

---

# CANopen User Manual

**MOONS'**  
*moving in better ways*

AMP & MOONS' Automation

## Contents

<b>1 Introduction.....</b>	<b>5</b>
1.1 About this manual .....	5
1.2 Reference documents.....	5
1.3 Commonly used acronyms .....	5
1.4 Revision History.....	6
<b>2 CANopen protocol.....</b>	<b>7</b>
2.1 Overview of CANopen .....	7
2.2 CANopen network topology overview.....	7
<b>3 Drive setup.....</b>	<b>8</b>
3.1 Wiring the power and motor.....	8
3.2 Wiring the CANopen connector for drives .....	8
3.3 CANopen BitRate & NodeID.....	9
3.4 Drive configuration.....	9
<b>4 CANopen communication object.....</b>	<b>10</b>
4.1 CANopen message.....	10
4.2 SDO message .....	11
4.3 PDO message .....	12
4.4 Network management (NMT) .....	13
4.5 NMT state machine.....	14
<b>5 Motion control.....</b>	<b>15</b>
5.1 Controlling the power drive system.....	15
5.1.1 General.....	15
5.1.2 Finite state automaton .....	15
5.2 Profile position mode .....	17
5.2.1 General information .....	17
5.2.2 Main controlling object.....	17
5.2.3 Functional description.....	18
5.3 Homing mode .....	22

5.3.1	General information .....	22
5.3.2	Main controlling object.....	22
5.3.3	Functional description.....	23
5.3.4	Method 1 and 2.....	23
5.3.5	Method 3 and 4.....	24
5.3.6	Method 5 and 6.....	24
5.3.7	Method 7 and 8.....	25
5.3.8	Method 9 and 10.....	25
5.3.9	Method 11 and 12.....	26
5.3.10	Method 13 and 14.....	26
5.3.11	Method 15 and 16.....	27
5.3.12	Method 17 and 18.....	27
5.3.13	Method 19 and 20.....	27
5.3.14	Method 21 and 22.....	28
5.3.15	Method 23 and 24.....	28
5.3.16	Method 25 and 26.....	29
5.3.17	Method 27 and 28.....	29
5.3.18	Method 29 and 30.....	30
5.3.19	Method 33 and 34.....	30
5.3.20	Method 35.....	30
5.3.21	Method 36.....	30
5.3.22	Method 37.....	30
5.4	Profile velocity mode.....	31
5.4.1	General information .....	31
5.4.2	Main controlling object.....	31
5.5	Profile torque mode (StepSERVO only) .....	33
5.5.1	General information .....	33
5.5.2	Main controlling object.....	33
5.5.3	Functional description.....	34
5.6	Q program mode.....	35

5.6.1	General information .....	35
5.6.2	Normal Q execution.....	35
5.6.3	Synchronous Q program execution .....	36
5.7	Global control word and status word .....	37
<b>6</b>	<b>Application function .....</b>	<b>38</b>
6.1	Communication watchdog .....	38
<b>7</b>	<b>Object dictionary .....</b>	<b>39</b>
7.1	Object description .....	39
7.2	Electronic data sheet .....	40
7.2.1	Communication profile (CiA301).....	40
7.2.2	Motion control profile (CiA402).....	55
7.2.3	Manufacturer profile.....	65
7.2.4	Manufacturer parameter for StepSERVO drives .....	72
7.2.5	Manufacturer parameter for Stepper drives.....	82
<b>8</b>	<b>Appendix A – Example for SDO message control .....</b>	<b>88</b>
<b>9</b>	<b>Contact MOONS' .....</b>	<b>91</b>

# 1 Introduction

## 1.1 About this manual

This manual describes MOONS' CANopen implementation of CiA 301 and CiA 402 specifications. It is expected that the user fully understands both CiA standards and this manual will enable the user to develop a distributed motion control system.

The intent of this manual is to highlight manufacturer specific requirements as they pertain to MOONS' drives.

## 1.2 Reference documents

MOONS' CANopen Drive Hardware Manual

CiA 301

CiA 303

CiA 402

Bosch CAN Physical Layer Specifications 2.0B

MOONS' Host Command Reference

## 1.3 Commonly used acronyms

CAN	Controller area network
CiA	CAN in automation group (Standards body)
COB-ID	Communication object identification
EDS	Electronic data sheet
NMT	Network management
OD	Object dictionary
PDS	Power drive system
PDO	Process data object
RPDO	Receive (incoming) PDO
SDO	Service data object
TPDO	Transmit (outgoing) PDO
COS	TPDO detects the change of state
FSA	Finite state automaton

## 1.4 Revision History

Date	Revision	Changes
6/05/2020	1.0	Initial release
9/11/2020	1.1	Rectification

### Drive model

This manual mainly introduces the communication description of StepSERVO and Stepper drives.

Stepper drives:

Model	Firmware
STF03-C STF05-C STF06-C STF10-C	310A and later
STF03-C-mini	1.02A and later

StepSERVO drives:

Model	Firmware
SSDC03-C SSDC05-C SSDC06-C SSDC10-C	310B and later

## 2 CANopen protocol

### 2.1 Overview of CANopen

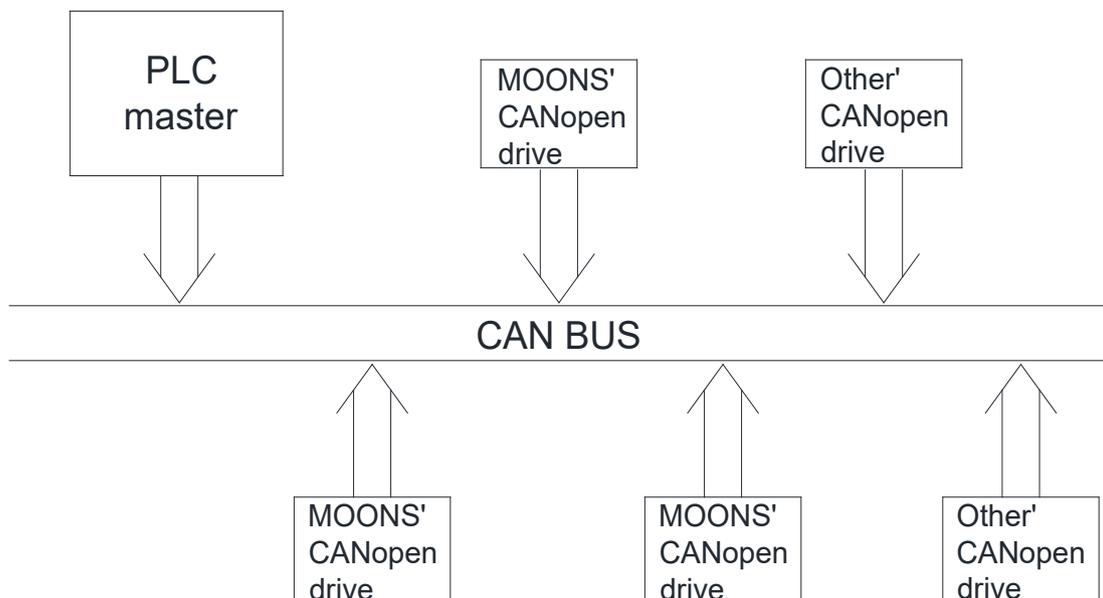
#### Reprinted by courtesy of CiA group

CANopen is a CAN-based communication system. It comprises higher-layer protocols and profile specifications. CANopen has been developed as a standardized embedded network with highly flexible configuration capabilities. It was designed originally for motion-oriented machine control systems, such as handling systems.

Today it is used in various application fields, such as medical equipment, off-road vehicles, maritime electronics, railway applications, or building automation.

### 2.2 CANopen network topology overview

MOONS' CANopen drives can be integrated into a CANopen system with other device types as shown below.



### 3 Drive setup

There are four phases to setting up MOONS' Stepper and StepSERVO drive:

- Wiring the power and motor
- Wiring the CANopen connector to the drive
- Setting the bit rate and Node ID
- Configuring the drive

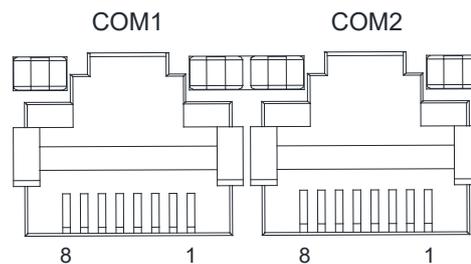
#### 3.1 Wiring the power and motor

Please refer to the hardware manual for this step. The appropriate manual can be found from our website <https://www.moonsindustries.com/>.

#### 3.2 Wiring the CANopen connector for drives

Please refer to the hardware manual of drive for the connection. Hereby we take the connection of STF drive for example.

STF drive uses a RJ45 connector, shown picture below, and conforms to CiA303 specification. The connector should be wired in a daisy-chain configuration, with a 120  $\Omega$  resistor used to terminate each end.



PIN	COM1	COM2	LINE
1	CAN_H	CAN_H	Orange/White
2	CAN_L	CAN_L	Orange
3	GND	GND	Green/White
4	RS-232_TX	NC	Blue
5	RS-232_RX	NC	Blue/White
6	NC	NC	Green
7	GND	GND	Brown/White
8	GND	GND	Brown

### 3.3 CANopen BitRate & NodeID

STF drive have three settings for CANopen communication, one for Bit Rate and two for Node-ID. The Bit Rate is configured using a 10-position switch. The Node-ID is configured using a 16-position switch to set the lower 4 bits and the upper 3 bits are configured using **Stepper Suite software**([Click here for download](#)).

Please reference the hardware manual for Node-ID switch configuration and setup. Valid ranges for the Node-ID are 01h through 7Fh. Node-ID 00h is reserved in accordance with the CiA 301 specification.

**Note:** The Node-ID and Bit Rate are captured only after a power cycle, or after a network reset command has been sent. Changing the switches while the drive is powered on will not change the Node-ID until one of those conditions has also been met.

Switch Setting	Resultant Bit Rate
0	1000Kbps
1	800Kbps
2	500Kbps
3	250Kbps
4	125Kbps
5	50Kbps
6	20Kbps
7	12.5Kps
8	Reserved
9	Reserved

### 3.4 Drive configuration

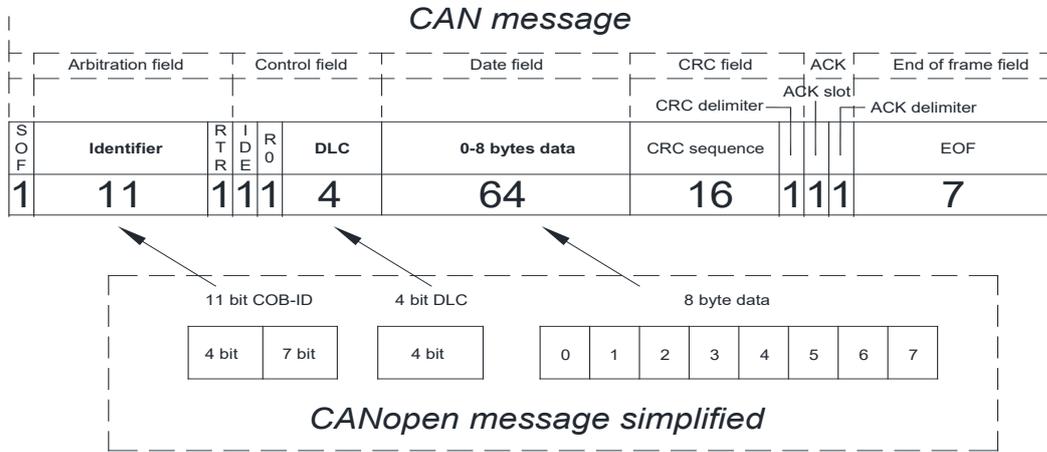
Once the CAN connector has been wired to the drive, and the Node-ID and Bit Rate have been set, the drive can be configured. Drive configuration for Stepper and StepSERVO using the **Stepper Suite** software, which can be found on MOONS' website. In all cases the drive will need to be connected to a Windows PC using the included RS-232 serial cable. Please refer to the appropriate hardware manual for details.

**Note:** When the CANopen drive is first powered on, the drive will automatically send a power-up packet over the RS-232 port. If a MOONS' application is present, it will send a response back to the drive over RS-232 and the drive will hold the CAN node in the Initialization state until the application is closed. If no response is detected, the drive will continue the normal CANopen startup procedure, the drive will power up into the Initialization state, send out a boot-up packet, move into the Pre-Operational state.

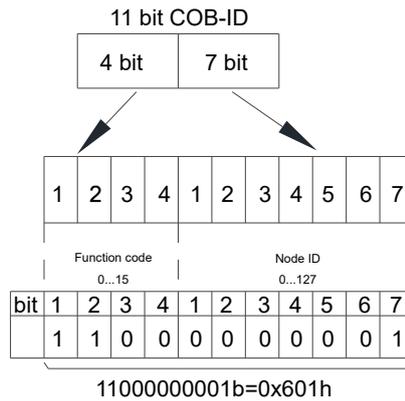
## 4 CANopen communication object

### 4.1 CANopen message

#### Simplified CANopen message over CAN message



The COB-ID interpretation:



COB	Function code	Resulting CAN-ID
NMT	0000b	0 (000h)
SYNC	0001b	128 (080h)
TIME	0010b	256 (100h)
EMCY	0001b	129 (081h) – 255 (0FFh)
PDO1(TX)	0011b	385 (181h) – 511 (1FFh)
PDO1(RX)	0100b	513 (201h) – 639 (27Fh)
PDO2(TX)	0101b	641 (281h) – 767 (2FFh)
PDO2(RX)	0110b	769 (301h) – 895 (37Fh)
PDO3(TX)	0111b	897 (381h) – 1023 (3FFh)
PDO3(RX)	1000b	1025 (401h) – 1151 (47Fh)
PDO4(TX)	1001b	1153 (481h) – 1279 (4FFh)
PDO4(RX)	1010b	1281 (501h) – 1407 (57Fh)
SDO(TX)	1011b	1409 (581h) – 1535 (5FFh)
SDO(RX)	1100b	1537 (601h) – 1663 (67Fh)
NMT error control	1110b	1793 (701h) – 1919 (77Fh)

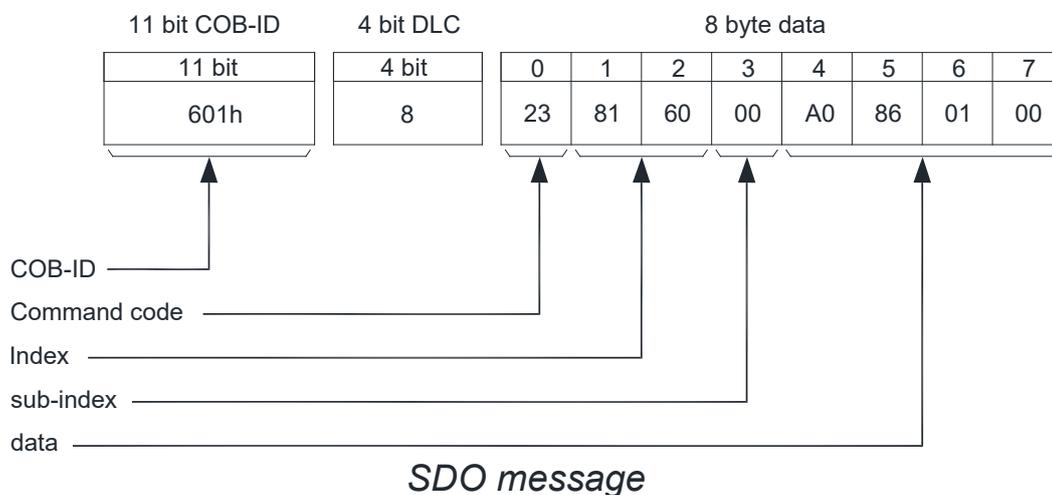
The NMT、SYNC、TIME is broadcast message, the others is peer-to-peer message.

## 4.2 SDO message

A SDO is providing direct access to object entries of a CANopen device's object dictionary. As these object entries may contain data of arbitrary size and data type. SDOs may be used to transfer multiple data sets (each containing an arbitrary large block of data) from a client to a server and vice versa. The client shall control via a multiplexer (index and sub-index of the object dictionary) which data set shall be transferred. The content of the data set is defined within the object dictionary.

PDO message											
	COB-ID		DLC	0	1	2	3	4	5	6	7
COB	4bit function code	7 bit Node ID	4bit	8 byte Data							
SDO(TX)	1011b(580h)	0...127	0...8								
SDO(RX)	1100b(600h)	0...127	0...8								

An example for SDO message interpretation:



The command code interpretation:

Command code	Description	Relationship
2Fh	Write parameter to object with 1 byte data	Master>slave
2Bh	Write parameter to object with 2 byte data	Master>slave
27h	Write parameter to object with 3 byte data	Master>slave
23h	Write parameter to object with 4 byte data	Master>slave
60h	Write response	Slave>master
40h	Read parameter from object	Master>slave
4Fh	Read response with 1byte data	Slave>master
4Bh	Read response with 2 byte data	Slave>master
47h	Read response with 3 byte data	Slave>master
43h	Read response with 4 byte data	Slave>master

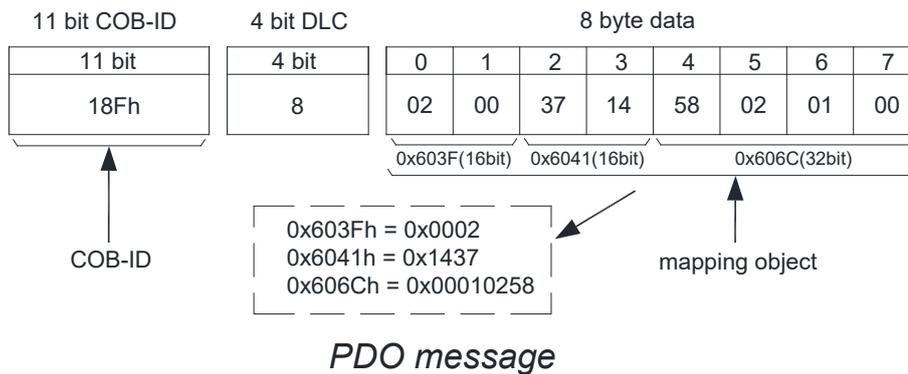
This example means the master write 4 byte data 0x000186A0 to the object 0x6081h (profile velocity) without sub-index to the slaver.

### 4.3 PDO message

The real-time data transfer is performed by means of "Process Data Objects (PDO)". The transfer of PDO is performed with no protocol overhead. The PDO correspond to objects in the object dictionary and provide the interface to the application objects. Data type and mapping of application objects into a PDO is determined by a corresponding default PDO mapping structure within the object dictionary. If variable PDO mapping is supported the number of PDO and the mapping of application objects into a PDO may be transmitted to a CANopen device during the configuration process by applying the SDO services to the corresponding objects of the object dictionary.

PDO message											
COB-ID		DLC	8 byte Data								
COB	4bit function code	7 bit Node ID	4bit	0	1	2	3	4	5	6	7
PDO1(TX)	0011b(180h)	0...127	0...8	mapping object							
PDO1(RX)	0100b(200h)	0...127	0...8								
PDO2(TX)	0101b(280h)	0...127	0...8								
PDO2(RX)	0110b(300h)	0...127	0...8								
PDO3(TX)	0111b(380h)	0...127	0...8								
PDO3(RX)	1000b(400h)	0...127	0...8								
PDO4(TX)	1001b(480h)	0...127	0...8								
PDO4(RX)	1010b(500h)	0...127	0...8								

An example for PDO message interpretation:



This example means that the slaver response the data of 603Fh、6041h、606Ch on PDO1 (TX) message with the Node ID is 0x0Fh.

## 4.4 Network management (NMT)

The network management (NMT) is CANopen device oriented and follows a master-slave structure. NMT objects are used for executing NMT services. Through NMT services, CANopen devices are initialized, started, monitored, reset or stopped. All CANopen devices are regarded as NMT slaves. An NMT slave is uniquely identified in the network by its Node-ID, a value in the range of [1...127]. NMT requires that one CANopen device in the network fulfils the function of the NMT master.

11 bit COB-ID		4 bit DLC	8 byte data		
4 bit	7 bit	4 bit	0	1	2.....7
0	0	2	NMT command	Node-ID	0

### *NMT message*

NMT command specific:

NMT command	Description
1(1h)	Start remote node
2(2h)	Stop remote node
128(80h)	Enter pre-operational
129(81h)	Reset node
130(82h)	Reset communication

The NMT state will be change when the NMT command has been implement. The state of the heartbeat producer.

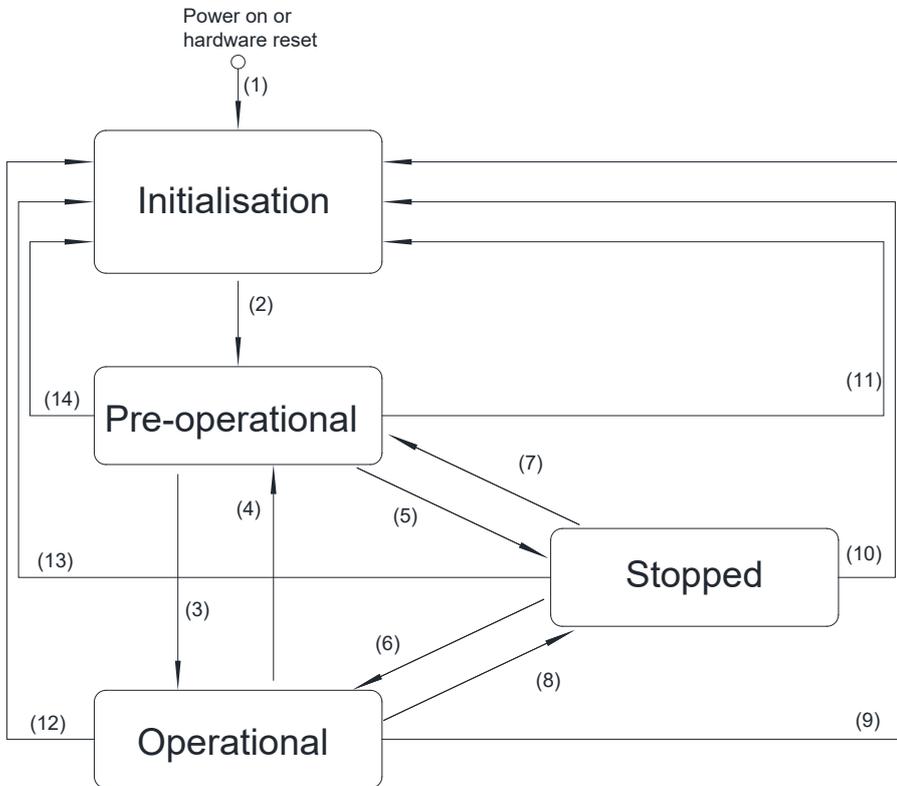
11 bit COB-ID		4 bit DLC	8 byte data	
4 bit	7 bit	4 bit	0	1.....7
1110b	Node ID	1	NMT state	0

### *Heartbeat message*

NMT state	Description
0	Boot-up
4	Stopped
5	Operational
127(7Fh)	Pre-operational

## 4.5 NMT state machine

CANopen devices enter the NMT state Pre-operational directly after finishing the CANopen devices initialization. During this NMT state CANopen device parameterization and CAN-ID-allocation via SDO (e.g. using a configuration tool) is possible. Then the CANopen devices may be switched directly into the NMT state Operational.



(1)	At power on the NMT state initialization is entered autonomously
(2)	NMT state initialization finished - enter NMT state pre-operational automatically
(3)	NMT service start remote indication or by local control
(4)(7)	NMT service enter pre-operational indication
(5)(8)	NMT service stop remote indication
(6)	NMT service start remote indication
(9)(10)(11)	NMT service reset indication
(12)(13)(14)	NMT service reset communication indication

	Pre-operational	Operational	Stopped
PDO		√	
SDO	√	√	
SYNC	√	√	
TIME	√	√	
EMCY	√	√	
Node control and error control	√	√	√

## 5 Motion control

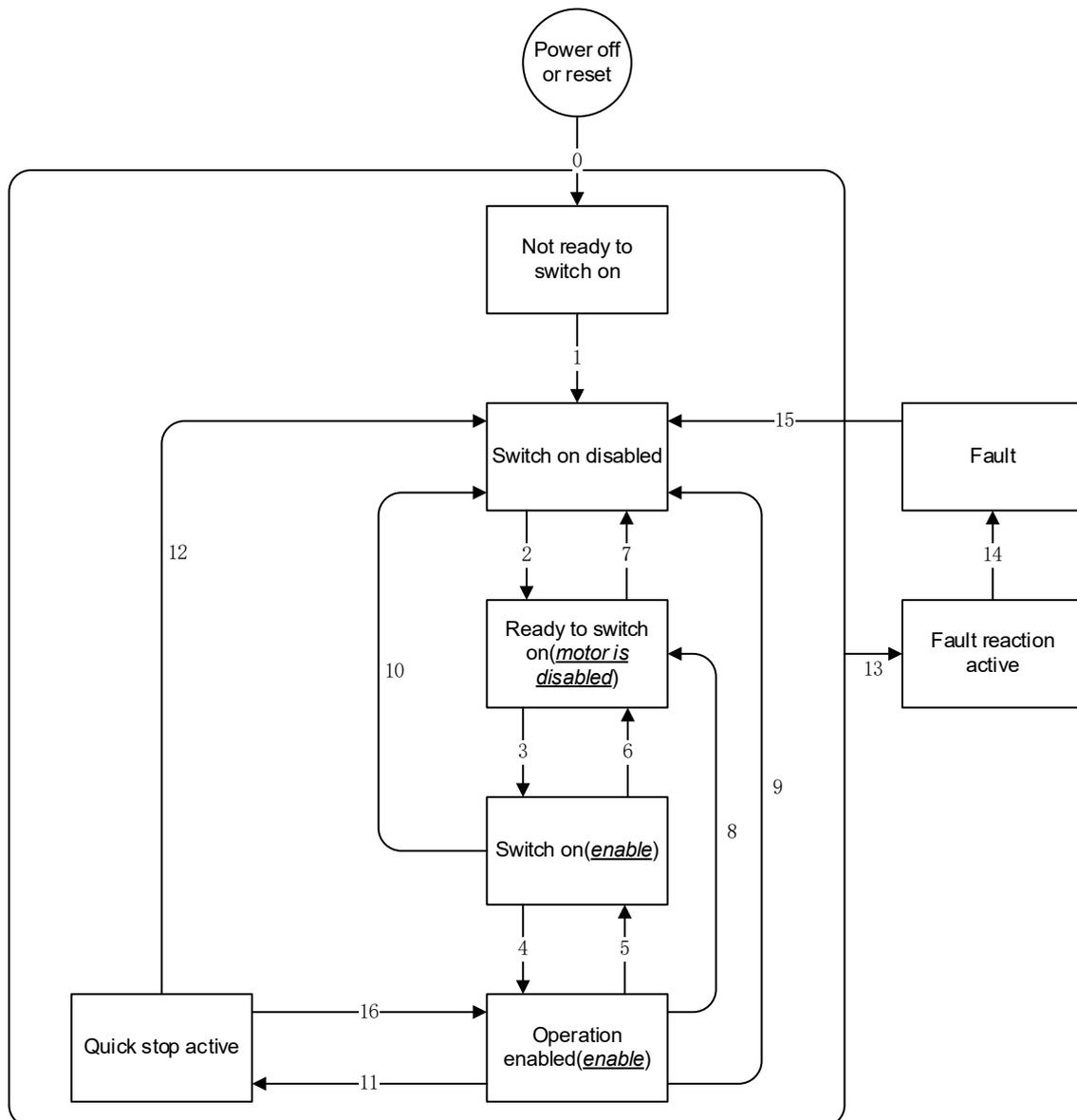
### 5.1 Controlling the power drive system

#### 5.1.1 General

The PDS FSA is an abstraction to define the behavior of a black box as a control device experiences the PDS. It defines the application behavior of the PDS. Due to the requirement that a PDS provides local control even when the communication network is not working properly, the communication FSA as defined in the communication system mapping specifications and the PDS FSA are only loosely coupled.

#### 5.1.2 Finite state automaton

This figure specifies the PDS FSA with respect to control of the power electronics as a result of user commands and internal drive faults.



Power drive system finite state automaton

Transition events and actions.

Transition	Event(s)	Action(s)
0	Automatic transition after power-on or reset application	Drive device self-test and/or self initialization shall be performed.
1	Automatic transition	Communication shall be activated.
2	Shutdown command from control device or local signal	None
3	Switch on command received from control device or local signal	The high-level power shall be switched on, if possible.
4	Enable operation command received from control device or local signal	The drive function shall be enabled and all internal set-points cleared.
5	Disable operation command received from control device or local signal	The drive function shall be disabled.
6	Shutdown command received from control device or local signal	The high-level power shall be switched off, if possible.
7	Quick stop or disable voltage command from control device or local signal	None
8	Shutdown command from control device or local signal	The drive function shall be disabled, and the high-level power shall be switched off, if possible.
9	Disable voltage command from control device or local signal	The drive function shall be disabled, and the high-level power shall be switched off, if possible.
10	Disable voltage or quick stop command from control device or local signal	The high-level power shall be switched off, if possible.
11	Quick stop command from control device or local signal	The quick stop function shall be started.
12	Automatic transition when the quick stop function is completed and quick stop option code is 1, 2, 3 or 4, or disable voltage command received from control device (depends on the quick stop option code)	The drive function shall be disabled, and the high-level power shall be switched off, if possible.
13	Fault signal (see also /CiA402-3/)	The configured fault reaction function shall be executed.
14	Automatic transition	The drive function shall be disabled; the high-level power shall be switched off, if possible.
15	Fault reset command from control device or local signal	A reset of the fault condition is carried out, if no fault exists currently on the drive device; after leaving the fault state, the fault reset bit in the controlword shall be cleared by the control device.
16	Enable operation command from control device, if the quick stop option code is 5, 6, 7, or 8	The drive function shall be enabled.

The command codes with object at 0x6040h.

Command	Bits of the controlword					Transitions
	bit7	bit3	bit2	bit1	bit0	
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
Quick stop	0	x	x	0	x	7,9,10,12
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4,16
Fault reset	↑	x	x	x	x	15

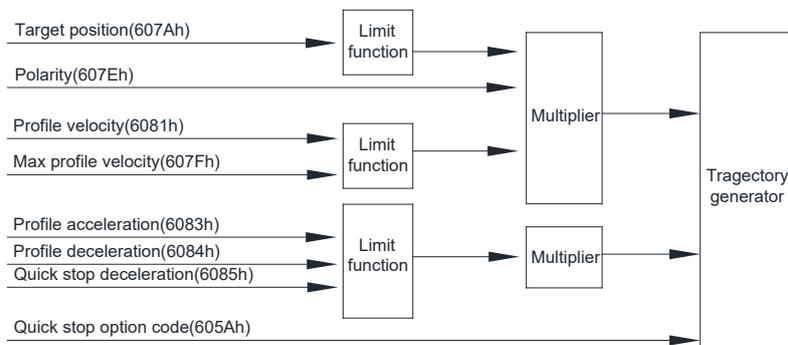
The state coding in PDS FSA.

Statusword(6041h)	PDS FSA state
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switch on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

## 5.2 Profile position mode

### 5.2.1 General information

Profile position mode is a point to point operating mode using set points which consist of velocity, acceleration, deceleration, and target position. Once all these parameters have been set, the drive buffers the commands and begins executing the set point. When using a set of set points method, a new set point can be sent to the drive while a previously sent set point is still executing.



Profile position mode(PP mode)

### 5.2.2 Main controlling object

Index	Name	Type	Access	Mapping
0x6040	Controlword	UINT16	WO	YES
0x6041	Statusword	UINT16	RO	YES
0x6060	Modes of operation	INT8	WO	YES
0x6061	Modes of operation display	INT8	RO	YES
0x607A	Target position	INT32	RW	YES
0x6081	Profile velocity	UINT32	RW	YES
0x6083	Profile acceleration	UINT32	RW	YES
0x6084	Profile deceleration	UINT32	RW	YES
0x6085	Quick stop deceleration	UINT32	RW	YES
0x605A	Quick stop code	INT16	RW	NO

#### Controlword of profile position mode (6040h)

15	10	9	8	7	6	5	4	3	0
***	Change on set point	Halt	***	Abs/Rel	Change set immediately	New set point	***		

\*\*\*: See object description

Bit	Name	Value	Description
4	New set point	0	Toggle this bit from 0->1 to clock in a new set point
		1	
5	Change set point immediately	0	Positioning shall be completed (target reached) before the next one gets started
		1	Next positioning shall be started immediately
6	Abs/Rel	0	Target position shall be an absolute value
		1	Target position shall be an relative value
8	Halt	0	positioning shall be executed or continued
		1	Axis shall be stopped
9	Change of set point	0	The previous set-point will be completed and the motor will come to rest before a new set point is processed
		1	The motor will continue at the speed commanded by the previous set point until it has reached the position commanded by the previous set point, then transition to the speed of the new set point

Statusword of profile position mode (6041h).

15	14	13	12	11	10	9	0
****	Following error	Set point acknowledge	****	Target reached	****		

\*\*\*\*: See object description

Bit	Name	Value	Description
10	Target reached	0	Halt (bit 8 in controlword) = 0: Target position not reached
			Halt (bit 8 in controlword) = 1: Axis decelerates
		1	Halt (bit 8 in controlword) = 0: Target position reached
			Halt (bit 8 in controlword) = 1: Velocity of axis is 0
12	Set point ACK	0	Previous set-point already processed, waiting for new set-point
		1	Previous set-point still in process, set-point overwriting shall be accepted
13	Following error	0	No following error
		1	Following error

### 5.2.3 Functional description

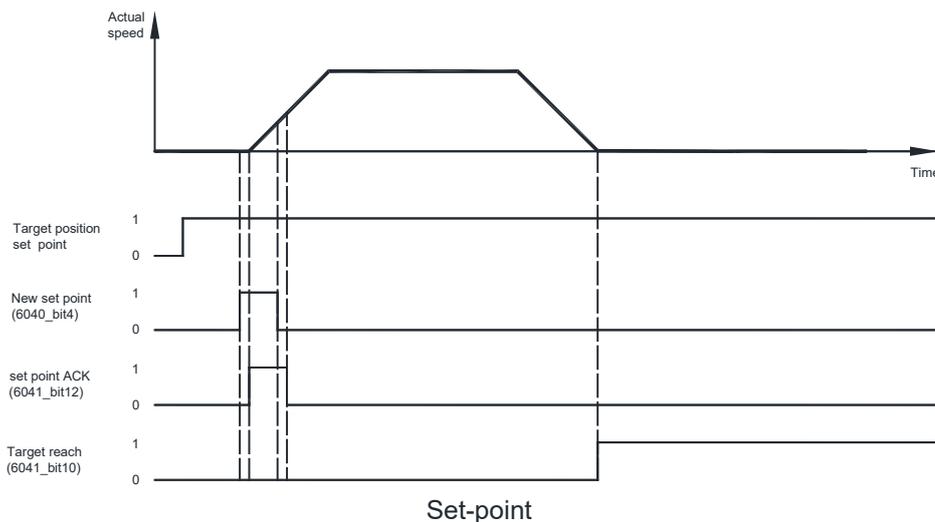
#### General

The setting of set-points is controlled by the timing of the new set-point bit and the change set immediately bit in the controlword as well as the set-point acknowledge bit in the statusword.

If the change set immediately bit of the controlword is set to 1, a single set-point is expected by the drive device. If the change set immediately bit of the controlword is set to 0, a set of set-points is expected by the drive device.

#### Set point

After a set-point is applied to the drive device, the control device signals that the set-point is valid by a rising edge of the new set-point bit in the controlword. The drive device sets the set-point acknowledge bit in the statusword to 1, and afterwards, the drive device signals with the set-point acknowledge bit set to 0 its ability to accept new set-points.



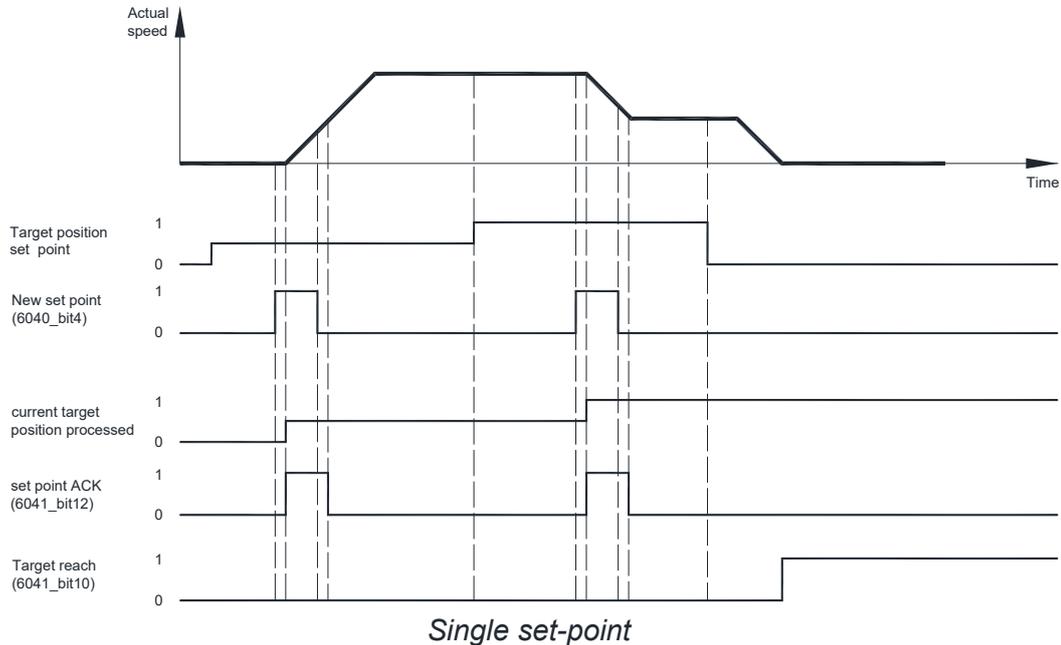
**The data of controlling object:**

Event	Set parameter	
Enable motor power	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + enable operation	0x6040=15(0Fh)
Set mode of operation	Profile position mode	0x6060=1(01h)
Set motion parameters	Distance	0x607A=100000(0186A0h)
	Velocity	0x6081=20000(4E20h)
	Acceleration	0x6083=50000(C350h)
	Deceleration	0x6084=50000(C350h)
Set point absolute	New set point	0x6040=31(1Fh)
	Clear new set point	0x6040=15(0Fh)
Set point relative	New set point	0x6040=95(5Fh)
	Clear new set point	0x6040=79(4Fh)

If one set-point is still in progress and a new one is validated, two methods of handling are supported: single set-point (change set immediately bit of controlword is 1) and set of set points (change set immediately bit of controlword is 0).

**Single set-point**

When a set-point is in progress and a new set-point is validated by the new set-point (bit 4) in the controlword, the new set-point shall be processed immediately.

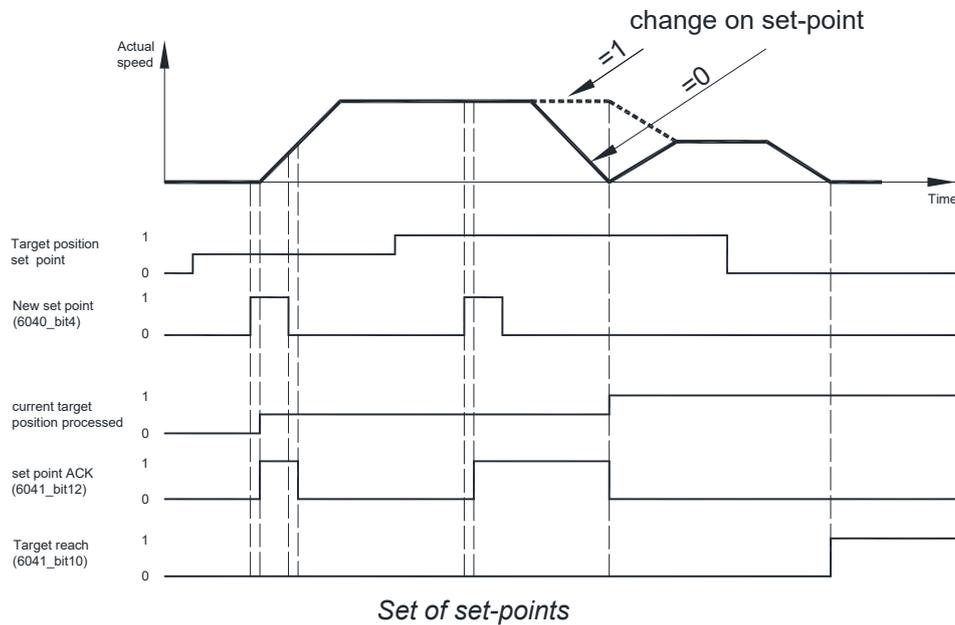


**The data of controlling object:**

	Event	Set parameter
Enable motor power	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + enable operation	0x6040=15(0Fh)
Set mode of operation	Profile position mode	0x6060=1(01h)
Set motion parameters	Acceleration	0x6083=50000(C350h)
	Deceleration	0x6084=50000(C350h)
Single set point	First part of velocity	0x6081=30000(7530h)
	First part of distance	0x607A=200000(030D40h)
	New set point	0x6040=639(27Fh)
	Clear new set point	0x6040=623(26Fh)
	Second part of velocity	0x6081=20000(4E20h)
	Second part of distance	0x607A=100000(0186A0h)
	New set point	0x6040=639(27Fh)
	Clear new set point	0x6040=623(26Fh)

**Set of set-points**

When a set-point is in progress and a new set-point is validated by the new set-point (bit 4) in the controlword, the new set-point shall be processed only after the previous has been reached.



**The data of controlling object:**

Event	Set parameter	
Enable motor power	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + enable operation	0x6040=15(0Fh)
Set mode of operation	Profile Torque mode	0x6060=1(01h)
Set motion parameters	Acceleration	0x6083=50000(C350h)
	Deceleration	0x6084=50000(C350h))
Set of set-points with change on set-point=0	First part of velocity	0x6081=30000(7530h)
	First part of distance	0x607A=400000(061A80h)
	New set point	0x6040=95(5Fh)
	Clear new set point	0x6040=79(4Fh)
	Second part of velocity	0x6081=20000(4E20h)
	Second part of distance	0x607A=300000(7530h)
	New set point	0x6040=95(5Fh)
	Clear new set point	0x6040=79(4Fh)
Set of set-points with change on set-point=1	First part of velocity	0x6081=30000(7530h)
	First part of distance	0x607A=400000(061A80h)
	New set point	0x6040=607(25Fh)
	Clear new set point	0x6040=591(24Fh)
	Second part of velocity	0x6081=20000(4E20h)
	Second part of distance	0x607A=300000(0493E0h)
	New set point	0x6040=607(25Fh)
	Clear new set point	0x6040=591(24Fh)

**NOTE:** MOONS' CANopen drive can be set up with two set-points, if the bit 12 of statusword is 1, then the buffer is full and another set-point will be ignored.

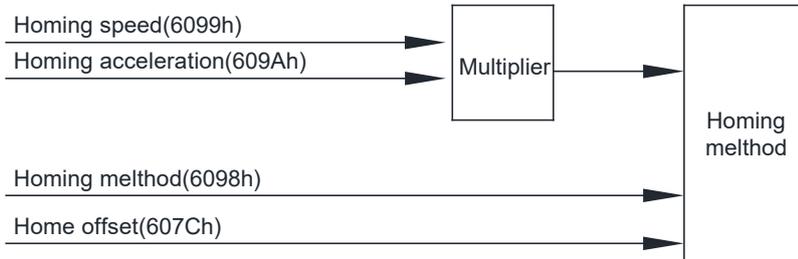
**Stop the motor with halt (0x6040\_bit8)**

If a drive is processing a set-point, the halt bit (bit 8 of the controlword) can be used to stop the motor and keep it in position control. After releasing the halt bit the processing of the actual set-point is continued.

## 5.3 Homing mode

### 5.3.1 General information

This clause describes the method by which a drive seeks the home position (also called, the datum, reference point or zero point). There are various methods of achieving this using limit switches at the ends of travel or a home switch (zero point switch) in mid-travel, most of the methods also use the index (zero) pulse train from an incremental encoder.



*Homing mode(HM mode)*

### 5.3.2 Main controlling object

Index	Name	Type	Access	Mapping
0x6040	Controlword	UINT16	WO	YES
0x6041	Statusword	UINT16	RO	YES
0x6060	Mode of operation	INT8	WO	YES
0x6098	Home method	INT8	RW	NO
0x6099	Homing speed	ARRAY	-	YES
0x609A	Homing acceleration	INT32	RW	YES
0x2001	Home switch	INT8	RW	YES
0x607C	Homing offset	INT32	RW	YES

#### Controlword of homing mode



\*\*\*: See object description

Bit	Name	Value	Description
4	Homing operation start	0	Do not start homing procedure
		1	Start or continue homing procedure
8	Halt	0	Enable bit4
		1	Stop axis according to halt option code (0x605D)

#### Statusword of homing mode



\*\*\*: See object description

Bit13	Bit12	Bit10	definition
0	0	0	Homing procedure is in progress
0	0	1	Homing procedure is interrupted or not started
0	1	0	Homing is attained, but target is not reached
0	1	1	Homing is procedure is completed successfully
1	0	0	Homing error occurred, velocity is not 0
1	0	1	Homing error occurred, velocity is 0
1	1	X	Reserved

### 5.3.3 Functional description

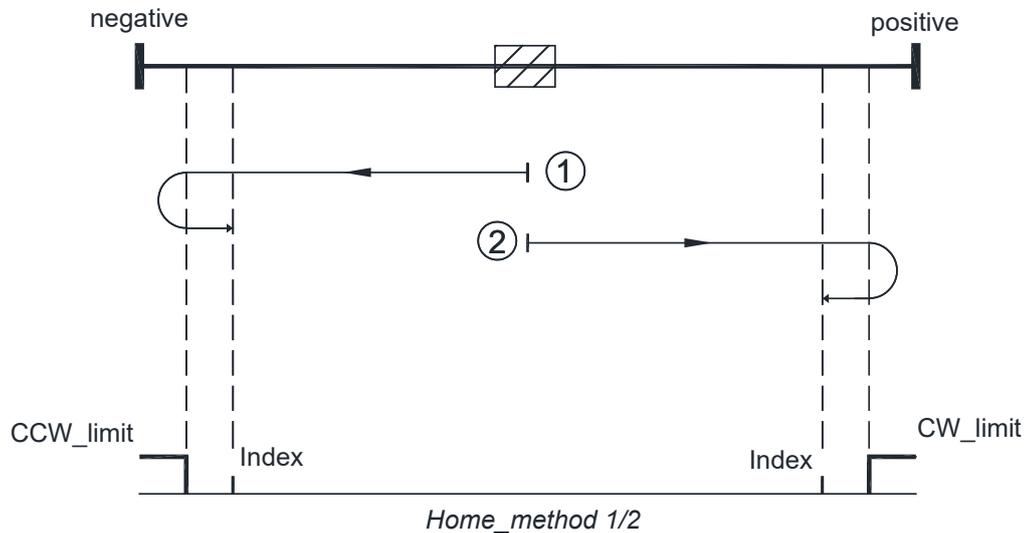
The homing modes are working on logical values of the limit and homing switches (object 60FD<sub>h</sub>).

**The data of controlling object:**

Event	Set parameter	
Enable motor power	Shut down	0x6040=06(06h)
	Switch on	0x6040=07(07h)
	Switch on+ enable operation	0x6040=15(0Fh)
Set mode of operation	Homing mode	0x6060=06(06h)
Set home method	Home method=13	0x6098=13(0Dh)
Set motion parameters	Homing acceleration	0x609A=200000(030D40h)
	Velocity for switch	0x6099_sub1=20000(4E20h)
	Velocity for index	0x6099_sub2=2000(07D0h)
	Homing offset	0x607C=100000(0186A0h)
	Homing switch	0x2001=5(05h)
Homing mode	Homing start	0x6040=31(1Fh)
	Homing stop	0x6040=287(011Fh)

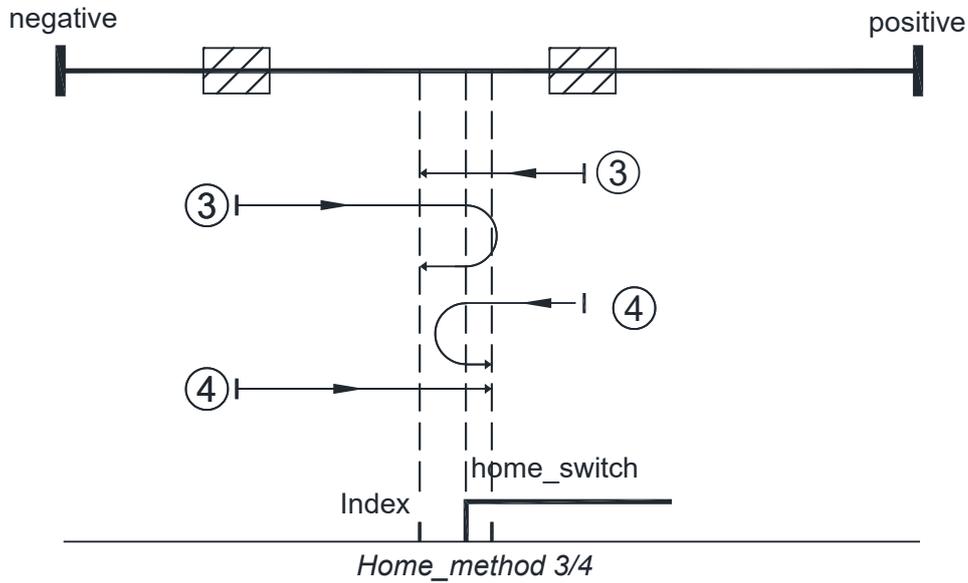
### 5.3.4 Method 1 and 2

The initial direction of movement shall be leftward (method 1) or rightward (method 2) if the limit switch is inactive. The position of home shall be at the first index pulse to the limit switch becomes inactive.



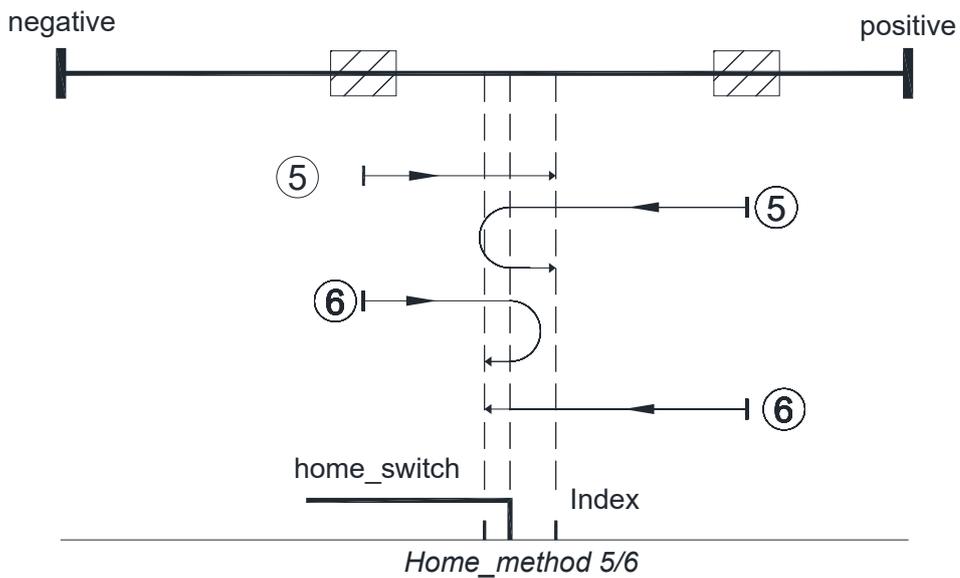
### 5.3.5 Method 3 and 4

The initial direction of movement shall be dependent on the state of the home switch. The home position shall be at the index pulse to either to the left or the right of the point where the home switch changes state. If the initial position is situated so that the direction of movement shall reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.



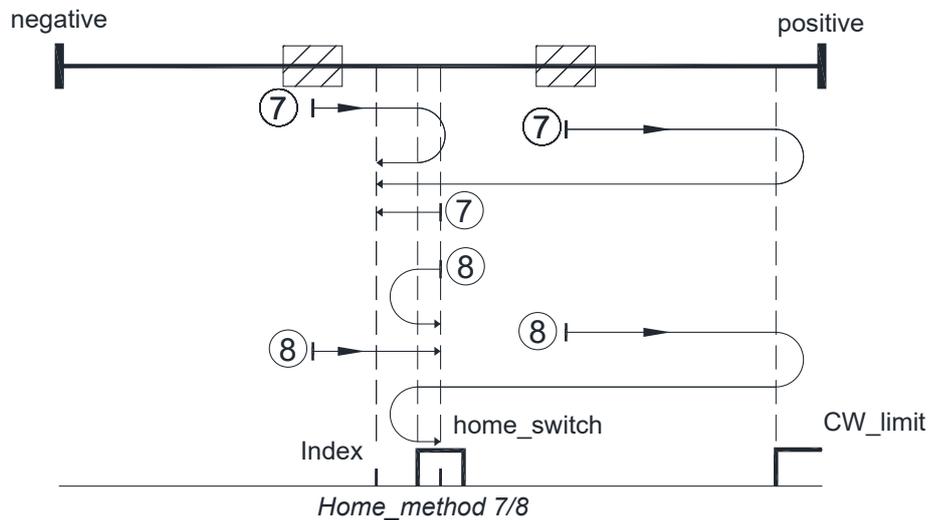
### 5.3.6 Method 5 and 6

The initial direction of movement shall be dependent on the state of the home switch. The home position shall be at the index pulse to either to the left or the right of the point where the home switch changes state. If the initial position is situated so that the direction of movement shall reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.



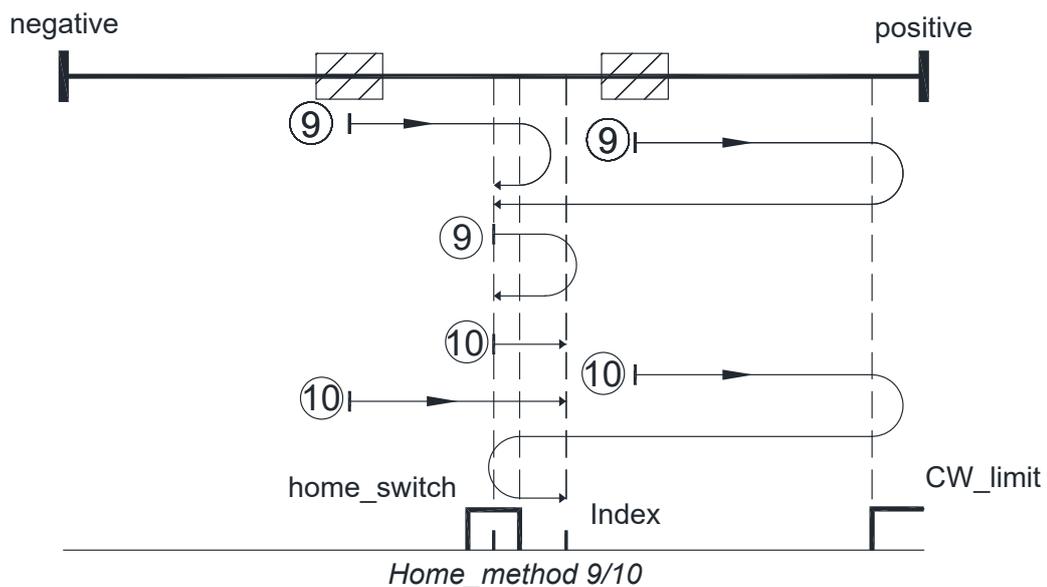
### 5.3.7 Method 7 and 8

The initial direction of movement shall be rightward if the positive limit switch is inactive. With the method 7, the home position shall be at the first index pulse to the left side of the home switch which the changes status is on falling edge. The home position shall be at the first index pulse to the left side of home switch which the changes status is on rising edge that moved from negative to positive on method 8 .



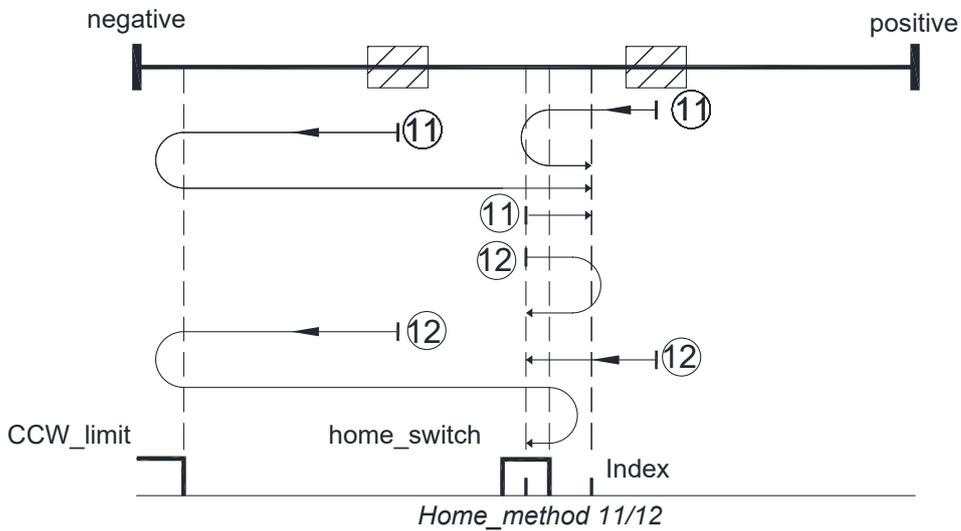
### 5.3.8 Method 9 and 10

The initial direction of movement shall be rightward if the positive limit switch is inactive. With the method 9, the home position shall be at the first index pulse to the right side of the home switch which the changes status is on rising edge that moved from positive to negative. The home position shall be at the first index pulse to the right side of home switch which the changes status is on falling edge with method 10.



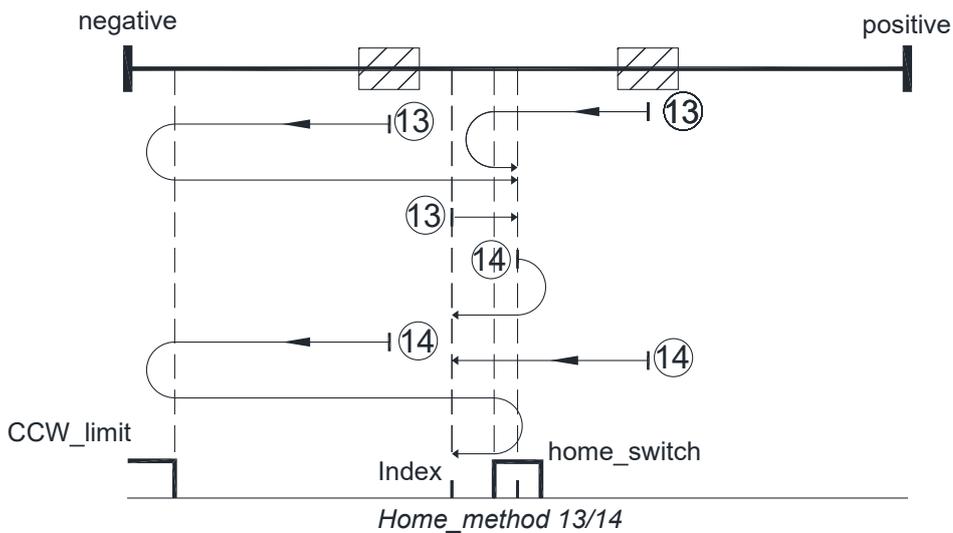
### 5.3.9 Method 11 and 12

The initial direction of movement shall be leftward if the negative limit switch is inactive. With the method 11, the home position shall be at the first index pulse to the right side of the home switch which the changes status is on falling edge. The home position shall be at the first index pulse on the right side of home switch which the changes status is on rising edge with method 12.



### 5.3.10 Method 13 and 14

The initial direction of movement shall be leftward if the negative limit switch is inactive. With the method 13, the home position shall be at the first index pulse to the left side of the home switch which the changes status is on falling edge. The home position shall be at the first index pulse to the left side of home switch which the changes status is on rising edge that moved from negative to positive on method 14.

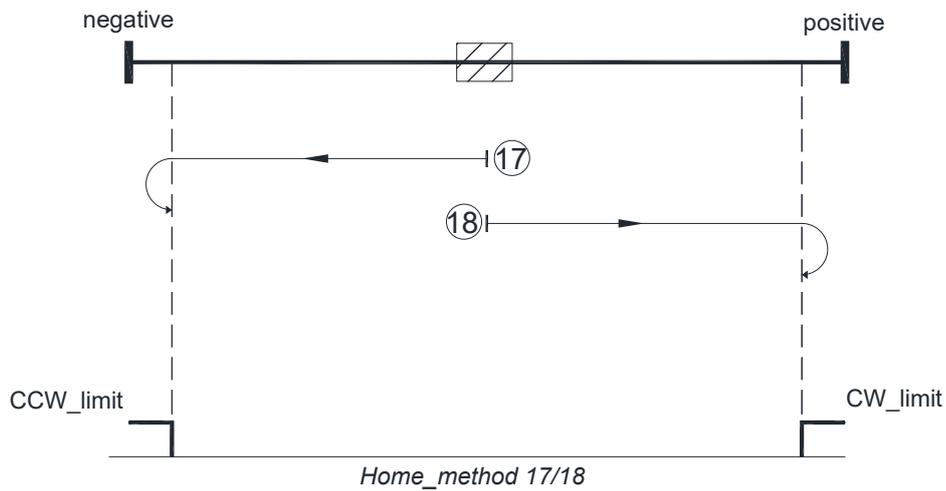


### 5.3.11 Method 15 and 16

Reserved

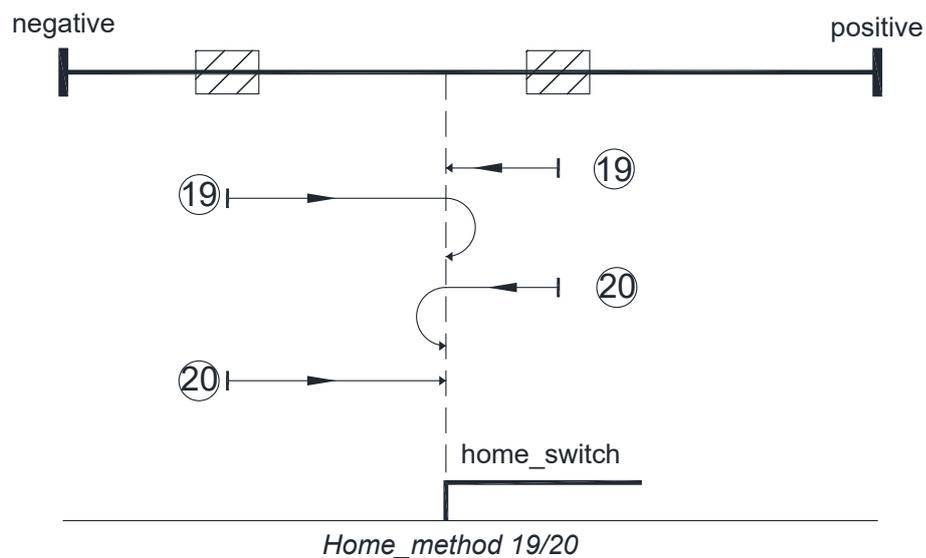
### 5.3.12 Method 17 and 18

The method 17 and 18 are similar to methods 1 and 2 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions.



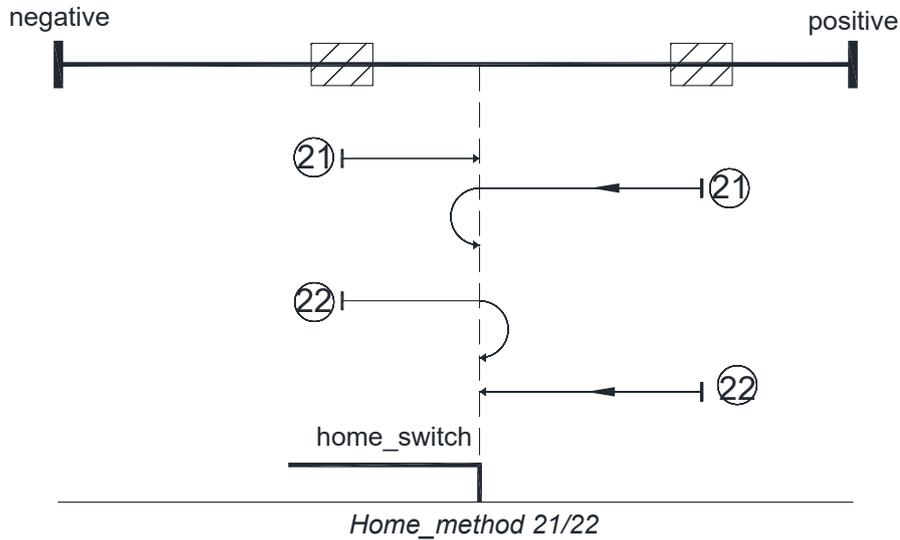
### 5.3.13 Method 19 and 20

The method 19 and 20 are similar to methods 3 and 4 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions.



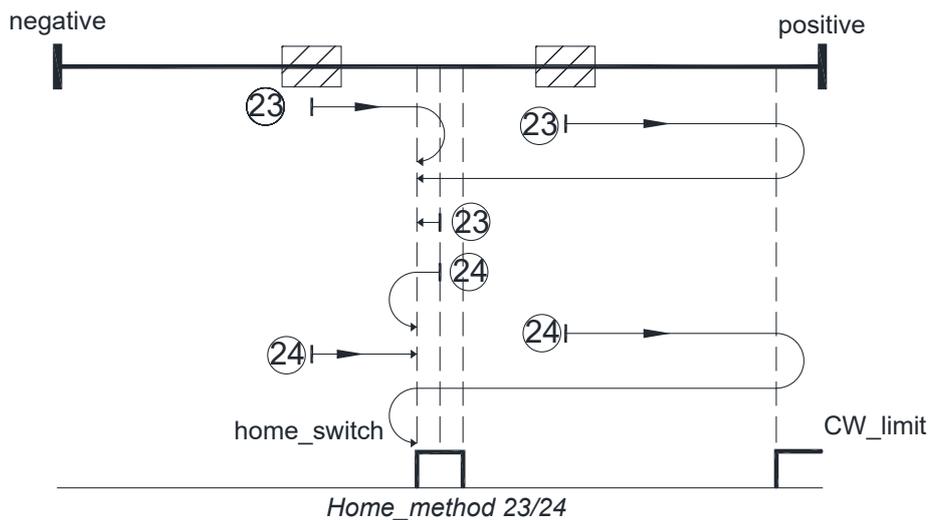
### 5.3.14 Method 21 and 22

The method 21 and 22 are similar to methods 5 and 6 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions.



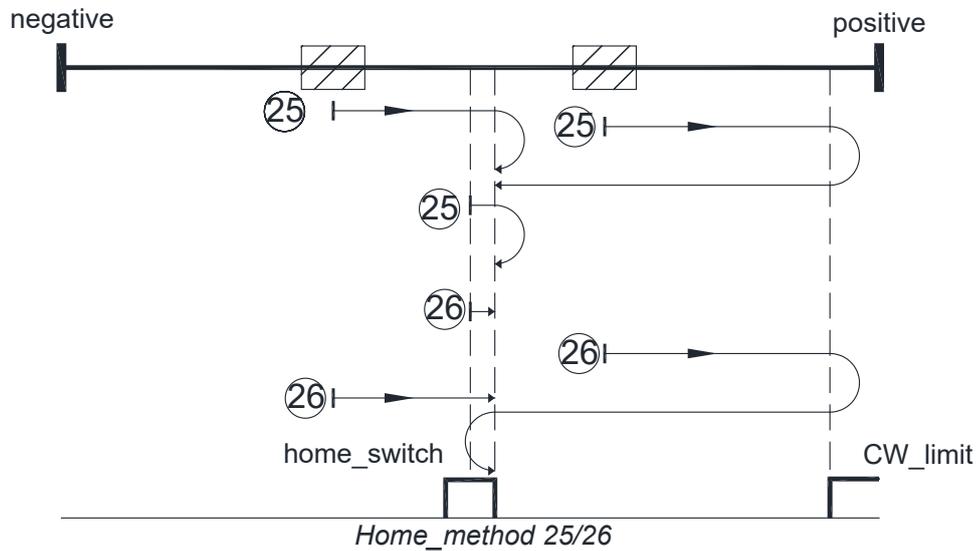
### 5.3.15 Method 23 and 24

The method 23 and 24 are similar to methods 7 and 8 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions.



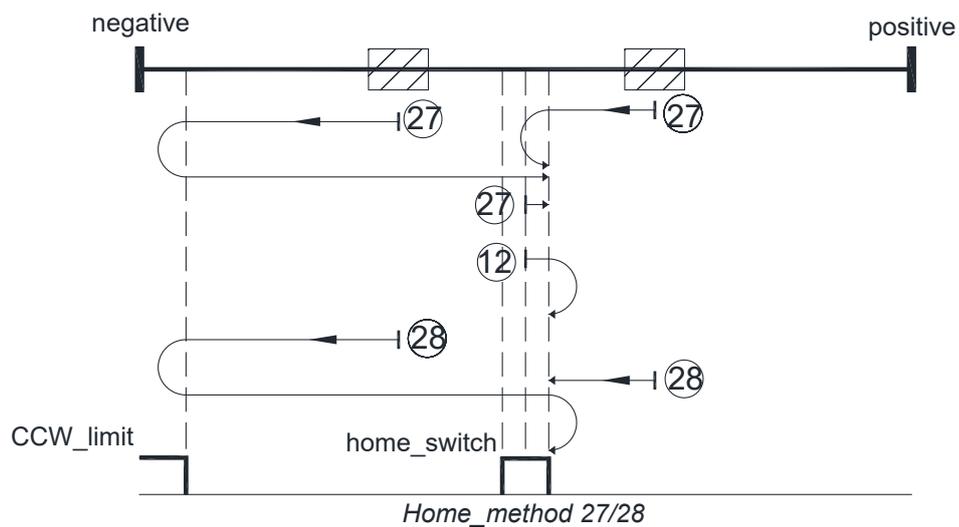
### 5.3.16 Method 25 and 26

The method 25 and 26 are similar to methods 9 and 10 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions.



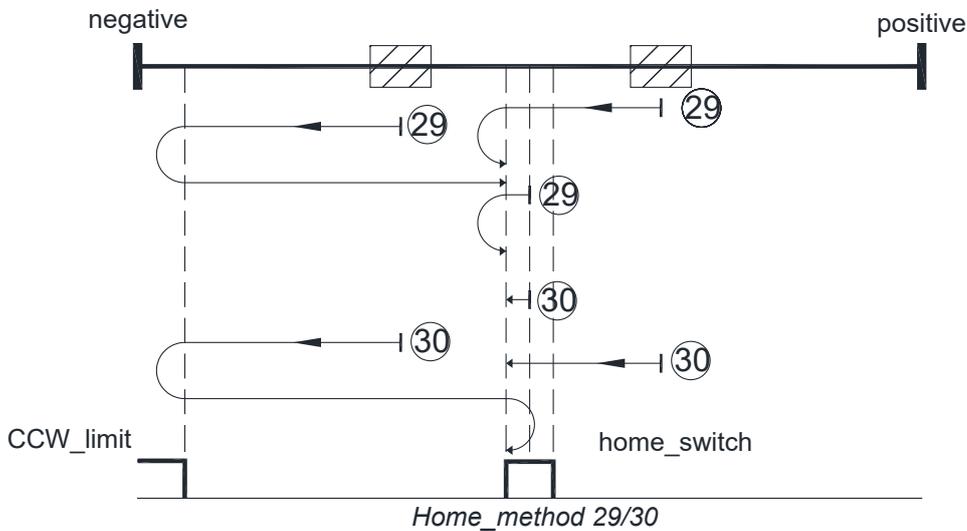
### 5.3.17 Method 27 and 28

The method 27 and 28 are similar to methods 11 and 12 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions.



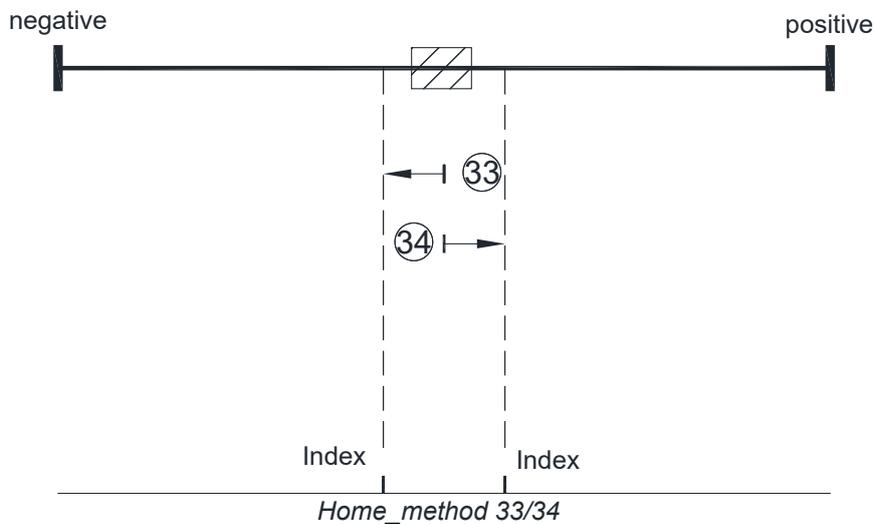
### 5.3.18 Method 29 and 30

The method 29 and 30 are similar to methods 13 and 14 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions.



### 5.3.19 Method 33 and 34

Using these methods, the direction of homing is negative or positive respectively. The home position shall be at the index pulse found in the selected direction



### 5.3.20 Method 35

In this method, the current position shall be taken to be the home position. This method does not require the drive device to be in operational enabled state.

### 5.3.21 Method 36

Reserved for compatibility reasons

### 5.3.22 Method 37

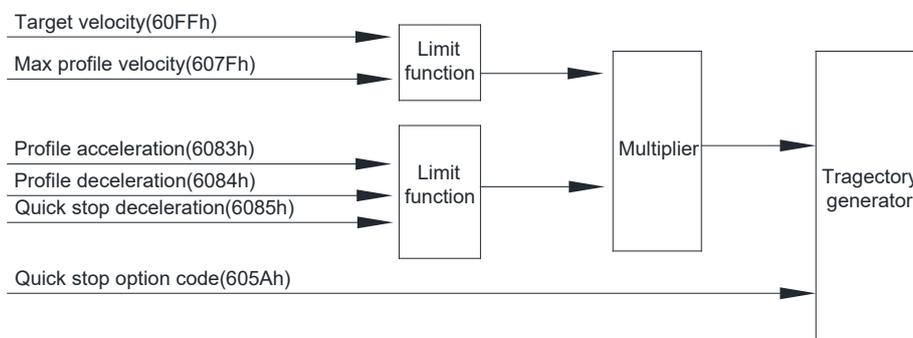
In this method, the position sensor information (converted in user-defined position units) shall be taken to be the home position. This method does not require the drive device to be in operation enabled state. At the home position (i.e. after the homing process) the position actual value (6064h) is calculated as follows:

$$\text{Position actual value (6064h)} = \text{Home offset (607Ch)}$$

## 5.4 Profile velocity mode

### 5.4.1 General information

Profile Velocity Mode is a relatively simple operating mode. Once the velocity, acceleration, and deceleration are set, the drive will either command the motor to accelerate to the running velocity according to the acceleration parameter, or to halt movement according to the deceleration parameter.



*Profile velocity mode(PV mode)*

### 5.4.2 Main controlling object

Index	Name	Type	Access	Mapping
0x6040	Controlword	UINT16	WO	YES
0x6041	Statusword	UINT16	RO	YES
0x6060	Modes of operation	INT8	WO	YES
0x6061	Modes of operation display	INT8	RO	YES
0x60FF	Target velocity	UINT32	RW	YES
0x6083	Profile acceleration	UINT32	RW	YES
0x6084	Profile deceleration	UINT32	RW	YES

#### Controlword of profile velocity mode

15	9	8	7	6	5	4	3	0
***		Halt	****		***	***		***

\*\*\*: See object description

Bit	Name	Value	Description
8	Halt	0	The motion shall be executed or continued
		1	Axis shall be stopped according to the halt option code (0x605D)

#### Statusword of profile velocity mode

15	14	13	12	11	10	9	0
***		Max slippage error	Speed	***	Target reached		***

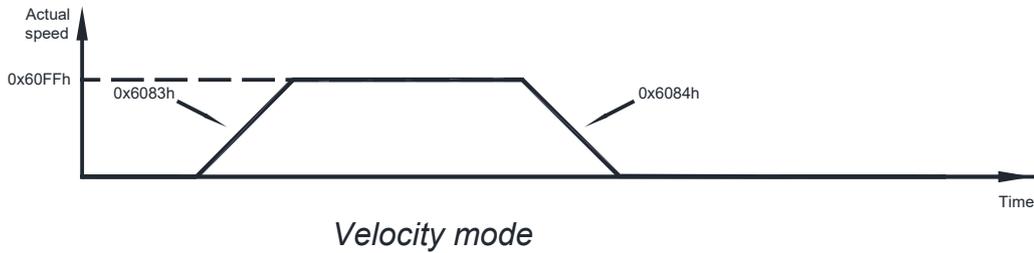
\*\*\*: See object description

Bit	Name	Value	Description
10	Target reached	0	Halt (bit 8 in controlword) = 0: Target not reached Halt (bit 8 in controlword) = 1: Axis decelerates
		1	Halt (bit 8 in controlword) = 0: Target reached Halt (bit 8 in controlword) = 1: Velocity of axis is 0
12	Speed	0	Speed is not equal 0
		1	Speed is equal 0
13	Max slippage error	0	Maximum slippage not reached
		1	Maximum slippage reached

### 5.4.3 Functional description

Profile velocity mode is according to the specified velocity, acceleration and deceleration value for moving. And to stopped with halt (6040\_bit8) control.

1. Target velocity (60FFh)
2. Profile acceleration (6083h)
3. Profile deceleration (6084h)

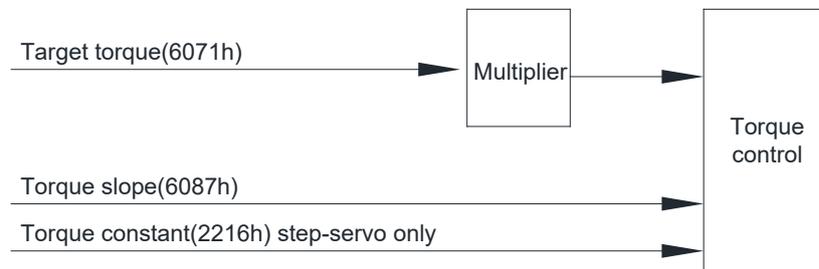


	Event	Set parameter
Enable motor power	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + enable operation +halt	0x6040=271(010Fh)
Set mode of operation	Profile velocity mode	0x6060=3(3h)
Set motion parameter	Target velocity	0x60FF=20000(4E20h)
	Acceleration	0x6083=50000(C350h)
	Deceleration	0x6084=50000(C350h)
Velocity mode	Start	0x6040=F(15h)
	Stop	0x6040=271(010Fh)

## 5.5 Profile torque mode (StepSERVO only)

### 5.5.1 General information

Profile Torque mode is a servo-control torque operating mode. It requires knowledge of the Torque Constant of the motor in mN/A. This information can be found in the motor print.



*Profile torque mode(PT mode)*

### 5.5.2 Main controlling object

Index	Name	Type	Access	Mapping
0x6040	Controlword	UINT16	WO	YES
0x6041	Statusword	UINT16	RO	YES
0x6071	Target torque	INT16	RW	NO
0x6087	Torque slope	INT8	WO	YES
0x2216	Torque constant	INT8	RO	YES

#### Controlword of profile torque mode

15	9	8	7	6	5	4	3	0
***		Halt	****	***	***	***	***	***

\*\*\*: See object description

Bit	Name	Value	Description
8	Halt	0	The motion shall be executed or continued
		1	Axis shall be stopped according to the halt option code (0x605D)

#### Statusword of profile torque mode

15	14	13	12	11	10	9	0
***		Reserved		***	Target reached		***

\*\*\*: See object description

Bit	Name	Value	Description
10	Target reached	0	Halt (bit 8 in controlword) = 0: Target not reached
			Halt (bit 8 in controlword) = 1: Axis decelerates
		1	Halt (bit 8 in controlword) = 0: Target reached
			Halt (bit 8 in controlword) = 1: Velocity of axis is 0

### 5.5.3 Functional description

To operate in profile torque mode, the following parameters must be set:

Index	Name	Description
0x2216	Torque constants	Motor parameter, found on the motor print
0x6071	Target torque	Torque to be applied to the motor
0x6087	Torque slope	Rate at which to ramp torque to new target

#### Parameter calculations – example

An application requires a torque of 0.353 Nm, and torque slope of 0.177 Nm/sec. we found the torque constants is 0.07Nm/A. then we write the value to the object:

Index	value	Units
0x2216	70	m•Nm/A
0x6071	353	m•Nm
0x6087	177	m•Nm/sec

#### Current verification – example

It is important to check that the current required of the drive is within the limits of the servo amplifier. The drive being used, for example, has a continuous rating of 7 amps, and a peak current of 14 amps, which may be held continuously for 2 seconds. This means that a current of 7 amps can be held indefinitely, and currents between 7 and 14 amps may be used in short bursts.

Using the target torque and torque constant from the example above the current draw can be checked, as shown:

$$0.353\text{Nm}/(0.07 \text{ Nm/A}) = 5.044\text{A}$$

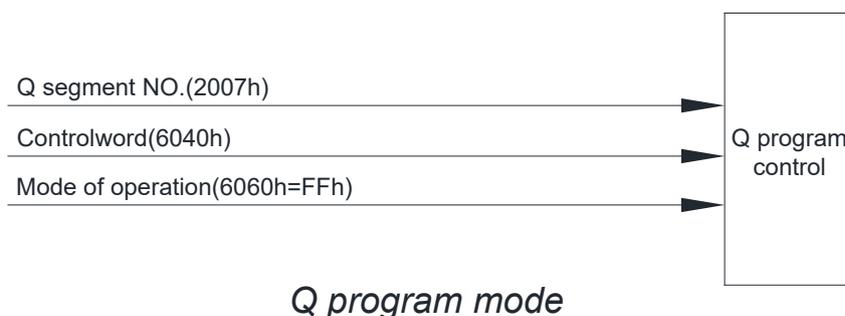
The resultant current, 5.044A, is below the 7A continuous current rating of the drive, and well below the peak current rating of 14A. It is possible for the drive to maintain a current of 7A indefinitely, and peak up to 14A for up to two seconds continuously. Values between 7A and 14A may be held proportionally long.

	Event	Set parameter
Enable motor power	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + enable operation +halt	0x6040=271(010Fh)
Set mode of operation	Profile velocity mode	0x6060=4(04h)
Set motion parameter	Target torque	0x6071=20(14h)
	Torque slope	0x6087=500(01F4h)
	Torque constant	0x2216=100(64h)
Profile torque mode	Start	0x6040=15(0Fh)
	Stop	0x6040=271(10Fh)

## 5.6 Q program mode

### 5.6.1 General information

In order to expand the functionality of MOONS' CANopen drives, the Q programming language may be used to execute complex motion profiles that may not be possible within the scope of CiA 402. The Q program must be written and pre-loaded into the CANopen drive using Q Programmer. Q Programs may also access and manipulate the CANopen General Purpose registers for use in stored programs.



### 5.6.2 Normal Q execution

To execute a stored Q program on a single drive, a value of -1 (FFh) must be written to the mode of operation OD entry, located at dictionary address 6060h. The mode of operation can be verified using OD entry 6061<sub>h</sub> - mode of operation display - which is updated when the current operation mode is accepted.

#### The data of controlling object example:

	Event	Set parameter
Enable motor power	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + enable operation +halt	0x6040=271(10Fh)
Set mode of operation	Normal Q mode	0x6060= -1(FFh)
Set parameter	Q procedure segment	0x2007=1(01h)
Normal Q mode	Start	0x6040=31(1Fh)
	Halt	0x6040=287(011Fh)

### 5.6.3 Synchronous Q program execution

To execute a stored Q program on a single drive, a value of -2 (FEh) must be written to the mode of operation OD entry, located at dictionary address 6060h. The mode of operation can be verified using OD entry 6061h - mode of operation display - which is updated when the current operation mode is accepted.

**The data of controlling object example:**

Event	Set parameter	
Enable motor power	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + enable operation +halt	0x6040=271(10Fh)
Set mode of operation	Normal Q mode	0x6060=-2(FEh)
Set parameter	SYNC message	0x1005=128(80h)
	Q procedure segment	0x2007=1(01h)
SYNCQ mode	Starts	COB-ID sync message"80"
	Halts	0x6040=271(10Fh)

## 5.7 Global control word and status word

BIT	Control word (0x6040h)					Status word (0x6041h)				
	Homing	Position	Velocity	Torque	Q	Homing	Position	Velocity	Torque	Q
15	Reserved					Reserved				
14										
13						Homing error	Following error	MAX slippage error	Reserved	Reserved
12						homing attained	set point acknowledge	speed	Reserved	Reserved
11						Internal limit active				
10						Target reached				
9	Reserved	Change on set point	Reserved	Reserved	Reserved	Remote				
8	Halt					Reserved				
7	Fault reset					Warning				
6	Reserved	Abs/Rel	Reserved	Reserved	Reserved	Switched on disabled				
5	Reserved	Change set immediately	Reserved	Reserved	Reserved	Quick stop				
4	Homing operation start	New set point	Reserved	Reserved	Q program start	Voltage enabled				
3	Enable operation					Fault				
2	Quick stop					Operation enabled				
1	Enable voltage					Switched on				
0	Switch on					Ready to switch on				

## 6 Application function

### 6.1 Communication watchdog

The Communication Watchdog monitors the state of the CANopen connection. If a break in connectivity is detected and lasts for a certain period (defined by 0x2060\_03h), then an action can be automatically triggered (defined by 0x2060\_05h). This allows a managed shutdown of the drive's motion in the event of communication loss.

Example for how to use communication watchdog (0x2060h) for safely control.

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	CONST	NO	NO	4
1	Watchdog enable	UINT8	RW	NO	NO	0
2	Watchdog status	UINT8	RW	NO	NO	0
3	Watchdog timeout	UINT16	RW	NO	NO	0x01F4
4	Watchdog trigger event	UINT8	RW	NO	NO	0x0F
5	Timeout option code	UINT16	RW	NO	NO	0

First of all we set the parameter of communication watchdog:

Object	Name	Description
0x2060_03h	Watchdog timeout	This object is used to set the cycle time of the watchdog. the units is 1ms.
0x2060_04h	Watchdog trigger event	This object is used to set the cycle trigger event
		0x01: RPDO1 for cycle trigger
		0x02: RPDO2 for cycle trigger
		0x04: RPDO3 for cycle trigger
		0x08: RPDO4 for cycle trigger
0x2060_05h	Watchdog timeout option code	0x00: the motor will be stopped and disabled
		0x01: the motor will be stopped and enable
		0x2-0x14: the motor will execute Q program with 1-12 segment

Then set 0x2060\_01h to 1 for enable watchdog and then the 0x2060\_02h shall be set to 0x01h. the object 0x2060\_01h must be set on **operation mode**.

When the CANopen connection has been broken, then 0x2060\_01h will be set to zero and 0x2060\_02h set to 1.

Write 1 to 0x2060\_02h can rest communication watchdog and write 1 to 0x2060\_01h can active the watchdog.

## 7 Object dictionary

The Object dictionary (OD) is the core of any CANopen node. It provides links to all the communication and running parameters of a node. The Object dictionary is defined in the Electronic Data Sheet (EDS), which lists all supported objects, along with any sub-objects.

Any OD entry may be accessed using the standard Service Data Object (SDO) protocol, while some may be accessed using the low-overhead Process Data Object (PDO) protocol.

### 7.1 Object description

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 flash. Each object within the dictionary is addressed using a 16-bit index. The overall layout of the standard Object Dictionary is shown below. This layout closely conforms to other industrial serial bus system concepts:

Index (hex)	Object
0	Not used
0001-001F	Static data types
0020-003F	Complex data Types
0040-005F	Manufacturer Specific Complex Data Types
0060-007F	Device Profile Specific Static Data Types
0080-009F	Device Profile Specific Complex Data Types
00A0-0FFF	Reserved for further use
1000-1FFF	Communication Profile Area
2000-5FFF	Manufacturer Specific Profile Area
6000-9FFF	Standardized Device Profile Area
A000-FFFF	Reserved for further use

## 7.2 Electronic data sheet

The EDS available on MOONS' website, lists all the properties of every supported object in the OD, here following the list of Object Dictionary description for step and step-servo drive.

### 7.2.1 Communication profile (CiA301)

Index	Sub	Name	Type	Access	Mapping
<a href="#">0x1000</a>	-	Device type	UINT32	RO	NO
<a href="#">0x1001</a>	-	Error register	UINT8	RO	YES
<a href="#">0x1002</a>	-	Manufacture status register	UINT32	RO	YES
<a href="#">0x1003</a>	8	Pre-defined error field	ARRAY	-	-
<a href="#">0x1005</a>	-	COB-ID SYNC message	UINT32	RW	NO
<a href="#">0x1006</a>	-	Communication cycle period	UINT32	RW	NO
<a href="#">0x1007</a>	-	Synchronous window length	UINT32	RW	NO
<a href="#">0x1008</a>	-	Manufacturer device name	STRING	RO	NO
<a href="#">0x1009</a>	-	Manufacturer hardware version	STRING	RO	NO
<a href="#">0x100A</a>	-	Manufacturer software version	STRING	RO	NO
<a href="#">0x1010</a>	2	Store parameters	ARRAY	RO	NO
<a href="#">0x1011</a>	2	Restore default parameters	ARRAY	RO	NO
<a href="#">0x1014</a>	-	COB-ID EMCY	UINT32	RO	NO
<a href="#">0x1017</a>	-	Producer heartbeat time	UINT16	RW	NO
<a href="#">0x1018</a>	4	Identity object	ARRAY	-	-
<a href="#">0x1019</a>	-	Synchronous counter overflow value	UINT8	RW	NO
<a href="#">0x1029</a>	6	Error behavior object	ARRAY	-	-
<a href="#">0x1200</a>	2	SDO server parameter	ARRAY	-	-
<a href="#">0x1400</a>	2	RPDO communication parameter1	ARRAY	-	-
<a href="#">0x1401</a>	2	RPDO communication parameter2	ARRAY	-	-
<a href="#">0x1402</a>	2	RPDO communication parameter3	ARRAY	-	-
<a href="#">0x1403</a>	2	RPDO communication parameter4	ARRAY	-	-
<a href="#">0x1600</a>	8	RPDO mapping parameter1	ARRAY	-	-
<a href="#">0x1601</a>	8	RPDO mapping parameter2	ARRAY	-	-
<a href="#">0x1602</a>	8	RPDO mapping parameter3	ARRAY	-	-
<a href="#">0x1603</a>	8	RPDO mapping parameter4	ARRAY	-	-
<a href="#">0x1800</a>	6	TPDO communication parameter1	ARRAY	-	-
<a href="#">0x1801</a>	6	TPDO communication parameter2	ARRAY	-	-
<a href="#">0x1802</a>	6	TPDO communication parameter3	ARRAY	-	-
<a href="#">0x1803</a>	6	TPDO communication parameter4	ARRAY	-	-
<a href="#">0x1A00</a>	8	TPDO mapping parameter1	ARRAY	-	-
<a href="#">0x1A01</a>	8	TPDO mapping parameter2	ARRAY	-	-
<a href="#">0x1A02</a>	8	TPDO mapping parameter3	ARRAY	-	-
<a href="#">0x1A03</a>	8	TPDO mapping parameter4	ARRAY	-	-

### 0x1000 Device type

Contains information about the device type. The object at index 1000h describes the type of device and its functionality. It is composed of a 16-bit field which describes the device profile that is used and a second 16-bit field which gives additional information about optional functionality of the device. The Additional Information parameter is device profile specific. Its specification does not fall within the scope of this document, it is defined in the appropriate device profile..

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	NO	NO	-

Bit 0-15: Device profile number

Bit 16-31: Additional information

**Note 1:** COS means: TPDO detects the change of State

### 0x1001 Error register

This object is an error register for the device. The device can map internal errors in this byte. This entry is mandatory for all devices. It is a part of an Emergency object.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RO	YES	NO	-

Bit 0: generic error

Bit 1: current

Bit 2: voltage

Bit 3: temperature

Bit 4: communication error (overrun, error state)

Bit 5: Reserved (always 0)

Bit 6: Reserved (always 0)

Bit 7: Reserved (always 0)

## 0x1002 Manufacturer status register

This object is a common status register for manufacturer specific purposes.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	YES	NO	0

This object always is 0.

## 0x1003 Pre-defined error field

The object at index 1003h holds the errors that have occurred on the device and have been signaled via the Emergency Object. In doing so it provides an error history.

1. The entry at sub-index 0 contains the number of actual errors that are recorded in the array starting at sub-index 1.
2. Every new error is stored at sub-index 1, the older ones move down the list.
3. Writing a "0" to sub-index 0 deletes the entire error history (empties the array). Values higher than 0 are not allowed to write. This have to lead to an abort message (error code: 06090030h).
4. The error numbers are of type UNSIGNED32 and are composed of a 16 bit error code and a 16 bit additional error information field which is manufacturer specific. The error code is contained in the lower 2 bytes (LSB) and the additional information is included in the upper 2 bytes (MSB).

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT32	RW	NO	NO	8

Sub index	Name	Data Type	Access Type	Default Value	PDO Mapping
0	Number of errors	UNSIGNED8	RW	0	NO
1	Standard error field	UNSIGNED32	RO	0	NO
2	Standard error field	UNSIGNED32	RO	0	NO
3	Standard error field	UNSIGNED32	RO	0	NO
4	Standard error field	UNSIGNED32	RO	0	NO
5	Standard error field	UNSIGNED32	RO	0	NO
6	Standard error field	UNSIGNED32	RO	0	NO
7	Standard error field	UNSIGNED32	RO	0	NO
8	Standard error field	UNSIGNED32	RO	0	NO

Number of Errors (sub index = 0):

Bit 0-7: Zero can be written to erase error history

Standard Error Field (sub index = 1~ 8):

Bit 0-15: Error code as transmitted in the Emergency object

Bit 16-31: Manufacturer specific additional information

## 0x1005 COB-ID SYNC message

This object shall indicate the configured COB-ID of the synchronization object (SYNC). Further, it defines whether the CANopen device generates the SYNC.

31	30	29	28 11	10 0
x	gen.	frame	0 0000 <sub>h</sub>	11-bit CAN-ID
29-bit CAN-ID				

Bit number	Value	Description
X	X	Do not care
Gen.	0 1	CANopen device does not generate SYNC message CANopen device generates SYNC message
Frame	0 1	11-bit ID valid (CAN base frame) 29-bit ID valid (CAN extended frame)
29-bit CAN-ID	X	29-bit CAN-ID of the CAN extended frame
11-bit CAN-ID	X	11-bit CAN-ID of the CAN base frame

Bits 29 (frame) and bit 30 (gen.) may be static (not changeable). If a CANopen device is not able to generate SYNC messages, an attempt to set bit 30 (gen.) to 1b is responded with the SDO abort transfer service (abort code: 0609 0030h). CANopen devices supporting the CAN base frame type only, an attempt to set bit 29 (frame) to 1b is responded with the SDO abort transfer service (abort code: 0609 0030h). The first transmission of SYNC object starts within 1 sync cycle after setting bit 30 to 1b. By setting bit 30 to 1b while the synchronous counter overflow value is greater than 0 the first SYNC message shall start with the counter reset to 1. It is not allowed to change bits 0 to 29, while the object exists (bit 30 = 1b).

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	0x00000080

Bit 0-10: COB-ID for SYNC object

Bit 11-29: set to 0

Bit 30: 1(0) - node generates (does NOT generate) SYNC object

Bit 31: set to 0

### 0x1006 Communication cycle period

This object defines the communication cycle period in ms. This period defines the SYNC interval. It is 0 if not used. If the communication cycle period on sync producer is changed to a new value unequal 0, the transmission of sync object resumes within 1 sync cycle of the new value.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	0

Bit 0-31: period of SYNC transmission in ms (0 = no transmission, no checking)

### 0x1007 Synchronous window length

Contains the length of the time window for synchronous PDOs in ms. It is 0 if not used.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	0

### 0x1008 Manufacturer device name

Contains the manufacturer device name.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	Visible string	CONST	NO	NO	AMA CANopen Motor Driver

Name of the manufacturer as string

### 0x1009 Manufacturer hardware version

Contains the manufacturer hardware version description.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	Visible string	CONST	NO	NO	-

Name of the hardware version as string.

### 0x100A Manufacturer software version

Contains the manufacturer software version description.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	Visible string	CONST	NO	NO	-

Name of the hardware version as string.

## 0x1010 Store parameters

This object supports the saving of parameters in non-volatile memory. By read access the device provides information about its saving capabilities. Several parameter groups are distinguished:

In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate Sub-Index. The signature is “save”.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT32	RW	NO	NO	2

Sub-index	Name	Data Type	Access Type	Default value	PDO Mapping
0	Highest sub-index supported	UNSIGNED8	RO	2	NO
1	Save all parameter	UNSIGNED32	RW	0	NO
2	Save communication parameters	UNSIGNED32	RW	0	NO

	MSB		LSB	
Signature ISO	e	v	a	s
8859("ASCII")hex	65h	76h	61h	73h

Storage write access signature

## 1 Save config parameters

### Stepper drives

Index	Name	Note
0x2601	Idle current	
0x2603	Acceleration current	
0x2604	Steps per revolution	
0x2607	Brake configuration	
0x2609	Alarm reset input	
0x260A	Jerk delay time	
0x260B	Define limits	
0x260C	Input filter	sub-index 3...8
0x260D	Motor selection	
0x260E	Open winding detect velocity limit	
0x260F	Drive configuration	
0x2610	Alarm mask	
0x2611	Load ratio	
0x2612	Custom motor	sub-index 1...15
0x2613	Pulse noise filter	
0x2614	Smooth filter	
0x2615	Move output	
0x2616	Alarm output	
0x2617	Servo enable	

## Step-servo drives

Index	Name	Note
0x2201	Peak current	
0x2202	Hard stop current	
0x2203	Idle current	
0x2204	In position error range	
0x2205	In position time	
0x2208	Rotation reverse	
0x2209	Velocity max	
0x220A	Acceleration max	
0x220B	Smooth filter	
0x220C	Brake configuration	
0x220D	Servo enable	
0x220E	Alarm reset input	
0x220F	Define limits	
0x2210	Input filter	sub-index 3...8
0x2211	Notch filter	
0x2212	Analog configuration	
0x2213	Alarm output	
0x2214	Move output	
0x2215	Jog mode	
0x221A	Alarm mask	
0x2220	Position gain	
0x2221	Position derigain	
0x2222	Position derifilter	
0x2224	Velocity gain	
0x2225	Velocity integergain	
0x2226	Acceleration feedforward	
0x2227	PID filter	
0x2252	In position counts	

## 2 Save communication parameters

Index	Name	Note
0x1005	COB-ID SYNC message	
0x1006	Communication cycle period	
0x1007	Synchronous window length	
0x1014	COB-ID EMCY	
0x1017	Producer default parameters	
0x1019	Synchronous counter overflow value	
0x1029	Error behavior	
0x1200	SDO server parameter	
0x1400-0x1403	RPDO communication parameter	
0x1600-0x1603	RPDO mapping parameter	
0x1800-0x1803	TPDO communication parameter	
0x1a00-0x1a03	TPDO mapping parameter	

## 0x1011 Restore default parameters

With this object the default values of parameters according to the communication or device profile are restored.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT32	RW	NO	NO	2

Sub-index	Name	Data Type	Access Type	Default value	PDO Mapping
0	Highest sub-index supported	UNSIGNED8	RO	2	NO
1	Restore all parameter	UNSIGNED32	RO	0	NO
2	Restore communication default parameters	UNSIGNED32	RO	0	NO

	MSB		LSB	
Signature ISO	d	a	o	l
8859("ASCII")hex	64h	61h	6Fh	6Ch

Storage write access signature

## 0x1014 COB-ID emergency object

Index 1014h defines the COB-ID of the Emergency Object (EMCY).

Object Type	Data Type	Access Type	PDO mapping	COS	Default value
ARRAY	UINT32	RO	NO	NO	0x80+\$NODEID

	UNSIGNED32				
	MSB			LSB	
bits	31	30	29	28-11	10-0
11-bit-ID	0/1	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-bit Identifier
29-bit-ID	0/1	0	1	29-bit Identifier	

Structure of the EMCY Identifier entry

Bit number	Value	Meaning
31 (MSB)	0	EMCY exists / is valid
	1	EMCY does not exist / is not valid
30	0	Reserved(always 0)
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
28-11	0	If bit 29=0
	X	If bit 29=1: bits 28-11 of 29-bit-COB-ID
10-0 (LSB)	X	Bits 10-0 of COB-ID

Description of EMCY COB-ID entry

Devices supporting the standard CAN frame type only, an attempt to set bit 29 is responded with an abort message (abort code: 0609 0030h). It is not allowed to change Bits 0-29, while the object exists (Bit 31=0).

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	NO	NO	0x80+\$NODEID

Bit 0-10: COB-ID

Bit 11-30: set to 0 for 11 bit COB-ID

Bit 31: 0(1) - node uses (does NOT use) Emergency object

### 0x1017 Producer heartbeat time

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

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	1000

0 = disable transmission

### 0x1018 Identity object

The object at index 1018h contains general information about the device.

1. The Vendor ID (sub-index 1h) contains a unique value allocated to each manufacturer.
2. The manufacturer-specific Product code (sub-index 2h) identifies a specific device version.
3. The manufacturer-specific Revision number (sub-index 3h) consists of a major revision number and a minor revision number. The major revision number identifies a specific CANopen behaviour. If the CANopen functionality is expanded, the major revision has to be incremented. The minor revision number identifies different versions with the same CANopen behaviour.
4. The manufacturer-specific Serial number (sub-index 4h) identifies a specific device.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT32	RW	NO	NO	4

Sub-index	Name	Data Type	Access Type	Default value	PDO Mapping
0	Max sub-index	UNSIGNED8	RO	4	NO
1	Vendor-ID	UNSIGNED32	RO	-	NO
2	Product code	UNSIGNED32	RO	-	NO
3	Revision number	UNSIGNED32	RO	-	NO
4	Serial number	UNSIGNED32	RO	-	NO

Max sub-index

Vendor-ID

Bit 0-31: Assigned by CiA (here is 0x000002D9 for Shanghai AMP and Moons' Automation)

Product code

Bit 0-31: Manufacturer specific

Revision number

Bit 0-15: Minor revision num. (CANopen behavior has not changed)

Bit 16-31: Major revision number (CANopen behavior has changed)

Serial number

Bit 0-31: Manufacturer specific (Lot Number)

### 0x1019 Synchronous counter overflow value

This object contains the max counters of SYNC message. If it is 0 the producer will send the SYNC message without any data, otherwise with data length 1.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RW	NO	NO	0

If value is zero, then SYNC message is transmitted with data length 0. If value is from 2 to 240, then SYNC message has one data byte, which contains the counter. Other values are reserved.

## 0x1029 Error behavior

This object contains the error information of the CANopen drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RW	NO	NO	6

Sub-index	Name	Data Type	Access Type	Default value	PDO Mapping
0	Highest sub-index supported	UNSIGNED8	RO	6	NO
1	Communication	UNSIGNED8	RO	1	NO
2	Communication other	UNSIGNED8	RO	1	NO
3	Communication passive	UNSIGNED8	RO	1	NO
4	Generic	UNSIGNED8	RO	1	NO
5	Device profile	UNSIGNED8	RO	1	NO
6	Manufacturer specific	UNSIGNED8	RO	1	NO

Value definition for all sub-indexes:

0x00 - if operational, switch to NMT pre-operational

0x01 - do nothing

0x02 - switch to NMT stopped

01 - Communication error - bus off or Heartbeat consumer error.

02 - Communication other error (critical errors - see 'Error status bits') except CAN bus passive but including bus off or Heartbeat consumer.

03 - Communication passive - any communication error including CAN bus passive.

04 - Generic error (critical errors - see 'Error status bits').

05 - Device profile error - bit 5 in error register is set.

06 - Manufacturer specific error - bit 7 in error register is set.

## 0x1200 SDO server parameter

The object holds the COB-ID (communication object ID, also known as CAN message ID) values used to access the drive's SDO.

1. Sub-index 0 contains the number of sub-elements of this record.
2. Sub-index 1 used by the driver to receive SDO packets. The value is  $0x600 + \$NODEID$ .
3. Sub-index 2 gives the COB-ID used by the driver to transmit SDO packets. The value is  $0x580 + \$NODEID$ .

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT32	RW	NO	NO	2

sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UNSIGNED8	RO	NO	NO	2
1	COB-ID client to server	UNSIGNED32	RO	NO	NO	$0x600 + \$NODEID$
2	COB-ID server to client	UNSIGNED32	RO	NO	NO	$0x580 + \$NODEID$

0x1400~0x1403 Receive PDO communication parameter

Contains the communication parameters for the PDOs the device is able to receive.

Object Type	Date type	Access Type	PDO mapping	COS	Sub Number
RECORD	-	-	NO	NO	2

Sub-index	Name	Date type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UNSIGNED8	RO	NO	NO	2
1	COB-ID used by RPDO	UNSIGNED32	RO	NO	NO	0x1400h: 0x200+\$NODEID 0x1401h: 0x300+\$NODEID 0x1402h: 0x400+\$NODEID 0x1403h: 0x500+\$NODEID
2	Transmission type	UNSIGNED32	RO	NO	NO	0xFF

Sub-index 0 contains the number of the valid sub-index, the value must more the 2 at least.

Sub-index 01 contains the COB-ID of RPDO.

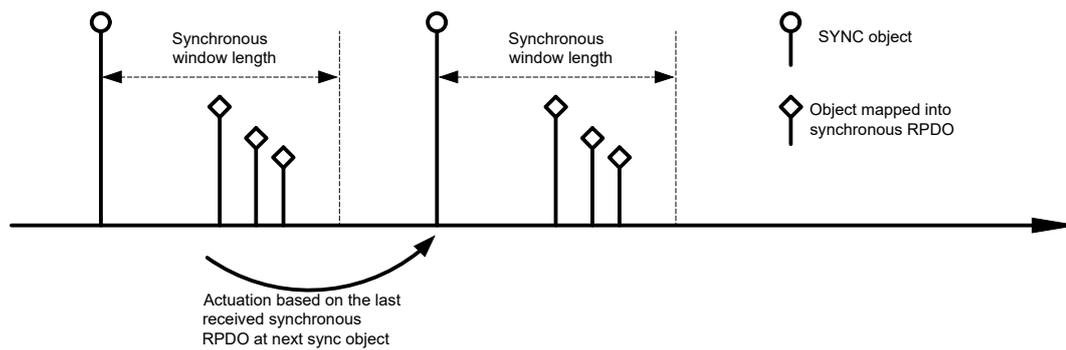
31	30	29	28	11	10	0
Valid	Reserved	Frame	0 0000h		11-bit CAN-ID	
29-bit CAN-ID						

Bit	Value	Description
Valid	0	EMCY exists / is valid
	1	EMCY does not exist / is not valid
Reserved	x	Do not care
Frame	0	11-bit ID valid(CAN base frame)
	1	29-bit ID valid(CAN extended frame)
29-bit CAN-ID	x	29-bit CAN-ID of the CAN extended frame
11-bit CAN-ID	X	11-bit CAN-ID of the CAN base frame

Sub-index 02 defines the reception character of the RPDO. An attempt to change the value of the transmission type to any not supported value shall be responded with the SDO abort transfer service.

Value	Description
00 <sub>h</sub>	Synchronous
...	...
F0 <sub>h</sub>	Synchronous
F1 <sub>h</sub>	Reserved
...	...
FD <sub>h</sub>	Reserved
FE <sub>h</sub>	Event-driven(manufacturer-specific)
FF <sub>h</sub>	Event-driven(device profile and application profile specific)

- Synchronous means that the CANopen device shall actuate the received data with the reception of the next SYNC (see below.)
- Event-driven means that the PDO may be received at any time. The CANopen device will actualize the data immediately.



0x1600~0x1603 Receive PDO mapping parameter

This object contains the mapping parameters for the PDOs the CANopen device is able to receive.

Object Type	Date type	Access Type	PDO mapping	COS	Sub Number
RECORD	-	-	NO	NO	8

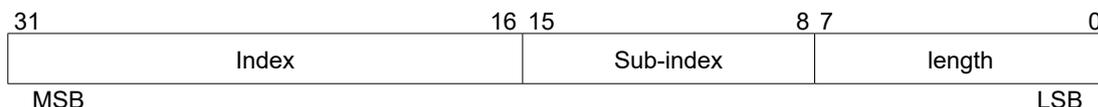
Sub-index	Name	Date type	Access Type	PDO Mapping	COS	Default value
0	Number of mapped application object in RPDO	UNSIGNED8	RW	NO	NO	1
1	1st application object	UNSIGNED32	RW	NO	NO	0x60400010
2	2st application object	UNSIGNED32	RW	NO	NO	0x00000000
3	3st application object	UNSIGNED32	RW	NO	NO	0x00000000
4	4st application object	UNSIGNED32	RW	NO	NO	0x00000000
5	5st application object	UNSIGNED32	RW	NO	NO	0x00000000
6	6st application object	UNSIGNED32	RW	NO	NO	0x00000000
7	7st application object	UNSIGNED32	RW	NO	NO	0x00000000
8	8st application object	UNSIGNED32	RW	NO	NO	0x00000000

Sub-index 0 contains the number of valid object entries within the mapping record or a specific value.

Value	Description
00	Mapping disabled
01	Sub-index 01h valid
02	Sub-index 01h to 02h valid
03	Sub-index 01h to 03h valid
04	Sub-index 01h to 04h valid
05	Sub-index 01h to 05h valid
06	Sub-index 01h to 06h valid
07	Sub-index 01h to 07h valid
08	Sub-index 01h to 08h valid

Sub-index from 01h to 40h:

contains the information of the mapped application objects. The object describes the content of the PDO by their index, sub-index and length. The length contains the length of the application object in bit. This may be used to verify the mapping.



The following procedure shall be used for re-mapping, which may take place during the NMT state Pre-operational and during the NMT state Operational, if supported:

1. Destroy RPDO by setting bit valid to 1b of sub-index 01h of the according RPDO communication parameter.
2. Disable mapping by setting sub-index 00h to 00h.
3. Modify mapping by changing the values of the corresponding sub-indices.
4. Enable mapping by setting sub-index 00h to the number of mapped objects.
5. Create RPDO by setting bit valid to 0b of sub-index 01h of the according RPDO communication parameter.

## 0x1800~0x1803 Transmit PDO communication parameter

Contains the communication parameters for the PDOs the device is able to transmit.

Object Type	Date type	Access Type	PDO mapping	COS	Sub Number
RECORD	-	-	NO	NO	6

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Max sub-index	UNSIGNED8	RO	NO	NO	6
1	COB-ID used by TPDO	UNSIGNED32	RW	NO	NO	Index 1800h 0x180+\$NODE-ID Index 1801h 0x280+\$NODE-ID Index 1802h 0x380+\$NODE-ID Index 1803h 0x480+\$NODE-ID
2	Transmission type	UNSIGNED8	RW	NO	NO	255
3	Inhibit time	UNSIGNED16	RW	NO	NO	100
4	Compatibility entry	UNSIGNED8	RW	NO	NO	0
5	Event timer	UNSIGNED16	RW	NO	NO	0
6	SYNC start value	UNSIGNED8	RW	NO	NO	0

Max sub-index:

Largest sub-index supported.

COB-ID:

Bit 0-10: COB-ID for PDO, to change it bit 31 must be set

Bit 11-29: set to 0 for 11 bit COB-ID

Bit 30: 0(1) - RTR are allowed (are NOT allowed) for PDO

Bit 31: 0(1) - node uses (does NOT use) PDO

Transmission type:

Value	Description
00	Synchronous(acyclic)
01	Synchronous(cyclic every SYNC)
02	Synchronous(cyclic every 2 <sup>nd</sup> SYNC)
03	Synchronous(cyclic every 3 <sup>rd</sup> SYNC)
04	Synchronous(cyclic every 4 <sup>th</sup> SYNC)
...	...
F0	Synchronous(cyclic every 240 <sup>th</sup> SYNC)
F1	Reserved
...	...
FB	Reserved
FC	RTR-only(synchronous, not supported)
FD	RTR-only(even-driven, not supported)
FE	Event-driven(manufacturer-specific)
FF	Event-driven(device profile and application)

Inhibit time:

Bit 0-15: Minimum time between transmissions of the PDO in 100us. Zero disables functionality.

Compatibility entry:

Bit 0-7: Not used.

Event timer:

Bit 0-15: Time between periodic transmissions of the PDO in ms. Zero disables functionality.

SYNC start value:

Value = 0: Counter of the SYNC message shall not be processed.

Value = 1-240: The SYNC message with the counter value equal to this value shall be regarded as the first received SYNC message.

0x1A00~0x1A03 Transmit PDO mapping parameter

Contains the mapping for the PDOs the device is able to transmit.

Object Type	Date type	Access Type	PDO mapping	COS	Sub Number
RECORD	-	-	NO	NO	8

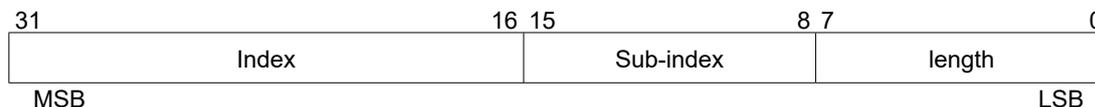
Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Number of mapped application object in TPDO	UNSIGNED8	RW	NO	NO	1
1	1st application object	UNSIGNED32	RW	NO	NO	0x60410010
2	2st application object	UNSIGNED32	RW	NO	NO	0x00000000
3	3st application object	UNSIGNED32	RW	NO	NO	0x00000000
4	4st application object	UNSIGNED32	RW	NO	NO	0x00000000
5	5st application object	UNSIGNED32	RW	NO	NO	0x00000000
6	6st application object	UNSIGNED32	RW	NO	NO	0x00000000
7	7st application object	UNSIGNED32	RW	NO	NO	0x00000000
8	8st application object	UNSIGNED32	RW	NO	NO	0x00000000

Sub-index 00h contains the number of valid object entries within the mapping record or a specific value.

Value	Description
00	Mapping disabled
01	Sub-index 01h valid
02	Sub-index 01h to 02h valid
03	Sub-index 01h to 03h valid
04	Sub-index 01h to 04h valid
05	Sub-index 01h to 05h valid
06	Sub-index 01h to 06h valid
07	Sub-index 01h to 07h valid
08	Sub-index 01h to 08h valid

Sub-index from 01h to 40h:

contains the information of the mapped application objects. The object describes the content of the PDO by their index, sub-index and length. The length contains the length of the application object in bit. This may be used to verify the mapping.



The following procedure shall be used for re-mapping, which may take place during the NMT state Pre-operational and during the NMT state Operational, if supported:

1. Destroy RPDO by setting bit valid to 1b of sub-index 01h of the according RPDO communication parameter.
2. Disable mapping by setting sub-index 00h to 00h.
3. Modify mapping by changing the values of the corresponding sub-indices.
4. Enable mapping by setting sub-index 00h to the number of mapped objects.
5. Create RPDO by setting bit valid to 0b of sub-index 01h of the according RPDO communication parameter.

## 7.2.2 Motion control profile (CiA402)

Index	Sub	Name	Type	Access	Mapping
<a href="#">0x603F</a>	-	Error code	UINT16	RO	YES
<a href="#">0x6040</a>	-	Controlword	UINT16	WO	YES
<a href="#">0x6041</a>	-	Statusword	UINT16	RO	YES
<a href="#">0x605A</a>	-	Quick stop option code	INT16	RW	NO
<a href="#">0x6060</a>	-	Modes of operation	INT8	WO	YES
<a href="#">0x6061</a>	-	Modes of operation display	INT8	RO	YES
<a href="#">0x6064</a>	-	Position actual value	INT32	RO	YES
<a href="#">0x6065</a>	-	Following error window	UINT32	RW	NO
<a href="#">0x606C</a>	-	Velocity actual value	INT32	RO	YES
<a href="#">0x6071</a>	-	Target torque	INT16	RW	YES
<a href="#">0x6073</a>	-	Max current	UINT16	RW	YES
<a href="#">0x6074</a>	-	Torque demand	INT16	RO	YES
<a href="#">0x6078</a>	-	Current actual value	INT16	RO	YES
<a href="#">0x607A</a>	-	Target position	INT32	RW	YES
<a href="#">0x607C</a>	-	Home offset	INT32	RW	YES
<a href="#">0x607E</a>	-	Polarity	UINT8	RW	YES
<a href="#">0x607F</a>	-	Max profile velocity	UINT32	RW	YES
<a href="#">0x6081</a>	-	Profile velocity	UINT32	RW	YES
<a href="#">0x6083</a>	-	Profile acceleration	UINT32	RW	YES
<a href="#">0x6084</a>	-	Profile deceleration	UINT32	RW	YES
<a href="#">0x6085</a>	-	Quick stop deceleration	UINT32	RW	YES
<a href="#">0x6087</a>	-	Torque slope	UINT32	RW	YES
<a href="#">0x6098</a>	-	Homing method	INT8	RW	YES
<a href="#">0x6099</a>	2	Homing speed	ARRAY	-	NO
<a href="#">0x609A</a>	-	Homing acceleration	UINT32	RW	YES
<a href="#">0x60F4</a>	-	Following error actual value	INT32	RO	YES
<a href="#">0x60FD</a>	-	Digital inputs	UINT32	RW	YES
<a href="#">0x60FE</a>	2	Digital outputs	ARRAY	-	NO
<a href="#">0x60FE</a>	-	Target velocity	INT32	RW	YES
<a href="#">0x6502</a>	-	Supported drive modes	UINT32	RO	NO

0x603F Error code

The error code captures the DSP alarm code of the last error that occurred in the drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RO	YES	YES	0

Each bit in DSP error code indicate one type alarm or faults status.

BIT	StepSERVO Drive	Stepper Drive
0	Position limit	
1	CCW limit	
2	CW limit	
3	Over temp	
4	Internal Voltage	
5	Over Voltage	
6	Under Voltage	
7	Over Current	
8	Open Motor Winding	
9	Bad Encoder	Not Used
10	Comm Error	
11	Bad Flash	
12	No Move	
13	Current Foldback	Not Used
14	Blank Q Segment	
15	Not Used	

0x6040 Controlword

This object shall indicate the received command controlling the PDS FSA. The bits 7, 3, 2, 1, and 0 shall be supported. The bits 0 to 9 shall be supported according to the mode of operation. If the related functionality is not available, an appropriate emergency message shall be generated. The manufacturer-specific bits may be supported. All implemented bits of the controlword are valid independent of the PDS FSA state. Starting of any movement is operation mode specific and is described in the related clause.

The controlword consist of bits for:

- The controlling of the state
- The controlling of operating modes
- Manufacturer specific options

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	WO	YES	NO	0

The bits of the controlword are defined as follows:

15	11	10	9	8	7	6	4	3	2	1	0
Manufacturer specific	Reserved		Halt	Fault reset	Operation mode specific	Enable operation	Quick stop	Enable voltage	switch on		
O	O	O	M	O	M	M	M	M	M	M	M
MSB	0	-	Optional		M	-	Mandatory				LSB

## 0x6041 Statusword

The status word indicates the current state of the drive. No bits are latched. The status word consist of bits for:

- The current state of the drive
- The operating state of the mode
- Manufacturer specific options

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RO	YES	YES	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms		oms		ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rtso

Key:

ms	manufacturer specific
oms	operation mode specific
ila	internal limit active
tr	target reached
rm	remote
w	warning
sod	switch on disabled
qs	quick stop
ve	voltage enabled
f	fault
oe	operation enabled
so	switched on
rtso	ready to switch on

Statusword(6041h)	PDS FSA state
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switch on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

## 0x605A Quick stop option code

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

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT16	RW	NO	NO	0

quick stop option code	Action
-32768...-1	Manufacturer Specific
0	Disable drive function
1	Slow down on slow down ramp and transit into switch on disabled
2	Slow down on quick stop ramp and transit into switch on disabled
3	Slow down on the current limit and transit into switch on disabled
4	Slow down on the voltage limit and transit into switch on disabled
5	Slow down on slow down ramp and stay in quick stop active
6	Slow down on quick stop ramp and stay in quick stop active
7	Slow down on slow current limit and stay in quick stop active
8	Slow down on voltage limit and stay in quick stop active
9...32767	Reserved

It is only supported of option code 1 and 2 feature at this moment

## 0x6060 Mode of operation

The parameter modes of operation switches the actually chosen operation mode.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT8	WO	YES	NO	0

Value	Action
-2	SYNC Q Mode(manufacturer specific)
-1	Normal Q Mode (manufacturer specific)
1	Profile Position Mode
3	Profile Velocity Mode
4	Torque Profile Mode(step servo only)
6	Homing Mode

-1: Normal Q mode (manufacturer specific mode)

-2: SYNC Q mode (manufacturer specific mode)

Velocity Mode and Interpolated Position Mode are not supported in this CANopen drive. Also the Torque profile Mode is only supported for StepSERVO drive.

### 0x6061 Mode of operation display

The modes of operation display shows the current mode of operation. The meaning of the returned value corresponds to that of the modes of operation option code (index 6060<sub>h</sub>).

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT8	RO	YES	YES	0

### 0x6064 Position actual value

This object represents the actual value of the position.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT32	RO	YES	YES	0

### 0x6065 Following error window

The following error window defines a range of tolerated position values symmetrically to the position demand value. It is the position fault limit in encoder counts. 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
- At wrong closed loop coefficients

If the value of the following error window is 0 ,the following control is switched off.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	-

### 0x606C Velocity actual value

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

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT32	RO	YES	YES	0

The unit of this object is in counts/s.

### 0x6071 Target torque

This parameter is the input value for the torque controller in profile torque mode and the value is given per thousand of rated torque. The units should be mN·m.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT16	RW	YES	NO	0

This object only can be accessed in step servo driver and this object parameters is related to the other torque values, such as current actual value (index 0x6078) and torque constant (index 0x2216).

0x6073 Max current

This value represents the maximum permissible torque creating current in the motor.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	YES	NO	0

The unit of this object is 0.01Amps.

0x6074 Torque demand value

This parameter is the output value of the torque limit function (if the torque control and power-stage function are available).

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT16	RO	YES	NO	0

This object is only available on step servo drives and the unit of this object is mNm.

0x6078 Current actual value

The current actual value refers to the instantaneous current in the drive motor.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT16	RO	YES	NO	0

This object is only available on step servo drives and the unit of object is 0.01Amps.

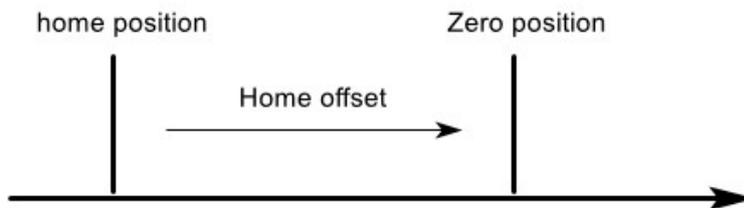
0x607A Target position

This object 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.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT32	RW	YES	NO	0

0x607C Home offset

The home offset object is the difference between the zero position for the application and the machine home position (found during homing), it is measured in position units. During homing the machine home position is found and once the homing is completed the zero position is offset from the home position by adding the home offset to the home position. All subsequent absolute moves shall be taken relative to this new zero position. This is illustrated in the following diagram.

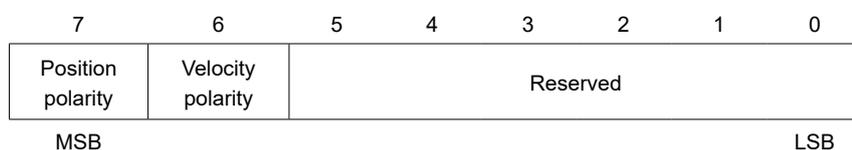


Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT32	RW	YES	NO	0

### 0x607E Polarity

Position demand value and position actual value are multiplied by 1 or -1 depending on the value of the polarity flag. PP PV Mode.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RO	YES	NO	0



Value	Description
0	Multiply by 1
1	Multiply by -1

### 0x607F Max profile speed

The max profile velocity is the maximum allowed speed in either direction during a profiled move. It is given in the same units as profile velocity.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

The unit of this object is in counts/s

### 0x6081 Profile velocity

The profile velocity is the velocity normally attained at the end of the acceleration ramp during a profiled move and is valid for both directions of motion.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

The unit of this object is in counts/s.

### 0x6083 Profile acceleration

The profile acceleration is given in counts/s<sup>2</sup>. It is converted to position increments per second<sup>2</sup> using the normalizing factors.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

### 0x6084 Profile deceleration

The profile deceleration is given in the same units as profile acceleration.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

### 0x6085 Quick stop deceleration

The quick stop deceleration is the deceleration used to stop the motor if the 'Quick Stop' command is given and the quick stop option code (see 605Ah) is set to 2. The quick stop deceleration is given in the same units as the profile acceleration.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

### 0x6087 Torque slope

This parameter describes the rate of change of torque in units of per thousand of rated torque per second.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

The units is Nm/s.

### 0x6098 Home method

The homing method object determines the method that will be used during homing.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT8	RW	YES	NO	0

Value	Description
128	Manufacturer specific
0	No homing operation required
1...37	Methods 1 to 35
38-127	Reserved

### 0x6099 Homing speed

This entry in the object dictionary defines the speeds used during homing.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	2

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Number of mapped entries	UNSIGNED8	RO	NO	NO	2
1	Speed during search for switch	UNSIGNED32	RW	YES	NO	0
2	Speed during search for zero	UNSIGNED32	RW	YES	NO	0

The value shall be given in counts/s.

### 0x609A Homing acceleration

The homing acceleration establishes the acceleration to be used for all accelerations and decelerations with the standard homing modes.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

The value units should be the same as profile acceleration/deceleration objects.

### 0x60F4 Following error actual value

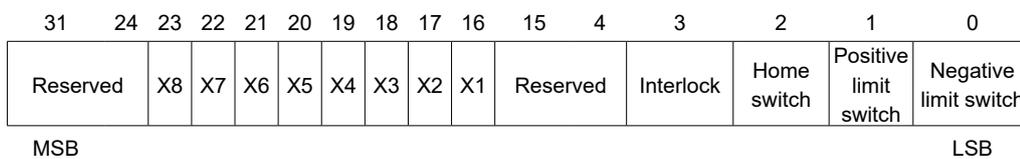
This object shall provide the actual value of the following error. The value shall be given in user-defined position units.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT32	RO	YES	NO	0

### 0x60FD Digital inputs

This object shall provide digital inputs. This object shall represent the logical input levels.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	YES	NO	0



Bit 3 (interlock) provides the state of the interlock input. If the logical input signal changes to not activated, the drive shall enter the switch on disabled or fault reaction active state. This means the power stage of the drive is disabled and locked against switching on.

### 0x60FE Digital outputs

This object shall command the digital outputs. This object shall represent the logical output levels.

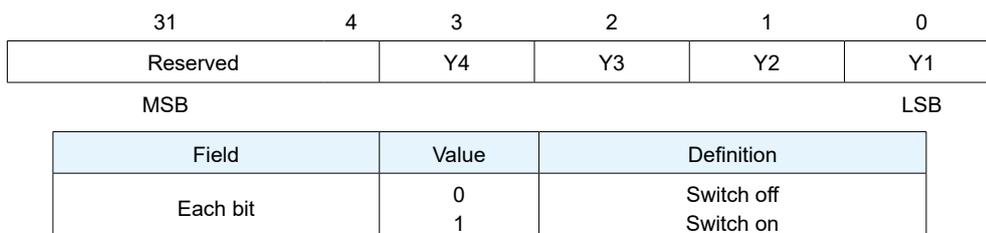
Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	2

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	1...2	RO	NO	NO	2
1	Physical outputs	UNSIGNED32	RW	YES	NO	0
2	Bit mask	UNSIGNED32	RW	NO	NO	0

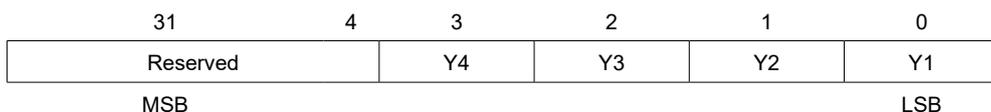
The first sub-index defines the assigned outputs. The second sub-index describes a mask to specify which of the outputs shall be used.

Note: the second sub-index are edge triggered, you must set the second sub-index first and then set the bit of first sub-index for change the status of output.

Physical outputs:



Bit mask:



Field	Value	Definition
Each bit	0	Disable output
	1	Enable output

### 0x60FF Target velocity

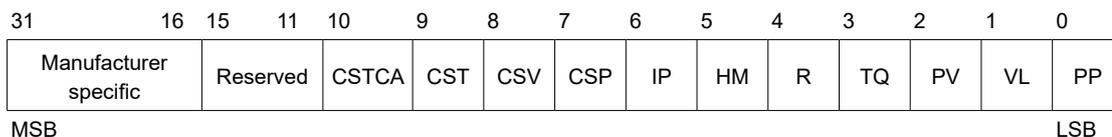
This object shall indicate the configured target velocity and shall be used as input for the trajectory generator. The value shall be given in counts/s.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT32	RW	YES	NO	0

### 0x6502 Supported drive modes

This object shall provide information on the supported drive modes.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	NO	NO	0



The supported mode in Moons' drive:

- Bit0: Profile Position Mode
- Bit2: Profile Velocity Mode
- Bit3: Profile Torque Mode ( Step-Servo only)
- Bit5: Homing Mode

### 7.2.3 Manufacturer profile

Index	Sub	Name	Type	Access	Mapping
<a href="#">0x2001</a>	-	Home switch	UINT8	RW	YES
<a href="#">0x2006</a>	-	DSP clear alarm	UINT8	WO	NO
<a href="#">0x2007</a>	-	Q segment NO.	UINT8	RW	YES
<a href="#">0x2009</a>	-	Homing offset mode	UINT16	RW	NO
<a href="#">0x200B</a>	-	DSP status code	UINT32	RO	YES
<a href="#">0x200C</a>	-	Zero position	UINT8	WO	NO
<a href="#">0x200F</a>	-	DSP alarm code	UINT32	RO	YES
<a href="#">0x2019</a>	4	Device temperature	ARRAY	-	NO
<a href="#">0x2020</a>	-	Node ID	UINT16	RW	NO
<a href="#">0x2021</a>	-	Bit rate	UINT16	RW	NO
<a href="#">0x2030</a>	-	DC bus voltage	UINT16	RO	NO
<a href="#">0x2031</a>	-	DSP version	STRING(4)	RO	NO
<a href="#">0x2040</a>	2	PDO1 transmit mask	ARRAY	-	NO
<a href="#">0x2041</a>	2	PDO2 transmit mask	ARRAY	-	NO
<a href="#">0x2042</a>	2	PDO3 transmit mask	ARRAY	-	NO
<a href="#">0x2043</a>	2	PDO4 transmit mask	ARRAY	-	NO
<a href="#">0x2050</a>	-	Product series	STRING(4)	RO	NO
<a href="#">0x2051</a>	-	Customer name	STRING(4)	RO	NO
<a href="#">0x2060</a>	5	Comm. watchdog	ARRAY	-	NO
<a href="#">0x2070</a>	-	Switch value	UINT32	RO	NO
<a href="#">0x2100</a>	23	User registers	ARRAY	-	NO

The object describe this section is manufacturer specific by MOONS' to configure or monitor the MOONS' CANopen drive.

#### 0x2001 Home switch

This object shall configure the number of Inputs as the Home switch in Homing.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RW	YES	NO	3

Value	Home sensor
1	X1
2	X2
3	X3
4	X4
5	X5
6	X6
7	X7
8	X8

0x2006 DSP clear alarm

This object provides the feature to clear alarm of the drives. Set this value to 01h can clear the error code in object 0x603F and object 0x200F.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	WO	NO	COS	0

0x2007 Q segment NO.

This object shall configure the number of Q Segment will be executed in Q mode.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RW	YES	NO	0

0x2009 Homing offset mode

This object is used to set homing offset mode.

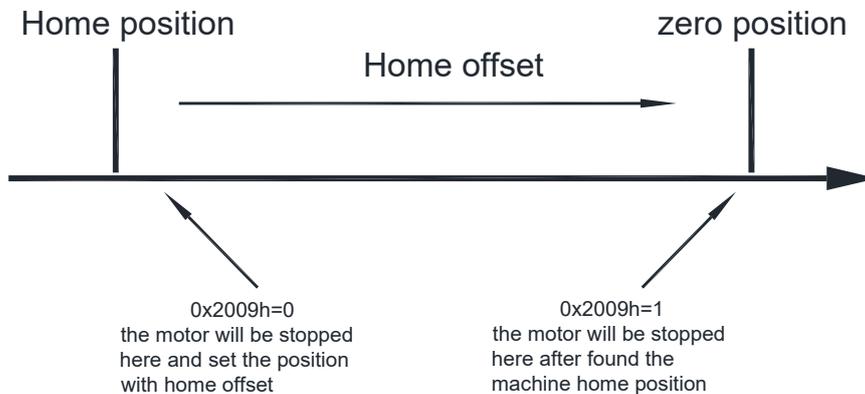
Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	0

Set the value to 0:

The motor will be stopped on machine home position and the current position value is the home offset.

Set the value to 1:

The motor will moving with a distance that home offset provided after the machine home position has found. The new position is the zero position.



### 0x200B DSP status code

This object represents the current status code of the drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	YES	YES	0

BIT	Status Code Bit Definition
0	Motor Enabled - motor disabled is this bit = 0
1	Sampling - for Quick Tuner
2	Drive Fault - check alarm code
3	In Position - motor is in position
4	Moving - motor is moving
5	Jogging - currently in jog mode
6	Stopping - in the process of stopping from a stop command
7	Waiting - for an input
8	Saving - parameter data is being saved
9	Alarm present - check alarm code
10	Homing - executing an SH command
11	Wait Time - executing a WT command
12	Wizard running - timing wizard is running
13	Checking encoder - timing wizard is running
14	Q Program is running
15	Initializing

### 0x200C Zero position

This object provides the feature to zero all position parameters, such as position actual value (which index is 0x6064h).

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	WO	NO	NO	0

Set this value to 01h can zero all position parameters.

### 0x200F DSP alarm code

This object shall indicate the high 16bit field of alarm code about the object at 0x603F.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	WO	NO	NO	0

### 0x2019 Device temperature

This object contains the information of device temperature.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	4

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UNSIGNED8	RO	NO	NO	4
1	Drive temperature	UNSIGNED16	RO	NO	NO	0
2	DSP temperature	UNSIGNED16	RO	NO	NO	0
3	Reserved 1	UNSIGNED16	RO	NO	NO	0
4	Reserved 2	UNSIGNED16	RO	NO	NO	0

The unit of this object is in 0.1 centigrade.

## 0x2020 Node ID

The object allows the user to set the high 4 bit of CAN Node ID, the change takes effect at next power cycle and store the parameter with the object at 0x1010h.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RW	NO	NO	0

Value	ID range
0	0x00-0x0F
1	0x10-0x1F
2	0x20-0x2F
3	0x30-0x3F
4	0x40-0x4F
5	0x50-0x5F
6	0x60-0x6F
7	0x70-0x7F

## 0x2021 Bit rate

The object allows the user to set the CAN bit rate of the Node, the change takes effect at next power cycle and store the parameter with the object at 0x1010h.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RW	NO	NO	0

## 0x2030 DC bus voltage

This object shall provide the present value of drive's DC bus voltage.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RO	NO	NO	0

The voltage reads out in 0.1 volts revolution.

## 0x2031 DSP version

This object shall provide the DSP version of the drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	STRING4	RO	NO	NO	0

### 0x2040...2043 PDOx transmit mask

The “TPDO detects COS” (means “on change data”) transmit event for PDOs can be mask off by these indexes. By default, all data in a PDO is considered when checking for change.

Each Index has 2 sub-indexes, sub-index 1 is used to mask (or reveal) bits in the lower 32bits (4bytes) of data. Sub-index 2 is the same for the upper 32bits.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	2

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	2
1	PDOx mask lowerBytes	UING32	RW	NO	NO	0xFFFFFFFF
2	PDOx mask upperBytes	UINT32	RW	NO	NO	0xFFFFFFFF

#### PDOx mask LowerBytes

Bit0-31: mask off the lower 4 bytes of a TPDO when the data is “on change”

#### PDOx mask upperBytes

Bit0-31: mask off the upper 4 bytes of a TPDO when the data is “on change”

### 0x2050 Product series

This object contains the product series of the drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	STRING4	RO	NO	NO	--

### 0x2051 Customer name

This object contains the name of the custom drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	STRING4	RO	NO	NO	--

## 0x2060 Comm. watchdog

This object can implement a safety feature in Profile Velocity mode. The watchdog shall only be reset by the particular RPOD messages or synchronization messages (SYNC message with COB-ID 0x80), if no particular messages are received for longer than the set and activated the watchdog time, e.g. in the event of a line interruption, the watchdog is triggered and the motor speed is decelerated to ZERO with a quick stop deceleration (OD 0x6085).

- The watchdog is configurable by the following,
- The watchdog can be configured to be enable or disable.
- The watchdog time out window is configurable.
- The particular RPDO or SYNC message is configurable independently. RPDO1, 2, 3, 4 and SYNC message can be selected to be monitored to trigger the watchdog or not.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	4

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	CONST	NO	NO	4
1	Watchdog enable	UINT8	RW	NO	NO	0
2	Watchdog status	UINT8	RW	NO	NO	0
3	Watchdog timeout	UINT16	RW	NO	NO	0x01F4
4	Watchdog trigger event	UINT8	RW	NO	NO	0x0F
5	Timeout option code	UINT16	RW	NO	NO	0

## Sub-index 1:

Value = 0      watchdog disable  
           = 1      watchdog enable

This object just can be set with operation mode.

## Sub-index 2:

Value = 0      normal status  
           = 1      watchdog is triggered  
           = 2      The watchdog is enabled/engaged and monitoring the CAN-bus message, but it's not triggered

## Sub-index 3:

Set the parameter of watchdog time, the unit of this object is 1ms.

## Sub-index 4:

This object is used to choose the message for cyclic trigger.

Value = 0x01    RPDO1  
           = 0x02    RPDO2  
           = 0x04    RPDO3  
           = 0x08    RPDO4  
           = 0x10    SYNC message

## Sub-index 5:

Value = 0x0      Stop moving and disable  
           = 0x01      Stop moving and enable  
           = 0x2-0x0Eh    Execute Q segments

## 0x2070 Switch value

The object contains the rotary switch, such as Node ID and Bit Rate switches setting information.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	NO	NO	3

The low 16 bit filed is the setting information.

## 0x2100 User registers

This object provide user 23 general purpose registers. They are volatile, so the information sent there will not be saved after a power cycle.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	23

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UNSIGNED8	RO	YES	NO	23
1	User register1	UNSIGNED32	RW	YES	NO	0
2	User register2	UNSIGNED32	RW	YES	NO	0
3	User register3	UNSIGNED32	RW	YES	NO	0
4	User register4	UNSIGNED32	RW	YES	NO	0
5	User register5	UNSIGNED32	RW	YES	NO	0
6	User register6	UNSIGNED32	RW	YES	NO	0
7	User register7	UNSIGNED32	RW	YES	NO	0
8	User register8	UNSIGNED32	RW	YES	NO	0
9	User register9	UNSIGNED32	RW	YES	NO	0
10	User register10	UNSIGNED32	RW	YES	NO	0
11	User register11	UNSIGNED32	RW	YES	NO	0
12	User register12	UNSIGNED32	RW	YES	NO	0
13	User register13	UNSIGNED32	RW	YES	NO	0
14	User register14	UNSIGNED32	RW	YES	NO	0
15	User register15	UNSIGNED32	RW	YES	NO	0
16	User register16	UNSIGNED32	RW	YES	NO	0
17	User register17	UNSIGNED32	RW	YES	NO	0
18	User register18	UNSIGNED32	RW	NO	NO	0
19	User register19	UNSIGNED32	RW	NO	NO	0
20	User register20	UNSIGNED32	RW	NO	NO	0
21	User register21	UNSIGNED32	RW	NO	NO	0
22	User register22	UNSIGNED32	RW	NO	NO	0
23	User register23	UNSIGNED32	RW	NO	NO	0

## 7.2.4 Manufacturer parameter for StepSERVO drives

Index	Sub	Name	Type	Access	Mapping
<a href="#">0x2201</a>	-	Peak current	UINT16	RW	YES
<a href="#">0x2202</a>	-	Torque limit of hard stop homing	UINT16	RW	NO
<a href="#">0x2203</a>	-	Idle current	UINT16	RW	NO
<a href="#">0x2204</a>	-	In position error range	UINT16	RW	NO
<a href="#">0x2205</a>	-	In position time	UINT16	RW	NO
<a href="#">0x2206</a>	-	Acceleration current	UINT16	RW	NO
<a href="#">0x2207</a>	-	Steps per revolution	UINT16	RO	NO
<a href="#">0x2208</a>	-	Rotation reverse	UINT8	RW	NO
<a href="#">0x2209</a>	-	Velocity max	UINT32	RW	NO
<a href="#">0x220A</a>	-	Acceleration max	UINT32	RW	NO
<a href="#">0x220B</a>	-	Smooth filter	UINT16	RW	NO
<a href="#">0x220C</a>	3	Brake configuration	UINT8	RO	NO
<a href="#">0x220D</a>	-	Servo enable	UINT16	RW	NO
<a href="#">0x220E</a>	-	Alarm reset input	UINT16	RW	NO
<a href="#">0x220F</a>	-	Define limits	UINT16	RW	NO
<a href="#">0x2210</a>	8	Digital inputs filter	ARRAY	-	NO
<a href="#">0x2211</a>	8	Notch filter	ARRAY	-	NO
<a href="#">0x2212</a>	4	Analog configuration	ARRAY	-	NO
<a href="#">0x2213</a>	-	Alarm output	UINT16	RW	NO
<a href="#">0x2214</a>	-	Move output	UINT16	RW	NO
<a href="#">0x2215</a>	-	Jog mode	UINT16	RW	NO
<a href="#">0x2216</a>	-	Torque constant	UINT16	RO	NO
<a href="#">0x2217</a>	-	Misc flag	UINT16	RW	NO
<a href="#">0x2218</a>	-	Encoder resolution	UINT16	RO	NO
<a href="#">0x221A</a>	-	Alarm mask	UINT16	RW	NO
<a href="#">0x221C</a>	4	Analog input	ARRAY	-	NO
<a href="#">0x2220</a>	-	Position gain	UINT16	RW	NO
<a href="#">0x2221</a>	-	Position derigain	UINT16	RW	NO
<a href="#">0x2222</a>	-	Position derifilter	UINT16	RW	NO
<a href="#">0x2224</a>	-	Velocity gain	UINT16	RW	NO
<a href="#">0x2225</a>	-	Velocity inteergain	UINT16	RW	NO
<a href="#">0x2226</a>	-	Acceleration feedforward	UINT16	RW	NO
<a href="#">0x2227</a>	-	PID filter	UINT16	RW	NO
<a href="#">0x2252</a>	-	In position counts	UINT16	RW	NO

### 0x2201 Peak current

This object is the peak (RMS) current of stepper servo. The peak current sets the maximum current should be used with a given motor.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	YES	NO	-

The value shall be given in mA.

### 0x2202 Torque limit of hard stop homing

This object is used for hard stop homing mode that setting the current when the motor hit the hard stop position.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

### 0x2203 Idle current

This object configures monitors the motor holding current of the device in idle mode. The units of this object is 0.01Amps.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

### 0x2204 In position error range

This object is used to set static in-position error range. When the actual position is within the target In-position error range for a time duration that exceeds the specified timing (0x2205), then the drive will define the motion complete or motor in-position.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

### 0x2205 In position time

This object is used to set the time duration for in range determination. When in-position error 0x2204h is defined, this parameter will set the duration for in-position test condition.

Time is counted as processor cycles, one cycle refers to 200usec.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

If the reading value is 100, means 20ms.

### 0x2206 Acceleration current

This object shall provide the acceleration current when the motor is running with stepper mode.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

If the reading value is 100 that means 1.00Amps.

### 0x2207 Steps per revolution

This object is used to get the steps per revolution about motor.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RO	NO	NO	-

The value shall be given in steps/rev

### 0x2208 Rotation reverse

This object is used to reverse motor rotating direction.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RW	NO	NO	0

The value can be set to 0 - 1.

Value = 0          default rotating direction

      = 1          reverse rotating direction

### 0x2209 Velocity max

The object at Index 0x2209h is used to set the maximum velocity of the drive. And it is used in analog velocity mode to limit the maximum speed of the drive too.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	-

The unit shall be given in counts/s.

### 0x220A Acceleration max

This object at index 0x220A is used to set the maximum acceleration/deceleration.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	-

The value shall be given in counts/s<sup>2</sup>.

### 0x220B Smooth filter

This object provides a filter to change in the position command to cause a smoother movement of the motor.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	2500

Please note that a lower filter value will result in much smoother motion but will cause a lag in response.

## 0x220C Brake configuration

This object at 0x220C is used to set the parameter of brake configuration. There has 3 sub-index to configure brake, the first is used for brake output, and the second is used for disengage delay. The last is used to set brake engage delay.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	3

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	3
1	Brake output	UINT16	RW	NO	NO	0
2	Move command waiting time when brake release	UINT16	RW	NO	NO	0
3	Servo-off waiting time after brake engagement	UINT16	RW	NO	NO	0

### Brake output:

- Value = 1      output is closed when drive is enabled, and open when the drive is disabled.  
 = 2      output is open when drive is enabled, and closed when the drive is disabled.  
 = 3      output is not used as a brake output and can be used as g general purpose output.

### Move command waiting time when brake release/Servo-off waiting time

The units is 1ms, if write 100 to the index means 0.1s.

## 0x220D Servo enable

This object is used to set the usage of the enable input. Input X5 is the default enable input on drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	3

### Details:

- Value = 1      input x5 is used for drive enabled when open  
 = 2      input x5 is used for drive enabled when closed  
 = 3      input x5 is used for general purpose

## 0x220E Alarm reset input

This object is used to set usage of the alarm reset input. Input X6 is the default AR input on drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	3

- Value = 1      input x6 is used for drive alarm reset when open  
 = 2      input x6 is used for drive alarm reset when closed  
 = 3      input x6 is used for general purpose

## 0x220F Define limits

This object is used to set the definition of limit.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

Value	Definition
1	End of travel limit occurs when an input is closed
2	End of travel limit occurs when an input is open
3	Inputs are not used as end of travel limit inputs and can be used as a general purpose
4...20	See the details in "host command reference"

## 0x2210 Input filter

This object is used to set a digital filter to the input.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	3

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	8
1	Filter input 1	UINT16	RW	NO	NO	0
2	Filter input 2	UINT16	RW	NO	NO	0
3	Filter input 3	UINT16	RW	NO	NO	0
4	Filter input 4	UINT16	RW	NO	NO	0
5	Filter input 5	UINT16	RW	NO	NO	0
6	Filter input 6	UINT16	RW	NO	NO	0
7	Filter input 7	UINT16	RW	NO	NO	0
8	Filter input 8	UINT16	RW	NO	NO	0

The unit of this parameter is 200us. If the value you set to 100, means 20ms delay.

## 0x2211 Notch filter

These eight objects shall configure the Notch Filter parameters. This object is only available on StepSERVO drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	8

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	8
1	Notch filter A tuning	UINT16	RW	NO	NO	-
2	Notch filter B tuning	UINT16	RW	NO	NO	-
3	Notch filter C tuning	UINT16	RW	NO	NO	-
4	Notch filter D tuning	UINT16	RW	NO	NO	-
5	Notch filter E tuning	UINT16	RW	NO	NO	-
6	Notch filter F tuning	UINT16	RW	NO	NO	-
7	Notch filter G tuning	UINT16	RW	NO	NO	-
8	Notch filter H tuning	UINT16	RW	NO	NO	-

## 0x2212 Analog configuration

This object shall indicate the configuration of running mode about analog. We should set the value of this object when running with analog velocity/position/torque mode.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	4

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	4
1	Analog type	UINT16	RW	NO	NO	0
2	Analog deadband	UINT16	RW	NO	NO	0
3	Analog offset	UINT16	RW	NO	NO	0
4	Analog filter	UINT16	RW	NO	NO	0

### Analog type:

Value = 0	single-ended +/- 10 volts
= 1	single-ended 0 - 10 volts
= 2	single-ended +/- 5 volts
= 3	single-ended 0 - 5 volts
= 4	differential +/- 10 volts
= 5	differential 0 - 10 volts
= 6	differential +/- 5 volts
= 7	differential 0 - 5 volts

### Analog deadband

The unit is in 0.001volit

### Analog offset

The unit is in 0.001volit

### Analog filter

Filter value =  $72090 / [ (1400 / x ) + 2.2 ]$

Where x = desired value of the analog filter in Hz

## 0x2213 Alarm output

This object is used to set the alarm output when the drive has been fault. Output Y1 is the default alarm output for drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	3

Value = 1	output Y1 is open when the driver is fault
= 2	output Y1 is closed when the driver is fault
= 3	output Y1 is used for general purpose

### 0x2214 Motion output

This object is used to define the drive Motion output function.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	3

SSDC has 4 digital output, they can be set to various functions.

Output	Y4				Y3				Y2				Y1			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

For example:

The value of Y4 is 0011B means that the functions of Y4 is general purpose.

- Value = 1      Open when static position error less than in-position counts.
- = 2      Closed when static position error less than in-position counts.
- = 3      General purpose (fault output or brake output)
- = 4      Tach output with 100 pulses/rev
- = 5      Tach output with 200 pulses/rev
- = 6      Tach output with 400 pulses/rev
- = 7      Tach output with 800 pulses/rev
- = 8      Tach output with 1600 pulses/rev
- = 9      Closed (energized) when dynamic position error is less than set value.
- = 10      Open (de-energized) when dynamic position error is less than set value.
- = 11      Timing out (50 pulses/rev)

### 0x2215 Jog mode

This object is used to set the jog mode. There has two mode for moons drive.

- Value = 1      position-type servo control when jogging
- = 2      velocity-type servo control when jogging

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	3

### 0x2216 Torque constant

This object shall configure the motor's torque constant in manufacturer specific units. The units should be mN·m/Amps.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RO	NO	NO	0

This object only supported in step-servo drives. 0x2216=MAX toque / MAX continuous current (base on different motor parameters).

## 0x2217 Misc flag

This object shall provide the other status and error of the drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT	RW	NO	NO	0

The value of this object is a hexadecimal sum of various drive states. Write 0 can clear the value.

Description	Hex Value	Decimal Value
DISTANCE LIMIT FLAG	0x0001	1
SENSOR FOUND FLAG	0x0002	2
LOWSIDE OVERCURRENT	0x0004	4
HIGHSIDE OVERCURRENT	0x0008	8
OVER CURRENT READING	0x0010	16
BAD CURRENT OFFSET - Phase A	0x0020	32
BAD CURRENT OFFSET - Phase B	0x0040	64
OPEN WINDING - Phase A	0x0080	128
OPEN WINDING - Phase B	0x0100	256
LOGIC SUPPLY	0x0200	512
GATE SUPPLY	0x0400	1024
BAD FLASH ERASE	0x4000	16384
BAD FLASH SAVE	0x8000	32768

## 0x2218 Encoder resolution

This object shall provide the encoder configuration of the motor. It contains how many counts per revolution.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RO	NO	NO	-

## 0x221A Alarm mask

This object is used to mask the LED flashing of the corresponding alarm.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

Details:

bit	15	14	13	12	11	10		6	5		2	
		Blank Q segment	Current foldback	Move while disabled	Flash memory	Comm. error		Under voltage	CCW limit		CW limit	

## 0x221C Analog input

This object provide analog input value.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	4

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	4
1	Analog input 1	UINT16	RO	YES	YES	-
2	Analog input 2	UINT16	RO	YES	YES	-
3	Analog input 3	UINT16	RO	YES	YES	-
4	Analog input 4	UINT16	RO	YES	YES	-

### 0x2220 Position gain

This object shall configure the proportional Gain in Position loop to step-servo drive. This object is only available on step-servo drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

### 0x2221 Position derigain

This object shall configure the Derivative Gain in Position loop to step-servo drive. This object is only available on step-servo drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

### 0x2222 Position derifilter

This object provides a very simple single-pole low pass filter that is used to limit this high frequency noise and make the system quieter and more stable.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

### 0x2224 Velocity gain

This object shall configure the proportional Gain in Velocity loop to step-servo drive. This object is only available on step-servo drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

### 0x2225 Velocity integergain

This object shall configure the Integral Gain in Velocity loop to step-servo drive. This object is only available on step-servo drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

### 0x2226 Acceleration feed forward

This object shall configure to add a feed forward acceleration/deceleration to the torque command to faster the system's response. This object is only available on step-servo drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

### 0x2227 PID filter

This object provide a torque command over-all filter at the end of Velocity loop. The filter is a very simple single-pole low pass filter that is used to limit the high frequency response of the Velocity and therefore the Position control loops. This object is only available on step-servo drive.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

### 0x2252 In position counts (position limit)

This object is used to set the static in-position error range. This is used by the servo for determining the state of motion output.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

## 7.2.5 Manufacturer parameter for Stepper drives

Index	Sub	Name	Type	Access	mapping
<a href="#">0x2601</a>		Idle current	UINT8	RW	NO
<a href="#">0x2603</a>		Acceleration current	UINT16	RW	NO
<a href="#">0x2604</a>		Steps per revolution	UINT16	RW	NO
<a href="#">0x2607</a>		Brake configuration	ARRAY	-	NO
<a href="#">0x2609</a>		Alarm reset input	UINT16	RW	NO
<a href="#">0x260A</a>		Jerk delay time	UINT16	RW	NO
<a href="#">0x260B</a>		Define limits	UINT16	RW	NO
<a href="#">0x260C</a>		Input filter	ARRAY	-	NO
<a href="#">0x260D</a>		Motor selection	UINT16	RW	NO
<a href="#">0x260E</a>		Open winding detect velocity limit	UINT32	RW	NO
<a href="#">0x260F</a>		Drive configuration	ARRAY	-	NO
<a href="#">0x2610</a>		Alarm mask	UINT16	RW	NO
<a href="#">0x2611</a>		Load ratio	UINT16	RW	NO
<a href="#">0x2612</a>		Custom motor	ARRAY	-	NO
<a href="#">0x2614</a>		Smooth filter	UINT16	RW	NO
<a href="#">0x2615</a>		Motion output	UINT16	RW	NO
<a href="#">0x2616</a>		Alarm output	UINT16	RW	NO
<a href="#">0x2617</a>		Servo enable	UINT16	RW	NO

[0x2601 Idle current](#)

This object configures/monitors the motor holding current of the device in idle mode.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

The units of this object is current in Amps multiply 100.

[0x2603 Acceleration current](#)

This object shall configure the current of drive during in acceleration state.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

It is given the same units as the objects max current and idle current whose value should be divided by 100.

[0x2604 Steps per revolution](#)

This object shall indicate the steps of revolution. It allows you to adjust the way that the drive responds to incoming step pulses.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

The value shall be given in steps/rev

## 0x2607 Brake configuration

This object at 0x2607 is used to set the parameter of brake configuration. It is same as object at 0x220C.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	3

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	3
1	Brake output	UINT16	RW	NO	NO	0
2	Move command waiting time when brake release	UINT16	RW	NO	NO	0
3	Servo-off waiting time after brake engagement	UINT16	RW	NO	NO	0

### Brake output:

- Value = 1      output is closed when drive is enabled, and open when the drive is disabled.
- = 2      output is open when drive is enabled, and closed when the drive is disabled.
- = 3      output is not used as a brake output and can be used as g general purpose output.

### Move command waiting time when brake release/Servo-off waiting time

The units is 1ms, if write 100 to the index that means 0.1s.

## 0x2609 Alarm reset input

This object is used to set usage of the alarm reset input. The same as the object at 0x220E

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	3

- Value = 1      input x6 is used for drive alarm reset when open
- = 2      input x6 is used for drive alarm reset when closed
- = 3      input x6 is used for general purpose

## 0x260A Jerk delay time

This object shall provide the jerk delay time.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	3

The value shall be given in ms.

## 0x260B Define limits

This object is used to set the definition of limit.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UNSIGNED16	RW	NO	NO	-

Value	Definition
1	End of travel limit occurs when an input is closed
2	End of travel limit occurs when an input is open
3	Inputs are not used as end of travel limit inputs and can be used as a general purpose
4...20	See the details in "host command reference"

## 0x260C Input filter

This object is used to set a digital filter to the input.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	8

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	8
1	Filter input 1	UINT16	RW	NO	NO	60
2	Filter input 2	UINT16	RW	NO	NO	0
3	Filter input 3	UINT16	RW	NO	NO	0
4	Filter input 4	UINT16	RW	NO	NO	0
5	Filter input 5	UINT16	RW	NO	NO	0
6	Filter input 6	UINT16	RW	NO	NO	0
7	Filter input 7	UINT16	RW	NO	NO	0
8	Filter input 8	UINT16	RW	NO	NO	0

The unit of this parameter is 200us. If the value you set to 100, means 20ms delay.

## 0x260D Motor selection

This object is used to set the motor.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

Details:

Value	Selection	Value	Selection
0	AM8HY2050	14	AM23HS2450
1	AM8HY4043	15	AM23HS3454
2	AM11HS008	16	AM23HS3455
3	AM11HS3007	17	AM23HS5412
4	AM11HS5008	18	AM23HS84A0
5	AM14HYB401	19	AM23HS84B0
6	AM17HD2438	20	AM23HSA4A0
7	AM17HD4452	21	AM23HSA4B0
8	AM17HDB410	22	AM24HS2402
9	AM23HS0420	23	AM24HS5401
10	AM23HS0421	24	AM34HD0404
11	AM23HS04A0	25	AM34HD1404
12	AM23HS04B0	26	AM34HD2403
13	AM23HS2449	27	AM34HD3402

### 0x260E Open winding detect velocity limit

This object is used to set the velocity limit of the open winding status detect when the motor is moving.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	-

The unit of this object is counts/s.

### 0x260F Drive configuration

This object shall be used to set the parameter of drive configuration.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RW	NO	NO	8

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	8
1	Reverse motion DIR	UINT16	RW	NO	NO	0
2	Wave form smooth	UINT16	RW	NO	NO	0
3	Electronic damping	UINT16	RW	NO	NO	0
4	DYNC open detected	UINT16	RW	NO	NO	0
5	Reserved 1	UINT16	RW	NO	NO	0
6	Reserved 2	UINT16	RW	NO	NO	0
7	Reserved 3	UINT16	RW	NO	NO	0
8	Reserved 4	UINT16	RW	NO	NO	0

Description of sub-index:

Sub-index	Name	Description
1	Reverse motion DIR	0: keep the current direction of motor 1: change the current direction of motor
2	Wave form smooth	0: wave smoothing off 1: wave smoothing on
3	Electronic damping	0: enable electronic damping 1: disable electronic damping
4	DYNC open detected	0: disable open winding detect functions 1: enable open winding detect functions

### 0x2610 Alarm mask

This object is used to mask the LED flashing of the corresponding alarm. The same as the object at 0x221A

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

Details:

bit	15	14	13	12	11	10		6	5		2	
		Blank Q segment	Current foldback	Move while disabled	Flash memory	Comm. error		Under voltage	CCW limit		CW limit	

## 0x2611 Load ratio

This object sets the ratio of the load inertia to the rotor inertia.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	1

Value = 1	1x rotor inertia
= 2	1x-3x rotor inertia
= 3	3x-5x rotor inertia
= 4	5x-7x rotor inertia
= 5	7x-10x rotor inertia

## 0x2612 Custom motor

This object shall indicate the motor of custom.

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UNSIGNED32	RW	NO	NO	22

Sub index	Name	Data Type	Access Type	Default Value	PDO Mapping
0	Highest sub-index supported	UNSIGNED8	R0	0	NO
1	Motor name 1	UNSIGNED32	RW	0	NO
2	Motor name 2	UNSIGNED32	RW	0	NO
3	Motor name 3	UNSIGNED32	RW	0	NO
4	Motor name 4	UNSIGNED32	RW	0	NO
5	Motor parameter 1	UNSIGNED16	RW	0	NO
6	Motor parameter 2	UNSIGNED16	RW	0	NO
7	Motor parameter 3	UNSIGNED16	RW	0	NO
8	Motor parameter 4	UNSIGNED16	RW	0	NO
9	Motor parameter 5	UNSIGNED16	RW	0	NO
A	Motor parameter 6	UNSIGNED16	RW	0	NO
B	Motor parameter 7	UNSIGNED16	RW	0	NO
C	Motor parameter 8	UNSIGNED16	RW	0	NO
D	Motor parameter 9	UNSIGNED16	RW	0	NO
E	Motor parameter 10	UNSIGNED16	RW	0	NO
F	Motor parameter 11	UNSIGNED16	RW	0	NO
10	Motor parameter 12	UNSIGNED16	RW	0	NO
11	Reserved 1	UNSIGNED16	RW	0	NO
12	Reserved 2	UNSIGNED16	RW	0	NO
13	Reserved 3	UNSIGNED16	RW	0	NO
14	Reserved 4	UNSIGNED16	RW	0	NO
15	Reserved 5	UNSIGNED32	RW	0	NO

## 0x2614 Smooth filter

This object provides a filter to change in the position command to cause a smoother movement of the motor. The same as the object at 0x220B.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UNSIGNED16	RW	NO	NO	2500

## 0x2615 Motion output

The same as the object at 0x2214.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UNSIGNED16	RW	NO	NO	0

STF has 4 digital output that can be set to various functions.

Output	Y4				Y3				Y2				Y1			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

For example:

The value of Y4 is 0011B means that the functions of Y4 is general purpose.

Value = 1	Open when motor is moving
= 2	Closed when motor is moving
= 3	General purpose (fault output or brake output)
= 4	Tach output with 100 pulses/rev
= 5	Tach output with 200 pulses/rev
= 6	Tach output with 400 pulses/rev
= 7	Tach output with 800 pulses/rev
= 8	Tach output with 1600 pulses/rev
= 9	Reserved
= 10	Reserved
= 11	Timing out (50 pulses/rev)

## 0x2616 Alarm output

This object is used to set the alarm output when the drive has been fault.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UNSIGNED16	RW	NO	NO	3

Value = 1	output Y1 is open when the driver is fault
= 2	output Y1 is closed when the driver is fault
= 3	output Y1 is used for general purpose

## 0x2617 Servo enable

This object is used to set the usage of the enable input. The same as the object at 0x220D.

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UNSIGNED16	RW	NO	NO	3

Details:

Value = 1	input x5 is used for drive enabled when open
= 2	input x5 is used for drive enabled when closed
= 3	input x5 is used for general purpose

## 8 Appendix A – Example for SDO message control

### Profile Position Mode

Enable Motor Power - CiA 402 State Machine

ID(Hex)	DLC	Data(Hex)	Description
603	8	2B 40 60 00 06 00 00 00	'Ready to Switch on
603	8	2B 40 60 00 07 00 00 00	'Switched on
603	8	2B 40 60 00 0F 00 00 00	'Operation Enabled
<b>Set to Profile Position Mode</b>			
603	8	2F 60 60 00 01 00 00 00	'Set to Profile Position Mode
<b>Set Motion Parameters</b>			
603	8	23 81 60 00 20 4E 00 00	'Set Profile Velocity to 20000 step/s
603	8	23 83 60 00 40 9C 00 00	'Set Acceleration to 40000 step/s <sup>2</sup>
603	8	23 84 60 00 40 9C 00 00	'Set Deceleration to 40000 step/s <sup>2</sup>
<b>Single Move Absolute</b>			
603	8	23 7A 60 00 40 0D 03 00	'Set Target Position to 200000 steps
603	8	2B 40 60 00 1F 00 00 00	'Set New Set Point Bit to 1
603	8	2B 40 60 00 0F 00 00 00	'Clear New Set Point Bit
<b>Single Move Relative</b>			
603	8	23 7A 60 00 40 0D 03 00	'Set Target Position to 200000 steps
603	8	2B 40 60 00 5F 00 00 00	'Set New Set Point Bit to 1
603	8	2B 40 60 00 4F 00 00 00	'Clear New Set Point Bit
<b>Multiple Move, Stopping between Moves</b>			
603	8	23 81 60 00 40 9C 00 00	'Set Profile Velocity to 40000 step/s
603	8	23 7A 60 00 40 0D 03 00	'Set Target Position to 200000 steps
603	8	2B 40 60 00 5F 00 00 00	'Set New Set Point Bit to 1
603	8	2B 40 60 00 4F 00 00 00	'Clear New Set Point Bit
603	8	23 81 60 00 80 38 01 00	'Set Profile Velocity to 80000 step/s
603	8	23 7A 60 00 C0 27 09 00	'Set Target Position to 600000 steps
603	8	2B 40 60 00 5F 00 00 00	'Set New Set Point Bit to 1
603	8	2B 40 60 00 4F 00 00 00	'Clear New Set Point Bit
<b>Multiple Move, Continuous Motion</b>			
603	8	23 81 60 00 40 9C 00 00	'Set Profile Velocity to 40000 step/s
603	8	23 7A 60 00 40 0D 03 00	'Set Target Position to 200000 steps
603	8	2B 40 60 00 5F 02 00 00	'Set New Set Point Bit to 1
603	8	2B 40 60 00 4F 02 00 00	'Clear New Set Point Bit
603	8	23 81 60 00 80 38 01 00	'Set Profile Velocity to 80000 step/s <sup>2</sup>
603	8	23 7A 60 00 C0 27 09 00	'Set Target Position to 600000 steps
603	8	2B 40 60 00 5F 02 00 00	'Set New Set Point Bit to 1
603	8	2B 40 60 00 4F 02 00 00	'Clear New Set Point Bit
<b>Multiple Move, Immediate Change in Motion</b>			
603	8	23 81 60 00 40 9C 00 00	'Set Profile Velocity to 40000 step/s
603	8	23 7A 60 00 40 0D 03 00	'Set Target Position to 200000 steps
603	8	2B 40 60 00 7F 02 00 00	'Set New Set Point Bit to 1
603	8	2B 40 60 00 6F 02 00 00	'Clear New Set Point Bit
603	8	23 81 60 00 80 38 01 00	'Set Profile Velocity to 80000 step/s
603	8	23 7A 60 00 C0 27 09 00	'Set Target Position to 600000 steps
603	8	2B 40 60 00 7F 02 00 00	'Set New Set Point Bit to 1
603	8	2B 40 60 00 6F 02 00 00	'Clear New Set Point Bit

## Profile Velocity Mode

### Enable Motor Power - CiA 402 State Machine

ID(Hex)	DLC	Data(Hex)	Description
603	8	2B 40 60 00 06 00 00 00	'Ready to Switch on
603	8	2B 40 60 00 07 00 00 00	'Switched on
603	8	2B 40 60 00 0F 01 00 00	'Operation Enabled; Motion Halted

### Set to Profile Velocity Mode

603	8	2F 60 60 00 03 00 00 00	'Set to Profile Velocity Mode
-----	---	-------------------------	-------------------------------

### Set Motion Parameters

603	8	23 FF 60 00 20 4E 00 00	'Set Target Velocity to 20000 step/s
603	8	23 83 60 00 40 9C 00 00	'Set Acceleration to 40000 step/s <sup>2</sup>
603	8	23 84 60 00 40 9C 00 00	'Set Deceleration to 40000 step/s <sup>2</sup>

### Start/Stop Motion

603	8	2B 40 60 00 0F 00 00 00	'Motion Starts
603	8	23 FF 60 00 80 38 01 00	'Change Target Velocity to 80000 step/s
603	8	2B 40 60 00 0F 01 00 00	'Motion Halts

## Homing Mode

### Enable Motor Power - CiA 402 State Machine

ID(Hex)	DLC	Data(Hex)	Description
603	8	2B 40 60 00 06 00 00 00	'Ready to Switch on
603	8	2B 40 60 00 07 00 00 00	'Switched on
603	8	2B 40 60 00 0F 00 00 00	'Operation Enabled

### Set to Homing Mode

603	8	2F 60 60 00 06 00 00 00	'Set to Homing Mode
603	8	2F 98 60 00 13 00 00 00	'Set Homing Method to 19

### Set Motion Parameters

603	8	23 9A 60 00 50 C3 00 00	'Set Homing Acceleration to 50000 step/s <sup>2</sup>
603	8	23 99 60 01 10 27 00 00	'Set Homing Velocity (Search for Switch) to 10000 step/s
603	8	23 99 60 02 88 13 00 00	'Set Index Velocity (Search for Index) to 5000 step/s
603	8	23 7C 60 00 40 9C 00 00	'Set Homing Offset to 40000 Steps
603	8	2F 01 20 00 03 00 00 00	'Set Homing Switch to Input 3

### Start/Stop Homing

603	8	2B 40 60 00 1F 00 00 00	'Homing Starts
603	8	2B 40 60 00 1F 01 00 00	'Homing Stops

## Normal Q Mode

### Enable Motor Power - CiA 402 State Machine

ID(Hex)	DLC	Data(Hex)	Description
603	8	2B 40 60 00 06 00 00 00	'Ready to Switch on
603	8	2B 40 60 00 07 00 00 00	'Switched on
603	8	2B 40 60 00 0F 00 00 00	'Operation Enabled

### Set to Normal Q Mode

603	8	2F 60 60 00 FF 00 00 00	'Set to Normal Q Mode
603	8	2F 07 20 00 01 00 00 00	'Set Q Segment Number to 1

### Start/Stop Q Program

603	8	2B 40 60 00 1F 00 00 00	'Q Program Starts
603	8	2B 40 60 00 1F 01 00 00	'Q Program Halts

## Sync Q Mode

### Enable Motor Power - CiA 402 State Machine

ID(Hex)	DLC	Data(Hex)	Description
603	8	2B 40 60 00 06 00 00 00	'Ready to Switch on
603	8	2B 40 60 00 07 00 00 00	'Switched on
603	8	2B 40 60 00 0F 00 00 00	'Operation Enabled

### Set to Sync Q Mode

603	8	2F 60 60 00 FE 00 00 00	'Set to Sync Q Mode
603	8	2F 07 20 00 01 00 00 00	'Set Q Segment Number to 1
603	8	23 05 10 00 80 00 00 00	'Set Sync Pulse to 0x80

### Start/Stop Q Program

80	0		'Q Program Starts
603	8	2B 40 60 00 0F 01 00 00	'Q Program Halts

## PDO Mapping

### Mapping TPDO2

ID(Hex)	DLC	Data(Hex)	Description
000	2	80 03	'Return back to "Pre-Operation" Mode
603	8	23 01 18 01 80 02 00 80	'Turn off the TPDO2
603	8	2F 01 1A 00 00 00 00 00	'Set Number of Mapped objects to zero
603	8	23 01 1A 01 10 00 41 61	'Map object 1(0x6041) to TPDO2 subindex1.
603	8	23 01 1A 02 20 00 0A 70	'Map object 2(0x700A) to TPDO2 subindex2.
603	8	2F 01 1A 00 02 00 00 00	'Set Number of total Mapped objects to two
603	8	23 01 18 01 80 02 00 00	'Turn on the TPDO2
603	8	23 10 10 01 73 61 76 65	'Save all parameter

## 9 Contact MOONS'

# Customer Service Center



# 400-820-9661

### ■ MOONS' Headquarter

168 Mingjia Road, Minhang District, Shanghai  
201107, P.R. China

### ■ Domestic Offices

#### Shenzhen

Rm.401, Building 53, Zhongchuang Park, Jiyue City, No.13,  
Xinyi 5th Road, Taoyuan Street, Nanshan Dist, Shenzhen  
518000, P.R. China

#### Beijing

Room 1206, Jing Liang Mansion, No.16 Middle Road of East,  
3rd Ring, Chaoyang District, Beijing 100022, P.R. China

#### Nanjing

Room 1101-1102, Building 2, New Town Development  
Center, No.126 Tianyuan Road, Moling Street,  
Jiangning District, Nanjing 211106, P.R. China

#### Qingdao

Room 1710, Zhuoyue Tower, No.16 Fengcheng Road,  
Shibei District, Qingdao 266000 P.R. China

#### Wuhan

Room 3001, World Trade Tower, 686 Jiefang Avenue,  
Jiangnan District, Wuhan 430022, P.R. China

#### Chengdu

Rm. 3907, Maoye Plaza, No.19, Dongyu Street,  
Jinjiang Distrit, Chengdu 610066, P.R. China

#### Xi'an

Room 1006, Tower D, Wangzuo International City,  
1 Tangyuan Road, Xi'an 710065, P.R. China

#### Ningbo

Room 309, Tower B, Taifu Plaza, 565 Jiangjia Road,  
Jiangdong District, Ningbo, 315040, P.R. China

#### Guangzhou

Room 4006, Tower B, China Shine Plaza, 9 Linhe Xi Road,  
Tianhe District, Guangzhou 510610, P.R. China

#### Chongqing

Rm. 2108, South yuanzhu Buliding 20, No.18 Fuquan Rd.,  
Jiangbei District, Chongqing 400000, P.R. China

### ■ North America Company

**MOONS' INDUSTRIES (AMERICA), INC.** (Chicago)  
1113 North Prospect Avenue, Itasca, IL 60143 USA

**MOONS' INDUSTRIES (AMERICA), INC.** (Boston)  
36 Cordage Park Circle, Suite 310 Plymouth, MA 02360 USA

**APPLIED MOTION PRODUCTS, INC.**  
404 Westridge Dr. Watsonville, CA 95076 USA

**LIN ENGINEERING, INC.**  
16245 Vineyard Blvd., Morgan Hill, CA 95037 USA

### ■ European Company

**MOONS' INDUSTRIES (EUROPE) S.R.L.**  
Via Torri Bianche n.1 20871 Vimercate(MB) Italy

**AMP & MOONS' AUTOMATION (GERMANY) GMBH**  
Börsenstr. 14  
60313 Frankfurt am Main Germany

### ■ South-East ASIA Company

**MOONS' INDUSTRIES (SOUTH-EAST ASIA) PTE. LTD.**  
33 Ubi Avenue 3 #08-23 Vertex Singapore 408868

### ■ Japan Company

**MOONS' INDUSTRIES JAPAN CO., LTD.**  
Room 602, 6F, Shin Yokohama Koushin Building,  
2-12-1, Shin-Yokohama, Kohoku-ku, Yokohama,  
Kanagawa, 222-0033, Janpan

### ■ India Company

**MOONS' INTELLIGENT MOTION SYSTEM INDIA PVT. LTD.**  
Rm. 908, 9th Floor, Amar Business Park,  
Tal. Haveli, Baner, Pune-411045, Maharashtra, India



<https://www.moonsindustries.com/>

E-mail: [ama-info@moons.com.cn](mailto:ama-info@moons.com.cn)

**MOONS'**  
*moving in better ways*

- All the specifications, technical parameters of the products provided in this catalog are for reference only, subject to change without notice. For the latest details, please contact our sales department.