Embedded system solutions
Emsiso

Emsiso d.o.o.
Zagrebška cesta 20
SI-2000 Maribor
Slovenia

tel. +386 2 46 129 07
fax. +386 2 46 129 08
www.emsiso.com

Member of
Seidel Electronics
www.seidel.at

SEIDEL GROUP

emDrive
Firmware Specifications

Project name	emDrive
Version	1.0
Author	Gregor Kosic
Date	22.07.2014

History

Version	Author	Description	Date
1.0	Gregor Kosic	Initial document creation	22.07.2014

Table of contents

TABLE OF FIGURES

1 INTRODUCTION

1.1 General Information

This documentation “Firmware description” provides the firmware details of the emDrive150 controller. It contains detailed description of architecture, device states, operation modes, error handling and object dictionary.

EmDrive family are advanced BLDC motor controllers for e-drive systems. Drives are designed to power brushless and induction motos, it supports standard three hall sensor with 60 and 120 degree configuration, absolute digital SSI encoder with RS422 interface, resolver, 5V analogue sin/cos, and encoder. Due the vector current control, torque ripple and vibrations in motor, there is no audible noise. Integrated current, velocity and position current control functionality allows accurate sophisticated positioning applications.

emDrives are specially designed to be controlled over CANopen network as a slave node and additionally can be operated through any RS232 communication port and local control.

1.2 Operating Principles

The CiD DSP 402 CANopen *Device Profile for Drives and Motion Control* is used to provide drives in a CAN network with an understandable and consistent behavior. The profile is built on top of a CAN communication profile, called CANopen, which describes the basic communication mechanisms common to all devices in the CAN network.

The purpose of the drive units is to connect axle controllers or other motion control products to the CAN bus. They usually receive configuration information via service data objects for I/O configuration, limit parameter for scaling, or application-specific parameters. At run time, data can be obtained from the drive unit via CAN bus either by polling or in event-driven mode (with properly-mapped TPDOs).

The motion control products use process-data object mapping for real-time operation, which may be configured using service data objects (SDOs). This communication channel is used to interchange real-time data-like set-points or actual values such as position actual values.

The most important part of a device profile is the object dictionary description. The object dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. The DSP 402 standard objects of single-axis drives, like the
Harmonica, are all in the index range of 0x6000 to 0x67ff.

2 SYSTEM OVERVIEW

3.1 Device Architecture

CAN Communication interface of the emDrive BLDC controller follows the CiA CANopen specifications:

Support of all CiA 301 services
Compliant to CiA-301 V4.2
LED CiA-303

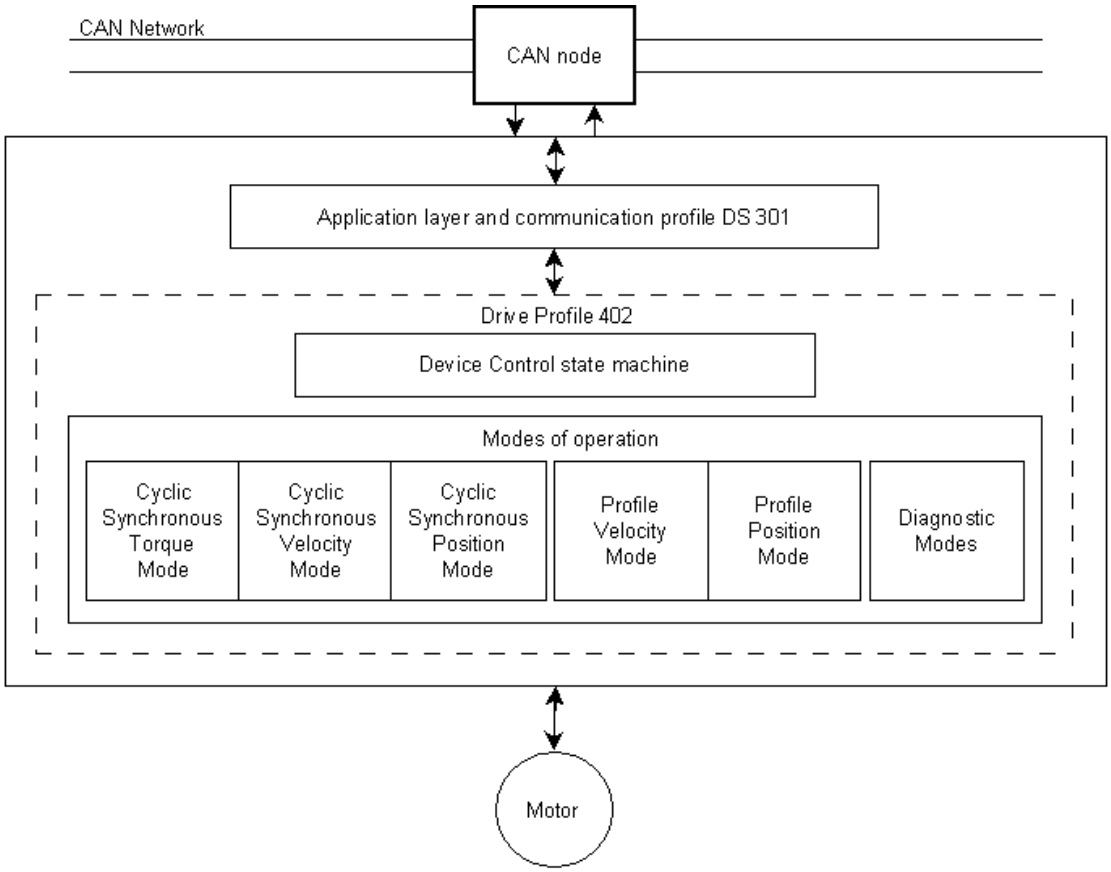


Figure 2-1: Communication architecture

Device Control
Drive managing and mode-specific command are executed by the state machine.

Modes of operation
Mode of operation is defining the behavior of the drive.

3.2 Device Control

Figure 2-2 contains detailed description of the device states and all possible controlling sequences for operating the emDrive.

Also states of the drives are determining which commands are accepted.

States may be changed using the *Controlword* and/or according to internal events. The current state can be read using the *Statusword*.

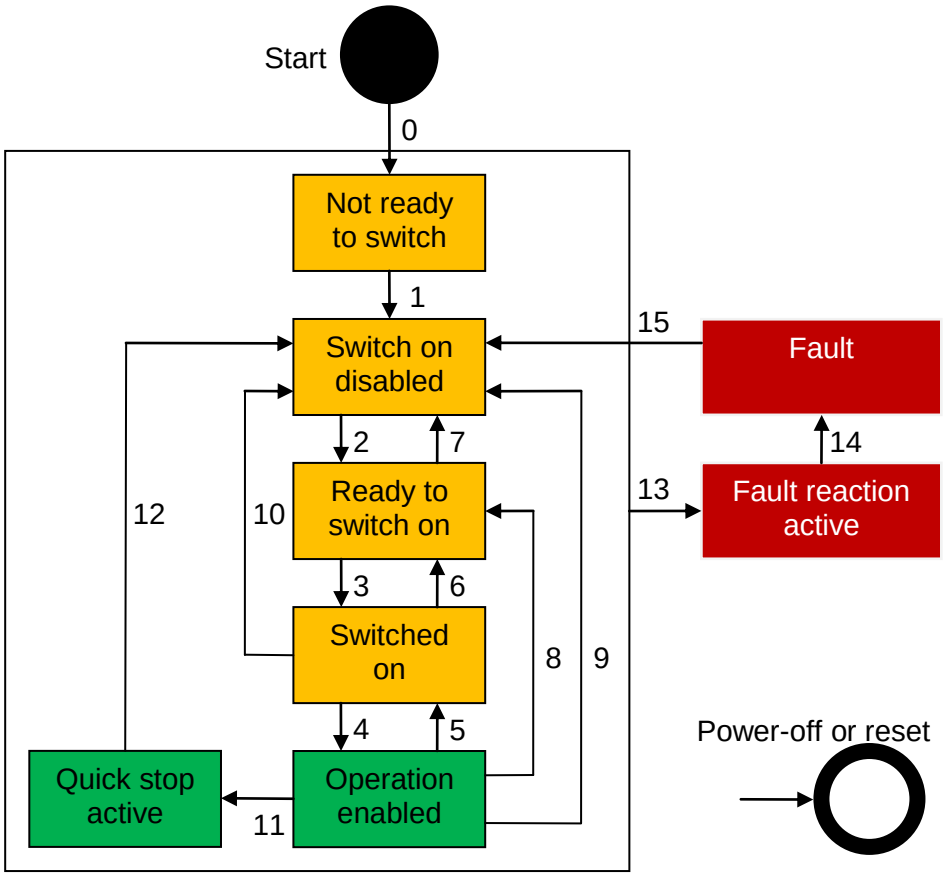


Figure 2-2: Device State Machine Block Diagram

3.2.1 State Transitions

State transitions are caused by internal events in the device or by command from host via the Controlword.

If command is received which causes a change of state, this command will be processed completely and the new state attained before the next command can be processed.

Table 2-1: Transition events and actions

Transition	Event(s)	Action(s)
0	Automatic transition after power-on or reset	Initialize drive, enable breaks, start local control if enabled
1	Automatic transition	Start safety procedure, disable drive function, initialize feedback, internal set-points cleared
2	“Shutdown” command from control device or local signal received	Disable drive function
3	“Switch on” command received from control device or local signal	Disable breaks
4	“Enable operation” command received from control device or local signal	Enable drive function
5	“Disable operation” command received from control device or local signal	Drive function disabled, disable breaks
6	“Shutdown” command received from control device or local signal	Drive functions disabled, enable breaks
7	“Quick stop” or “disable voltage” command received from control device or local signal	Drive functions disabled, enable breaks, initialize feedback, internal set-points cleared
8	“Shutdown” command received from control device or local signal	Drive functions disabled, enable breaks
9	“Disable voltage” command received from control device or local signal	Drive functions disabled, enable breaks, initialize feedback, internal set-points cleared
10	“Disable voltage” or “quick stop” command received from control device or local signal	Drive functions disabled, enable breaks, initialize feedback, internal set-points cleared
11	“Quick stop” command received from control device or local signal	none
12	Automatic transition	Disable drive function, initialize feedback, internal set-points cleared
13	Fault signal	None
14	Automatic transition	Disable drive function, enable break
15	“Fault reset” command received from control device or local signal	Clear fault condition if no fault is present

3.2.2 Device Control Commands

Device control commands are executed by writing in the *Controlword*, codes given in Table 2-2

Table 2-2: Command coding

Command	Bit of the *Controlword*					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3
Switch on + enable operation	0	1	1	1	1	3+4 (NOTE 1)
Disable voltage	0	X	X	0	X	7, 9, 10, 12
Quick stop	0	X	0	1	X	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4
Fault reset	From 0 to 1 (NOTE 2)	X	X	X	X	15
NOTE 1: Automatic transition to enable operation state after executing SWITCHED ON state functionality. NOTE 2: Bit 7 transition from 0 to 1						

4 ERROR CODES AND ERROR BEHAVIOR

4.1 Emergency Message Frame

When device-internal failure is detected emDrive will transmit emergency message frames over the CANopen with highest priority. Message frame shown in Table 4-1.
An emergency message frame will be transmitted only once per error event and consists of the *Error Code* and the actual *Error Register*.

Table 4-1: Emergency message frame

Byte	0	1	2	3	4	5	6	7
Content	Emergency error code		Error register	Not used (always “0”)				

emDrive can detect several variety of device error. The reaction to an error depends on error type. After execution of the fault reaction, the device changes to fault state and the drive will be disabled.

4.2 Device Emergency Error Codes

4.2.1 0x1000 Generic error

Error Code	0x1000
Cause	Unspecific error occurred
Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.2 0x2220 Overcurrent error

Error Code	0x2220
Cause	short circuit in motor winding controller gain to high (Current control parameters, Velocity control parameters) power stage damaged
Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.3 0x3210 DC link over voltage

Error Code	0x3210
Cause	Power supply voltage to high
Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.4 0xFF01 Phase A current measurement

Error Code	0xFF01
Cause	Current phase A hall sensor missing or damaged
Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.5 0xFF02 Phase B current measurement

Error Code	0xFF02
Cause	Current phase A hall sensor missing or damaged
Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.6 0xFF03 High side Fet short circuit

Error Code	0xFF03
Cause	DC voltage not applied to bridge or to low motor phases not connected to controller internal low side fet error circuit damaged damaged high side fets
Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.7 0xFF04 Low side Fet short circuit

Error Code	0xFF04
Cause	DC voltage not applied to bridge or to low motor phases not connected to controller damaged low side fets
Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.8 0xFF05 Low side Fet phase 1 short circuit

Error Code	0xFF05
Cause	motor phases not connected damaged low side fets on phase 1
Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.9 0xFF06 Low side Fet phase 2 short circuit

Error Code	0xFF06
Cause	motor phases not connected damaged low side fets on phase 2

Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.10 0xFF07 Low side Fet phase 3 short circuit

Error Code	0xFF07
Cause	motor phases not connected damaged low side fets on phase 3
Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.11 0xFF08 High side Fet phase 1 short circuit

Error Code	0xFF08
Cause	motor phases not connected damaged low side fets on phase 1
Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.12 0xFF09 High side Fet phase 2 short circuit

Error Code	0xFF09
Cause	motor phases not connected damaged low side fets on phase 2
Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.13 0xFF0A High side Fet phase 3 short circuit

Error Code	0xFF0A
Cause	motor phases not connected damaged low side fets on phase 3
Effect	device disabled red error led “on” error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.14 0xFF0B Motor Feedback

Error Code	0xFF0C
Cause	Wrong feedback selected (check feedback type) Feedback damaged or not connected
Effect	- device disabled - red error led “on” - error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.15 0xFF0C DC link undervoltage

Error Code	0xFF0C

Cause	DC voltage not applied to bridge or to low
Effect	- device disabled - red error led “on” - error flag set in *Statusword* (bit 3)
Error Recovery	Reset fault with *Controlword*

4.2.16 **0xFF0D Puls mode finished**

Error Code	0xFF0D
Cause	Puls mode finished It’s used for fine adjustments and troubleshooting
Effect	- device disabled - red error led “on”
Error Recovery	Reset fault with *Controlword*

4.2.17 **0xFF0E Emergency button pressed**

Error Code	0xFF0E
Cause	Emergency button pressed. If not used pin must be connected to GND
Effect	- device disabled - red error led “on”
Error Recovery	Reset fault with *Controlword*

5 MODES OF OPERATION

0x6060: Modes of operation
0x6061: Modes of operation display

5.1 Functional Description

The drive behavior depends on the activated modes of operation. Different modes can be implemented, although not in parallel. Therefore, the user must activate the required function by selecting modes of operation, through the object *Modes of Operation* and it's by default set to *"no mode"* (value – 0). Modes can be set in any state excluding *OPERATION ENABLED* in this state switching between modes of operation is not recommended. Actual modes of operation is displayed in *Modes of Operation Display.*

5.1.1 0x6060: Modes of operation

Object description:

Index	0x6060
Name	Modes of Operation
Data type	INTEGER8
Saved in EEPROM	Yes

Entry description:

Access	Read / write
PDO mapping	Allowed
Value range	INTEGER8
Default value	0

Data description:

Value	Description
-128…-6	Reserved
-5	Induction VF – testing purposes only
-4	Aligning rotor position
-3	Current stepper – testing purposes only
-2	Voltage stepper – testing purposes only
-1	Voltage constant angle – testing purposes only
0	No mode
1…7	Reserved
8	Cyclic sync position
9	Cyclic sync velocity
10	Cyclic sync torque
11...127	Reserved

- A read of this object shows only the new value of modes of operation. The actual mode of the drive is reflected in the *modes of operation display* (0x6061). It may be changed by writing to modes of operation.
- An attempt to access an unsupported mode causes the *modes operation* set to *no mode*.

5.1.2 0x6061 Modes of operation display

This object shows the current mode of operation. The meaning of the returned value corresponds to *modes of operation* option code (0x6060).

Object description:

Index	0x6061
Name	Modes of Operation display
Data type	INTEGER8
Saved in EEPROM	No

Entry description:

Access	Read only
PDO mapping	Not allowed
Value range	INTEGER8
Default value	0

Data description:
Similar to object 0x6060, *modes of operation.*

5.1.2.1 Induction VF

Induction VF mode uses *Test Mode Command* (0x2031) to command the Phase Voltage calculation function. For enabling *Induction VF* mode *Test Stepper Enable* (0x2040,10) must be set.
In this mode NO protections and limits are active. **Proposed use only for developers.**

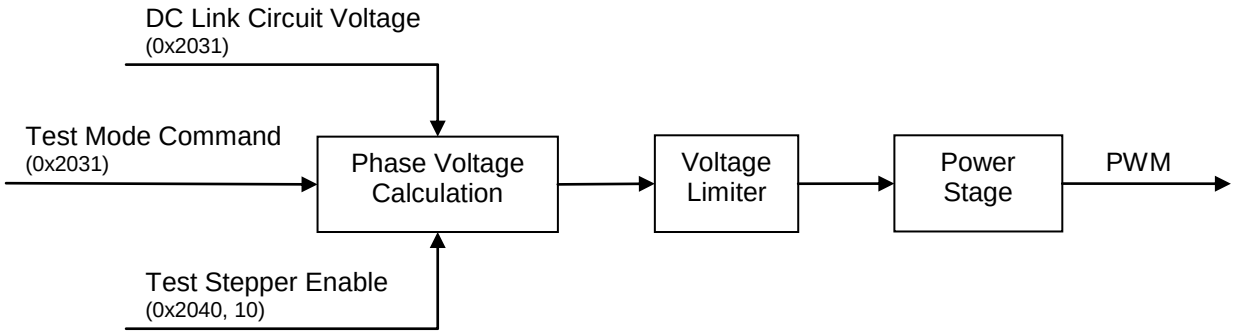


Figure 0-1 Induction VF – Block Diagram

5.1.2.2 Aligning Rotor Position

Aligning Rotor Position is used for determination of rotor position offset. Motor must be able to free spin.
Minimal requested settings for proper determination must be set:

- *Motor Type* (0x2034),
- *Motor Rated Current* (0x6075),
- *Motor Pole Pairs* (0x2033),
- *Feedback type* (0x2040, 1),
- *Feedback Resolution* (0x2040, 7),
- *Auto aligning rotor position current* (0x2040, 11).

Internal state machine is started as soon as proper *Modes of Operation* (0x6060) is selected and NMT state is changed to *operational*.
Motor will spin with 20 RPM in both directions for few seconds and calculate rotor position offset. If the procedure will be successful, *Motor Phase Offset* (0x2040, 3) and *Feedback Diretion* (0x2040, 4) will be set, drive will go into *preoperation mode*. *Modes of Operation* (0x6060) will be set to 0 (*No Mode)*.
In case that procedure will fail, drive will go into *Fail State*. In *Error Register* (0x1001) error code 0xFF0B (Error Motor Feedback) will be displayed.

5.1.2.3 Current Stepper

Current stepper mode uses *Test Mode Command* (0x2031) to command the current stepper control function. For enabling stepper, *Test Stepper Enable* (0x2040, 10) and *Test stepper Frequency* (0x2040, 5) must be set.
In this mode only maximum controller current limit is active, other protections and limits are inactive. **Proposed use only for developers.**

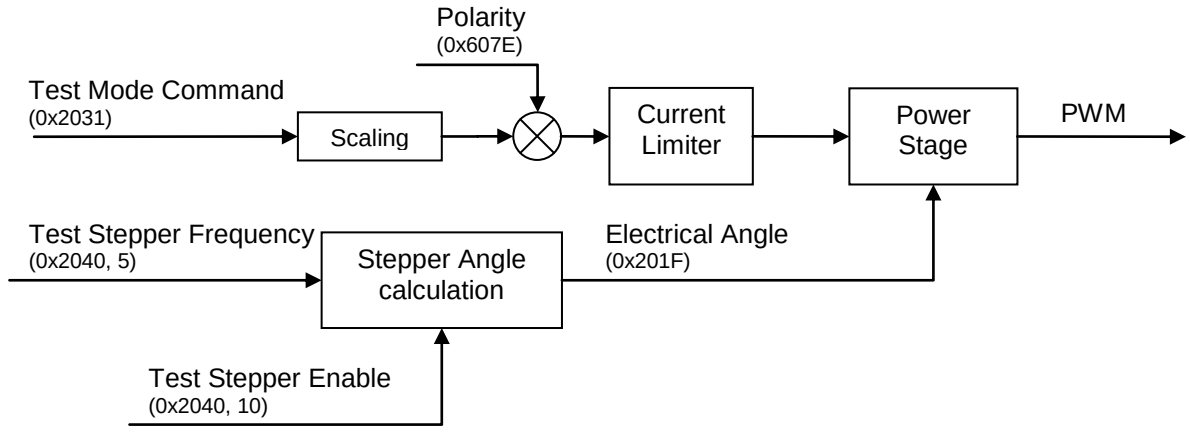


Figure 0-2 Current Stepper – Block Diagram

5.1.2.4 Voltage Stepper

Voltage stepper mode uses *Test Mode Command* (0x2031) to command the voltage stepper control function. For enabling stepper, *Test Stepper Enable* (0x2040, 10) and *Test stepper Frequency* (0x2040, 5) must be set.
In this mode NO protections and limits are active. **Proposed use only for developers.**

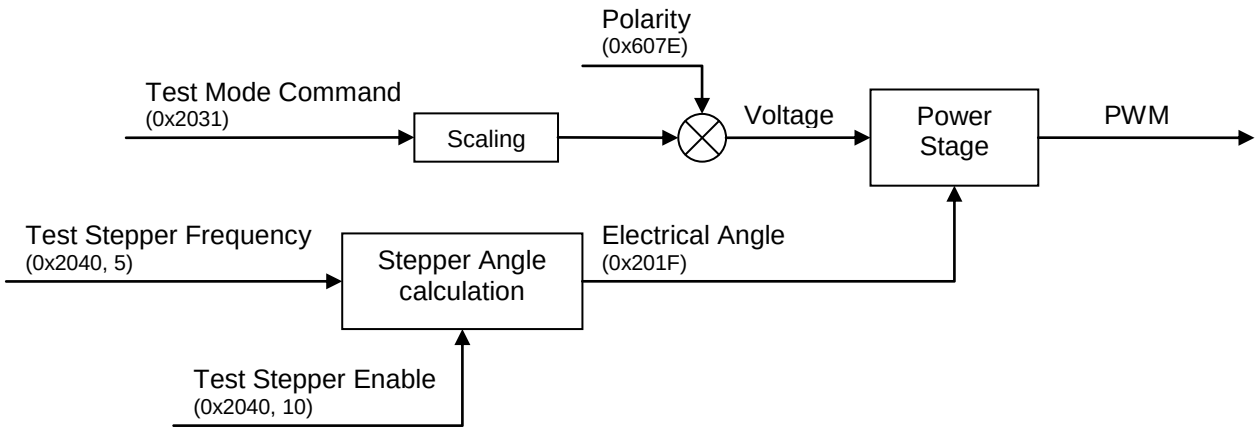


Figure 0-3 Voltage Stepper – Block Diagram

5.1.2.5 Cyclic Sync Torque

Cyclic Sync Torque mode uses *Target Torque* (0x6071) to command the current control function.
Find description for block "*Limiter"* in paragraph 5.1.3 Output current Limitation.
Optionally local control can be enabled and an analog input can be used to command the current control function.

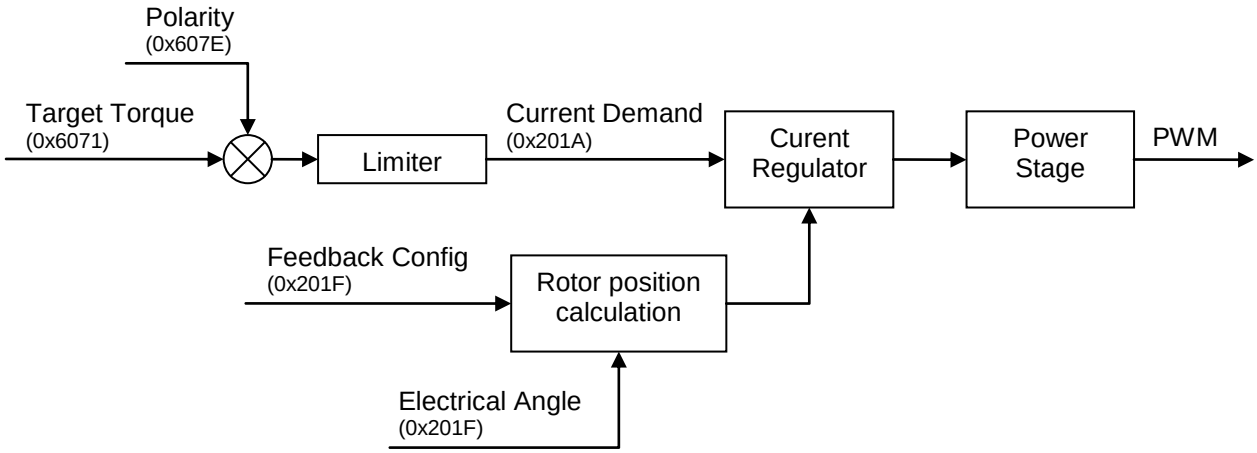


Figure 0-4 Cyclic Sync Torque – Block Diagram

5.1.2.6 Cyclic Sync Velocity

Cyclic Sync Velocity mode uses *Target Velocity* (0x60FF) to command the velocity control function.
Find description for block "*Limiter"* in paragraph 5.1.3 Output current Limitation.
Optionally local control can be enabled and an analog input can be used to command the velocity control function.

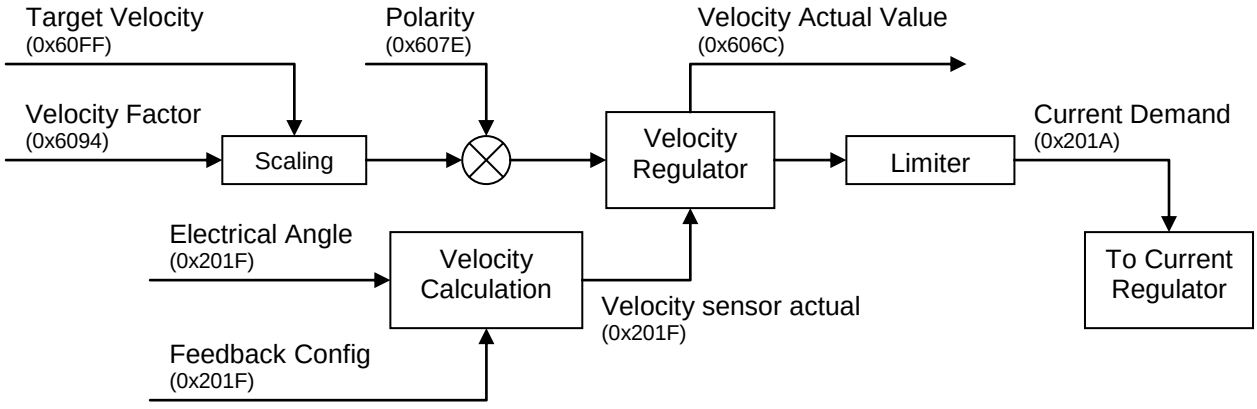


Figure 0-5 Cyclic Sync Velocity – Block Diagram

5.1.2.7 Cyclic Sync Position

Cyclic Sync Position mode uses *Target Position* (0x607A) to command the position control function.
Find description for block "*Limiter"* in paragraph 5.1.3 Output current Limitation.
Optionally local control can be enabled and an analog input can be used to command the position control function.

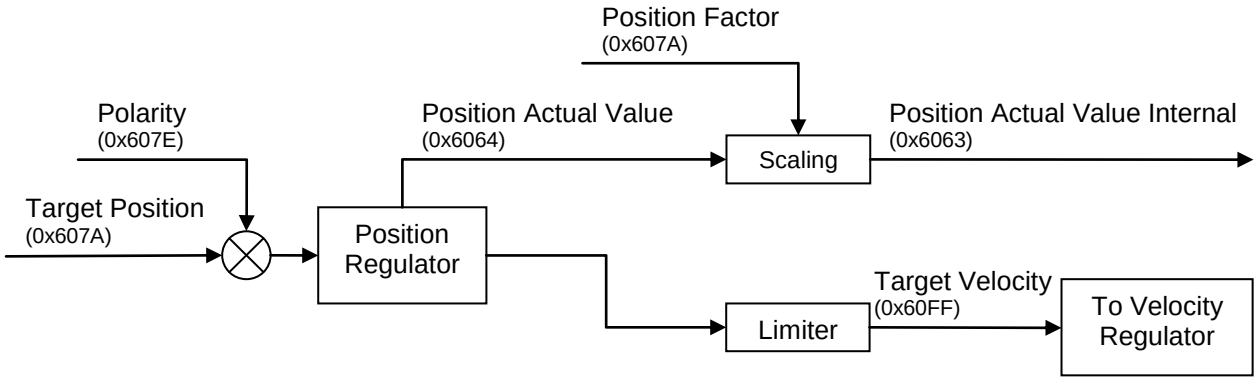


Figure 0-6 Cyclic Sync Position – Block Diagram

5.1.3 Output current Limitation

Output current limitation function is limiting *Current Demand* (0x201a) according limit values which can be set in objects:

- *Motor Maximum Temperature* (0x2058),
- *Motor Maximum Temperature Gain* (0x2059),
- *Maximum Velocity* (0x2052, 1),
- *Maximum Velocity Gain* (0x2052, 2),
- *Undervoltage Limit* (0x2055, 1),
- *Undervoltage Limit Gain* (0x2055, 2),
- *Overvoltage Limit* (0x2054),
- *Stall Protection Current* (0x205c),
- *Stall Protection Time* (0x205b).

Controller temperature is also monitored and it's linearly for each degree of over temperature (85°C) limiting maximum current for 20% of maximum controller current.
If any of current limits are active, Warning Yellow LED will be turned on, also bit will be set in *Warning* (0x2027) according active limits.

Object description:

Index	0x2027
Name	Warning
Data type	UNSIGNED16
Saved in EEPROM	No

Entry description:

Access	Read
PDO mapping	Allowed
Default value	0

Data description:

Bit patern (MSB...LSB)	**Description**
0000 0000 0000 0001	Controller Temperature
0000 0000 0000 0010	Motor Temperature
0000 0000 0000 0100	DC Link Under Voltage
0000 0000 0000 1000	DC Link Over Voltage
0000 0000 0001 0000	Stall
0000 0000 0010 0000	Maximum Velocity

6 FEEDBACK CONFIG

0x2040: Feedback Config

In object Feedback Config (0x2040) user is able to set all settings according position feedback:

- Feedback Type
- Position Observer Enable
- Motor Phase Offset
- Feedback Direction
- Test Stepper Frequency
- Speed Filtering
- Motor Phase Offset Compensation
- Test Stepper Enable
- Auto Align Rotor Position Current

All feedback setting must be done in NMT state Preoperational. Parameters must be saved and the emDrive must be restarted.

6.1 0x2040, 1 Feedback Type

emDrive is supporting several different feedback types which are selectable through object Feedback Type (0x2040, 1).
Selection must be done in NMT state Preoperational.

Object description:

Index	0x2040
Name	Feedback Type
Data type	UNSIGNED8
Saved in EEPROM	Yes

Entry description:

Access	Read / Write
PDO mapping	Allowed
Default value	3

Data description:

Register data	**Feedback type**
0	Constant Angle
1	Hall Six Step Commutation
2	SSI
3	Resolver
4	Sin / Cos
5	Encoder
6	Hall Predictive
7 … 255	Reserved

Hint:
If using Motor with hall feedback (selected feedback Type Hall six step commutation or Hall predictive) Feedback resolution must be set to 6.

7 INPUTS AND OUTPUTS

7.1 Analog Inputs

The drive supports three analog inputs with resolution of 12bit (5V). One is used for motor temperature measurements, other two may be used for general purpose process values like reference for desired speed, torque or position etc.

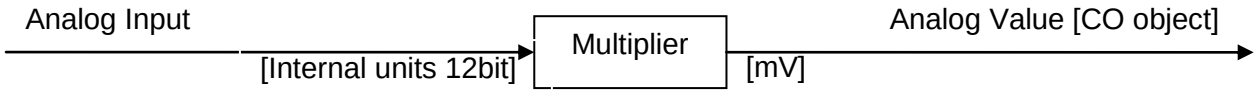


Figure 7 Analog Inputs Block Diagram

7.1.1 Output data description

The output values are given in the object Analog Values. For each analog input voltage on AD pin can be read in corresponding Analog value.

7.2 Digital Inputs

The drive supports 3-6 digital inputs (depends of emDrive type). They may be used for general purpose process like enable FW, RW, Brake etc.

7.2.1 Output data description

The output values are given in the object Digital Values. Corresponding flag is set in object. Polarity of digital input is here already incorporated.

emDrive 400:

Din num.	Reserved										Din6	Din5	Din4	Din3	Din2	Din1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

emDrive150, emDrive H300:

Din num.	Reserved													Din3	Din2	Din1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Example:
If Din3 is active bit 13 will be set to 1 and value in register will be 0x0002.

8 OBJECT DICTIONARY

8.1 0x1000 Device Type

Name	Device Type	
Index	*0x1000*	
Subindex	*0x00*	
Type	*UNSIGNED32*	
Access	*RO*	
Stored in EEPROM	*Yes*	
Default Value	*0x0192*	
Value range	*0*	*4294967295*
Map to PDO	*No*	

Description
This constant describes the device type. The value 0x0192 (400) mean that device follows the CiA Draft Standard Proposal 402, Device Profile Driver and Motion Control.

8.2 0x1001 Error register

Name	Error register	
Index	*0x1001*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*0x00*	
Value range	*0*	*255*
Map to PDO	*Yes*	

Description
This object is an error register for the device. The device maps internal errors in this byte.

Related Objects

Table 2 Error register bits

Bit	Description
0	Generic Error
1	Current
2	Voltage
3	Temperature
4	Communication Error
5	Device Profile Defined Error
6	Reserved (always zero)
7	Manufacturer specific

8.3 0x1005 COB ID SYNC

Name	COB ID SYNC	
Index	*0x1005*	
Subindex	*0x00*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*0x80*	
Value range	*0*	*4294967295*
Map to PDO	*No*	

Description
Communication Object Identifier of synchronization object.

8.4 0x1008 Manufacturer Device Name

Name	Manufacturer Device Name	
Index	*0x1008*	
Subindex	*0x00*	
Type	*VISIBLE_STRING*	
Access	*RO*	
Stored in EEPROM	*Yes*	
Default Value	*EMSISO*	
Value range	*0*	*0*
Map to PDO	*No*	

Description
The product name is “EMSISO”

8.5 0x100b Node-ID

Name	Node-ID	
Index	*0x100b*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0x01*	
Value range	*1*	*255*
Map to PDO	*No*	

8.6 0x1010 Store Parameter Field

Name	**Store Parameter Field**
Index	*0x1010*
Number of entries	*0x04*

Name	**Save all Parameters**	
Index	*0x1010*	
Subindex	*0x01*	
Type	*VISIBLE_STRING*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*0*	
Value range	-	-
Map to PDO	*No*	

Description
All parameters of device where stored in non volatile memory, if the code “save” is written to this object.

Byte	MSB			LSB
Character	‘e’	‘v’	‘a’	‘s’
Hexvalue	0x65	0x76	0x61	0x73

Controller must be restarted after this action is performed.

8.7 0x1011 Restore Default Parameters

Name	**Restore Default Parameters**
Index	*0x1011*
Number of entries	*0x04*

Name	**Restore Default Parameters**	
Index	*0x1011*	
Subindex	*0x01*	
Type	*VISIBLE_STRING*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*0*	
Value range	-	-
Map to PDO	*No*	

Description
All parameters of device where restored with default values, if code “load” is written to this object.

Byte	MSB			LSB
Character	‘d’	‘a’	‘o’	‘l’
Hexvalue	0x64	0x61	0x6F	0x6C

Controller must be restarted after this action is performed.

8.8 0x1014 COB ID EMCY

Name	COB ID EMCY	
Index	*0x1014*	
Subindex	*0x00*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*0x80 + node ID*	
Value range	-	-
Map to PDO	*No*	

Description
Communication Object Identifier of emergency object.

8.9 0x1017 Producer Heartbeat Time

Name	Producer Heartbeat Time	
Index	*0x1017*	
Subindex	*0x00*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*0*	*65535*
Map to PDO	*No*	

Description
This producer heartbeat time defines the cycle time of heartbeat. The producer heartbeat time is 0 if not used. The time has to be a multiple of 1ms.

Remarks
It's not allowed for one device to use both error control mechanisms Guarding Protocol and Heartbeat Protocol at the same time. If the heartbeat producer time is equal to 0 the heartbeat protocol is used and the guarding protocol is disabled.

8.10 0x1018 Identity Object

Name	Identity Object
Index	*0x1018*
Number of entries	*0x04*

Name	Vendor Id
Index	*0x1018*
Subindex	*0x01*
Type	*UNSIGNED32*
Access	*RO*
Stored in EEPROM	*-*

Default Value	*000003C6*	
Value range	*0*	*4294967295*
Map to PDO	*No*	

Description
The CANopen vendor identification of “Emsiso emDrives”

Name	**Product Code**	
Index	*0x1018*	
Subindex	*0x02*	
Type	*UNSIGNED32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**Revision number**	
Index	*0x1018*	
Subindex	*0x03*	
Type	*UNSIGNED32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*4294967295*
Map to PDO	*No*	

Description

CAN revision	Drive Major revision	Drive Minor revision
xx	xx	xx

8.11 0x1021 Store EDS

Name	**Store EDS**	
Index	*0x1021*	
Subindex	*0x00*	
Type	*DOMAIN*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*Not defined*	
Value range	*-*	*-*
Map to PDO	*No*	

Description
The Store EDS includes Electronic Datasheet which is used for the node.

Remarks

The electronic datasheet is read and written according to the format specified in entry 0x1022.

8.12 0x1022 Storage Format

Name	**Store EDS**	
Index	*0x1022*	
Subindex	*0x00*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*Not defined*	
Value range	*-*	*-*
Map to PDO	*No*	

Description
From FW version newer than 1.1.7 this entry defines the format that electronic datasheet is read from and written to entry 0x1021

Value	Description
0x0000	Uncompressed ASCII
0x0001	.zip

0x1029 Error behavior

Name	**Error behavior**	
Index	*0x1029*	
Number of entries	*0x01*	

Name	**Communication Error**	
Index	*0x1029*	
Subindex	*0x01*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*0*	*1*
Map to PDO	*No*	

Description

Value	Description
0x0000	Switches to Pre-operational
0x0001	Does not change states

8.13 0x1200 Server SDO parameter

Name	**Server SDO parameter**
Index	*0x1200*

Number of entries	*0x02*

Name	**COB ID Client to Server (Receive SDO)**	
Index	*0x1200*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*Node ID + 0x600*	
Value range	*-*	*-*
Map to PDO	*No*	

Description
The communication Object Identifier of service data objects from master to device is shown here.

Name	**COB ID Server to Client (Transmit SDO)**	
Index	*0x1200*	
Subindex	*0x02*	
Type	*UNSIGNED32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*Node ID + 0x580*	
Value range	*-*	*-*
Map to PDO	*No*	

Description
The communication Object Identifier of service data objects from device to master is shown here.

8.14 0x1400 Receive PDO 1 Parameter

Name	**Receive PDO 1 Parameter**
Index	*0x1400*
Number of entries	*0x05*

Name	**COB ID receive PDO 1**	
Index	*0x1400*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*Node ID + 0x200*	
Value range	*0*	*4294967295*
Map to PDO	*No*	

Description
Communication Object Identifier of receive process data object 1.

Name	Transmission type receive PDO 1	
Index	*0x1400*	
Subindex	*0x02*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0xFE*	
Value range	*0*	*255*
Map to PDO	*No*	

Description
The transmission type describes how PDO communication works. The following types are supported:

Value	Description
1	synchron
255	asynchron

8.15 0x1401 Receive PDO 2 parameter

Name	Receive PDO 2 Parameter
Index	*0x1401*
Number of entries	*0x02*

Name	COB ID receive PDO 2	
Index	*0x1401*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*Node Id + 0x300*	
Value range	*0*	*4294967295*
Map to PDO	*No*	

Description
Communication Object Identifier of receive process data object 2.

Name	Transmission type receive PDO 2	
Index	*0x1401*	
Subindex	*0x02*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0xFE*	
Value range	*0*	*255*
Map to PDO	*No*	

Description
The transmission type describes how PDO communication works. The following types are supported:

Value	Description

1	synchron
255	asynchron

8.16 0x1402 Receive PDO 3 parameter

Name	**Receive PDO 3 Parameter**
Index	*0x1402*
Number of entries	*0x02*

Name	**COB ID receive PDO 3**	
Index	*0x1402*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*Node Id + 0x300*	
Value range	*0*	*4294967295*
Map to PDO	*No*	

Description
Communication Object Identifier of receive process data object 3.

Name	**Transmission type receive PDO 3**	
Index	*0x1402*	
Subindex	*0x02*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0xFE*	
Value range	*0*	*255*
Map to PDO	*No*	

Description
The transmission type describes how PDO communication works. The following types are supported:

Value	**Description**
1	synchron
255	asynchron

8.17 0x1403 Receive PDO 4 parameter

Name	**Receive PDO 4 Parameter**
Index	*0x1403*
Number of entries	*0x02*

Name	**COB ID receive PDO 4**
Index	*0x1403*
Subindex	*0x01*
Type	*UNSIGNED32*

Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*Node Id + 0x300*	
Value range	*0*	*4294967295*
Map to PDO	*No*	

Description
Communication Object Identifier of receive process data object 2.

Name	**Transmission type receive PDO 4**	
Index	*0x1403*	
Subindex	*0x02*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0xFE*	
Value range	*0*	*255*
Map to PDO	*No*	

Description
The transmission type describes how PDO communication works. The following types are supported:

Value	Description
1	Synchron
255	Asynchrony

8.18 0x1600 Receive PDO 1 Mapping

Name	**Receive PDO 1 Parameter**
Index	*0x1600*
Number of entries	*0x08*

Name	**Number of mapped Aplication Objects in receive PDO**	
Index	*0x1600*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*3*	
Value range	*0*	*255*
Map to PDO	*No*	

Description
Changes in mapping are only possible in NMT state pre-operational. Before it's possible to enable PDO, it's necessary to map objects.

Value	Description
0	PDO is disabled
1-8	One to eight objects are mapped

Name	1st mapped object	
Index	*0x1600*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0x60400010*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	2nd mapped object	
Index	*0x1600*	
Subindex	*0x02*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	3rd mapped object	
Index	*0x1600*	
Subindex	*0x03*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	4th mapped object	
Index	*0x1600*	
Subindex	*0x04*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	5th mapped object	
Index	*0x1600*	
Subindex	*0x05*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	7th mapped object

Index	*0x1600*	
Subindex	*0x07*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**8th mapped object**	
Index	*0x1600*	
Subindex	*0x08*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Description
The objects that are supported to map are marked in each object Id description at the bottom area of table description: Map to PDO.

Remarks
Changes in mapping are only possible in NMT state pre-operational. To change a mapped object it is necessary to disable PDO. The maximal length of a process data object is 64 bit; because of this it is only possible to map two 32-bit values or two 16-bit values and one 32-bit value and so on.

8.19 0x1601 Receive PDO 2 mapping

Name	**Receive PDO 2 Parameter**	
Index	*0x1601*	
Number of entries	*0x08*	

Name	**Number of mapped Aplication Objects in receive PDO**	
Index	*0x1601*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*1*	
Value range	*0*	*255*
Map to PDO	*No*	

Description
Changes in mapping are only possible in NMT state pre-operational. Before it's possible to enable PDO, it's necessary to map objects.

Value	Description
0	PDO is disabled

1-8	One to eight objects are mapped

Name	**1st mapped object**	
Index	*0x1601*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*0x60400010*	
Value range	-	-
Map to PDO	*No*	

Name	**2nd mapped object**	
Index	*0x1601*	
Subindex	*0x02*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*0*	
Value range	-	-
Map to PDO	*No*	

Name	**3rd mapped object**	
Index	*0x1601*	
Subindex	*0x03*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*0*	
Value range	-	-
Map to PDO	*No*	

Name	**4th mapped object**	
Index	*0x1601*	
Subindex	*0x04*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*0*	
Value range	-	-
Map to PDO	*No*	

Name	**5th mapped object**	
Index	*0x1601*	
Subindex	*0x05*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*0*	
Value range	-	-

Map to PDO	*No*

Name	**7th mapped object**	
Index	*0x1601*	
Subindex	*0x07*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**8th mapped object**	
Index	*0x1601*	
Subindex	*0x08*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Description
The objects that are supported to map are marked in each object Id description at the bottom area of table description: Map to PDO.

Remarks
Changes in mapping are only possible in NMT state pre-operational. To change a mapped object it is necessary to disable PDO. The maximal length of a process data object is 64 bit; because of this it is only possible to map two 32-bit values or two 16-bit values and one 32-bit value and so on.

8.20 0x1602 Receive PDO 3 Mapping

Name	**Receive PDO 3 Parameter**
Index	*0x1602*
Number of entries	*0x08*

Name	**Number of mapped Aplication Objects in receive PDO**	
Index	*0x1602*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*3*	
Value range	*0*	*255*
Map to PDO	*No*	

Description
Changes in mapping are only possible in NMT state pre-operational. Before it's possible to enable PDO, it's necessary to map objects.

Value	Description
0	PDO is disabled
1-8	One to eight objects are mapped

Name	**1st mapped object**	
Index	*0x1602*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0x60400010*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**2nd mapped object**	
Index	*0x1602*	
Subindex	*0x02*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**3rd mapped object**	
Index	*0x1602*	
Subindex	*0x03*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**4th mapped object**	
Index	*0x1602*	
Subindex	*0x04*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**5th mapped object**
Index	*0x1602*
Subindex	*0x05*
Type	*UNSIGNED32*
Access	*RW*

Stored in EEPROM	-	
Default Value	*0*	
Value range	-	-
Map to PDO	*No*	

Name	**7th mapped object**	
Index	*0x1602*	
Subindex	*0x07*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*0*	
Value range	-	-
Map to PDO	*No*	

Name	**8th mapped object**	
Index	*0x1602*	
Subindex	*0x08*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*0*	
Value range	-	-
Map to PDO	*No*	

Description
The objects that are supported to map are marked in each object Id description at the bottom area of table description: Map to PDO.

Remarks
Changes in mapping are only possible in NMT state pre-operational. To change a mapped object it is necessary to disable PDO. The maximal length of a process data object is 64 bit; because of this it is only possible to map two 32-bit values or two 16-bit values and one 32-bit value and so on.

8.21 0x1603 Receive PDO 4 Mapping

Name	**Receive PDO 4 Parameter**	
Index	*0x1603*	
Number of entries	*0x08*	

Name	**Number of mapped Aplication Objects in receive PDO**	
Index	*0x1603*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*3*	
Value range	*0*	*255*
Map to PDO	*No*	

Description

Changes in mapping are only possible in NMT state pre-operational. Before it's possible to enable PDO, it's necessary to map objects.

Value	Description
0	PDO is disabled
1-8	One to eight objects are mapped

Name	**1st mapped object**	
Index	*0x1603*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0x60400010*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**2nd mapped object**	
Index	*0x1600*	
Subindex	*0x02*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**3rd mapped object**	
Index	*0x1603*	
Subindex	*0x03*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**4th mapped object**	
Index	*0x1603*	
Subindex	*0x04*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**5th mapped object**
Index	*0x1603*
Subindex	*0x05*
Type	*UNSIGNED32*

Access	*RW*	
Stored in EEPROM	-	
Default Value	*0*	
Value range	-	-
Map to PDO	*No*	

Name	**7th mapped object**
Index	*0x1603*
Subindex	*0x07*
Type	*UNSIGNED32*
Access	*RW*
Stored in EEPROM	-
Default Value	*0*
Value range	- -
Map to PDO	*No*

Name	**8th mapped object**
Index	*0x1603*
Subindex	*0x08*
Type	*UNSIGNED32*
Access	*RW*
Stored in EEPROM	-
Default Value	*0*
Value range	- -
Map to PDO	*No*

Description
The objects that are supported to map are marked in each object Id description at the bottom area of table description: Map to PDO.

Remarks
Changes in mapping are only possible in NMT state pre-operational. To change a mapped object it is necessary to disable PDO. The maximal length of a process data object is 64 bit; because of this it is only possible to map two 32-bit values or two 16-bit values and one 32-bit value and so on.

8.22 0x1800 Transimt PDO 1 parameter

Name	**Transmit PDO 1 Parameter**
Index	*0x1800*
Number of entries	*0x06*

Name	**COB ID**
Index	*0x1800*
Subindex	*0x01*
Type	*UNSIGNED32*
Access	*RW*
Stored in EEPROM	-
Default Value	*Node ID + 0x180*
Value range	- -

Map to PDO	*No*

Description
Communication Object Identifier of transmit process data object 1.

Name	**Transmission type transmit PDO 1**	
Index	*0x1800*	
Subindex	*0x02*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0xFE*	
Value range	*0*	*255*
Map to PDO	*No*	

Description
The transmission type describes how PDO communication works. The following types are supported:

Value	Description
1	Synchron
253	Asynchron on RTR only
255	Asynchron on change

Remarks
The transmission type 253 means that the PDO is only transmitted on remote transmission request (RTR). If transmission type 255 is selected the PDO is transmitted if the data's change its values. The inhibit time defines a minimum interval therefore.

Name	**Inhibit time transmit PDO 1**	
Index	*0x1800*	
Subindex	*0x03*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*65535*	
Value range	*-*	*-*
Map to PDO	*No*	

Description
The time is the minimum interval for event triggered PDO transmison. The value is defined as multiple of 100µs.

Remarks
Event triggered PDOs can generate a huge CAN bus load also devise load especially if the inhibit time of different PDOs are set to a small value.

8.23 0x1801 Transmit PDO 2 parameter

Name	**Transmit PDO 2 parameter**
Index	*0x1801*
Number of entries	*0x03*

Name	COB ID transmit PDO 2	
Index	*0x1801*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*Node ID + 0x0280*	
Value range	*0x00000181*	*0x0000057F*
Map to PDO	*No*	

Description
Communication Object Identifier of transmit data object 2.

Name	Transmission type transmit PDO 2	
Index	*0x1801*	
Subindex	*0x02*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*253*	
Value range	*-*	*-*
Map to PDO	*No*	

Description
The transmission type describes how PDO communication works. The following types are supported:

Value	Description
1	Synchron
253	Asynchron on RTR only
255	Asynchron on change

Remarks
The transmission type 253 means that the PDO is only transmitted on remote transmission request (RTR). If transmission type 255 is selected the PDO is transmitted if the data's change its values. The inhibit time defines a minimum interval therefore.

Name	Inhibit time transmit PDO 2	
Index	*0x1801*	
Subindex	*0x02*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*253*	
Value range	*-*	*-*
Map to PDO	*No*	

Description
The transmission type describes how PDO communication works. The following types are supported:

Value	Description
1	Synchron

253	Asynchron on RTR only
255	Asynchron on change

Remarks
The transmission type 253 means that the PDO is only transmitted on remote transmission request (RTR). If transmission type 255 is selected the PDO is transmitted if the data's change its values. The inhibit time defines a minimum interval therefore.

8.24 0x1802 Transmit PDO 3 parameter

Name	**Transmit PDO 3 parameter**
Index	*0x1802*
Number of entries	*0x03*

Name	**COB ID transmit PDO 3**	
Index	*0x1802*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*Node ID + 0x0380*	
Value range	*0x00000181*	*0x0000057F*
Map to PDO	*No*	

Description
Communication Object Identifier of transmit process data object 3.

Name	**Transmission type transmit PDO 3**	
Index	*0x1802*	
Subindex	*0x02*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*253*	
Value range	*-*	*-*
Map to PDO	*No*	

Descritpion
The transmission type describes how PDO communication works. The following types are supported:

Value	Description
1	Synchron
253	Asynchron on RTR only
255	Asynchron on change

Remarks
The transmission type 253 means that the PDO is only transmitted on remote transmission request (RTR). If transmission type 255 is selected the PDO is transmitted if the data's change its values. The inhibit time defines a minimum interval therefore.

Name	**Inhibit time transmit PDO 3**

Index	*0x1802*	
Subindex	*0x03*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*65535*	
Value range	*-*	*-*
Map to PDO	*No*	

Description
This time is the minimum interval for event triggered PDO transmission. The value is defined as multiple of 100µs.

Remarks
Event triggered PDOs can generate a huge CAN bus load and also device load especially if the inhibit time of different PDOs are set to a small value.

8.25 0x1803 Transmit PDO 4 parameter

Name	**Transmit PDO 4 parameter**	
Index	*0x1803*	
Number of entries	*0x03*	

Name	**COB ID transmit PDO 4**	
Index	*0x1803*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*Node ID + 0x0480*	
Value range	*0x00000181*	*0x0000057F*
Map to PDO	*No*	

Description
Communication Object Identifier of transmit process data object 3.

Name	**Transmission type transmit PDO 3**	
Index	*0x1803*	
Subindex	*0x02*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*253*	
Value range	*-*	*-*
Map to PDO	*No*	

Descritpion
The transmission type describes how PDO communication works. The following types are supported:

Value	Description

1	Synchron
253	Asynchron on RTR only
255	Asynchron on change

Remarks
The transmission type 253 means that the PDO is only transmitted on remote transmission request (RTR). If transmission type 255 is selected the PDO is transmitted if the data's change its values. The inhibit time defines a minimum interval therefore.

Name	**Inhibit time transmit PDO 4**	
Index	*0x1803*	
Subindex	*0x03*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*65535*	
Value range	-	-
Map to PDO	*No*	

Description
This time is the minimum interval for event triggered PDO transmission. The value is defined as multiple of 100µs.

Remarks
Event triggered PDOs can generate a huge CAN bus load and also device load especially if the inhibit time of different PDOs are set to a small value.

8.26 0x1A00 Transmit PDO 1 mapping

Name	**Transmit PDO 1 mapping**
Index	*0x1A00*
Number of entries	-

Name	**Number of mapped Application Objects in transmit PDO**	
Index	*0x1A00*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*1*	
Value range	*0*	*8*
Map to PDO	*No*	

Remarks
Changes in mapping are only possible in NMT state pre-operational. Before it is possible to enable PDO, it is necessary to map objects.

Value	Description
0	PDO is disabled
1-8	One to eight objects are mapped

Name	1st mapped object	
Index	*0x1A00*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0x60400010*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	2nd mapped object	
Index	*0x1A00*	
Subindex	*0x02*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	3rd mapped object	
Index	*0x1A00*	
Subindex	*0x03*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	4th mapped object	
Index	*0x1A00*	
Subindex	*0x04*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	5th mapped object	
Index	*0x1A00*	
Subindex	*0x05*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	7th mapped object

Index	*0x1A00*	
Subindex	*0x07*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**8th mapped object**	
Index	*0x1A00*	
Subindex	*0x08*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Description
The objects that are supported to map are marked in each object Id description at the bottom area of table description: Map to PDO.

Remarks
Changes in mapping are only possible in NMT state pre-operational. To change a mapped object it is necessary to disable PDO. The maximal length of a process data object is 64 bit; because of this it is only possible to map two 32-bit values or two 16-bit values and one 32-bit value and so on.

8.27 0x1A01Transmit PDO 2 mapping

Name	**Transmit PDO 2 mapping**	
Index	*0x1A01*	
Number of entries	*-*	

Name	**Number of mapped Application Objects in transmit PDO**	
Index	*0x1A01*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*1*	
Value range	*0*	*8*
Map to PDO	*No*	

Remarks
Changes in mapping are only possible in NMT state pre-operational. Before it is possible to enable PDO, it is necessary to map objects.

Value	Description
0	PDO is disabled
1-8	One to eight objects are mapped

Name	1st mapped object	
Index	*0x1A01*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0x60400010*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	2nd mapped object	
Index	*0x1A01*	
Subindex	*0x02*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	3rd mapped object	
Index	*0x1A01*	
Subindex	*0x03*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	4th mapped object	
Index	*0x1A01*	
Subindex	*0x04*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	5th mapped object	
Index	*0x1A01*	
Subindex	*0x05*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	7th mapped object	
Index	*0x1A01*	
Subindex	*0x07*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*0*	
Value range	-	-
Map to PDO	*No*	

Name	8th mapped object	
Index	*0x1A01*	
Subindex	*0x08*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*0*	
Value range	-	-
Map to PDO	*No*	

Description
The objects that are supported to map are marked in each object Id description at the bottom area of table description: Map to PDO.

Remarks
Changes in mapping are only possible in NMT state pre-operational. To change a mapped object it is necessary to disable PDO. The maximal length of a process data object is 64 bit; because of this it is only possible to map two 32-bit values or two 16-bit values and one 32-bit value and so on.

8.28 0x1A02 Transmit PDO 3 mapping

Name	Transmit PDO 3 mapping
Index	*0x1A02*
Number of entries	-

Name	Number of mapped Application Objects in transmit PDO	
Index	*0x1A02*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*1*	
Value range	*0*	*8*
Map to PDO	*No*	

Remarks
Changes in mapping are only possible in NMT state pre-operational. Before it is possible to enable PDO, it is necessary to map objects.

Value	Description

0	PDO is disabled
1-8	One to eight objects are mapped

Name	**1st mapped object**	
Index	*0x1A02*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0x60400010*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**2nd mapped object**	
Index	*0x1A02*	
Subindex	*0x02*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**3rd mapped object**	
Index	*0x1A02*	
Subindex	*0x03*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**4th mapped object**	
Index	*0x1A02*	
Subindex	*0x04*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**5th mapped object**
Index	*0x1A02*
Subindex	*0x05*
Type	*UNSIGNED32*
Access	*RW*
Stored in EEPROM	*-*
Default Value	*0*

Value range	-	-
Map to PDO	*No*	

Name	**7th mapped object**
Index	*0x1A02*
Subindex	*0x07*
Type	*UNSIGNED32*
Access	*RW*
Stored in EEPROM	-
Default Value	*0*
Value range	- -
Map to PDO	*No*

Name	**8th mapped object**
Index	*0x1A02*
Subindex	*0x08*
Type	*UNSIGNED32*
Access	*RW*
Stored in EEPROM	-
Default Value	*0*
Value range	- -
Map to PDO	*No*

Description
The objects that are supported to map are marked in each object Id description at the bottom area of table description: Map to PDO.

Remarks
Changes in mapping are only possible in NMT state pre-operational. To change a mapped object it is necessary to disable PDO. The maximal length of a process data object is 64 bit; because of this it is only possible to map two 32-bit values or two 16-bit values and one 32-bit value and so on.

8.29 0x1A03 Transmit PDO 4 mapping

Name	**Transmit PDO 3 mapping**
Index	*0x1A03*
Number of entries	-

Name	**Number of mapped Application Objects in transmit PDO**	
Index	*0x1A03*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	-	
Default Value	*1*	
Value range	*0*	*8*
Map to PDO	*No*	

Remarks

Changes in mapping are only possible in NMT state pre-operational. Before it is possible to enable PDO, it is necessary to map objects.

Value	Description
0	PDO is disabled
1-8	One to eight objects are mapped

Name	**1st mapped object**	
Index	*0x1A03*	
Subindex	*0x01*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0x60400010*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**2nd mapped object**	
Index	*0x1A03*	
Subindex	*0x02*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**3rd mapped object**	
Index	*0x1A03*	
Subindex	*0x03*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**4th mapped object**	
Index	*0x1A03*	
Subindex	*0x04*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**5th mapped object**
Index	*0x1A03*
Subindex	*0x05*
Type	*UNSIGNED32*

Access	*RW*	
Stored in EEPROM	-	
Default Value	*0*	
Value range	-	-
Map to PDO	*No*	

Name	**7th mapped object**
Index	*0x1A03*
Subindex	*0x07*
Type	*UNSIGNED32*
Access	*RW*
Stored in EEPROM	-
Default Value	*0*
Value range	- -
Map to PDO	*No*

Name	**8th mapped object**
Index	*0x1A03*
Subindex	*0x08*
Type	*UNSIGNED32*
Access	*RW*
Stored in EEPROM	-
Default Value	*0*
Value range	- -
Map to PDO	*No*

Description
The objects that are supported to map are marked in each object Id description at the bottom area of table description: Map to PDO.

Remarks
Changes in mapping are only possible in NMT state pre-operational. To change a mapped object it is necessary to disable PDO. The maximal length of a process data object is 64 bit; because of this it is only possible to map two 32-bit values or two 16-bit values and one 32-bit value and so on.

8.30 0x2001 CAN Bitrate

Name	**CAN bitrate**
Index	*0x2001*
Subindex	*0x00*
Type	*UNSIGNED16*
Access	*RW*
Stored in EEPROM	*Yes*
Default Value	*500*
Value range	- -
Map to PDO	*No*

Description
The bit rate of the CAN interface can be changed with the CAN bitrate parameter.

Remarks
The changes to this object take only effect after restart. Therefore it's necessary to store all parameters after changing and then restart.

Value	Bit rate
20	20 kbit/s
50	50 kbit/s
125	125 kbit/s
250	250 kbit/s
500	500 kbit/s
800	800 kbit/s
1000	1Mbit/s

8.31 0x2002 RS232

Name	**RS232 baudrate**	
Index	*0x2002*	
Subindex	*0x00*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*115200*	
Value range	-	-
Map to PDO	*No*	

Description
The baud rate of the serial communication interface can be changed with the RS232 baudrate parameter.

8.32 0x2003 CPU load

Name	**CPU load**	
Index	*0x2003*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*Yes*	
Default Value	-	
Value range	*0*	*100*
Map to PDO	*No*	

Description:
Calculated CPU load displayed in this object.

8.33 0x200C Custom persistent memory

Name	**Custom persistent memory**

Index	*0x200C*
Number of entries	*12*

Name	**Custom persistent memory unsigned-8 1**	
Index	*0x200C*	
Subindex	*0x01*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**Custom persistent memory unsigned-8 2**	
Index	*0x200C*	
Subindex	*0x02*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**Custom persistent memory signed-8 1**	
Index	*0x200C*	
Subindex	*0x03*	
Type	*INTEGER8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**Custom persistent memory signed-8 2**	
Index	*0x200C*	
Subindex	*0x04*	
Type	*INTEGER8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**Custom persistent memory unsigned-16 1**	
Index	*0x200C*	
Subindex	*0x05*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-*	*-*

Map to PDO	*No*

Name	**Custom persistent memory unsigned-16 2**	
Index	*0x200C*	
Subindex	*0x06*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**Custom persistent memory signed-16 1**	
Index	*0x200C*	
Subindex	*0x07*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**Custom persistent memory signed-16 2**	
Index	*0x200C*	
Subindex	*0x08*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**Custom persistent memory unsigned-32 1**	
Index	*0x200C*	
Subindex	*0x09*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**Custom persistent memory unsigned-32 2**	
Index	*0x200C*	
Subindex	*0x0A*	
Type	*UNSIGNED32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Name	**Custom persistent memory signed-32 1**	
Index	*0x200C*	
Subindex	*0x0B*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	
Name	**Custom persistent memory signed-32 2**	
Index	*0x200C*	
Subindex	*0x0C*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*No*	

Description
This persistent memory can be used to store custom values. These values would not be evaluated by the firmware, but they will be cleared by setting default parameters.

8.34 0x200F Phase A current

Name	**Phase A current**	
Index	*0x200F*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-32768*	*-32767*
Map to PDO	*Yes*	

Description
Actual measured current on phase A in quants.

8.35 0x2010 Phase B current

Name	**Phase B current**	
Index	*0x2010*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-32768*	*-32767*

Map to PDO	*Yes*

Description
Actual measured current of phase B in quants.

8.36 0x201A Current demand

Name	**Current demand**	
Index	*0x201A*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-32768*	*-32767*
Map to PDO	*Yes*	

Description
The desired value of motor current which is inserted into current regulator in 1/1000 Motor rated current.
Object is read only and is set according selected mode of operation.
In test modes of operation current limitation is not done.

8.37 0x201B Torque regulator

Name	**Torque regulator**
Index	*0x201B*
Number of entries	*-*

Name	**Number of entries**	
Index	*0x201B*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*7*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Torque regulator requested**	
Index	*0x201B*	
Subindex	*0x01*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Object is read only and is calculated from Current demand 0x201A in quants.

Name	**Torque regulator actual**	
Index	*0x201B*	
Subindex	*0x02*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Object is read only in quants.

Name	**Torque regulator out**	
Index	*0x201B*	
Subindex	*0x03*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Object is read only in quants. Value is output of the regulator.

Name	**Torque regulator error**	
Index	*0x201B*	
Subindex	*0x04*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Object is read only in quants. Value is presenting error – difference between requested and actual value.

Name	**Calculated fly on torque regulator-I**	
Index	*0x201B*	
Subindex	*0x05*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

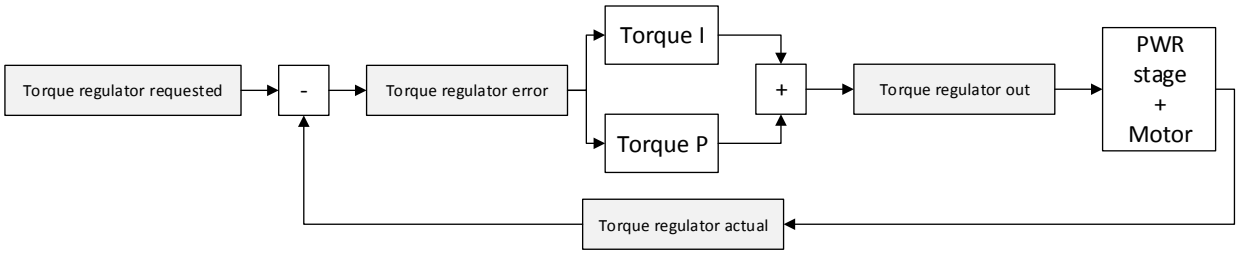
Description:
Object is read only in 1/sec. Calculated value is used for fly on initialization of I part of torque regulator.

Name	**Calculated fly on torque regulator-P**	
Index	*0x201B*	
Subindex	*0x06*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	-	
Default Value	-	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Object is read only in 1/1000. Calculated value is used for fly on initialization of P part of torque regulator.

Name	**Calculated fly on torque internal**	
Index	*0x201B*	
Subindex	*0x07*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	-	
Default Value	-	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Object is read only in quants. Calculated value is used for fly on initialization of internal value in torque regulator.



8.38 0x201C Flux regulator

Name	**Flux regulator**
Index	*0x201C*
Number of entries	-

Name	**Number of entries**
Index	*0x201C*
Subindex	*0x00*
Type	*UNSIGNED8*

Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*7*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Flux regulator requested**	
Index	*0x201C*	
Subindex	*0x01*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Object is read only in quants.

Name	**Flux regulator actual**	
Index	*0x201C*	
Subindex	*0x02*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Object is read only in quants.

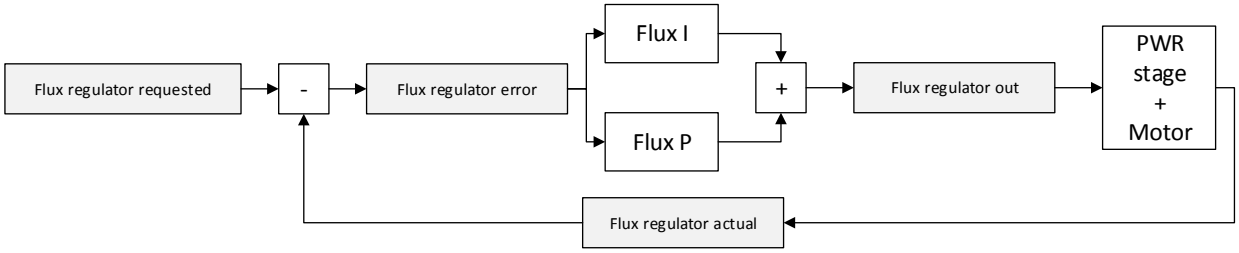
Name	**Flux regulator out**	
Index	*0x201C*	
Subindex	*0x03*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Object is read only in quants. Value is output of the regulator.

Name	**Flux regulator error**	
Index	*0x201C*	
Subindex	*0x04*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	

Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Object is read only in quants. Value is presenting error – difference between requested and actual value.



8.39 0x201D Motor Field Settings

Name	**Motor Field Settings**
Index	*0x201D*
Number of entries	*-*

Name	**Number of entries**	
Index	*0x201D*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*6*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**PMSM Flux mode**	
Index	*0x201D*	
Subindex	*0x01*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*0*	*255*
Map to PDO	*No*	

Description:
emDrive si supporting different field weakening modes:
0 - no field weakening
1 - Manual field weakening
2 - Auto field weakening

Name	**PMSM Flux Manual current**
Index	*0x201D*
Subindex	*0x02*
Type	*INTEGER32*
Access	*RO*

Stored in EEPROM	-	
Default Value	0	
Value range	0	255
Map to PDO	No	

Description:
Negative is field weakening, positive is field reinforcement
in auto mode this is maximum field weakening current

Name	**Flux actual current**	
Index	0x201D	
Subindex	0x03	
Type	INTEGER32	
Access	RO	
Stored in EEPROM	-	
Default Value	0	
Value range	0	255
Map to PDO	No	

Description:
Positive is field wekening, value is actual field weakening current, value displayed in quants.

Name	**Flux I increment**	
Index	0x201D	
Subindex	0x04	
Type	INTEGER32	
Access	RO	
Stored in EEPROM	-	
Default Value	0	
Value range	0	255
Map to PDO	No	

Description:
When motor is transitioning from normal operation to field weakening this is gain for adding flux current, unit: mA/s.

Name	**Flux pulse mode counter**	
Index	0x201D	
Subindex	0x05	
Type	UNSIGNED16	
Access	RO	
Stored in EEPROM	-	
Default Value	0	
Value range	0	255
Map to PDO	No	

Description:
Used for tuning Flux current regulator. Internal value is incremented each FOC cycle, when this value is higher than entered value in this parameter flux current is inverted.

Name	**Flux pulse mode num of pulses**

Index	*0x201D*	
Subindex	*0x06*	
Type	*UNSIGNED16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*0*	*255*
Map to PDO	*No*	

Description:
Number how many times current is inverted (refer to Flux pulse mode counter).

8.40 0x201E BMS Data

Name	**Flux regulator**
Index	*0x201E*
Number of entries	*-*

Name	**Number of entries**	
Index	*0x201E*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*2*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**BMS Battery Voltage**	
Index	*0x201E*	
Subindex	*0x01*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*NO*	

Desciription:
Value is read only in Volts. It’s the sum of all cell voltages, representing total battery voltage.

Name	**BMS min cell voltage**	
Index	*0x201E*	
Subindex	*0x02*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*NO*	

Description:
Minimimum cell voltage in mV.

Name	**BMS min cell voltage ID**	
Index	*0x201E*	
Subindex	*0x03*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*NO*	

Description:
Index of minimum cell voltage.

Name	**BMS max cell voltage**	
Index	*0x201E*	
Subindex	*0x04*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*NO*	

Description:
Maximum cell voltage in mV.

Name	**BMS max cell voltage ID**	
Index	*0x201E*	
Subindex	*0x05*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*NO*	

Description:
Index of maximum cell voltage.

Name	**BMS battery current**	
Index	*0x201E*	
Subindex	*0x06*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*NO*	

Description:
Value is read only in mA. Positive value when current flows into battery.

Name	**BMS unit temperature**	
Index	*0x201E*	
Subindex	*0x07*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*NO*	

Description:
Value is read only in °C.

Name	**BMS state of charge**	
Index	*0x201E*	
Subindex	*0x08*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*NO*	

Description:
Value is read only in %.

Name	**BMS charging**	
Index	*0x201E*	
Subindex	*0x09*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*NO*	

Description:
Indication if charger is present:
1 - charger connected, 0 - charger not connected.

Name	**BMS battery charging current**	
Index	*0x201E*	
Subindex	*0x0A*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*NO*	

Description:
When charging, value is charging current in mA.

Name	BMS proposed power	
Index	*0x201E*	
Subindex	*0x0B*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*NO*	

Description:
Propose power is calculated from minimum cell voltage of battery stack and external temperature. If minimum cell voltage is below value of parameter Ucell power decreasing (BMS parameters) then the BMS starts to decreasing power. Also if temperature is out of range BMS is decreasing propose power.
When BMS starts to decreasing proposed power, warning LED become active and motor current will be limited.
Value in %.

8.41 0x201F Electrical angle

Name	Electrical angle	
Index	*0x201F*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Value is in quants, representing raw rotor position, get from feedback. If stepper is enabled position is stepper position.
Used in FOC.

8.42 0x2021 Safety execute

Name	Safety execute	
Index	*0x2021*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*0*	*2*

Map to PDO	*No*

Description:
Safety procedure will be executed according entered value:

Value	Description
0	Execute always
1	Execute only when motor is stopped
2	Never execute
3..255	Reserved

8.43 0x2022 Hall sensor

Name	**Flux regulator**	
Index	*0x2022*	
Number of entries	*-*	

Name	**Number of entries**	
Index	*0x201B*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*2*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Hall sensor bit pattern**	
Index	*0x2022*	
Subindex	*0x01*	
Type	*UNSIGNED16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*NO*	

Description:
Direct value on microprocessor Hall input pins. If halls are working correctly number from 1-6 will appear, 0 and 7 are invalid sates in 120° hall configuration.

Name	**Hall noise counter**	
Index	*0x2022*	
Subindex	*0x02*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*-21474283648*	*21474283647*
Map to PDO	*NO*	

Description:
Used for identification of HALL errors. Each time when error is detected value is increased. Object is read only and it's not used in FOC.

8.44 0x2023 DC current

Name	**DC current**	
Index	*0x2023*	
Subindex	*0x00*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-21474283648*	*21474283647*
Map to PDO	*Yes*	

Description:
DC current in mA. Motor current is measured each FOC cycle, averaging is performed each velocity loop from which DC current is calculated.

8.45 0x2024 Electric power

Name	**Electric power**	
Index	*0x2024*	
Subindex	*0x00*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-21474283648*	*21474283647*
Map to PDO	*Yes*	

Description:
Calculated from DC current and DC voltage value in W.

8.46 0x2025 Motor temperature

Name	**Motor temperature**	
Index	*0x2025*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*0*	*255*
Map to PDO	*Yes*	

Description:

Actual motor temperature calculated from input pin on microprocessor value in °C. Proper sensor type must be selected, if displayed value is 200 there is an error on sensor – not connected or wrong type.

8.47 0x2026 Controller temperature

Name	**Controller temperature**	
Index	*0x2026*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*0*	*255*
Map to PDO	*Yes*	

Description:
Actual controller temperature – internal sensor, value in °C.

8.48 0x2027 Warnings

Name	**Warnings**	
Index	*0x2027*	
Subindex	*0x00*	
Type	*UNSIGNED16*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*Yes*	

Description:
If any of protections are active corresponding bit is set.

Bit	7	6	5	4	3	2	1	0
Description	x	Max velocity exceeded	Stall protection active	DC over current	DC link over voltage	DC link under voltage	Motor temperature exceeded	Controller temperature exceeded

8.49 0x2028 Motor current limit

Name	**Motor current limit**
Index	*0x2028*
Subindex	*0x00*
Type	*INTEGER16*
Access	*RO*
Stored in EEPROM	*No*
Default Value	*-*

Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Maximum motor current – output current after limitations in 1/1000 controller rated current.
If any of warning are active, highest limitation is determining this value.

8.50 0x2029 Logic power supply voltage

Name	**Logic power supply**	
Index	*0x2029*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Internaly measured logic supply voltage in V.

8.51 0x2031 Test mode command

Name	**Test mode command**	
Index	*0x2031*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*0*	*1000*
Map to PDO	*Yes*	

Description:
Different functionality according test mode of operation. For detailed description refer modes of operation chapter.

8.52 0x2032 Pulse mode

Name	**Pulse mode**
Index	*0x2032*
Number of entries	*-*

Name	**Number of entries**
Index	*0x2032*
Subindex	*0x00*

Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*2*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Pulse mode counter**
Index	*0x2032*
Subindex	*0x01*
Type	*UNSIGNED16*
Access	*RO*
Stored in EEPROM	*-*
Default Value	*0*
Value range	*0* *65535*
Map to PDO	*Yes*

Desctiption:
When internal counter value is exceeded polarity of motor current is inverted. Inverting value is selectable by Modes of operation. Used for regulator fine tuning and testing purposes.

Name	**Pulse mode number of pulses**
Index	*0x2032*
Subindex	*0x02*
Type	*UNSIGNED8*
Access	*RO*
Stored in EEPROM	*-*
Default Value	*0*
Value range	*0* *255*
Map to PDO	*Yes*

Description:
Set number of pulses when inverting current or velocity – inverting value is selectable by modes of operation. When number of pulses is exceeded error PULSE MODE FINISHED is displayed if entered value is 0, number of pulses is not limited.
Used for regulator fine tuning and testing purposes.

8.53 0x2033 Motor pole pairs

Name	**Motor pole pairs**
Index	*0x2033*
Subindex	*0x00*
Type	*UNSIGNED8*
Access	*RW*
Stored in EEPROM	*Yes*
Default Value	*6*
Value range	*0* *255*
Map to PDO	*No*

Description:
Number of motor pole pairs, must be set by user.

8.54 0x2034 Motor type

Name	Motor type	
Index	*0x2034*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*NO*	
Default Value	*0*	
Value range	*-*	*-*
Map to PDO	*Yes*	

Description:
Used for selecting motor type:
0 – PMS motor
1 – induction motor

8.55 0x2040 Feedback config

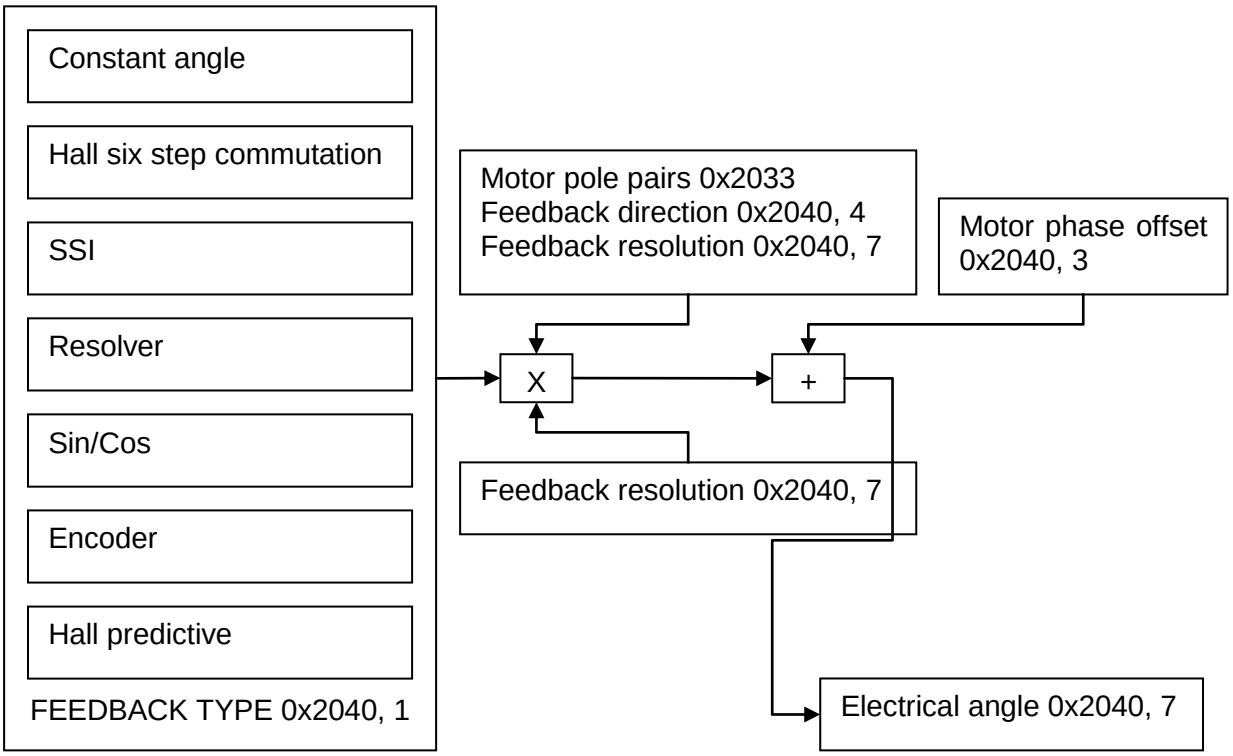
Name	Feedback config	
Index	*0x2040*	
Number of entries	*-*	

Name	Number of entries	
Index	*0x2040*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*13*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	Feedback type	
Index	*0x2040*	
Subindex	*0x01*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*3*	
Value range	*0*	*6*
Map to PDO	*No*	

Description:
Used for selection of proper feedback:

Value	0	1	2	3	4	5	6
Description	Constant angle	Hall six step	SSI	Resolver	Sin/cos	Encode	Hall predictive



Name	**Motor hase offset**	
Index	*0x2040*	
Subindex	*0x02*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*55*	
Value range	*-180*	*180*
Map to PDO	*No*	

Description:
Rotor position offset of feedback sensor in degrees. Can be measured or set over modes of operation – Auto aligning rotor position.

Name	**Feedback direction**	
Index	*0x2040*	
Subindex	*0x03*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*1*	
Value range	*0*	*1*
Map to PDO	*No*	

Description:
If 1 is entered feedback position is inverted, 0 Is not inverting feedback position.

Name	**Test stepper frequency**

Index	*0x2040*	
Subindex	*0x04*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*10*	
Value range	*1*	*100*
Map to PDO	*No*	

Description:
Used in test mode of operation if stepper is enabled this value is used in Hz.

Name	**Hall configuration**	
Index	*0x2040*	
Subindex	*0x05*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*10*	
Value range	*1*	*100*
Map to PDO	*No*	

Description:
0 – 120° hall configuration selected
1 – 60° hall configuration selected

Name	**Feedback Resolution**	
Index	*0x2040*	
Subindex	*0x06*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*4096*	
Value range	*6*	*8192*
Map to PDO	*No*	

Description:
Enter feedback resolution provided by feedback producer, if halls are used resolution is 6.

Name	**Electrical angle filter**	
Index	*0x2040*	
Subindex	*0x07*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*0*	*1024*
Map to PDO	*No*	

Description:
Value enables filter with corner frequency 1Hz, 10 Hz or 100 Hz

Value	Description

0	Filter off
1	1 Hz
2	10 Hz
3	100 Hz

Filter with 1 Hz of corner frequency seriously decrease dynamics, use only in low dynamic systems.

Name	**Motor phase offset compensation**	
Index	*0x2040*	
Subindex	*0x08*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-32768*	*32767*
Map to PDO	*No*	

Description:
Position delay in µs.

Name	**Test stepper enable**	
Index	*0x2040*	
Subindex	*0x09*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*0*	*1*
Map to PDO	*No*	

Description:
Used for enabling steeper angle instead of feedback angle.

Name	**Auto aligning rotor position current**	
Index	*0x2040*	
Subindex	*0x010*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*1000*	
Value range	*0*	*2000*
Map to PDO	*No*	

Deascription:
Value set motor current which is used in mode of operation – auto aligning rotor position. Value in 1/1000 Motor rated current.

8.56 0x2050 Maximum controller current

Name	**Maximum controller current**

Index	*0x2050*	
Subindex	*0x00*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*250000*	
Value range	*10000*	*600000*
Map to PDO	*Yes*	

Descrition:
Value is determining global maximum motor current for controller.

8.57 0x2051 Secondary current protection

Name	**Secondary current protection**	
Index	*0x2051*	
Subindex	*0x00*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*300000*	
Value range	*50000*	*650000*
Map to PDO	*Yes*	

Description:
This protection is last stage of protection. It must not be changed – it's set according HW limitations.

8.58 0x2052 Velocity control parameter

Name	**Velocity control parameter**	
Index	*0x2052*	
Number of entries	*-*	

Name	**Number of entries**	
Index	*0x2052*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*2*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Maximum velocity**	
Index	*0x2052*	
Subindex	*0x01*	
Type	*INTEGER32*	

Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*5000*	
Value range	*1*	*2147483647*
Map to PDO	*Yes*	

Description:
Value is determining maximum velocity of motor in RPM. If value is exceeded motor current will be limited.

Name	**Maximum velocity gain**	
Index	*0x2052*	
Subindex	*0x01*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*10*	
Value range	*0*	*100*
Map to PDO	*Yes*	

Description:
If Maximum velocity parameter is exceeded current will be limited according entered value in A/RPM. If entered value is 0 limitation is disabled.

8.59 0x2053 DC current limit

Name	**DC current limit**
Index	*0x2053*
Number of entries	*-*

Name	**Number of entries**	
Index	*0x2053*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*2*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Maximum DC current**	
Index	*0x2053*	
Subindex	*0x01*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*250000*	
Value range	*1000*	*250000*
Map to PDO	*Yes*	

Description:
Maximum DC current in mA. If current is exceeded limit of motor current will be active. Limitation gain is determined in Maximum DC current gain. If enter value is 0, limit is not active.

Name	**Maximum DC current gain**	
Index	*0x2053*	
Subindex	*0x01*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*10*	
Value range	*0*	*100*
Map to PDO	*Yes*	

Description:
When DC current is exceeding Maximum DC current, limit will become active.
For each Amper over the Maximum DC current, controller current will be reduced for entered value (in $1/1000_{max\_controller\_current}$).

Example:
Maximum DC current = 100000mA
Maximum DC current gain = 50
Actual DC current = 110A
Calculated Max Controller current = (110 A - 100000mA) * 50 = 500
This means that maximum controller current is reduced to 500/1000 = 0,5 max controller current

8.60 0x2054 Overvoltage limit

Name	**Overvoltage limit**	
Index	*0x2054*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*60*	
Value range	*0*	*63*
Map to PDO	*Yes*	

Description:
If DC link voltage is exceeded drive functionality is disabled and error DC link overvoltage is active. In 1/10V

8.61 0x2055 Undervoltage limitation parameter

Name	**Undervoltage limitation parameter**
Index	*0x2055*
Number of entries	-

Name	**Number of entries**
Index	*0x2055*

Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*3*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Undervoltage Limit**	
Index	*0x2055*	
Subindex	*0x01*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*40*	
Value range	*0*	*63*
Map to PDO	*No*	

Description:
DC link voltage below this value will start limiting motor current. If entered value is 0 limit is disabled. Value is in V.

Name	**Undervoltage gain**	
Index	*0x2055*	
Subindex	*0x02*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*20*	
Value range	*0*	*100*
Map to PDO	*No*	

Description:
When DC link voltage is lower than Undervoltage Limit, limit will become active.
For each Volt under the undervoltage limit maximum controller will be reduced for entered value (in $1/1000_{max\_controller\_current}$).

Example:
Undervoltage limit = 100 V
Undervoltage gain = 50
DC voltage = 90A
Calculated Max Controller current = (100 V – 90 V) * 50 = 500;
This means that maximum controller current is reduced to 500/1000 = 0,5 max controller current.

Name	**Undervoltage min voltage**	
Index	*0x2055*	
Subindex	*0x03*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*33*	

Value range	*0*	*63*
Map to PDO	*No*	

Description:
If DC link voltage will be lower that entered in V, drive mode will be disabled and under voltage error will appear.

8.62 0x2057 Motor temperature config

Name	**Motor temperature config**
Index	*0x2057*
Number of entries	*-*

Name	**Number of entries**	
Index	*0x2057*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*3*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Motor temperature sensor type**	
Index	*0x2057*	
Subindex	*0x01*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*2*	
Value range	*0*	*4*
Map to PDO	*No*	

Description:

value	Temperature sensor type
0	No sensor (displayed temperature will always be 25°C – no temperature protection for motor)
1	NTC 10k
2	KTY 84-130
3	KTY 81-210
4..255	reserved

Name	**Motor maximum temperature**
Index	*0x2057*
Subindex	*0x02*
Type	*UNSIGNED8*
Access	*RW*
Stored in EEPROM	*Yes*
Default Value	*100*

Value range	*50*	*150*
Map to PDO	*No*	

Description:
If motor temperature is exceeding entered value in °C controller current will be limited. If entered value is 0 limit is not active.

Name	**Motor maximum temperature gain**	
Index	*0x2057*	
Subindex	*0x03*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*10*	
Value range	*0*	*10*
Map to PDO	*No*	

Description:
When Motor temperature is lower than Motor maximum temperature, limit will become active.
For each °C under the Motor maximum temperature maximum controller will be reduced for entered value (in $1/1000_{max\_controller\_current}$).

Example:
Motor maximum temperature = 100°C
Motor maximum temperature gain = 50
Motor temperature = 110°C
Calculated Max Controller current = (100°C – 90°C) * 50 = 500;
This means that maximum controller current is reduced to 500/1000 = 0,5 max controller current.

8.63 0x205B Stall Config

Name	**Stall Config**
Index	*0x205B*
Number of entries	*-*

Name	**Number of entries**	
Index	*0x205B*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*3*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Stall protection time**
Index	*0x205B*
Subindex	*0x00*
Type	*UNSIGNED8*
Access	*RW*

Stored in EEPROM	*Yes*	
Default Value	*5*	
Value range	*0*	*100*
Map to PDO	*No*	

Description:
Stall is detected when motor is not commutating for time entered here in seconds and current is higher that value set in Stall protection current (0x205C, 0).

Name	**Stall protection current**	
Index	*0x205B*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*500*	
Value range	*100*	*1000*
Map to PDO	*No*	

Descritpion:
If motor current exceed entered value in 1/10A and motor is not commutating for time longer than value entered in Stall protection time (0x205B, 0) maximum controller will be reduced to entered value.

8.64 0x2060 OC out config

Name	**Local control config**
Index	*0x2060*
Number of entries	*-*

Name	**Number of entries**	
Index	*0x2060*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*2*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**OC out 1 enable**	
Index	*0x2060*	
Subindex	*0x02*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*1*	
Value range	*0*	*2*
Map to PDO	*No*	

Name	OC out 2 enable	
Index	*0x2060*	
Subindex	*0x03*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*1*	
Value range	*0*	*2*
Map to PDO	*No*	

8.65 0x2061 Brake config

Name	Brake config
Index	*0x2061*
Number of entries	*-*

Name	Number of entries	
Index	*0x2061*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*3*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	Brake nominal voltage	
Index	*0x2061*	
Subindex	*0x01*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*24*	
Value range	*6*	*48*
Map to PDO	*No*	

Description:
Entered value set nominal output voltage on OC1 and OC2 output in V if break functionality is selected with object OC1/2 out enable (0x2060, 2/3).

Name	Brake reduced voltage	
Index	*0x2061*	
Subindex	*0x02*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*12*	
Value range	*6*	*48*
Map to PDO	*No*	

Description:
Entered value set reduced voltage on OC1 and OC2 output in V if break functionality is selected with object OC1/2 out enable (0x2060, 2/3), after time defined in Brake time to reduce voltage (0x2061,3).

Name	**Brake time to reduce voltage**	
Index	*0x2061*	
Subindex	*0x03*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*-*	
Default Value	*12*	
Value range	*6*	*48*
Map to PDO	*No*	

Descritpion:
After entered value in seconds OC1/2 reduced voltage will be set on OC1/2 output if break functionality is selected with object OC1/2 out enable (0x2060, 2/3).

8.66 0x6070 Analog inputs

Name	**Analog inputs**
Index	*0x2070*
Number of entries	*-*

Name	**Number of entries**	
Index	*0x2070*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*3*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Throttle voltage**	
Index	*0x2070*	
Subindex	*0x01*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Value measured on microprocessor throttle AD input pin in mV.

Name	**Aux voltage**
Index	*0x2070*

Subindex	*0x02*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Voltage measured on microprocessor Aux AD input pin in mV.

Name	**Brake voltage**	
Index	*0x2070*	
Subindex	*0x03*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Voltage measured on microprocessor Brake AD pin in mV

8.67 0x2072 AD resolver

Name	AD resolver	
Index	*0x2072*	
Number of entries	*-*	

Name	**Number of entries**	
Index	*0x2072*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*2*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**AD resolver sin**	
Index	*0x2072*	
Subindex	*0x01*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Voltage measured on input AD pin resolver sin in quants.

Name	**AD resolver cos**	
Index	*0x2072*	
Subindex	*0x02*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description:
Voltage measured on input AD pin resolver cos in quants.

8.68 0x2076 Digital inputs

Name	**Digital Inputs**	
Index	*0x2076*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*Yes*	

Description:

The output values are given in the object Digital Values. Corresponding flag is set in object. Polarity of digital input is here already incorporated.

emDrive 400:

Din num.	Reserved										Din6	Din5	Din4	Din3	Din2	Din1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

emDrive150, EmDrive H300:

Din num.	Reserved													Din3	Din2	Din1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

8.69 0x2085 Actual FOC angle

Name	Actual FOC angle
Index	*0x2085*
Subindex	*0x00*
Type	*UNSIGNED8*
Access	*RO*

Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*Yes*	

Description:
It is angle which is actually used in FOC.

Example:
Selected feedback in Feedback Type (0x2040, 0x01) is SSI and stepper is enabled. Value in Actual FOC angle is stepper angle, but value in Electrical angle (0x201f, 0x00) is actually angle from SSI feedback.

8.70 0x2086 Velocity actual value averaged

Name	Velocity actual value averaged	
Index	*0x2086*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*0*	*65535*
Map to PDO	*Yes*	

Description:
Over 1 second averaged Velocity actual value (0x606c, 0). Used for more accurate velocity reading if resolution of feedback is low.

8.71 0x2091 Induction motor config

Name	**Induction motor config**
Index	*0x2091*
Number of entries	*-*

Name	**Number of entries**	
Index	*0x2091*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*10*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Induction motor rotor time constant**
Index	*0x2091*

Subindex	*0x01*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*70000*	
Value range	*100*	*1000000*
Map to PDO	*Yes*	

Description:

R_1 stator resistance
X_1 stator leakage reactance
X_m magnetizing reactance
R_c machine core loss resistor
X_2 rotor leakage reactance
R_2 rotor resistance

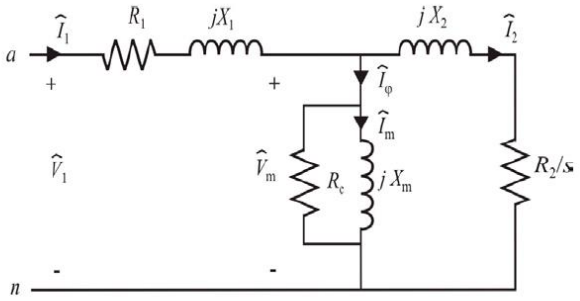


Figure 1. Per-phase equivalent circuit of an induction motor.

Rotor time constant [s] = (Lm+L2)/R2

Name	**Induction motor rated velocity**	
Index	*0x2091*	
Subindex	*0x02*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*3000*	
Value range	*0*	*2147483647*
Map to PDO	*Yes*	

Description:
Used for entering field weakening mode if motor velocity is higher than this value. See also description

Name	**Induction motor Flux weakening enabled**	
Index	*0x2091*	
Subindex	*0x03*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*1*	
Value range	*0*	*0*
Map to PDO	*Yes*	

Description:

Value	Description

0	Field weakening disabled
1	Field weakening enabled

Name	**Induction motor flux off at zero torque**	
Index	*0x2091*	
Subindex	*0x04*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*0*	*1*
Map to PDO	*Yes*	

Description:
0 – disabled
1 – enabled

If enabled, flux current will be disabled when no torque is demanded.

Name	**Induction motor V to Hz ratio**	
Index	*0x2091*	
Subindex	*0x05*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*155*	
Value range	*1*	*10000*
Map to PDO	*Yes*	

Description:
Used in motor UF control. Voltage is measured in mV.

Name	**Induction motor rated flux current**	
Index	*0x2091*	
Subindex	*0x06*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*54000*	
Value range	*10*	*54000*
Map to PDO	*Yes*	

Description:
In 1/100 motor rated current. Used for FOC control.

Name	**Induction motor parameter flux regulator P gain**
Index	*0x2091*
Subindex	*0x07*
Type	*INTEGER16*
Access	*RW*
Stored in EEPROM	*Yes*

Default Value	*1000*	
Value range	*1*	*1000*
Map to PDO	*No*	

Description:
Gain for P part of flux current PI regulator in units 1/1000.

Name	**Induction motor parameter flux regulator I gain**	
Index	*0x2091*	
Subindex	*0x08*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*200*	
Value range	*1*	*23767*
Map to PDO	*No*	

Description:
Gain for I part of flux current PI regulator in units 1/second.

8.72 0x603F Error code

Name	**Error Code**	
Index	*0x603F*	
Subindex	*0x00*	
Type	*UNSIGNED16*	
Access	*RO*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*0*	*65535*
Map to PDO	*Yes*	

Description
In this object error code is displayed if drive goes into error mode. You can find error code description in 4.2 Device Emergency Error Codes paragraph.

8.73 0x6040 Controlword

Name	**Controlword**	
Index	*0x6040*	
Subindex	*0x00*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*0*	
Value range	*0*	*65535*
Map to PDO	*Yes*	

Description

The controlword consists of bits for:

- The Table 2-2: Command coding (bits 0-3 and 7)
- The controlling of operation modes (bits 4-6 and 8) (State Transitions on page 11)

8.74 0x6041 Statusword

Name	**Statusword**	
Index	*0x6041*	
Subindex	*0x00*	
Type	*UNSIGNED16*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	-	
Value range	*0*	*65535*
Map to PDO	*Yes*	

Description

Data description

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB															**LSB**

Bits in the statusword

Bit	Description	M/O
0	Ready to switch on	M
1	Switched on	M
2	Operation enabled	M
3	Fault	M
4	Voltage enabled	M
5	Quick stop	M
6	Switch on disabled	M
7	Warning	O
8	Manufacturer specific	O
9	Remote	M
10	Target reached	M
11	Internal limit active	M
12-13	Operation mode specific	O
14-15	Manufacturer specific	O

BIT 4: VOLTAGE ENABLED
High voltage is applied to the drive when this bit is set to 1.

BIT 5: QUICK STOP
When reset, this bit indicates that the drive is reacting on a quick stop request. Bits 0, 1 and 2 of the
statusword must be set to 1 to indicate that the drive is capable to regenerate. The setting of the other
bits indicates the status of the drive (e.g. the drive is performing a quick stop as result of a reaction to
a non-fatal fault. The fault bit is set as well as bits 0, 1 and 2).

BIT 7: WARNING
A drive warning is present if bit 7 is set. The cause means no error but a state that has to be mentioned, e.g. temperature limit, job refused. The status of the drive does not change. The cause of
this warning may be found by reading the fault code parameter. The bit is set and reset by the device.

BIT 8:
This bit may be used by a drive manufacturer to implement any manufacturer specific functionality.

BIT 9: REMOTE
If bit 9 is set, then parameters may be modified via the CAN-network, and the drive executes the
content of a command message. If the bit remote is reset, then the drive is in local mode and will not
execute the command message. The drive may transmit messages containing valid actual values like
a *position actual value*, depending on the actual drive configuration. The drive will accept accesses via
SDO in local mode.

BIT 10: TARGET REACHED
If bit 10 is set by the drive, then a set-point has been reached. The set-point is dependent on the
operating mode. The description is situated in the chapter of the special mode. The change of a target
value by software alters this bit.
If *quick stop option code* is 5, 6, 7 or 8, this bit must be set, when the quick stop operation is finished
and the drive is halted.
If halt occurred and the drive has halted then this bit is set too.

BIT 11: INTERNAL LIMIT ACTIVE
This bit set by the drive indicates, that an internal limitation is active (e.g. *position range limit*).

BIT 12 AND 13:
These bits are operation mode specific. The description is situated in the chapter of the special mode
The following table gives an overview:

Mode specifics bits in the status word

Bit	Operation mode					
	vl	**pp**	**pv**	**tq**	**hm**	**ip**
12	Reserved	Set-point acknowledge	Speed	Reserved	Homing attained	Ip mode active
13	Reserved	Following error	Max slippage error	Reserved	Homing error	Reserved

BIT 14 AND 15:
These bits may be used by a drive manufacturer to implement any manufacturer specific functionality.

8.75 0x605A Quick stop option code

Name	Quick stop option code	
Index	*0x605A*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*0*	
Value range	*-32768*	*32767*
Map to PDO	*No*	

Description

The parameter quick stop option code determines what action should be taken if the Quick Stop Function is executed.

Data description

Value	Description
-32768… -1	Manufacturer specific
0	Disable drive function
1	Slow down on slow ramp slow dosn on quick stop ramp
2	Slow down on current limit
3	Slow down on the current limit
4	Slow down on the voltage limit
5	Slow down on slow down ramp and stay in QUICK STOP
6	Slow down on quick stop ramp and stay in QUICK STOP
7	Slow down on the current limit and stay in QUICK STOP
8	Slow down on the voltage limit and stay in QUICK STOP
9… 32767	Reserved

8.76 0x605b Shut down option code

Name	Shut down option code	
Index	*0x605B*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*0*	
Value range	*-32768*	*32767*
Map to PDO	*No*	

Description

The parameter shutdown option code determines what action should be taken if there is a transition

OPERATION ENABLE ⇒ READY TO SWITCH ON.

Data description

Value	Description
-32768… -1	Manufacturer specific

0	Disable drive function
1	Slow down with slow down ramp; disable of the drive function
2… 32767	Reserved

8.77 0x605C Disable operation option code

Name	**Disable operation option code**	
Index	*0x605C*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*0*	
Value range	*-32768*	*32767*
Map to PDO	*No*	

Description
The parameter disable operation option code determines what action should be taken if there is a transition

OPERATION ENABLE ⇒ SWITCHED ON.

Data description

Value	**Description**
-32768… -1	Manufacturer specific
0	Disable drive function
1	Slow down with slow down ramp and then disabling of the drive function
2… 32767	Reserved

8.78 0x605E Fault reaction option code

Name	**Fault reaction option code**	
Index	*0x605E*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*0*	
Value range	*-32768*	*32767*
Map to PDO	*No*	

Description
The parameter fault reaction option code determines what action should be taken if a fault occurs in the drive.

Data description

Value	**Description**
-32768… -1	Manufacturer specific
0	Disable drive, motor is free rotating
1	Slow down on slow down ramp
2	Slow down on quick stop ramp

3	Slow down on the current limit
4	Slow down on the voltage limit
5… 32767	reserved

8.79 0x6060 Modes of operation

Name	**Modes of operation**	
Index	*0x6060*	
Subindex	*0x00*	
Type	*INTEGER8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-128*	*127*
Map to PDO	*Yes*	

Description
Detailed description in section 5.1.1 0x6060: Modes of operation.

8.80 0x6061 Modes of operation display

Name	**Modes of operation display**	
Index	*0x6061*	
Subindex	*0x00*	
Type	*INTEGER8*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	-	
Value range	-	-
Map to PDO	*Yes*	

Description:
Detailed description in section 5.1.2 0x6061 Modes of operation display.

8.81 0x6063 Position actual value internal

Name	**Position actual value internal**	
Index	*0x6063*	
Subindex	*0x00*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-2147483648*	*2147483647*
Map to PDO	*Yes*	

Description
The actual value of the position measurement device is one of the two input values of the closed loop position control. The data unit is defined as increments. If necessary, the data unit must be transformed with the position factor defined in chapter 11 from user defined units to increments.

8.82 0x6064 Position actual value

Name	**Position actual value**	
Index	*0x6064*	
Subindex	*0x00*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*-2147483648*	*2147483647*
Map to PDO	*Yes*	

Description
The actual value of the position measurement device is one of the two input values of the closed loop position control. The data unit is defined as increments. If necessary, the data unit must be transformed with the position factor defined in chapter 11 from user defined units to increments.

8.83 0x6065 Following error window

Name	**Following error window**	
Index	*0x6065*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*1000*	
Value range	*1*	*32767*
Map to PDO	*Yes*	

Description
The following error window defines a range of tolerated position values symmetrically to the position demand value. As it is in most cases used with user defined units, a transformation into increments with the position factor is necessary. If the position actual value is out of the following error window, a following error occurs A following error might occur when
• a drive is blocked,
• unreachable profile velocity occurs, or
• at wrong closed loop coefficients.
If the value of the following error window is 232-1, the following control is switched off.

8.84 Position window

Name	Position window	
Index	*0x6067*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*100*	
Value range	*1*	*32767*
Map to PDO	*Yes*	

Description
The position window defines a symmetrical range of accepted positions relatively to the target position. If the actual value of the position encoder is within the position window, this target position is regarded as reached. As the user mostly prefers to specify the position window in his application in user defined units, the position factor of chapter 11 must be used to transform this value into increments. The target position has to be handled in the same manner as in the Trajectory Generator concerning limiting functions and transformation into internal machine units before it can be used with this function.
If the value of the position window is 232-1, the position window control is switched off.

8.85 0x6069 Velocity sensor actual value

Name	Velocity sensor actual value	
Index	*0x6069*	
Subindex	*0x00*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-2147483648*	*2147483647*
Map to PDO	*Yes*	

Description
The velocity sensor actual value describes the value read from a velocity encoder (if present) in increments (in the case of encoders) and in increments per second (in the case of tachometers and AD converters). This value is scaled to the format of the position encoder using the scaling factor velocity factor 2.

8.86 0x606C Velocity actual value

Name	Velocity actual value
Index	*0x606C*
Subindex	*0x00*
Type	*INTEGER32*
Access	*RO*
Stored in EEPROM	*No*
Default Value	*-*

Value range	*-2147483648*	*2147483647*
Map to PDO	*Yes*	

Description
The velocity actual value is represented in velocity units and is coupled to the velocity used as input to the velocity controller.

8.87 0x6071 Target torque

Name	**Target torque**	
Index	*0x6071*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*0*	
Value range	*-1000*	*1000*
Map to PDO	*Yes*	

Description
This parameter is the input value for the torque controller in profile torque mode and the value is given per thousand of rated torque.

8.88 0x6075 Motor rated current

Name	**Motor rated current**	
Index	*0x6075*	
Subindex	*0x00*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*59800*	
Value range	*-2147483648*	*2147483647*
Map to PDO	*Yes*	

Description
This value is taken from the motor name plate and is entered as multiples of milliamp. Depending on the motor and drive technology this current may be either DC, peak or rms (root-mean-square) current. All relative current data refers to this value.

8.89 0x6076 Motor rated torque

Name	**Motor rated torque**
Index	*0x6076*
Subindex	*0x00*
Type	*INTEGER32*
Access	*RW*
Stored in EEPROM	*No*

Default Value	*59800*	
Value range	*-2147483648*	*2147483647*
Map to PDO	*Yes*	

Description
This value is taken from the motor name plate and is entered as multiples of mNm. All relative torque data refer to this value. For linear motors, the object name is not changed, but the motor rated force value must be entered as multiples of mN.

8.90 0x6077 Torque actual value

Name	**Torque actual value**	
Index	*0x6077*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description
The torque actual value is actual value of torque current. Value in 1/1000 motor current.

8.91 0x6078 Motor current Actual Values

Name	Throttle config	
Index	*0x6078*	
Number of entries	*-*	

Name	**Number of entries**	
Index	*0x6078*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*2*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Current torque actual value**	
Index	*0x6078*	
Subindex	*0x01*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-65536*	*65535*
Map to PDO	*Yes*	

Description
The torque actual value is actual value of torque current. Value in 1/1000 motor current.

Name	Current torque actual value mA	
Index	*0x6078*	
Subindex	*0x01*	
Type	*INTEGER32*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*0*	
Value range	*-65536*	*65535*
Map to PDO	*Yes*	

Description
The torque actual value mA is actual value of torque current in mA.

8.92 0x6079 DC link actual voltage

Name	DC link actual voltage	
Index	*0x6079*	
Subindex	*0x00*	
Type	*INTEGER16*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*-*	
Value range	*-32768*	*32767*
Map to PDO	*Yes*	

Description
Object is read only and it's internally measured and displayed in 1/10 V.

8.93 0x607A Target position

Name	Target position	
Index	*0x607a*	
Subindex	*0x00*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*0*	
Value range	*-2147483648*	*2147483647*
Map to PDO	*Yes*	

Description
The target position is the position that the drive should move to in position profile mode using the current settings of motion control parameters such as velocity, acceleration, deceleration, motion profile type etc. The target position is given in user defined position units. It is converted to position increments using the position factor (see chapter 11). The target position will be interpreted as absolute or relative depending on the 'abs / rel' flag in the controlword.

8.94 0x607E Polarity

Name	Polarity	
Index	*0x607E*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RW*	
Stored in EEPROM	*No*	
Default Value	*1*	
Value range	*0*	*1*
Map to PDO	*No*	

Description
Demand and return value of selected modes of operation will be inverted depending of the value of the polarity flag.

Data description

Value	Description
0	Not inverted
1	inverted

8.95 0x6093 Position factor

Name	Position factor
Index	*0x6093*
Number of entries	*-*

Name	Number of entries	
Index	*0x6093*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*2*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	Position factor numerator	
Index	*0x6093*	
Subindex	*0x01*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*60*	
Value range	*0*	*2147483647*
Map to PDO	*Yes*	

Name	Position factor divisor
Index	*0x6093*
Subindex	*0x02*

Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*4096*	
Value range	*0*	*2147483647*
Map to PDO	*Yes*	

Description
The position encoder factor converts the desired position (in position units) into the internal format (in increments).

$$PositionActualValue = \frac{PositionFactorNumerator \bullet PositionActualValue}{PositionFactorDevisor}$$

8.96 0x6094 Velocity Encoder factor

Name	**Velocity Encoder factor**
Index	*0x6094*
Number of entries	*-*

Name	**Number of entries**	
Index	*0x6094*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*2*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Velocity Encoder factor numerator**	
Index	*0x6094*	
Subindex	*0x01*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*60*	
Value range	*0*	*2147483647*
Map to PDO	*Yes*	

Name	**Velocity Encoder factor divisor**	
Index	*0x6094*	
Subindex	*0x02*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*4096*	
Value range	*0*	*2147483647*
Map to PDO	*Yes*	

Description
The velocity encoder factor converts the desired velocity (in velocity units) into the internal format (in increments).

$$VelocityActualValue = \frac{VelocityFactorNumerator \bullet VelocityActualValue}{VelocityFactorDevisor}$$

8.97 0x60f6 Current control parameter

Name	**Current control parameter**
Index	*0x60f6*
Number of entries	*-*

Name	**Number of entries**	
Index	*0x60f6*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*2*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	**Current control parameter set P-gain**	
Index	*0x60f6*	
Subindex	*0x01*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*1000*	
Value range	*1*	*32767*
Map to PDO	*Yes*	

Description:
Gain for P part of current PI regulator in units 1/1000.

Name	**Current control parameter set I-gain**	
Index	*0x60f6*	
Subindex	*0x01*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*200*	
Value range	*1*	*32767*
Map to PDO	*Yes*	

Description
Gain for I part of current PI regulator in units 1/second.

8.98 0x60f9 Velocity control parameter

Name	Velocity control parameter	
Index	*0x60f6*	
Number of entries	*-*	

Name	Number of entries	
Index	*0x60f6*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*2*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	Velocity control P-gain	
Index	*0x60f6*	
Subindex	*0x01*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*5000*	
Value range	*1*	*32767*
Map to PDO	*Yes*	

Description:
Gain for P part of velocity PI regulator in units 1/1000.

Name	Velocity control I-gain	
Index	*0x60f6*	
Subindex	*0x01*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*100*	
Value range	*1*	*32767*
Map to PDO	*Yes*	

Description
Gain for I part of velocity PI regulator in units 1/second.

8.99 0x60fb Position control parameter set

Name	Position control parameter set
Index	*0x60fb*
Number of entries	*-*

Name	Number of entries	
Index	*0x60fb*	
Subindex	*0x00*	
Type	*UNSIGNED8*	
Access	*RO*	
Stored in EEPROM	*-*	
Default Value	*4*	
Value range	*0*	*255*
Map to PDO	*No*	

Name	Position control parameter set P-gain	
Index	*0x60fb*	
Subindex	*0x01*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*1000*	
Value range	*1*	*32767*
Map to PDO	*Yes*	

Description:
Gain for P part of position PI regulator in units 1/1000

Name	Position control parameter set I-gain	
Index	*0x60fb*	
Subindex	*0x02*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*1*	*32767*
Map to PDO	*Yes*	

Description:
Gain for I part of velocity PI regulator in units 1/second.

Name	Position control parameter set D-gain	
Index	*0x60fb*	
Subindex	*0x03*	
Type	*INTEGER16*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*1*	*32767*
Map to PDO	*Yes*	

8.100x60ff Target Velocity

Name	Target Velocity
Index	*0x60ff*

Subindex	*0x00*	
Type	*INTEGER32*	
Access	*RW*	
Stored in EEPROM	*Yes*	
Default Value	*0*	
Value range	*10000*	*10000*
Map to PDO	*Yes*	

Description
The target velocity is the input for the trajectory generator and the value is given in velocity units.

8.101 0x6504 Drive manufacturer

Name	Drive manufacturer	
Index	*0x60ff*	
Subindex	*0x00*	
Type	*VISIBLE_STRING*	
Access	*RO*	
Stored in EEPROM	*No*	
Default Value	*EMSISO*	
Value range	*0*	*0*
Map to PDO	*Yes*	

Description
The drive manufacturer's name.