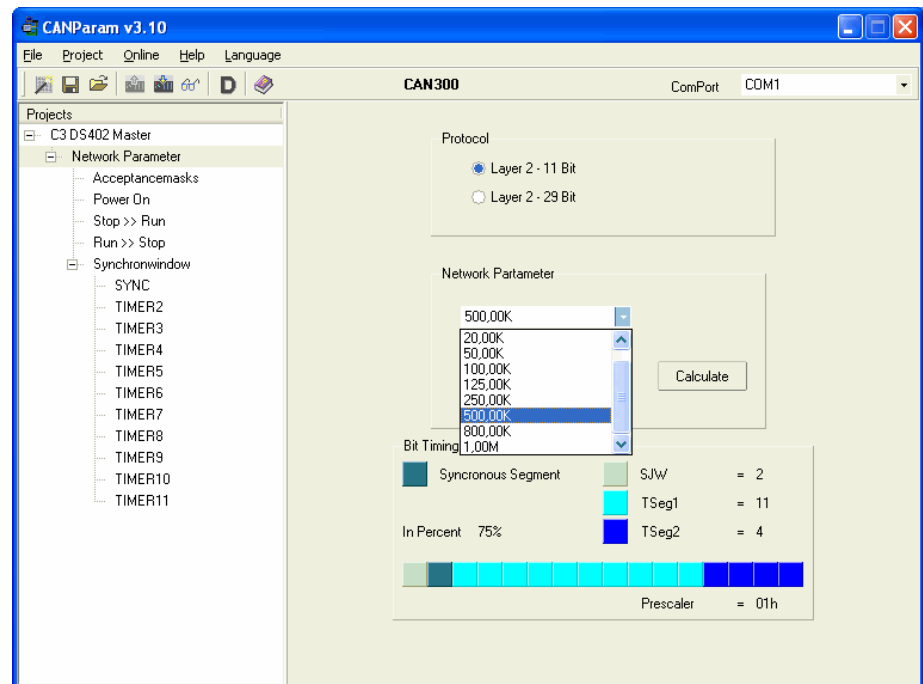
The TxPDOs is only transmitted from the EPOS if the EPOS is in CANopen mode “operational”.

2 Configuring the CAN module

To be able to use the example program, the attached CAN project “C3 DS402 Master.par” must be imported into the CAN 300 module. For the CAN 400 module, the “C4 DS402 Master.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.



The default setting of the Maxon motor is 1 Mbit.

2.2 Setting the transmission mode (protocol)

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

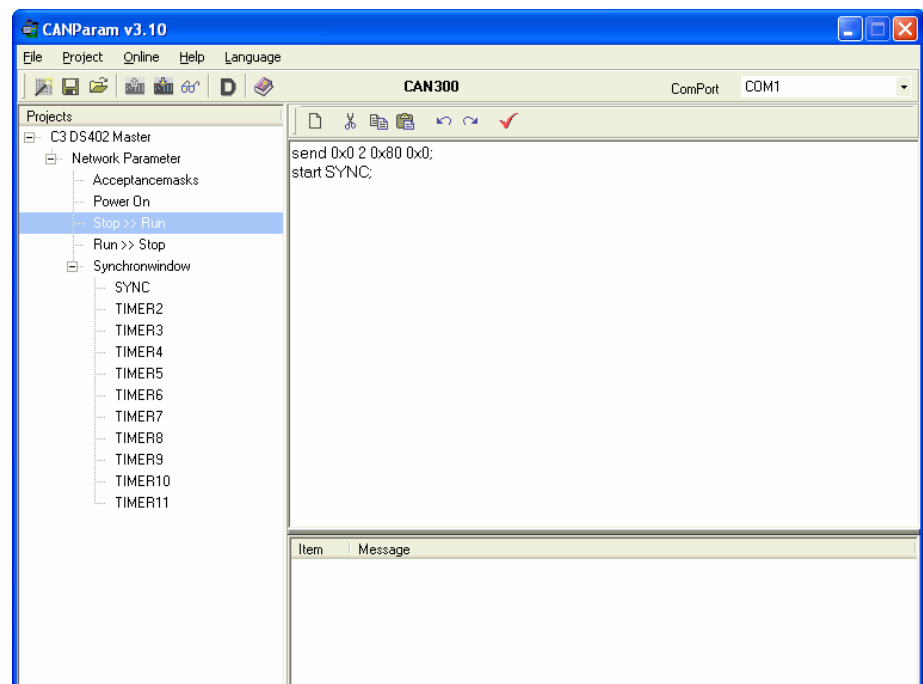
2.3 Acceptance masks

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

	Begin	End
<input checked="" type="checkbox"/> Mask 1	0x000	0x7FF
<input type="checkbox"/> Mask 2	0x000	0x000
<input type="checkbox"/> Mask 3	0x000	0x000
<input type="checkbox"/> Mask 4	0x000	0x000
<input type="checkbox"/> Express Mask	0x000	0x000
<hr/>		
<input type="checkbox"/> Mask 6	0x000	0x000
<input type="checkbox"/> Mask 7	0x000	0x000
<input type="checkbox"/> Mask 8	0x000	0x000
<input type="checkbox"/> Mask 9	0x000	0x000
<input type="checkbox"/> Mask 10	0x000	0x000
<input type="checkbox"/> Mask 11	0x000	0x000
<input type="checkbox"/> Mask 12	0x000	0x000
<input type="checkbox"/> Mask 13	0x000	0x000
<input type="checkbox"/> Mask 14	0x000	0x000
<input type="checkbox"/> Mask 15	0x000	0x000
<input type="checkbox"/> Mask 16	0x000	0x000

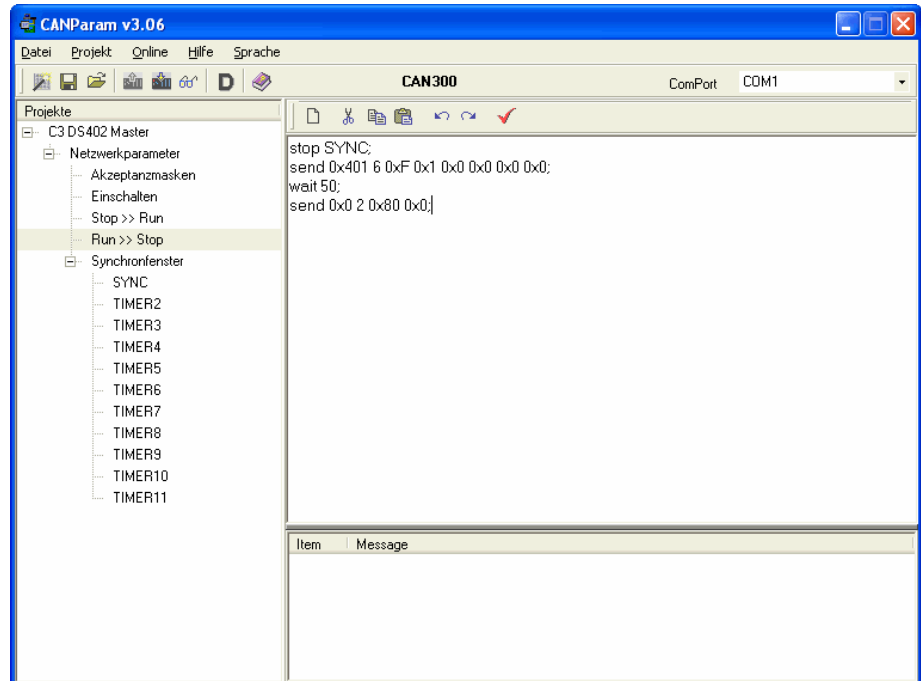
2.4 Network management

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.



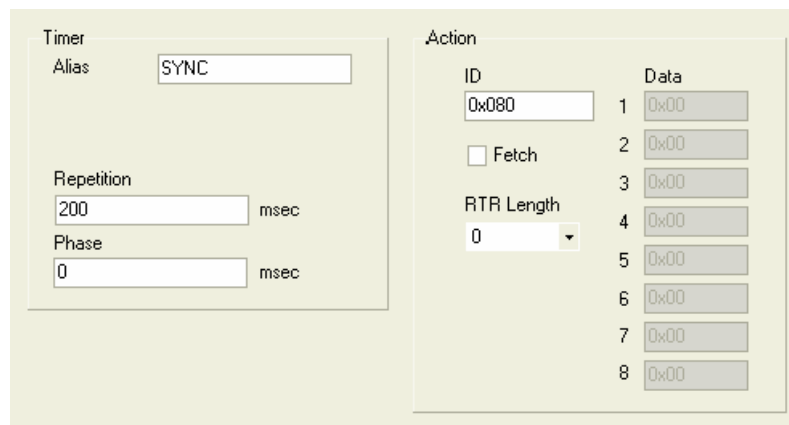
The CAN bus is started by the handling blocks.

If the PLC is stopped, the “On->Off” script is executed. This script stops the SYNC timer, transmits a telegram to the motor, with which current movements are stopped, and then stops the CANopen bus (NMT Stop all Nodes).



2.5 Timer

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



3 Programming in the PLC

3.1 Overview

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

FB30 and FB37 are the blocks specially created for handling the Maxon motor and they call the blocks of the CANopen Master handling.

3.2 Init Drive (FB 37)

The **INIT_DRIVE (FB 37)** function block initializes all important communication parameters of the drive.

```
CALL FB 37 , DB37
  Activate:=M109.7
  Node    :=1
  Busy    :=M111.5
  Done    :=M111.6
  Error   :=M111.7
  ErrorNo :=MW112
```

Parameter	Type		Function
Activate	IN	BOOL	Activation bit
Node	IN	INT	Node ID
Busy	OUT	BOOL	Flag bit for current function
Done	OUT	BOOL	Flag bit for initialization complete
Error	OUT	BOOL	Flag bit for error occurred
ErrorNo	OUT	WORD	Error number of CANopen handling blocks

The function block describes the following SDOs:

SDO 1801/1	TPDO2 COB-ID enable
SDO 1802/1	TPDO3 COB-ID enable
SDO 1803/1	TPDO4 COB-ID enable
SDO 1802/2 = 1	TPDO3 Transmission Type = Sync
SDO 1803/2 = 1	TPDO4 Transmission Type = Sync

At the end of the process, the drive is put in operational mode with “NMT start”.

3.3 Homing (FB 30, FB 31)

The **INIT_HOMING (FB 30)** function block activates the homing mode of the drive and transfers the necessary parameters.

```
CALL FB 30 , DB30
  Activate      :=M109.0
  Node          :=1
  HomeOffset    :=L#0
  HomingAcceleration:=L#2000
  SpeedSwitchSearch :=L#1000
  SpeedZeroSearch  :=L#1000
  CurrentThreshold :=L#100
  QuickStopDecel   :=L#4000
  Busy          :=M111.0
  Done           :=M111.1
  Error          :=M111.7
  ErrorNo        :=MW112
  Statusword     :=MW30
```

Parameter	Type		Function
Activate	IN	BOOL	Activation bit
Node	IN	INT ;	Node ID
HomeOffset	IN	DWORD	→ SDO 607C/0
HomingAcceleration	IN	DWORD	→ SDO 609A/0
SpeedSwitchSearch	IN	DWORD	→ SDO 6099/1
SpeedZeroSearch	IN	DWORD	→ SDO 6099/2
CurrentThreshold	IN	DWORD	→ SDO 2080/0
QuickStopDecel	IN	DWORD	→ SDO 6085/0
Busy	OUT	BOOL	Flag bit for current function
Done	OUT	BOOL	Flag bit for initialization complete
Error	OUT	BOOL	Flag bit for error occurred
ErrorNo	OUT	WORD	Error number of CANopen handling blocks
Statusword	OUT	WORD	Status word of the drive (from PDO1)

The **HOMING (FB 31)** function block performs a homing run.

```
CALL FB 31 , DB31
  Node          :=1
  HomePosition:=L#0
  HomingMethod:=MW28
  Activate      :=M109.1
  Halt          :=M109.2
  Statusword    :=MW30
  Position      :=MD32
  Velocity      :=MD36
  Done          :=M111.2
  Error         :=M110.7
```

Parameter	Type		Function
Node	IN	INT ;	Node ID
HomePosition	IN	DWORD	Value for home position
HominMethod	IN	INT ;	Homing mthod (→ SDO 6098/0)
Activate	IN	BOOL	Activation bit for homing start
Halt	IN	BOOL	Activation bit for homing stop
Statusword	OUT	WORD	Status word of the drive (from PDO1)
Position	OUT	WORD	Current position of the drive (from PDO3)
Velocity	OUT	WORD	Current speed of the drive (from PDO4)
Done	OUT	BOOL	Flag bit "homing reached"
Error	OUT	BOOL	Flag bit for error occurred

3.4 Profile position (FB 32, FB 33)

The **INIT_POSITION (FB 30)** function block activates the profile position mode of the drive and transfers the necessary parameters.

```
CALL FB 32 , DB32
  Activate      :=M108.4
  Node          :=1
  ProfileVelocity :=L#3000
  ProfileAcceleration:=L#1000
  ProfileDeceleration:=L#1000
  QuickStopDecel :=L#4000
  Busy          :=M110.4
  Done          :=M110.5
  Error         :=M111.7
  ErrorNo       :=MW112
  Statusword    :=MW30
```

Parameter	Type		Function
Activate	IN	BOOL	Activation bit
Node	IN	INT ;	Node ID
ProfileVelocity	IN	DWORD	→ SDO 6081/0
ProfileAcceleration	IN	DWORD	→ SDO 6083/0
ProfileDeceleration	IN	DWORD	→ SDO 6084/0
QuickStopDecel	IN	DWORD	→ SDO 6085/0
Busy	OUT	BOOL	Flag bit for current function
Done	OUT	BOOL	Flag bit for initialization complete
Error	OUT	BOOL	Flag bit for error occurred
ErrorNo	OUT	WORD	Error number of CANopen handling blocks
Statusword	OUT	WORD	Status word of the drive (from PDO1)

The **POSITION (FB 31)** function block performs positioning.

```
CALL FB 33 , DB33
  Node          :=1
  TargetPosition :=MD24
  Activate      :=M108.5
  RelativPositioning :=M108.7
  ImmediatePositioning:=FALSE
  Halt          :=M108.6
  Statusword    :=MW30
  Position      :=MD32
  Velocity      :=MD36
  Done          :=M110.6
  Error         :=M111.7
```

Parameter	Type		Function
Node	IN	INT ;	Node ID
TargetPosition	IN	DWORD	Target position
Activate	IN	BOOL	Activation bit for homing start
RelativPositioning	IN	BOOL	Control bit relative/absolute positioning
ImmediatePositioning	IN	BOOL	Control bit immediate positioning
Halt	IN	BOOL	Activation bit for positioning stop
Statusword	OUT	WORD	Status word of the drive (from PDO1)
Position	OUT	WORD	Current position of the drive (from PDO3)
Velocity	OUT	WORD	Current speed of the drive (from PDO4)
Done	OUT	BOOL	Flag bit "position reached"
Error	OUT	BOOL	Flag bit for error occurred

3.5 Profile velocity (FB 34, FB 35)

The **INIT_VELOCITY (FB 34)** function block activates the profile velocity mode of the drive and transfers the necessary parameters.

```
CALL FB 34 , DB34
Activate      :=M108.0
Node         :=1
MaxProfileVelocity :=L#10000
ProfileAcceleration:=L#1000
ProfileDeceleration:=L#1000
QuickStopDecel  :=L#4000
Busy         :=M110.0
Done         :=M110.1
Error        :=M111.7
ErrorNo      :=MW112
Statusword   :=MW30
```

Parameter	Type		Function
Activate	IN	BOOL	Activation bit
Node	IN	INT ;	Node ID
MaxProfileVelocity	IN	DWORD	→ SDO 607F/0
ProfileAcceleration	IN	DWORD	→ SDO 6083/0
ProfileDeceleration	IN	DWORD	→ SDO 6084/0
QuickStopDecel	IN	DWORD	→ SDO 6085/0
Busy	OUT	BOOL	Flag bit for current function
Done	OUT	BOOL	Flag bit for initialization complete
Error	OUT	BOOL	Flag bit for error occurred
ErrorNo	OUT	WORD	Error number of CANopen handling blocks
Statusword	OUT	WORD	Status word of the drive (from PDO1)

The **VELOCITY (FB 35)** function block activates a speed.

```
CALL FB 35 , DB35
Node         :=1
TargetVelocity:=MD20
Activate     :=M108.1
Halt        :=M108.2
Statusword  :=MW30
Position    :=MD32
Velocity    :=MD36
Done        :=M110.2
Error       :=M111.7
```

Parameter	Type		Function
Node	IN	INT ;	Node ID
TargetVelocity	IN	DWORD	Set speed
Activate	IN	BOOL	Activation bit for homing start
Halt	IN	BOOL	Activation bit for positioning stop
Statusword	OUT	WORD	Status word of the drive (from PDO1)
Position	OUT	WORD	Current position of the drive (from PDO3)
Velocity	OUT	WORD	Current speed of the drive (from PDO4)
Done	OUT	BOOL	Flag bit "velocity reached"
Error	OUT	BOOL	Flag bit for error occurred

3.6 Fault reset (FB 36)

The **FAULT_RESET (FB 35)** function block reset errors in the drive. Whether there is a “fault” in the drive can be seen from the status word (see also Ch. 3.9).

CALL	FB	36	, DB37
Node	:	=1	
Activate	:	=M109.6	
Statusword	:	=MW30	
Done	:	=M111.4	
Error	:	=M111.7	

Parameter	Type	Function
Node	IN INT ;	Node ID
Activate	IN BOOL	Activation bit for homing start
Halt	IN BOOL	Activation bit for positioning stop
Statusword	OUT WORD	Status word of the drive (from PDO1)
Done	OUT BOOL	Flag bit, fault state no longer present
Error	OUT BOOL	Flag bit for error occurred

3.7 Example FC 1 (Maxon_Test)

The functions of the handling blocks are triggered in FC1 by the bits of input bytes 8 and 9. The input bytes are copied onto the MB108 (in the OB1). The example can therefore also be used by controlling MB108 without an input or test module.

At the start of FC 1, the cycle block FC 49 is called to fetch telegrams received via the CAN bus or to execute transmit jobs.

The drive functions are activated with the following inputs and flags:

I 8.0 (M108.0):	Init Profile Velocity Mode (FB34)
I 8.1 (M108.1):	Set Velocity (FB35)
I 8.2 (M108.2):	Halt Velocity (FB35)
MD20	Target Velocity (FB35)
I 8.4 (M108.4):	Init Profile Position Mode (FB32)
I 8.5 (M108.5):	Activate Positioning (FB33)
I 8.6 (M108.6):	Halt Positioning (FB33)
MD24	Target Position (FB33)
I 8.7 (M108.7):	Relative/Absolute Positioning (FB33)
I 9.0 (M109.0):	Init Homing Mode (FB30)
I 9.1 (M109.1):	Activate Homing (FB31)
I 9.2 (M109.2):	Halt Homing (FB31)
MB28	Homing Method (FB31)
I 9.6 (M109.6):	Fault Reset (FB36)
I 9.7 (M109.7):	Init Drive (FB37)

3.8 Use with several drives

The function blocks can be called several times in succession for different drives. Only the parameter node has to be adapted.



Do not activate „INIT_“ blocks simultaneously.

Please note that simultaneous activation of the “INIT_...” block should be avoided because transmission of the SDOs executed in the blocks cannot be performed simultaneously. These blocks must only be called successively for each drive (node).

3.9 Content of the status word

The status word of the EPOS 24/5 is stored in MW30 in the application example.

Bit	Description	PPM	PVM	HMM
15	Position referenced to home position			
14	Refresh cycle of power stage			
13	<i>Operation mode specific</i>	Following error	Not used	Homing error
12	<i>Operation mode specific</i>	Set-point ack	Speed	Homing attained
11	not used (Internal limit active)			
10	<i>Operation mode specific</i>	Target reached	Target reached	Target reached
9	Remote (NMT operational state)			
8	Offset current measured			
7	Warning			
6	Switch on disable			
5	Quick stop			
4	Voltage enabled (power stage on)			
3	Fault			
2	Operation enable			
1	Switched on			
0	Ready to switch on			

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
Measure Init	x1xx xxx1 x011 0011	The power is applied to motor The motor resistance or the commutation delay is measured
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

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

Notes