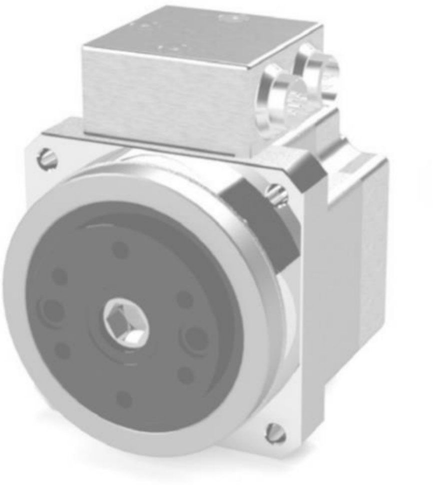
HARMONIC DRIVE LLC

Integrated Drive Technology

CANopen Firmware Manual

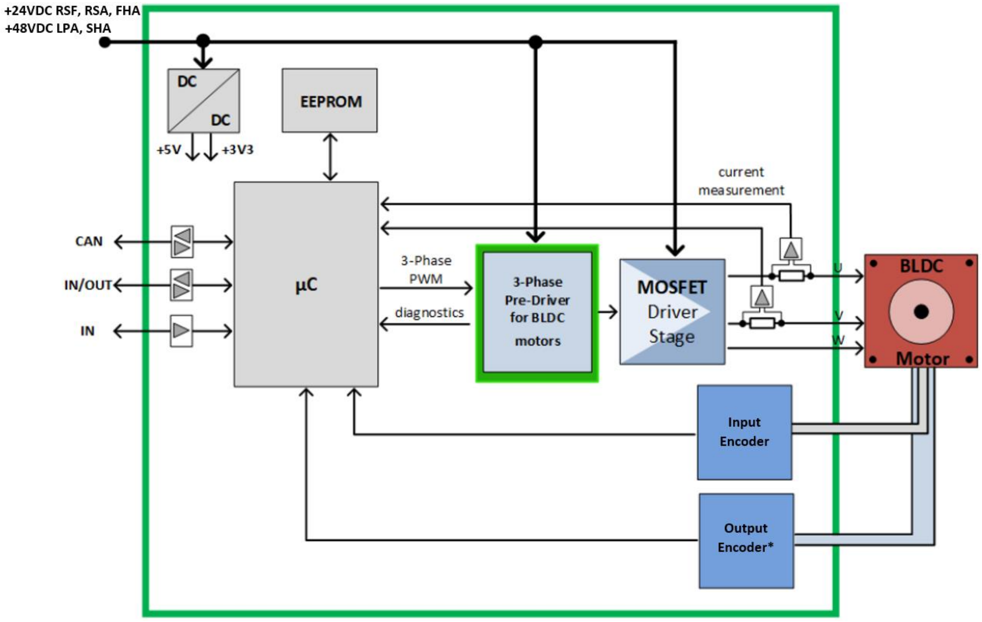


The Harmonic Drive LLC (HDLLC) Integrated Actuators include a single axis controller module for brushless DC (BLDC) motors. It offers field-oriented control (FOC) with a +24V DC supply, a single or dual absolute encoder depending on the model and I/O. A CAN interface allows communication with a CANopen master.

Features

- Single axis field-oriented control for BLDC/PMSM motors
- RSA/RSF Series: +7…30V DC supply voltage
- FHA-C Mini Series: +7…28V DC supply voltage
- LPA, SHA Series +24V…72V DC supply voltage
- Single or dual absolute encoders depending on model
- CAN interface
- CANopen CiA 402 drive profile
- PP, PV, CSP, CSV, CST modes
- Hard Stop Homing Modes
- Bode Plot, Step Response (CAN mode)
- Inputs:
  RSA/RSF: 1 programmable input (NPN/PNP)
  FHA-C mini, LPA, SHA series: 2 opto-isolated inputs (NPN/PNP)
- Outputs:
  RSA/RSF: 1 programmable IO
  FHA-C mini, LPA, SHA series: 2 programmable IO
- Brake on SHA
- Motor Torque Off (MTO) on LPA, SHA series

Simplified Block Diagram



*When available as described in the actuator’s specification

Contents

1 PREFACE ... 7

1.1 General Features of this CANopen Implementation ... 7
1.2 Abbreviations used in this Manual ... 8
1.3 Firmware Update ... 9

2 COMMUNICATION ... 10

2.1 Reference Model ... 10
2.2 NMT State Machine ... 11
2.3 System Model ... 12
2.4 Object Dictionary ... 13

3 COMMUNICATION AREA ... 15

3.1 Detailed object specifications ... 15
3.1.1 Object 1000h: Actuator Type ... 15
3.1.2 Object 1001$_h$: Error Register ... 15
3.1.3 Object 1005$_h$: COB-ID SYNC Message ... 16
3.1.4 Object 1008$_h$: HDLLC Actuator Type ... 17
3.1.5 Object 1009$_h$: HDLLC Hardware Version ... 17
3.1.6 Object 100A$_h$: HDLLC Firmware Version ... 18
3.1.7 Object 1010$_h$: Store Parameters ... 18
3.1.8 Object 1011$_h$: Restore Parameters ... 19
3.1.9 Object 1014$_h$: COB-ID Emergency Object ... 20
3.1.10 Object 1015$_h$: Inhibit Time EMCY ... 20
3.1.11 Object 1016$_h$: Consumer Heartbeat Time ... 21
3.1.12 Object 1017$_h$: Producer Heartbeat Time ... 21
3.1.13 Object 1018$_h$: Identity Object ... 22
3.1.14 Object 1029$_h$: Error Behavior ... 22
3.1.15 Objects 1400$_h$ – 1403$_h$: Receive PDO Communication Parameter ... 23
3.1.16 Objects 1600$_h$ – 1603$_h$: Receive PDO Mapping Parameter ... 24
3.1.17 Objects 1800$_h$ – 1803$_h$: Transmit PDO Communication Parameter ... 24
3.1.18 Objects 1A00$_h$ – 1A03$_h$: Transmit PDO Mapping Parameter ... 26

4 APPLICATION SPECIFIC ... 27

4.1 Detailed object specifications ... 27
4.1.1 Object 2005$_h$: Limit Switches ... 27
4.1.2 Object 200D$_h$: Status Flags ... 28
4.1.3 Object 200E$_h$: Supply Voltage ... 29
4.1.4 Object 200F$_h$: Driver Temperature ... 29
4.1.5 Object 2010$_h$: Motor Settings ... 30
4.1.6 Object 2015$_h$: Brake Settings ... 30
4.1.7 Object 2020$_h$: Limits ... 31
4.1.8 Object 2025$_h$: Homing Mode Settings ... 33
4.1.9 Object 2030$_h$: Torque Mode Settings ... 33
4.1.10 Object 2040$_h$: Velocity Mode Settings ... 34
4.1.11 Object 2050$_h$: Position Mode Settings ... 35
4.1.12 Object 2055$_h$: Commutation Mode ... 35
4.1.13 Object 2056$_h$: Velocity Ramp Mode ... 36
4.1.14 Object 2060$_h$: Open Loop Settings ... 36
4.1.15 Object 2080$_h$: Encoder Settings ... 36
4.1.16 Object 2090$_h$: IIt Monitor Settings ... 37
4.1.17 Object 2095$_h$: Velocity Window Settings ... 38

4.1.18 Object 2096h: Position Window Settings 38
4.1.19 Object 2100h: Home Offset Display 39
4.1.20 Object 2702h: Digital Inputs 39
4.1.21 Object 2703h: Device Digital Outputs 40
4.1.22 Object 2704h: CAN Bit Rate 40
4.1.23 Object 2705h: Node ID 41
4.1.24 Object 2706h: User Variables 41
4.1.25 Object 270Eh: Analog Inputs 41
4.1.26 Object 5FFFh: Bootloader Mode 42

5 PROFILE SPECIFIC AREA 43

5.1 DETAILED OBJECT SPECIFICATIONS 43
5.1.1 Object 605Ah: Quick Stop Option Code 43
5.1.2 Object 605Bh: Shutdown Option Code 44
5.1.3 Object 605Ch: Disable Operation Option Code 44
5.1.4 Object 605Dh: Halt Option Code 45
5.1.5 Object 605Eh: Fault Reaction Option Code 45
5.1.6 Object 6060h: Modes of Operation 46
5.1.7 Object 6061h: Modes of Operation Display 46
5.1.8 Object 608Fh: Position Encoder Resolution 47
5.1.9 Object 60FDh: Digital Inputs 48
5.1.10 Object 6502h: Supported Drive Modes 48

6 PROFILE POSITION MODE 50

6.1 DETAILED OBJECT SPECIFICATIONS 50
6.1.1 Object 6040h: Control Word 51
6.1.2 Object 6041h: Status Word 52
6.1.3 Object 6062h: Position Demand Value 53
6.1.4 Object 6063h: Position Actual Internal Value 54
6.1.5 Object 6064h: Position Actual Value 54
6.1.6 Object 6067h: Position Window 55
6.1.7 Object 606Bh: Velocity Demand Value 55
6.1.8 Object 606Ch: Velocity Actual Value 55
6.1.9 Object 607Ah: Target Position 56
6.1.10 Object 607Dh: Software Position Limit 56
6.1.11 Object 6081h: Profile Velocity (for pp mode) 57
6.1.12 Object 6082h: End Velocity 57
6.1.13 Object 6083h: Profile Acceleration 57
6.1.14 Object 6084h: Profile Deceleration 58
6.1.15 Object 6085h: Quick Stop Deceleration 58
6.2 EXAMPLE OF PP MODE OPERATION 59

7 PROFILE VELOCITY MODE 60

7.1 DETAILED OBJECT SPECIFICATIONS 60
7.1.1 Object 6040h: Control Word 60
7.1.2 Object 6041h: Status Word 61
7.1.3 Object 6062h: Position Demand Value 62
7.1.4 Object 6063h: Position Actual Internal Value 63
7.1.5 Object 6064h: Position Actual Value 63
7.1.6 Object 606Bh: Velocity Demand Value 63
7.1.7 Object 606Ch: Velocity Actual Value 64
7.1.8 Object 607Dh: Software Position Limit 64

7.1.9 Object 6083h: Profile Acceleration 64
7.1.10 Object 6085h: Quick Stop Deceleration 65
7.1.11 Object 60FFh: Target Velocity 65
7.2 EXAMPLE OF PV MODE OPERATION 66
8 HOMING MODE 67
8.1 HOMING METHODS 68
8.1.1 Homing Method 17 and 18: Homing without Index Pulse 68
8.1.2 Homing Method 35: Current Position as Home Position 68
8.1.3 Homing Method -1: Single Ended Clockwise Hard Stop Homing 69
8.1.4 Homing Method -2: Single Ended Counterclockwise Hard Stop Homing 69
8.1.5 Homing Method -3: Double Ended Clockwise Hard Stop Homing 70
8.1.6 Homing Method -4: Double Ended Counterclockwise Hard Stop Homing 70
8.1.7 Homing Method -5: Double Ended Clockwise Hard Stop Homing (compute scaler) 71
8.1.8 Homing Method -6: Double Ended Counterclockwise Hard Stop Homing (compute scaler) 71
8.1.9 Teach Mode 71
8.1.10 Parameter Saving After Homing 71
8.2 DETAILED OBJECT SPECIFICATIONS 72
8.2.1 Object 6040h: Control Word 72
8.2.2 Object 6041h: Status Word 73
8.2.3 Object 606Ch: Velocity Actual Value 74
8.2.4 Object 607Ch: Home Offset 75
8.2.5 Object 6098h: Homing Method 75
8.2.6 Object 6099h: Homing Speeds 75
8.2.7 Object 609Ah: Homing Acceleration 76
8.2.8 Object 2100h: Home Offset Display 76
8.3 EXAMPLE OF HOMING IN HM MODE 77
9 CYCLIC SYNCHRONOUS POSITION MODE 78
9.1 DETAILED OBJECT SPECIFICATIONS 78
9.1.1 Object 6040h: Control Word 78
9.1.2 Object 6041h: Status Word 79
9.1.3 Object 6062h: Position Demand Value 80
9.1.4 Object 6063h: Position Actual Internal Value 80
9.1.5 Object 6064h: Position Actual Value 81
9.1.6 Object 606Ch: Velocity Actual Value 81
9.1.7 Object 607Ah: Target Position 82
9.1.8 Object 607Dh: Software Position Limit 82
9.1.9 Object 60B0h: Position Offset 83
9.1.10 Object 60C2h: Interpolation Time Period 83
9.2 EXAMPLE OF CSP MODE OPERATION 84
10 CYCLIC SYNCHRONOUS VELOCITY MODE 85
10.1 DETAILED OBJECT SPECIFICATIONS 85
10.1.1 Object 6040h: Control Word 85
10.1.2 Object 6041h: Status Word 86
10.1.3 Object 606Ch: Velocity Actual Value 87
10.1.4 Object 60FFh: Target Velocity 88
10.1.5 Object 607Dh: Software Position Limit 88
10.1.6 Object 60B1h: Velocity Offset 89
10.1.7 Object 60C2h: Interpolation Time Period 89
10.2 EXAMPLE OF CSV MODE OPERATION 90
11 CYCLIC SYNCHRONOUS TORQUE MODE 91

11.1 DETAILED OBJECT SPECIFICATIONS 91
11.1.1 Object 6040h: Control Word 91
11.1.2 Object 6041h: Status Word 92
11.1.3 Object 6071h: Target Torque 93
11.1.4 Object 6077h: Torque Actual Value 94
11.1.5 Object 60B2h: Torque offset 94
11.2 EXAMPLE OF CST MODE OPERATION 94

12 IO SUMMARY 95

12.1 IO PORT MAPPING 95
12.2 DIGITAL INPUT AS LIMIT SWITCH 97
12.3 LED FOR NMT STATE 97
12.4 MOTOR TORQUE OFF 98

13 EMERGENCY MESSAGES (EMCY) 99

13.1 FORMAT OF EMCY MESSAGES 99
13.2 ERROR CODE 99

14 SDO ABORT CODES 101

15 FIGURES INDEX 103

16 TABLES INDEX 103

17 DOCUMENT REVISION 106

18 SUPPLEMENTAL DIRECTIVES 107

18.1 PRODUCER INFORMATION 107
18.2 COPYRIGHT 107
18.3 TRADEMARK DESIGNATIONS AND SYMBOLS 107
18.4 TARGET USER 107
18.5 DISCLAIMER: LIFE SUPPORT SYSTEMS 107
18.6 DISCLAIMER: INTENDED USE 108
18.7 SUPPLEMENTAL DOCUMENTS & TOOLS 108

1 Preface

This document specifies objects and modes of operation of the HDLLC actuator BLDC/PMSM motor control module with CANopen firmware. The CANopen firmware is designed to fulfill the CANopen DS402 and DS301 standards. This manual assumes that the reader is already familiar with the basics of the CANopen protocol, defined by the DS301 and DS402 standards of the CAN-CiA.

If necessary, it is always possible to convert the actuator from a CANopen version into an HDL version by loading the actuator HDL CAN firmware with the firmware update function of the HDL-IDE .

1.1 General Features of this CANopen Implementation

Main Characteristics

- Communication according to standard CiA-301 V4.1
- CAN bit rate: 20. . . 1000kBit/s
- CAN ID: 11 bit
- Node ID: 1. . . 127 (use vendor specific objects for changing the node ID)
- NMT services: NMT slave

SDO Communication

- 1 server
- Expedited transfer
- Segmented transfer
- No block transfer

PDO Communication

- Producer
- Consumer
- RPDOs
  - Supported RPDOs: 1, 2, 3, 4
  - Transmission modes: asynchronous, synchronous.
  - Dynamic mapping with max. 3 mapping entries.
  - Default mappings: according to CiA-402 for first three PDOs, HDLLC specific for other PDOs of each axis.
- TPDOs
  - Supported TPDOs: 1, 2, 3, 4
  - Transmission modes: asynchronous, asynchronous with event timer, synchronous.
  - Dynamic mapping with max. 3 mapping entries.
    Default mappings: according to CiA-402 for first three PDOs, HDLLC specific for other PDOs.

Further Characteristics

- SYNC: consumer
- Emergency: producer
- Heartbeat: consumer and producer

1.2 Abbreviations used in this Manual

Abbreviations	
CAN	Controller area network
CHGND	Chassis ground / earth ground
COB	Communication object
CST	Cyclic synchronous torque mode
CSV	Cyclic synchronous velocity mode
CSP	Cyclic synchronous position mode
EMCY	Emergency object
FOC	Field Oriented Control
FSA	Finite state automaton
FSM	Finite state machine
hm	Homing mode
ID	Identifier
LSB	Least significant bit
MSB	Most significant bit
NMT	Network management
PDO	Process data object
PDS	Power drive system
PMSM	Permanent Magnet Synchronous Motor
pp	Profile position mode
pv	Profile velocity mode
RPDO	Receive process data object
RTR	Remote Transmission Request
SDO	Service data object
TPDO	Transmit process data object
rw	Read and write
ro	Read only

Table 1: Abbreviations used in this Manual

1.3 Firmware Update

The software running on the microprocessor consists of two parts, a bootloader and the CANopen firmware itself. Whereas the bootloader is installed during production and testing and remains un-touched throughout the whole lifetime, the CANopen firmware can easily be updated by the user. The new firmware can be loaded into the module via the firmware update function of the HDL-IDE. Entering boot mode is also possible by writing 12345678_h to object $5FFF_h$ and use the CAN interface for a firmware update with the HDL-IDE.

2 Communication

2.1 Reference Model

The application layer comprises a concept to configure and communicate real-time-data as well as the mechanisms for synchronization between actuators. The functionality which the application layer offers to an application is logically divided over different service data objects (SDO) in the application layer. A service object offers a specific functionality and all the related services.

Applications interact by invoking services of a service object in the application layer. To realize these services this object exchanges data via the CAN Network with peer service object(s) using a protocol.

The application and the application layer interact with service primitives.

Service Primitives	
Primitive	Definition
Request	Issued by the application to the application layer to request a service.
Indication	Issued by the application layer to the application to report an internal event detected by the application layer or indicate that a service is requested.
Response	Issued by the application to the application layer to respond to a previously received indication.
Confirmation	Issued by the application layer to the application to report the result of a previously issued request.

Table 2: Service Primitives

A service type defines the primitives that are exchanged between the application layer and the cooperating applications for a particular service of a service object. Unconfirmed and confirmed services are collectively called remote services.

Service Types	
Type	Definition
Local service	Involves only the local service object. The application issues a request to its local service object that executes the requested service without communicating with peer service object(s).
Unconfirmed service	Involves one or more peer service objects. The application issues a request to its local service object. This request is transferred to the peer service object(s) that each passes it to their application as an indication. The result is not confirmed back.
Confirmed service	Can involve only one peer service object. The application issues a request to its local service object. This request is transferred to the peer service object that passes it to the other application as an indication. The other application issues a response that is transferred to the originating service object that passes it as a confirmation to the requesting application.
Provider initiated service	Involves only the local service object. The service object (being the service provider) detects an event not solicited by a requested service. This event is then indicated to the application.

Table 3: Service Types

2.2 NMT State Machine

The finite state machine (FSM) or simply state machine is a model of behavior composed of a finite number of states, transitions between those states, and actions. It shows which way the logic runs when certain conditions are met.

Starting and resetting the actuator is controlled via the state machine. The NMT state machine consists of the states shown in Figure 1.

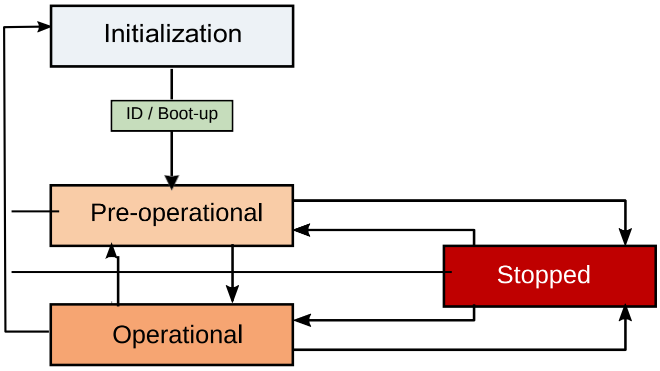


Figure 1: NMT State Machine

After power-on or reset the actuator enters the Initialization state. After the actuator initialization is finished, the actuator automatically transits to the **Pre-operational** state and indicates this state transition by sending the boot-up message. This way the actuator indicates that it is ready to work. An actuator that stays in Pre-operational state may start to transmit SYNC-, time stamp- or heartbeat message. In contrast to the PDO communication that is disabled in this state, the actuator can communicate via SDO.

The PDO communication is only possible within the **Operational** state. During Operational state the actuator can use all supported communication objects.

An actuator that was switched to the **Stopped** state only reacts on received NMT commands. In addition, the actuator indicates the current NMT state by supporting the error control protocol during Stopped state.

The transitions between states are made by issuing a network management (NMT) communication object to the actuator. The NMT protocols are used to generate state machine change commands (e.g. to start and stop the actuator), detect remote actuator boot-ups and error conditions.

The Heartbeat message of a CANopen actuator contains the actuator status of the NMT state machine and is sent cyclically by the CANopen actuator.

The NMT state machine (or DS301 state machine) is not to be confused with the DS402 state machine. There is only one NMT state machine for the entire system, but for each actuator, there is a DS402 state machine which controls the actuator. There are no links between these state machines, with one exception: When the NMT state machine is being switched to the stopped state, all DS402 state machines that are in OPERATION_ENABLED state will be switched to FAULT state.

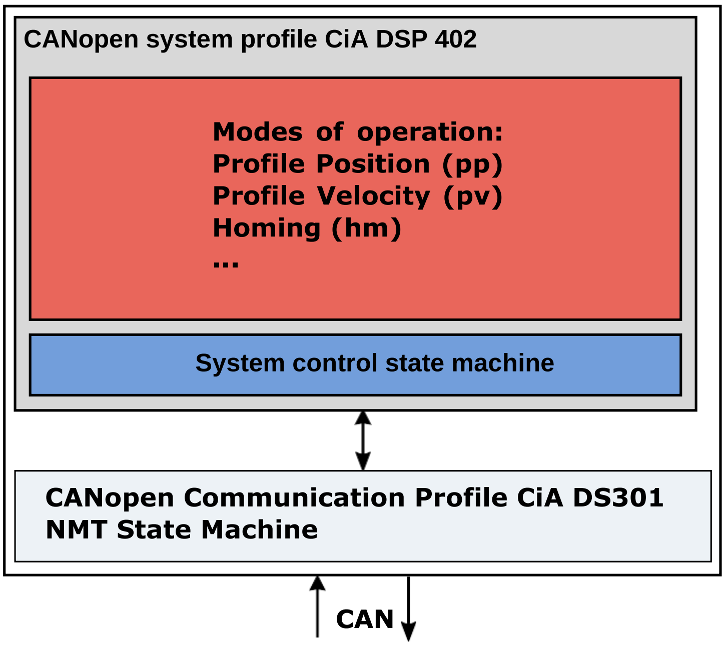

CANopen system profile CiA DSP 402
Modes of operation:
Profile Position (pp)
Profile Velocity (pv)
Homing (hm)
...
System control state machine
CANopen Communication Profile CiA DS301
NMT State Machine
CAN

Figure 2: Communication Architecture

2.3 System Model

A CANopen device mainly consists of the following parts:

- *Communication:* This function unit provides the communication objects and the appropriate functionality to transport data items via the underlying network structure.
- *Object dictionary:* The object dictionary is a collection of all the data items which have an influence on the behavior of the application objects, the communication objects and the state machine used on this actuator.
- *Application:* The application comprises the functionality of the actuator with respect to the interaction with the process environment.

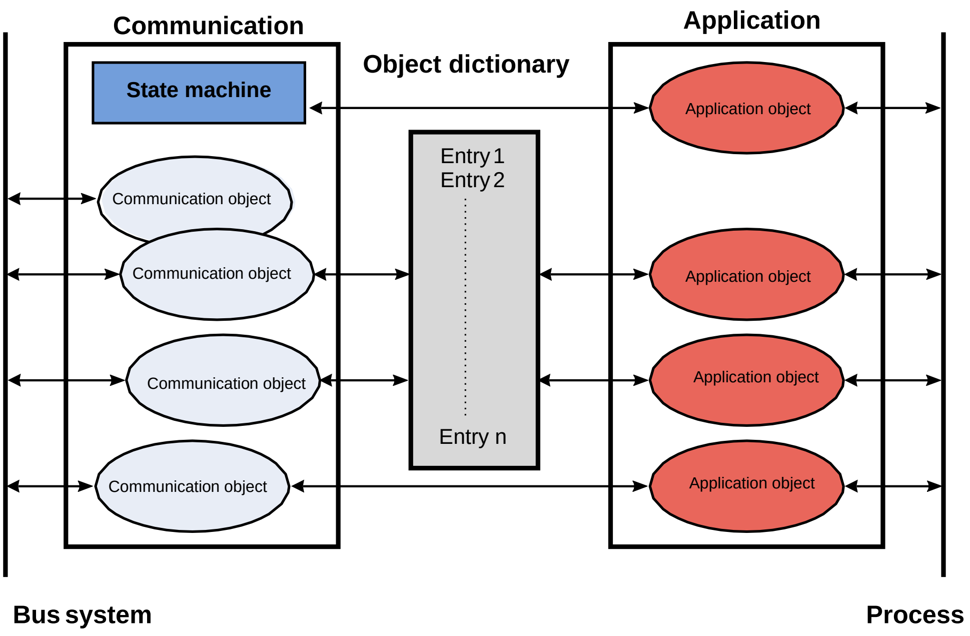


Figure 3: System Model

2.4 Object Dictionary

The most important part of an actuator profile is the object dictionary description. The object dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. Each object within the dictionary is addressed using a 16-bit index. The overall layout of the standard object dictionary is shown in Table 4:

Object Dictionary	
Index	Object
0000h	Not used.
0001h – 001Fh	Static data types.
0020h – 003Fh	Complex data types.
0040h – 005Fh	HDLLC specific complex data types.
0060h – 007Fh	Actuator profile specific static data types.
0080h – 009Fh	Actuator profile specific complex data types.
00A0h – 0FFFh	Reserved for further use.
1000h – 1FFFh	Communication profile area.
2000h – 5FFFh	HDLLC specific profile area.
6000h – 9FFFh	Standardized actuator profile area.
A000h – BFFFh	Standardized interface profile area.
C000h – FFFFh	Reserved for further use.

Table 4: Object Dictionary

The communication profile area at indices 1000_h through $1FFF_h$ contains the communication specific parameters for the CAN network. These entries are common to all actuators.

The HDLLC segment at indices 2000_h through $5FFF_h$ contains HDLLC specific objects. These objects control the special features of the HDLLC actuator.

The standardized actuator profile area at indices 6000_h through $9FFF_h$ contains all data objects common to a class of actuators that can be read or written via the network. They describe the actuator parameters and the actuator functionality of the actuator profile.

3 Communication area

The communication area contains all objects that define the communication parameters of the CANopen actuator according to the DS301 standard.

3.1 Detailed object specifications

3.1.1 Object 1000h: Actuator Type

This object contains information about the actuator type. The object 1000_h describes the type of actuator and its functionality. It is composed of a 16-bit field which describes the actuator profile that is used and a second 16-bit field which provides additional information about optional functionality of the actuator.

Object Description			
Index	Name	Object Type	Data Type
1000_h	Actuator type	Variable	UNSIGNED32

Table 5: Object Description (1000_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	UNSIGNED32	00420192_h

Table 6: Entry Description (1000_h)

3.1.2 Object 1001_h: Error Register

This object contains error information. The CANopen device maps internal errors into object 1001_h. It is part of an emergency object.

Object Description			
Index	Name	Object Type	Data Type
1001_h	Error register	Variable	UNSIGNED8

Table 7: Object Description (1001_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	UNSIGNED8	0

Table 8: Entry Description (1001_h)

Error Register Bits	
Bit	Definition
0	Generic error
1	Current
2	Voltage
3	Temperature
4	Communication error
5	Actuator profile specific
6	Reserved (always 0)
7	HDLLC specific

Table 9: Error Register Bits

3.1.3 Object 1005$_h$: COB-ID SYNC Message

This object defines the COB-ID of the synchronization object (SYNC). Further, it defines whether the module generates the SYNC.

		Value Definition
Bit	Name	Definition
30	Generate	0: Actuator does not generate SYNC message 1: Actuator generates SYNC message
29	Frame	Not supported, always set to 0.
28. . . 11	29 bit ID	Not supported, always set to 0.
10. . . 0	11 bit ID	11 bit COB-ID.

Table 10: Value Definition (1005$_h$)

Object Description			
Index	Name	Object Type	Data Type
1005$_h$	COB-ID SYNC message	Variable	UNSIGNED32

Table 11: Object Description (1005$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	80$_h$

Table 12: Entry Description (1005$_h$)

3.1.4 Object 1008$_h$: HDLLC Actuator Type

This object contains the HDLLC actuator type.

Object Description			
Index	Name	Object Type	Data Type
1008$_h$	HDLLC Actuator Type	Variable	Visible String

Table 13: Object Description (1008$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	Depends on actuator

Table 14: Entry Description (1008$_h$)

3.1.5 Object 1009$_h$: HDLLC Hardware Version

This object contains the hardware version description.

Object Description			
Index	Name	Object Type	Data Type
1009$_h$	HDLLC Hardware Version	Variable	Visible String

Table 15: Object Description (1009$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	Depends on actuator, e.g. 1.0.

Table 16: Entry Description (1009$_h$)

3.1.6 Object 100A$_h$: HDLLC Firmware Version

This object contains the firmware version description.

Object Description			
Index	Name	Object Type	Data Type
100A$_h$	HDLLC Firmware Version	Variable	Visible String

Table 17: Object Description (100A$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	Depends on actuator, e.g. 1.0.

Table 18: Entry Description (100A$_h$)

3.1.7 Object 1010$_h$: Store Parameters

This object supports the saving of parameters in non-volatile memory. By read access the actuator provides information about its saving capabilities.

There are several parameter groups:

- Sub-index 0$_h$: contains the largest sub-index that is supported.
- Sub-index 1$_h$: saves all parameters.
- Sub-index 2$_h$: saves communication parameters 2704$_h$ and 2705$_h$.
- Sub-index 3$_h$: saves actuator profile parameters (not used).
- Sub-index 4$_h$: saves actuator 0 parameters.

Note In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate sub-Index. This signature is "save" (65766173$_h$, see also table 19).

Save Signature			
e	v	a	s
65$_h$	76$_h$	61$_h$	73$_h$

Table 19: Save Signature

On reception of the correct signature in the appropriate sub-index the drive stores the parameter and then confirms the SDO transmission (initiate download response). If the storing failed, the drive responds with an abort SDO transfer (abort code: 06060000$_h$). If a wrong signature is written, the actuator refuses to store and responds with abort SDO transfer (abort code: 0800002x$_h$).

On read access, each sub-index provides information if it is possible to store the parameter group. It reads 1 if yes and 0 if no.

Object Description			
Index	Name	Object Type	Data Type
1010h	Store Parameters	Array	UNSIGNED32

Table 20: Object Description (1010h)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
00h	Highest supported sub-index	ro	no	UNSIGNED8	4
01h	Save all parameters	rw	no	UNSIGNED32	1
02h	Save communication parameters	rw	no	UNSIGNED32	1
03h	Save actuator profile parameters	rw	no	UNSIGNED32	0
04h	Save motor axis 0 parameters	rw	no	UNSIGNED32	1

Table 21: Entry Description (1010h)

3.1.8 Object 1011h: Restore Parameters

With this object the default values of parameters according to the communication or actuator profile are restored. By read access the actuator provides information about its capabilities to restore these values.

There are several parameter groups:

- Sub-index 0h: contains the largest sub-index that is supported.
- Sub-index 1h: restores all parameters (factory reset).
- Sub-index 2h: no function.
- Sub-index 3h: restores actuator profile parameters (not used).
- Sub-index 4h: restores actuator 0 parameters.

Note In order to avoid restoring the parameters by mistake, restoring is only executed when a specific signature is written to the appropriate sub-Index. This signature is "load" (64616F6Ch, see also table22).

Load Signature			
d	a	o	l
64h	61h	6Fh	6Ch

Table 22: Load Signature

On reception of the correct signature in the appropriate sub-index the actuator restores the parameter and then confirms the SDO transmission (initiate download response). If the restoring failed, the actuator responds with an abort SDO transfer (abort code: 06060000h). If a wrong signature is written, the actuator refuses to restore and responds with abort SDO transfer (abort code: 0800002xh).

On read access, each sub-index provides information if it is possible to restore the parameter group. It reads 1 if yes and 0 if no.

After the default values have been restored, they will become active after the next rest or power cycle of the actuator.

Object Description			
Index	Name	Object Type	Data Type
1011_h	Restore parameters	Array	UNSIGNED32

Table 23: Object Description (1011_h)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Highest supported sub-index	ro	no	UNSIGNED8	4
01_h	Restore all parameters	rw	no	UNSIGNED32	1
02_h	Restore communication parameters	rw	no	UNSIGNED32	1
03_h	Restore actuator profile parameters	rw	no	UNSIGNED32	0
04_h	Restore actuator axis 0 parameters	rw	no	UNSIGNED32	1

Table 24: Entry Description (1011_h)

3.1.9 Object 1014_h: COB-ID Emergency Object

This object defines the COB-ID of the emergency object (EMCY).

Object Description			
Index	Name	Object Type	Data Type
1014_h	COB-ID emergency object	Variable	UNSIGNED32

Table 25: Object Description (1014_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	80_h + Node ID

Table 26: Entry Description (1014_h)

3.1.10 Object 1015_h: Inhibit Time EMCY

The inhibit time for the EMCY message can be adjusted via this entry. The time has to be a multiple of $100\mu s$.

Object Description			
Index	Name	Object Type	Data Type
1015_h	COB-ID emergency object	Variable	UNSIGNED16

Table 27: Object Description (1015_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED16	0

Table 28: Entry Description (1015_h)

3.1.11 Object 1016$_h$: Consumer Heartbeat Time

The consumer heartbeat time defines the expected heartbeat cycle time and thus has to be higher than the corresponding producer heartbeat time configured on the module producing this heartbeat. The monitoring starts after the reception of the first heartbeat. If the consumer heartbeat time is 0 the corresponding entry is not used. The time has to be a multiple of 1ms.

Value Definition		
Bits	Name	Definition
31. . . 24	Reserved	—
23. . . 16	Node ID	Heartbeat Producer Node ID
15. . . 0	Heartbeat time	Time in 1ms

Table 29: Value Definition (1016$_h$)

Object Description			
Index	Name	Object Type	Data Type
1016$_h$	Consumer heartbeat time	Array	UNSIGNED32

Table 30: Object Description (1016$_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Number of entries	rw	no	UNSIGNED8	1
1	Consumer heartbeat time1	rw	no	UNSIGNED32	0

Table 31: Entry Description (1016$_h$)

3.1.12 Object 1017$_h$: Producer Heartbeat Time

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

Object Description			
Index	Name	Object Type	Data Type
1017$_h$	Producer heartbeat time	Variable	UNSIGNED16

Table 32: Object Description (1017$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED16	0

Table 33: Entry Description (1017$_h$)

3.1.13 Object 1018$_h$: Identity Object

The object 1018$_h$ contains general information about the actuator:

- The HDLLC ID (sub-index 01$_h$) contains the CiA assigned value of 443$_h$.
- The HDLLC specific product code (sub-index 2$_h$) identifies a specific actuator version.
- The HDLLC specific revision number (sub-index 3$_h$) consists of a major revision number and a minor revision number.

Object Description			
Index	Name	Object Type	Data Type
1018$_h$	Identity object	Record	Identity

Table 34: Object Description (1018$_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
00$_h$	Number of entries	ro	no	0. . . 3	3
01$_h$	Vendor ID	ro	no	UNSIGNED32	0443$_h$
02$_h$	Product code	ro	no	UNSIGNED32	e.g.10008 $_h$
03$_h$	Revision number	ro	no	UNSIGNED32	e.g. 20003$_h$ for version 2.3

Table 35: Entry Description (1018$_h$)

3.1.14 Object 1029$_h$: Error Behavior

If an actuator failure is detected in operational state, the actuator can be configured to enter alternatively the stopped state or remain in the current state in case of an actuator failure. Actuator failures include the following errors:

- Communication error
- Application error

Object Description			
Index	Name	Object Type	Data Type
1029$_h$	Error behavior	Array	UNSIGNED8

Table 36: Object Description (1029$_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
00$_h$	Number of error classes	ro	no	—	2
01$_h$	Communication error	rw	no	UNSIGNED8	0 (enter stopped state)
02$_h$	Application error	rw	no	UNSIGNED8	1 (remain in current state)

Table 37: Entry Description (1029$_h$)

3.1.15 Objects 1400h – 1403h: Receive PDO Communication Parameter

This object contains the communication parameters for the RPDOs which the actuator is able to receive. The sub-index 00h contains the number of valid entries within the communication record. Its value normally is 2, as this object consists of two other entries.

Sub-index 01h contains the COB-ID used by this PDO (in bits 10. . . 0). Bit 30 (RTR bit) defines if this PDO uses RTRs. As RTRs are not supported for PDOs by this CANopen implementation, this bit must always be set in order to turn off RTR support for this PDO. Bit 31 defines if this PDO is active or not. If this bit is set, the PDO is inactive, and if this bit is clear, the PDO is active. Before making any changes to a PDO definition, set this bit to inactivate the PDO.

Sub-Index 02h contains the transmission type of the RPDO. This can be FFh or FEh for event-driven, or 1…240 for synchronous (1 means that the PDO will be processed with every SYNC message, and 4 for example means that the PDO will be processed with every 4th SYNC message).. Other values are not supported.

Object Description			
Index	Name	Object Type	Data Type
1400h – 1403h	Receive PDO parameter	RECORD	RPDO CommPar
1400h	RPDO 1	RECORD	RPDO CommPar
1401h	RPDO 2	RECORD	RPDO CommPar
1402h	RPDO 3	RECORD	RPDO CommPar
1403h	RPDO 4	RECORD	RPDO CommPar

Table 38: Object Description (1400h)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00h	Largest sub-index supported	ro	2	2
01h	COB-ID used by PDO	rw	UNSIGNED32	Index 1400h: 40000200h + Node-ID Index 1401h: 40000300h + Node-ID Index 1402h: 40000400h + Node-ID Index 1403h: 40000500h + Node-ID
02h	Transmission type	rw	UNSIGNED8	Index 1400h: FFh Index 1401h: FFh Index 1402h: FFh Index 1403h: FFh

Table 39: Entry Description (1400h)

3.1.16 Objects 1600_h – 1603_h: Receive PDO Mapping Parameter

These objects contain the mapping parameters for the RPDOs the actuator is able to receive. The sub-index 00_h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be received with the corresponding RPDO. The sub-indices from 01_h to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length.

Object Description			
Index	Name	Object Type	Data Type
1600_h – 1603_h	Receive PDO mapping parameter	RECORD	PDO Mapping
1600_h	RPDO 1	RECORD	PDO Mapping
1601_h	RPDO 2	RECORD	PDO Mapping
1602_h	RPDO 3	RECORD	PDO Mapping
1603_h	RPDO 4	RECORD	PDO Mapping

Table 40: Object Description (1600_h)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00_h	Number of mapped application objects in PDO	rw	0. . . 3	Index 1600_h: 1 Index 1601_h: 2 Index 1602_h: 2 Index 1603_h: 2
01_h	Mapping entry 1	rw	UNSIGNED32	Index 1600_h: 60400010_h Index 1601_h: 60400010_h Index 1602_h: 60400010_h Index 1603_h: 60400010_h
02_h	Mapping entry 2	rw	UNSIGNED32	Index 1600_h: 0 Index 1601_h: 60600008_h Index 1602_h: $607A0020_h$ Index 1603_h: $60FF0020_h$
03_h	Mapping entry 3	rw	UNSIGNED32	Index 1600_h: 0_h Index 1601_h: 0_h Index 1602_h: 0_h Index 1603_h: 0_h

Table 41: Entry Description (1600_h)

Before making changes to PDO definitions, first mark the PDO as inactive by setting bit 31 of its COB-ID (see section 3.1.15). Then, set its number of mapped PDO entries to zero (sub-index 0 of the appropriate PDO mapping object). Now, the mappings themselves can be changed. After that, set the number of map objects to the desired value, and finally activate the PDO by clearing bit 31 of its COB-ID.

3.1.17 Objects 1800_h – 1803_h: Transmit PDO Communication Parameter

This object contains the communication parameters for the TPDOs which the actuator is able to transmit. The sub-index 00_h contains the number of valid entries within the communication record. Its value normally is 5, as this object consists of five other entries.

Sub-index 01_h contains the COB-ID used by this PDO (in bits 10. . . 0). Bit 30 (RTR bit) defines if this PDO uses RTRs. As RTRs are not supported for PDOs by this CANopen implementation, this bit must always be set in order to turn off RTR support for this PDO. Bit 31 defines if this PDO is active or not. If this bit is set, the PDO is inactive, and if this bit is clear, the PDO is active. Before making any changes to a PDO definition, set this bit to inactivate the PDO.

Sub-index 02_h contains the transmission type of the RPDO. This can be FF_h or FE_h for event-driven, or 1 …240 for synchronous (1 means that the PDO will be sent with every SYNC message, and 4 for example means that the PDO will be sent with every 4th SYNC message). Other values are not supported.

Sub-index 03_h contains the inhibit time, given in 0.1ms. After a TPDO has been sent, it will not be sent again before the inhibit time has elapsed.

Sub-index 04_h is not used.

Sub-index 05_h contains the event timer value in milliseconds. When this is set to a value greater than 0 the TPDO will be sent repeatedly each time the event timer has elapsed. For example, when this value is set to 250, the TPDO will be sent every 250ms. It will also be sent when the value has changed before the event timer has elapsed, but not before the inhibit time has elapsed

Object Description			
Index	Name	Object Type	Data Type
1800_h – 1803_h	Transmit PDO communication parameter	RECORD	TPDO CommPar
1800_h	TPDO 1	RECORD	TPDO CommPar
1801_h	TPDO 2	RECORD	TPDO CommPar
1802_h	TPDO 3	RECORD	TPDO CommPar
1803_h	TPDO 4	RECORD	TPDO CommPar

Table 42: Object Description (1800_h)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00_h	Largest sub-index supported	ro	5	5
01_h	COB-ID	rw	UNSIGNED32	Index 1800_h: 40000180_h + Node-ID Index 1801_h: 40000280_h + Node-ID Index 1802_h: 40000380_h + Node-ID Index 1803_h: 40000480_h + Node-ID
02_h	Transmission type	rw	UNSIGNED8	Index 1800_h: FF_h Index 1801_h: FF_h Index 1802_h: 01_h Index 1803_h: 01_h
03_h	Inhibit time	rw	UNSIGNED16	0
04_h	Compatibility entry	ro	UNSIGNED8	0
05_h	Event timer	rw	UNSIGNED16	0

Table 43: Entry Description (1800_h)

3.1.18 Objects $1A00_h$ – $1A03_h$: Transmit PDO Mapping Parameter

These objects contain the mapping parameters for the TPDOs the actuator is able to transmit. The sub-index 00_h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be transmitted with the corresponding TPDO. The sub-indices from 01_h to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length.

Object Description			
Index	Name	Object Type	Data Type
$1A00_h$ – $1A03_h$	Transmit PDO mapping parameter	RECORD	PDO Mapping
$1A00_h$	TPDO 1	RECORD	PDO Mapping
$1A01_h$	TPDO 2	RECORD	PDO Mapping
$1A02_h$	TPDO 3	RECORD	PDO Mapping
$1A03_h$	TPDO 4	RECORD	PDO Mapping

Table 44: Object Description ($1A00_h$)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00_h	Number of Mapped application objects in PDO	rw	0. . . 3	Index $1A00_h$: 1 Index $1A01_h$: 2 Index $1A02_h$: 2 Index $1A03_h$: 2
01_h	Mapping entry 1	rw	UNSIGNED32	Index $1A00_h$: 60410010_h Index $1A01_h$: 60410010_h Index $1A02_h$: 60410010_h Index $1A03_h$: 60410010_h
02_h	Mapping entry 2	rw	UNSIGNED32	Index $1A00_h$: 0 Index $1A01_h$: 60610008_h Index $1A02_h$: 60640020_h Index $1A03_h$: $606C0020_h$
03_h	Mapping entry 3	rw	UNSIGNED32	Index $1A00_h$: 0_h Index $1A01_h$: 0_h Index $1A02_h$: 0_h Index $1A03_h$: 0_h

Table 45: Entry Description ($1A00_h$)

Before making changes to PDO definitions, first mark the PDO as inactive by setting bit 31 of its COB-ID (see section 3.1.17). Then, set its number of mapped PDO entries to zero (sub-index 0 of the appropriate PDO mapping object). Now, the mappings themselves can be changed. After that, set the number of map objects to the desired value, and finally activate the PDO by clearing bit 31 of its COB-ID.

4 Application Specific

The application segment contains application specific objects. These objects control the special features of the actuators.

4.1 Detailed object specifications

4.1.1 Object 2005$_h$: Limit Switches

This object defines which limit switches are to be used. Bit 0 stands for the left and bit 1 stands for the right limit switch. If a bit is set, the corresponding limit switch will not be used. This object has to be set to the value 3 if limit switches are not connected. The object can only be written when the drive is in the SWITCHED_ON_DISABLED state (but is always readable).

The limit switches can also be inverted using bit 2 and bit 3:

- Bit 2 inverts the left limit switch
- Bit 3 inverts the right limit switch

Object Description			
Index	Name	Object Type	Data Type
2005$_h$	Limit switches	Variable	UNSIGNED32

Table 46: Object Description (2005$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0. . . 15	0

Table 47: Entry Description (2005$_h$)

Bit Definitions	
Bit	Definition
0	Left limit switch deactivated, if set.
1	Right limit switch deactivated, if set.
2	Left limit switch inverted, if set.
3	Right limit switch inverted, if set.

Table 48: Bit Definitions (2005$_h$)

4.1.2 Object 200D$_h$: Status Flags

This object provides information about the actual module status flags. (0: not active, 1: active). This object is organized bit-wise. The bits have the following meaning:

Status Flags		
Bit	Name	Definition
0	OVERCURRENT	Motor current too high.
1	UNDERVOLTAGE	Supply voltage too low.
2	OVERVOLTAGE	Supply voltage too high.
3	OVERTEMPERATURE	Driver temperature too high.
4	MOTORHALTED	Motor halted due to error.
5	DRIVER_ENABLED	Motor driver is enabled.
6	DRIVER_ERROR	Motor driver error.
7	INIT_ERROR	Error during motor initialization.
8	STOP_MODE	Drive in stop mode.
9	VELOCITY_MODE	Drive in velocity mode.
10	POSITION_MODE	Drive in position mode.
11	TORQUE_MODE	Drive in torque mode.
12	VELOCITY_WINDOW_ERROR	Velocity window has been exceeded.
13	POSITION_WINDOW_ERROR	Position window has been exceeded.
14	POSITION_END	Target position reached.
15	MODULE_INITIALIZED	Drive successfully initialized.
16	BRAKE_ACTIVE	Brake output on.
17	IIT_EXCEEDED_1	IIT1 limit has been exceeded.
18	IIT_EXCEEDED_2	IIT2 limit has been exceeded.
19	unused	
20	HOMED	Homing successfully finished.
21	HOMING	Homing active.
22	MIN_POS_LIMIT	Minimum position limit reached.
23	MAX_POS_LIMIT	Maximum position limit reached.
24	HOMING_ERROR	Error during homing.

Table 49: Status Flags (200D$_h$)

Object Description			
Index	Name	Object Type	Data Type
200D$_h$	Status Flags	Variable	UNSIGNED32

Table 50: Object Description (200D$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Status Flags	no	0	4294967295	0		ro

Table 51: Entry Description (200Dh)

4.1.3 Object 200Eh: Supply Voltage

The actual supply voltage in 100mV.

Object Description			
Index	Name	Object Type	Data Type
200Eh	Supply Voltage	Variable	UNSIGNED32

Table 52: Object Description (200Eh)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Supply Voltage	no	0	Depends on actuator	240	[100mV]	ro

Table 53: Entry Description (200Eh)

4.1.4 Object 200Fh: Driver Temperature

The actual temperature of the motor driver.

Object Description			
Index	Name	Object Type	Data Type
200Fh	Driver Temperature	Variable	SIGNED32

Table 54: Object Description (200Fh)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Driver Temperature	no	-20	120	0	[degree C]	ro

Table 55: Entry Description (200Fh)

4.1.5 Object 2010_h: Motor Settings

Object Description			
Index	Name	Object Type	Data Type
2010_h	Motor Settings	Variable	Record

Table 56: Object Description (2010_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Motor Poles	no	2	254	Depends on actuator		rw
2	Gear Ratio	no	1	200	Depends on actuator		rw

Table 57: Entry Description (2010_h)

4.1.6 Object 2015_h: Brake Settings

SHA series actuators come with a safety brake. This Object contains all settings for configuring the brake output

Object Description			
Index	Name	Object Type	Data Type
2015_h	Brake Settings	Variable	Record

Table 58: Object Description (2015_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	BrakeMode	no	0	2	0		rw

Table 59: Entry Description (2015_h)

Brake modes:
0: Apply (brake is engaged)
1: Hold (low current to maintain the brake release)
2: Release (high current for initial brake release)

For SHA-20 actuator, there are only two brake modes:
0: Apply (brake is engaged)
1: Release (current applied to maintain the brake release)

4.1.7 Object 2020$_h$: Limits

This object contains the software limits for torque, velocity and acceleration.

Object Description			
Index	Name	Object Type	Data Type
2020$_h$	Limits	Variable	Record

Table 60: Object Description (2020$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	MaxTorque	no	0	Depends on actuator	Depends on actuator	[mA] (peak)	rw
2	MaxVelocity	no	0	Depends on actuator	Depends on actuator	[rpm]	rw
3	MaxAcceleration	no	0	Depends on actuator	Depends on actuator	[rpm/s]	rw
4	DigitalInput Threshold	no	0	24000	10000	[mV]	rw

Table 61: Entry Description (2020$_h$)

Maximum Values for Object 2020h			
Unit(Ratio)	MaxTorque (Current Threshold, mA$_{Pk}$)	MaxVelocity(rpm)	MaxAcceleration (rpm/s)

RSF5(30)
RSF5(50)
RSF5(100)
	3252
	3111
	2404
		10000	400000

RSA8(30)
RSA8(50)
RSA8(100)
	5656
	5656
	4949
		8500	100000

FHA-8(30)
FHA-8(50)
FHA-8(100)
	4242
	4666
	3394
		6000	100000

FHA-11(30)
FHA-11(50)
FHA-11(100)
	11029
	11595
	7918
		6000	100000

FHA-14(30)
FHA-14(50)
FHA-14(100)
	20927
	23190
	17392
		6000	100000

LPA-20(51)
LPA-20(101)
	18948
	16261
		5000	100000

SHA-20(51)
SHA-20(81)
SHA-20(101)
SHA-20(121)
SHA-20(161)
	21069
	18665
	16968
	15271
	12867
		3232	100000

SHA-25(51)
SHA-25(81)
SHA-25(101)
SHA-25(121)
SHA-25(161)
	36057
	36057
	36057
	32522
	26866
		3864	100000

SHA-32(51)
SHA-32(81)
SHA-32(101)
SHA-32(121)
SHA-32(161)
	36057
	36057
	36057
	36057
	36057
		2576	100000

Table 62: Maximum Values for Object (2020$_h$)

4.1.8 Object 2025$_h$: Homing Mode Settings

Configuration settings for hard stop homing modes (homing modes -1 … -6)

Object Description			
Index	Name	Object Type	Data Type
2025$_h$	Homing Mode Settings	Variable	Record

Table 63: Object Description (2025$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
01$_h$	HomingState	no	0	255	0		ro
02$_h$	PositionOffset_CW	no	0	2147483647	40000		rw
03$_h$	PositionOffset_CCW	no	0	2147483647	40000		rw
04$_h$	CurrentThreshold	no	0	3000	Depends on actuator	[mA]	rw
05$_h$	TeachPositionLimit	no	0	3	0		rw

Table 64: Entry Description (2025$_h$)

Homing States:
0: NOT_HOMED
1: HOMED
2: START
3: MOVE_TO_END_POSITION
4: CHECK_NEGATIVE_STOP
5: CHECK_POSITIVE_STOP
253: STOP_TORQUE
254: STOP_VELOCITY
255: STOP_POSITION

4.1.9 Object 2030$_h$: Torque Mode Settings

This object contains all torque regulation parameters and all values that need to be monitored for tuning the torque regulation loop.

Object Description			
Index	Name	Object Type	Data Type
2030$_h$	Torque Mode Settings	Variable	Record

Table 65: Object Description (2030$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ActualCurrent	no	-2147483648	2147483647	0	[mA] (peak)	ro
2	TargetCurrent	no	Depends on actuator	Depends on actuator	0	[mA] (peak)	ro
3	RampTargetCurrent	no	Depends on actuator	Depends on actuator	0	[mA] (peak)	ro
4	P_Parameter	no	0	65535	500		rw
5	I_Parameter	no	0	65535	5000		rw
6	PI_Torque_Error	no	-2147483648	2147483647	0	[mA]	ro
7	PI_Torque_Error_Sum	no	-2147483648	2147483647	0		ro
8	PI_Flux_Error	no	-2147483648	2147483647	0	[mA]	ro
9	PI_Flux_Error_Sum	no	-2147483648	2147483647	0		ro

Table 66: Entry Description (2030$_h$)

4.1.10 Object 2040$_h$: Velocity Mode Settings

This object contains all velocity regulation parameters. It also contains all values that need to be monitored for tuning the velocity regulation loop.

Object Description			
Index	Name	Object Type	Data Type
2040$_h$	Velocity Mode Settings	Variable	Record

Table 67: Object Description (2040$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ActualVelocity	no	-2147483648	2147483647	0	[rpm]	ro
2	TargetVelocity	no	-200000	200000	0	[rpm]	ro
3	RampTargetVelocity	no	-2147483648	2147483647	0	[rpm]	ro
4	MotorHaltedVelocity	no	0	200000	5	[rpm]	rw
5	P_Parameter	no	0	65535	Depends on actuator		rw
6	I_Parameter	no	0	65535	Depends on actuator		rw
7	VelocityError	no	-2147483648	2147483647	0	[rpm]	ro
8	VelocityErrorSum	no	-2147483648	2147483647	0		ro

Table 68: Entry Description (2040$_h$)

4.1.11 Object 2050$_h$: Position Mode Settings

This object contains all position regulation parameters. It also contains all values that need to be monitored for tuning the position regulation loop.

Object Description			
Index	Name	Object Type	Data Type
2050$_h$	Position Mode Settings	Variable	Record

Table 69: Object Description (2050$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ActualPosition	no	-2147483648	2147483647	0		rw
2	TargetPosition	no	-2147483648	2147483647	0		ro
3	RampTargetPosition	no	-2147483648	2147483647	0		ro
4	P_Parameter	no	0	65535	100		rw
5	Position_Error	no	-2147483648	2147483647	0		ro
6	TargetReachedVelocity	no	0	200000	500	[rpm]	rw
7	TargetReachedDistance	no	0	100000	5		rw
8	PositionScaler	no	-32768	32767	32767		rw
9	ActualPositionSelection	no	0	1	0		rw
10	ActualMotorPosition	no	-2147483648	2147483647	0		ro
11	ActualGearboxPosition	no	-2147483648	2147483647	0		ro
12	RawMotorEncoderPosition	no	-2147483648	2147483647	0		ro
13	RawGearboxEncoderPosition	no	-2147483648	2147483647	0		ro

Table 70: Entry Description (2050$_h$)

4.1.12 Object 2055$_h$: Commutation Mode

Select a commutation mode that fits best to the actuator encoders. Possible values are

- 0: Disable
- 1: Open loop
- 2: Encoder

Object Description			
Index	Name	Object Type	Data Type
2055$_h$	Commutation Mode	Variable	Record

Table 71: Object Description (2055$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Commutation Mode	no	0	2	2		rw

Table 72: Entry Description (2055$_h$)

4.1.13 Object 2056$_h$: Velocity Ramp Mode

An activated ramp allows a defined acceleration for velocity and position mode, and always be active as the default

Object Description			
Index	Name	Object Type	Data Type
2056$_h$	Velocity Ramp Mode	Variable	UNSIGNED8

Table 73: Object Description (2056$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Velocity Ramp Mode	no	0	1	1		rw

Table 74: Entry Description (2056$_h$)

4.1.14 Object 2060$_h$: Open Loop Settings

This object contains the actual commutation angle

Object Description			
Index	Name	Object Type	Data Type
2060$_h$	Open Loop Settings	Variable	Record

Table 75: Object Description (2060$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ActualAngle	no	-32768	32767	0		ro

Table 76: Entry Description (2060$_h$)

4.1.15 Object 2080$_h$: Encoder Settings

This object contains all encoder specific settings and the actual commutation angle

Object Description			
Index	Name	Object Type	Data Type
2080$_h$	Encoder Settings	Variable	Record

Table 77: Object Description (2080$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ActualAngle	no	-32768	32767	0		ro
2	StepsPerRotation	no	0	2147483647	Depends on actuator (Refer to Table 126)		rw
3	Offset	no	0	2147483647	0		rw
4	Direction	no	0	1	1		rw
5	InitMode	no	0	2	1		rw
6	InitDelay	no	0	10000	1000	[ms]	rw
7	InitVelocity	no	-200000	200000	200	[rpm]	rw

Table 78: Entry Description (2080_h)

4.1.16 Object 2090_h: IIt Monitor Settings

This object controls the IIt monitor functions. The actual current is being monitored, and these values are being squared and summed up periodically over the configured winding time using a 1ms cycle. If one of the limits gets exceeded during this time, the actuator will be stopped and the IIt error flag will be set. The IIt error flag can be reset by writing any value to sub-index 7.

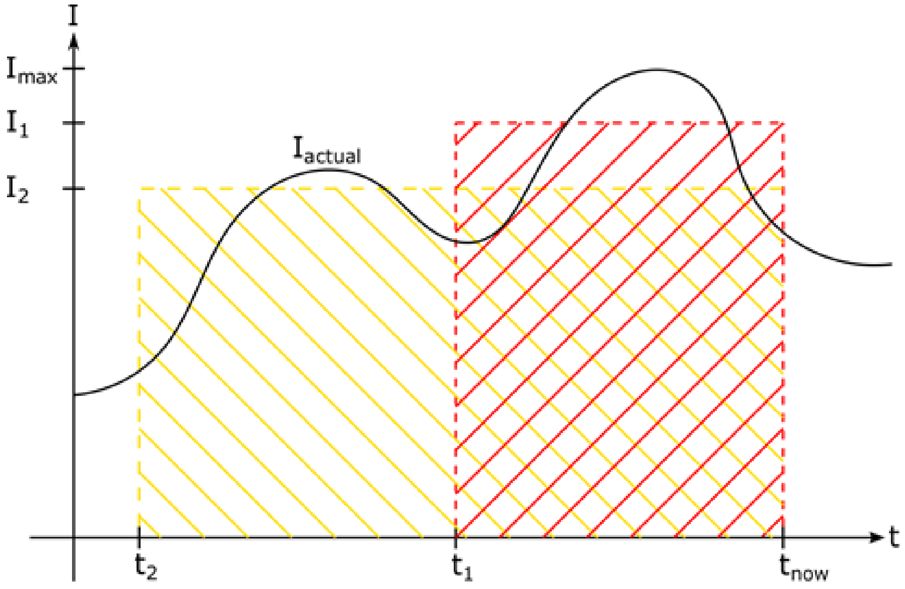
There are two IIt windows (see Figure 4). The first one directly uses the actual current, and the second one uses the actual current divided by $\sqrt{2}$ (less power over longer time). Sub-indices 3 and 6 show the actual integration sums.

Object Description			
Index	Name	Object Type	Data Type
2090_h	IIt Monitor Settings	Variable	Record

Table 79: Objects Description (2090_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	WindingTime_1	no	1000	60000	3000	[ms]	rw
2	IItLimit_1	no	0	54000000	Depends on actuator	[A^2ms]	rw
3	IItSum_1	no	0	4294967295	—	[mA]	ro
4	WindingTime_2	no	1000	60000	6000	[ms]	rw
5	IItLimit_2	no	0	54000000	Depends on actuator	[A^2ms]	rw
6	IItSum_2	no	0	4294967295	—	[mA]	ro
7	ClearExceededFlags	no	0	255	0	—	rw

Table 80: Entry Description (2090_h)



(I1: sub-index 2; I2: sub-index 5; t1: sub-index 1; t2: sub-index 4)

Figure 4 IIt Monitor Windows

4.1.17 Object 2095h: Velocity Window Settings

This object configures the maximum difference between actual velocity and ramp velocity. If this value is exceeded and error flag is set and the motor will be stopped. The error flag can be reset by writing any value to sub index2 of this object

Object Description			
Index	Name	Object Type	Data Type
2095h	Velocity Window Settings	Variable	Record

Table 81: Object Description(2095h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	VelocityWindow	no	0	65535	Depends on actuator	[rpm]	rw
2	ClearVelocityWindowError	no	0	255	0	—	rw

Table 82: Entry Description (2095h)

4.1.18 Object 2096h: Position Window Settings

This object configures the maximum difference between actual position and ramp position. If this value is exceeded an error flag is set and the motor will be stopped. The error flag can be reset by writing any value to sub-index 2 of this object.

Object Description			
Index	Name	Object Type	Data Type
2096h	Position Window Settings	Variable	Record

Table 83: Object Description (2096h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	PositionWindow	no	0	4294967295	Depends on actuator	[rpm]	rw
2	ClearPositionWindowError	no	0	255	0	—	rw

Table 84: Entry Description (2096$_h$)

4.1.19 Object 2100$_h$: Home Offset Display

This object shows the home offset. The value is given in encoder increments.

Object Description			
Index	Name	Object Type	Data Type
2100$_h$	Home Offset Display	Variable	SIGNED32

Table 85: Object Description (2100$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Home Offset Display	no	-2147483648	2147483647	0		ro

Table 86: Entry Description (2100$_h$)

4.1.20 Object 2702$_h$: Digital Inputs

Object Description			
Index	Name	Object Type	Data Type
2702$_h$	Digital Inputs	Variable	UNSIGNED32

Table 87: Object Description (2702$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Digital Inputs	yes	0	15	0		ro

Table 88: Entry Description (2702$_h$)

Bit mapping	
Bit	Input
0	IO$_1$ (configurable I/O used as input)
1	IO$_2$ (configurable I/O used as input)
2	IN$_3$ (optically isolated input)
3	IN$_4$ (optically isolated input

Table 89: Bit Mapping (2702$_h$)

4.1.21 Object 2703h: Device Digital Outputs

With this object the digital outputs (general purpose outputs) can be set. The bits of sub index 1 control the outputs of the module. The bits of sub index 2 determine which outputs can be switched. The number of available digital outputs depends on the module type.

Object Description			
Index	Name	Object Type	Data Type
2703h	Device Digital Outputs	Variable	ARRAY

Table 90: Object Description (2703h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Physical outputs	yes	0	2147483647	0	—	rw
2	Output mask	no	0	2147483647	0	—	rw

Table 91: Entry Description (2703h)

Bit mapping	
Bit	Output
0	OUT1 (configurable I/O used as output)
1	OUT2 (configurable I/O used as output)

Table 92:Bit Mapping(2703h)

Refer to IO port mapping for the bit mapping variance of RSF-5 and RSA-8 actuator.

4.1.22 Object 2704h: CAN Bit Rate

With this object it is possible to change the CAN bit rate.

To do this, first write the new value to this object. Then, store the new setting by writing the save signature to object 1010h. After that, reset the module. The new setting then becomes active.

(Available bit rates: 20, 50, 125, 250, 500, 1000 in kBit/s)

Object Description			
Index	Name	Object Type	Data Type
2704h	CAN Bit Rate	Variable	UNSIGNED16

Table 93: Object Description (2704h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	CAN Bit Rate	no	20	1000	1000		rw

Table 94: Entry Description (2704h)

4.1.23 Object 2705$_h$: Node ID

The node ID can be selected using this object.to change the node ID, first write the new node ID to this object. Then, store the new setting by writing the save signature to object 1010$_h$ sub index 2. After that, reset the module. The new setting then becomes active.

Object Description			
Index	Name	Object Type	Data Type
2705$_h$	Node ID	Variable	UNSIGNED8

Table 95: Object Description (2705$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Node ID	no	1	127	1		rw

Table 96: Entry Description (2705$_h$)

4.1.24 Object 2706$_h$: User Variables

Object Description			
Index	Name	Object Type	Data Type
2706$_h$	User Variables	Variable	Record

Table 97: Object Description (2706$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Address	no	0	255	0		rw
2	Data	no	-2147483648	2147483647	0		rw
3	Store	no	0	4294967295	305419896		rw

Table 98: Entry Description (2706$_h$)

4.1.25 Object 270E$_h$: Analog Inputs

Object Description			
Index	Name	Object Type	Data Type
270E$_h$	Analog Inputs	Variable	Record

Table 99: Object Description (270E$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ADC_phase_A	no	0	4095	0		ro
2	ADC_phase_B	no	0	4095	0		ro
3	ADC_Vsupply*	no	0	4095	0		ro
4	ADC_Temp*	no	0	4095	0		ro
5	ADC_IN_1*	yes	0	4095	0		ro
6	ADC_IN_2*	yes	0	4095	0		ro

Table 100: Entry Description (270E$_h$)

*Depends on actuator since there are some minor differences in subindex mapping

4.1.26 Object 5FFF$_h$: Bootloader Mode

This object resets the module into the bootloader mode. Write the hex code 12345678$_h$ into this object to perform the reset and update the firmware.

Object Description			
Index	Name	Object Type	Data Type
5FFF$_h$	Bootloader mode	Variable	UNSIGNED32

Table 101: Object Description (5FFF$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	12345678$_h$	0

Table 102: Entry Description (5FFF$_h$)

5 Profile specific area

The profile segment contains CiA-402 standard motion control objects. These objects control the motion control functions of the actuator. Since it is not possible to operate the modes in parallel, the user is able to activate the required function by selecting a mode of operation. The control device writes to the modes of operation object in order to select the operation mode. The drive device provides the modes of operation display object to indicate the actual activated operation mode. Controlword, statusword, and set-points are mode-specific. This implies the responsibility of the control device to avoid inconsistencies and erroneous behavior.

The following operating modes (selectable via object 6060$_h$, please see 5.1.6) are implemented on the actuator:

- Profile position mode (pp)
- Profile velocity mode (pv)
- Homing mode (hm)
- Cyclic synchronous position mode (csp)
- Cyclic synchronous velocity mode (csv)
- Cyclic synchronous torque mode (cst)

5.1 Detailed object specifications

5.1.1 Object 605A$_h$: Quick Stop Option Code

This object indicates what action is performed when the quick stop function is executed. The slow down ramp is the deceleration value of the used mode of operation. The following quick stop option codes are supported in the current version of the CANopen firmware:

Value Definition	
Value	Definition
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*
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*

Table 103: Value Description (605A$_h$)

Object Description			
Index	Name	Object Type	Data Type
605A$_h$	Quick Stop Option Code	Variable	SIGNED16

Table 104: Object Description (605A$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	1/2/5/6	2

Table 105: Entry Description (605A$_h$)

5.1.2 Object 605B$_h$: Shutdown Option Code

This object indicates what action is performed if there is a transition from *operation enabled* state to ready to *switch on* state. The shutdown option code always has the value 0 as only this is supported.

Value Definition	
Value	Definition
0	Disable drive function (switch off the power stage)

Table 106: Value Description (605B$_h$)

Object Description			
Index	Name	Object Type	Data Type
605B$_h$	Shutdown Option Code	Variable	UNSIGNED16

Table 107: Object Description (605B$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0	0

Table 108: Entry Description (605B$_h$)

5.1.3 Object 605C$_h$: Disable Operation Option Code

This object indicates what action is performed if there is a transition from *operation enabled* state to *switched on* state. The disable operation option code always has the value 1 as only this is supported. The slow down ramp is the deceleration value of the used mode of operation.

Value Definition	
Value	Definition
1	Slow down on slow down ramp

Table 109: Value Description (605C$_h$)

Object Description			
Index	Name	Object Type	Data Type
605C$_h$	Disable Operation Option Code	Variable	UNSIGNED16

Table 110: Object Description (605C$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	1	1

Table 111: Entry Description (605C$_h$)

5.1.4 Object 605D$_h$: Halt Option Code

This object indicates what action is performed when the halt function is executed. The slow down ramp is the deceleration value of the used mode of operation. The halt option code always has the value 1 as only this is supported.

Value Definition	
Value	Definition
1	Slow down on slow down ramp and stay in *operation enabled*

Table 112: Value Description (605D$_h$)

Object Description			
Index	Name	Object Type	Data Type
605D$_h$	Halt Option Code	Variable	UNSIGNED16

Table 113: Object Description (605D$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	1	1

Table 114: Entry Description (605D$_h$)

5.1.5 Object 605E$_h$: Fault Reaction Option Code

This object indicates what action is performed when fault is detected in the power drive system. The slow down ramp is the deceleration value of the used mode of operation. The fault reaction option code always has the value 2 as only this is supported.

Value Definition	
Value	Definition
2	Slow down on quick stop ramp

Table 115: Value Description (605E$_h$)

Object Description			
Index	Name	Object Type	Data Type
605E$_h$	Fault Reaction Option Code	Variable	UNSIGNED16

Table 116: Object Description (605E$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	2	2

Table 117: Entry Description (605E$_h$)

5.1.6 Object 6060$_h$: Modes of Operation

This object indicates the requested operation mode. Supported operating modes are:

Value Definition	
Value	Mode
0	No mode
1	Profile position mode (pp)
3	Profile velocity mode (pv)
6	Homing mode (hm)
8	Cyclic synchronous position mode (csp)
9	Cyclic synchronous velocity mode (csv)
10	Cyclic synchronous torque mode (cst)

Table 118: Value Description (6060$_h$)

The motor will not run when the operating mode is set to 0. It will be stopped when the motor is running in one of the supported operating modes and the operating mode is then switched to 0.

Object Description			
Index	Name	Object Type	Data Type
6060$_h$	Modes of Operation	Variable	SIGNED8

Table 119: Object Description (6060$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	see table 118	0

Table 120: Entry Description (6060$_h$)

5.1.7 Object 6061$_h$: Modes of Operation Display

This object shows the operating mode that is currently set.

Value Definition	
Value	Mode
0	No mode
1	Profile position mode (pp)
3	Profile velocity mode (pv)
6	Homing mode (hm)
8	Cyclic synchronous position mode (csp)
9	Cyclic synchronous velocity mode (csv)
10	Cyclic synchronous torque mode (cst)

Table 121: Value Description (6061$_h$)

The motor will not run when the operating mode is set to 0. It will be stopped when the motor is running in one of the supported operating modes and the operating mode is then switched to 0

Object Description			
Index	Name	Object Type	Data Type
6061_h	Modes of Operation Display	Variable	SIGNED8

Table 122: Object Description (6061_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	see table 121	0

Table 123: Entry Description (6061_h)

5.1.8 Object $608F_h$: Position Encoder Resolution

This object defines the resolution of the encoder. The position encoder resolution is calculated by the following formula:

Position encoder resolution = Encoder increments / Motor revolutions.

Object Description			
Index	Name	Object Type	Data Type
$608F_h$	Position Encoder Resolution	Array	UNSIGNED32

Table 124: Object Description ($608F_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Encoder increments	no	0	131072	Depends on actuator		rw
2	Motor revolutions	no	1	1	1		ro

Table 125: Entry Description ($608F_h$)

Position Encoder Resolution			
Actuator Family	Encoder	Encoder Bit Resolution	Encoder Counts
RSF	Absolute	14(motor)	16384
RSA	Absolute	14(motor)	16384
FHA-C mini	Dual Absolute	15(motor)/14(gear)	32768/16384
LPA	Dual Absolute	17(motor)/16(gear)	131072/65536
SHA	Dual Absolute	17(motor)/16(gear)	131072/65536

Table 126: Position Encoder Resolution

5.1.9 Object 60FD$_h$: Digital Inputs

This object contains the states of the digital inputs of the module. Starting from bit 0, every bit reflects the state of one digital input. The number of valid bits depends on the number of digital inputs of the module.

Object Description			
Index	Name	Object Type	Data Type
60FD$_h$	Digital Inputs	Variable	UNSIGNED32

Table 127: Object Description (60FD$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	UNSIGNED32	0

Table 128: Entry Description (60FD$_h$)

5.1.10 Object 6502$_h$: Supported Drive Modes

This object provides information on the supported drive modes (0: not supported, 1: supported). This object is organized bit-wise. The bits have the following meaning:

Value Definition	
Bit	Mode
0	Profile position mode (pp)
1	Velocity mode (vl), not used
2	Profile velocity mode (pv)
3	Torque mode (tq), not used
4	Reserved
5	Homing mode (hm)
6	Interpolated position mode (ip), not used
7	Cyclic synchronous position mode (csp)
8	Cyclic synchronous velocity mode (csv)
9	Cyclic synchronous torque mode (cst)
10-15	Reserved
16-31	Reserved

Table 129: Value Definition (6502$_h$)

Object Description			
Index	Name	Object Type	Data Type
6502$_h$	Supported Drive Modes	Variable	UNSIGNED32

Table 130: Object Description (6502$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Supported Drive Modes	no	0	$FFFFFFFF_h$	$3A5_h$		ro

Table 131: Entry Description (6502_h)

6 Profile Position Mode

A target position is applied to the trajectory generator. It is generating a position demand value for the position control loop described in the position control function.

Please refer to object 6060$_h$ (section 5.1.6) for information about how to choose an operation mode. Object 6061$_h$ (section 5.1.7) shows the operation mode that is set.

6.1 Detailed Object Specifications

The following text offers detailed object specifications. For a better understanding, it is necessary to see how the state machine works.

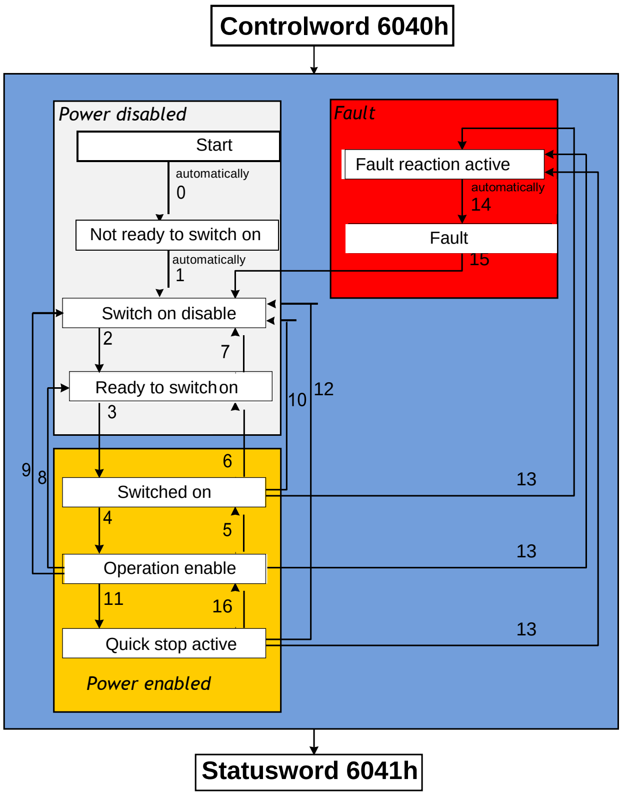


Figure 5: DS402 Finite State Machine

Notes on state transitions:

- Commands directing a change in state are processed completely and the new state achieved before additional state change commands are processed.
- Transitions 0 and 1 occur automatically at drive power-on or reset. Transition 14 occurs automatically, too. All other state changes must be directed by the host.
- Drive function disabled indicates that no current is being supplied to the motor.
- Drive function enabled indicates that current is available for the motor and profile position and profile velocity reference values may be processed.

6.1.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to Figure 5 for detailed information.

Structure of the Control Word									
15 11	10	9	8	7	6 4	3	2	1	0
nu	r	oms	h	fr	oms	eo	qs	ev	so
MSB									LSB

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 132: Structure of the Control Word in pp Mode

Operation Mode specific Bits in pp Mode		
Bit	Name	Definition
4	New set point	0-to-1: the next positioning will be started.
5	Change immediately	Not supported.
6	Absolute / relative	0: New position is absolute. 1: New position is relative.
9	Change set point	Not supported.

Table 133: Operation Mode specific Bits in pp Mode

Command Coding						
Command	Bits of Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
Disable voltage	0	x	x	0	x	7,9,10,12
Quick stop	0	x	0	1	x	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0-to-1	x	x	x	x	15

Table 134: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040_h	ControlWord	Variable	UNSIGNED16

Table 135: Object Description (6040_h in pp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See command coding above.	

Table 136: Entry Description (6040$_h$ in pp Mode)

6.1.2 Object 6041$_h$: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to Figure 5 for detailed information. The object is structured as defined below.

For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

Structure of the Status Word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dir	mot	oms		ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Legend: dir=direction of rotation; mot=motor activity; oms=operation mode specific; ila=internal limit active; tr=target reached; rm=remote; ms=manufacturer spec; w=warning; sod=switch on disabled; qs=quick stop; ve=voltage enabled; f=fault; oe=operation enabled; so=switch on; rtso=ready to switch on.

Table 137: Structure of the Status Word in pp Mode

HDLLC Specific Bits		
Bit	Name	Definition
14	Motor activity	0: Motor stands still. 1: Motor rotates.
15	Direction of rotation	This bit shows the direction of rotation.

Table 138: HDLLC Specific Bits

Operation Mode specific Bits in pp Mode		
Bit	Name	Definition
10	Target reached	Set when the motor is within the position window.
12	Set point acknowledged	0: Set point processed. 1: Set point still in process.
13	Following error	Not supported.

Table 139 Operation Mode specific Bits in pp Mode

State Coding	
Status word	FSA state
xxxx xxxx x0xx 0000$_b$	Not ready to switch on
xxxx xxxx x1xx 0000$_b$	Switch on disabled
xxxx xxxx x01x 0001$_b$	Ready to switch on
xxxx xxxx x01x 0011$_b$	Switched on
xxxx xxxx x01x 0111$_b$	Operation enabled
xxxx xxxx x00x 0111$_b$	Quick stop active
xxxx xxxx x0xx 1111$_b$	Fault reaction active
xxxx xxxx x0xx 1000$_b$	Fault

Table 140: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041$_h$	StatusWord	Variable	UNSIGNED16

Table 141: Object Description (6041$_h$ in pp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See state coding above.	

Table 142: Entry Description (6041$_h$ in pp Mode)

6.1.3 Object 6062$_h$: Position Demand Value

This object provides the demanded position value. The value is given in encoder steps. Object 6062$_h$ indicates the actual position that the motor should have. It is not to be confused with objects 6063$_h$ and 6064$_h$.

Object Description			
Index	Name	Object Type	Data Type
6062$_h$	Position Demand Value	Variable	SIGNED32

Table 143: Object Description (6062$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Demand Value	yes	-2147483648	2147483647	0		ro

Table 144: Entry Description (6062$_h$)

6.1.4 Object 6063h: Position Actual Internal Value

This object provides the actual position value of the motor.

Object Description			
Index	Name	Object Type	Data Type
6063h	Position Actual Internal Value	Variable	SIGNED32

Table 145: Object Description (6063h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Actual Internal Value	yes	-2147483648	2147483647	0		ro

Table 146: Entry Description (6063h)

6.1.5 Object 6064h: Position Actual Value

This object provides the actual value of the position measurement device. It always contains the same value as object 6063h.

Object Description			
Index	Name	Object Type	Data Type
6064h	Position Actual Value	Variable	SIGNED32

Table 147: Object Description (6064h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Actual Value	yes	-2147483648	2147483647	0		ro

Table 148: Entry Description (6064h)

6.1.6 Object 6067h: Position Window

This object indicates the configured symmetrical range of accepted positions relative to the target position. If the actual value of the position encoder is within the position window, this target position is regarded as having been reached. The value is given in increments. If the value of the position window is FFFFFFFFh, the position window control is switched off. If this object is set to zero, the target reached event will be signaled when the demand position (6062h) has reached the target position (6064h). When the position window is set to a value greater than zero, the target reached event will be signaled when the actual encoder position value (6064h) is within (target_position - position_window) and (target_position + position_window).

Object Description			
Index	Name	Object Type	Data Type
6067h	Position Window	Variable	UNSIGNED32

Table 149: Object Description (6067h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Window	no	0	4294967295	4294967295		rw

Table 150: Entry Description (6067h)

6.1.7 Object 606Bh: Velocity Demand Value

This object shows the velocity output value of the ramp generator.

Object Description			
Index	Name	Object Type	Data Type
606Bh	Velocity Demand Value	Variable	SIGNED32

Table 151: Object Description (606Bh)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Velocity Demand Value	no	-2147483648	2147483647	0	[rpm]	ro

Table 152: Entry Description (606Bh)

6.1.8 Object 606Ch: Velocity Actual Value

This object shows the actual velocity value derived from the velocity sensor.

Object Description			
Index	Name	Object Type	Data Type
606Ch	Velocity Actual Value	Variable	SIGNED32

Table 153: Object Description (606Ch)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Velocity Actual Value	yes	-2147483648	2147483647	0	[rpm]	ro

Table 154: Entry Description (606C$_h$)

6.1.9 Object 607A$_h$: Target Position

The target position is the position that the drive should move to in profile position mode using the actual settings of motion control parameters (such as velocity, acceleration, deceleration, etc.). The value of this object is interpreted as absolute or relative depending on the abs/rel flag in the ControlWord.

Object Description			
Index	Name	Object Type	Data Type
607A$_h$	Target Position	Variable	SIGNED32

Table 155: Object Description (607A$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Target Position	yes	-2147483648	2147483647	0		rw

Table 156: Entry Description (607A$_h$)

6.1.10 Object 607D$_h$: Software Position Limit

This object indicates the configured maximum and minimum software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

Corrected min position limit = min position limit — home offset
Corrected max position limit = max position limit — home offset

Object Description			
Index	Name	Object Type	Data Type
607D$_h$	Software Position Limit	Array	SIGNED32

Table 157: Object Description (607D$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Min Position Limit	no	-2147483648	2147483647	-2147483648		rw
2	Max Position Limit	no	-2147483648	2147483647	2147483647		rw

Table 158: Entry Description (607D$_h$)

6.1.11 Object 6081h: Profile Velocity (for pp mode)

This object indicates the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and is valid for both directions of motion. The profile velocity is the maximum velocity used when driving to a new position.

Object Description			
Index	Name	Object Type	Data Type
6081h	Profile Velocity (pp)	Variable	UNSIGNED32

Table 159: Object Description (6081h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Profile Velocity	no	0	200000	6000	[rpm]	rw

Table 160: Entry Description (6081h)

6.1.12 Object 6082h: End Velocity

This object indicates the configured velocity normally attained at the end of the deceleration ramp during a profiled motion and is valid for both directions of motion. The end velocity is the velocity used when reaching the new position.

Object Description			
Index	Name	Object Type	Data Type
6082h	End Velocity	Variable	SIGNED32

Table 161: Object Description (6082h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	End Velocity	no	-200000	200000	500	[rpm]	rw

Table 162: Entry Description (6082h)

6.1.13 Object 6083h: Profile Acceleration

This object indicates the configured acceleration. Object 6083h sets the maximum acceleration to be used in profile positioning mode, and profile velocity mode.

Object Description			
Index	Name	Object Type	Data Type
6083h	Profile Acceleration	Variable	UNSIGNED32

Table 163: Object Description (6083h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Profile Acceleration	no	0	100000	Depends on actuator	[rpm/s]	rw

Table 164: Entry Description (6083$_h$)

6.1.14 Object 6084$_h$: Profile Deceleration

This object indicates the configured deceleration, which will be the same as the Profile Acceleration.

Object Description			
Index	Name	Object Type	Data Type
6084$_h$	Profile Deceleration	Variable	UNSIGNED32

Table 165: Object Description (6084$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Profile Deceleration	no	0	100000	Depends on actuator	[rpm/s]	ro

Table 166: Entry Description (6084$_h$)

6.1.15 Object 6085$_h$: Quick Stop Deceleration

This object indicates the configured deceleration used to stop the motor when the quick stop function is activated and the quick stop code object 605A$_h$ is set to 2 (or 6).

Object Description			
Index	Name	Object Type	Data Type
6085$_h$	Quick Stop Deceleration	Variable	UNSIGNED32

Table 167: Object Description (6085$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Quick Stop Deceleration	no	0	100000	Depends on actuator	[rpm/s]	rw

Table 168: Entry Description (6085$_h$)

6.2 Example of pp Mode Operation

The following is an example of running in pp mode (the values are decimal), assume that the actuator has been reset and then switched to pre-operational or operational by NMT commands.

- If limit switches are not connected, first disable the limit switch inputs by writing 3 to object 2005$_h$.
- Select pp mode by writing 1 to object 6060$_h$.
- Write 6 to object 6040$_h$ to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040$_h$ to switch to SWITCHED_ON state.
- Write 15 to object 6040$_h$ to switch to OPERATION_ENABLED state.
- Write the desired target position (e.g. 500000) to object 607A$_h$.
- Mark the new target position as active by writing 31 to object 6040$_h$ and observe motion commencing.
- Reset the activation by writing 15 to object 6040$_h$ (this can be done while the actuator is still moving).

7 Profile Velocity Mode

The profile velocity mode is used to control the velocity of the drive independent of the position. It contains limit functions and trajectory generation.

The profile velocity mode covers the following sub-functions:

- Demand value input via trajectory generator.
- Monitoring of the profile velocity using a window-function.
- Monitoring of velocity actual value using a threshold.

The operation of the reference value generator and its input parameters include:

- Profile velocity
- Profile acceleration
- Motion profile type

7.1 Detailed Object Specifications

7.1.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to Figure 5 for detailed information.

In pv mode the control word does not contain any operation mode specific bits.

Structure of the Control Word									
15 11	10	9	8	7	6 4	3	2	1	0
nu	r	r	h	fr	r	eo	qs	ev	so
MSB									LSB

Legend: nu=not used; r=reserved; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 169: Structure of the Control Word in pv Mode

Command Coding						
Command	Bits of Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
Disable voltage	0	x	x	0	x	7,9,10,12
Quick stop	0	x	0	1	x	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0-to-1	x	x	x	x	15

Table 170: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040_h	Controlword	Variable	UNSIGNED16

Table 171: Object Description (6040_h in pv Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See command coding above.	

Table 172: Entry Description (6040_h in pv Mode)

7.1.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to Figure 5 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

Structure of the Status Word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dir	mot	oms		ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Legend: dir=direction of rotation; mot=motor activity; oms=operation mode specific; ila=internal limit active; tr=target reached; rm=remote; ms=manufacturer spec; w=warning; sod=switch on disabled; qs=quick stop; ve=voltage enabled; f=fault; oe=operation enabled; so=switch on; rtso=ready to switch on.

Table 173: Structure of the Status Word in pv Mode

HDLLC Specific Bits		
Bit	Name	Definition
14	Motor activity	0: Motor stands still. 1: Motor rotates.
15	Direction of rotation	This bit shows the direction of rotation.

Table 174: HDLLC Specific Bits

Operation Mode specific Bits in pv Mode		
Bit	Name	Definition
10	Target reached	Indicates that the target speed has been reached.
12	Speed	Not supported.
13	Max. slippage error	Not supported.

Table 175: Operation Mode Specific Bits in pv Mode

State Coding	
Status word	FSA state
xxxx xxxx x0xx 0000$_b$	Not ready to switch on
xxxx xxxx x1xx 0000$_b$	Switch on disabled
xxxx xxxx x01x 0001$_b$	Ready to switch on
xxxx xxxx x01x 0011$_b$	Switched on
xxxx xxxx x01x 0111$_b$	Operation enabled
xxxx xxxx x00x 0111$_b$	Quick stop active
xxxx xxxx x0xx 1111$_b$	Fault reaction active
xxxx xxxx x0xx 1000$_b$	Fault

Table 176: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041$_h$	StatusWord	Variable	UNSIGNED16

Table 177: Object Description (6041$_h$ in pv Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See state coding above	

Table 178: Entry Description (6041$_h$ in pv Mode)

7.1.3 Object 6062$_h$: Position Demand Value

This object provides the demanded position value. The value is given in encoder steps. Object 6062$_h$ indicates the actual position that the motor should have. It is not to be confused with objects 6063$_h$ and 6064$_h$.

Object Description			
Index	Name	Object Type	Data Type
6062$_h$	Position Demand Value	Variable	SIGNED32

Table 179: Object Description (6062$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Demand Value	yes	-2147483648	2147483647	0		ro

Table 180: Entry Description (6062$_h$)

7.1.4 Object 6063$_h$: Position Actual Internal Value

This object provides the actual position value of the motor.

Object Description			
Index	Name	Object Type	Data Type
6063$_h$	Position Actual Internal Value	Variable	SIGNED32

Table 181: Object Description (6063$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Actual Internal Value	yes	-2147483648	2147483647	0		ro

Table 182: Entry Description (6063$_h$)

7.1.5 Object 6064$_h$: Position Actual Value

This object provides the actual value of the encoder. It always contains the same value as object 6063$_h$.

Object Description			
Index	Name	Object Type	Data Type
6064$_h$	Position Actual Value	Variable	SIGNED32

Table 183: Object Description (6064$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Actual Value	yes	-2147483648	2147483647	0		ro

Table 184: Entry Description (6064$_h$)

7.1.6 Object 606B$_h$: Velocity Demand Value

This object provides the velocity output value of the ramp generator

Object Description			
Index	Name	Object Type	Data Type
606B$_h$	Velocity Demand Value	Variable	SIGNED32

Table 185: Object Description (606B$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Velocity Demand Value	no	-2147483648	2147483647	0	[rpm]	ro

Table 186: Entry Description (606B$_h$)

7.1.7 Object 606C$_h$: Velocity Actual Value

This object shows the actual velocity value derived from the velocity sensor.

Object Description			
Index	Name	Object Type	Data Type
606C$_h$	Velocity Actual Value	Variable	SIGNED32

Table 187: Object Description (606C$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Velocity Actual Value	yes	-2147483648	2147483647	0	[rpm]	ro

Table 188: Entry Description (606C$_h$)

7.1.8 Object 607D$_h$: Software Position Limit

This object indicates the configured maximum and minimum software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

Corrected min position limit = min position limit — home offset
Corrected max position limit = max position limit — home offset

Object Description			
Index	Name	Object Type	Data Type
607D$_h$	Software Position Limit	Array	SIGNED32

Table 189: Object Description (607D$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Min Position Limit	no	-2147483648	2147483647	-2147483648		rw
2	Max Position Limit	no	-2147483648	2147483647	2147483647		rw

Table 190: Entry Description (607D$_h$)

7.1.9 Object 6083$_h$: Profile Acceleration

This object indicates the configured acceleration. Object 6083$_h$ sets the maximum acceleration to be used in profile positioning mode, and profile velocity mode.

Object Description			
Index	Name	Object Type	Data Type
6083$_h$	Profile Acceleration	Variable	UNSIGNED32

Table 191: Object Description (6083$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Profile Acceleration	no	0	Table 62	Depends on actuator	[rpm/s]	rw

Table 192: Entry Description (6083$_h$)

7.1.10 Object 6085$_h$: Quick Stop Deceleration

This object indicates the configured deceleration used to stop the motor when the quick stop function is activated and the quick stop code object 605A$_h$ is set to 2 (or 6).

Object Description			
Index	Name	Object Type	Data Type
6085$_h$	Quick Stop Deceleration	Variable	UNSIGNED32

Table 193: Object Description (6085$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Quick Stop Deceleration	no	0	Same as Profile Acceleration Table 62	Depends on actuator	[rpm/s]	rw

Table 194: Entry Description (6085$_h$)

7.1.11 Object 60FF$_h$: Target Velocity

This object indicates the configured target velocity and is used as input for the trajectory generator. Object 60FF$_h$ sets the target velocity when using profile velocity mode. The drive then accelerates or decelerates to that velocity using the acceleration and deceleration set by objects 6083$_h$ and 6084$_h$.

Object Description			
Index	Name	Object Type	Data Type
60FF$_h$	Target Velocity	Variable	SIGNED32

Table 195: Object Description (60FF$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Target Velocity	yes	-200000	200000	0	[rpm]	rw

Table 196: Entry Description (60FF$_h$)

7.2 Example of pv Mode Operation

The following is an example of running in pv mode the values are decimal. Assume that the actuator had been reset and then switched to pre-operational or operational by NMT commands

- Select pv mode by writing 3 to object 6060h (Modes_of_Operation).
- Write 6 to object 6040h (Controlword) to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040h to switch to SWITCHED_ON state.
- Write the desired target velocity (e.g. 2000) to object 60FFh (Target_Velocity).
- Write 15 to object 6040h to switch to OPERATION_ENABLED state. The motor now accelerates to the target velocity.
- Stop the motor by writing 0 to object 60FFh.

8 Homing mode

This chapter describes the method by which a drive seeks the home position (reference point). There are various methods of achieving this using limit switches at the ends of travel or a home switch in mid-travel. Some methods also use the index (zero) pulse train from an incremental encoder. The user may specify the speeds, acceleration and the method of homing.

There is no output data except for those bits in the statusword which return the status or result of the homing process and the demand to the position control loops.

There are four sources of the homing signal available: these are positive and negative limit switches, the home switch and the index pulse from an encoder.

Figure 6 shows the defined input objects as well as the output objects. The user can specify the speeds, acceleration and method of homing. The home offset object 607Cₕ allows displacing the zero in point the coordinate system for the home position.

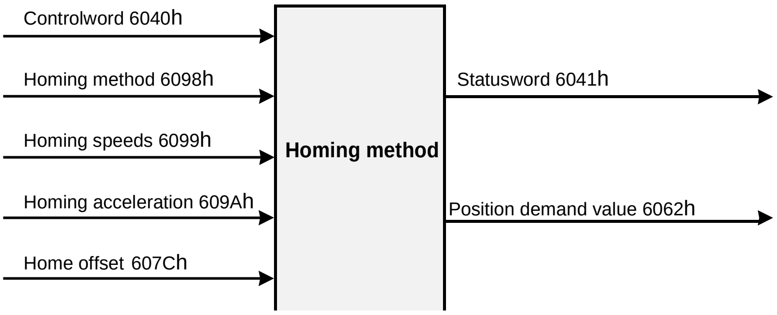


Figure 6: Homing Mode Function

Choosing a homing mode determines the following things:

- The homing signal (positive limit switch, negative limit switch, and home switch).
- The direction of actuation where appropriate.
- The position of the index pulse.

The home position and the zero position are offset by the home offset (see object 607Cₕ, section 0).

There are four sources of homing signals available:

- Negative and positive limit switches.
- Home switch.
- Index pulse of an encoder

For the operation of positioning drives, an exact knowledge of the absolute position is normally required

8.1 Homing Methods

A subset of different standard CANopen homing methods are supported. The homing method that is to be used can be chosen via object 6098h (section 8.2.5).

Supported Homing Methods	
Method	Description
0	No homing (default value for object 6098h).
17	Search the left end switch.
18	Search the right end switch.
35	The actual position is used as home position. All position values (objects 6062h, 6063 h, and 6064 h) are set to zero, but the motor will not move.
-1	Single Ended Clockwise Hard Stop Homing
-2	Single Ended Counterclockwise Hard Stop Homing
-3	Double Ended Clockwise Hard Stop Homing
-4	Double Ended Counterclockwise Hard Stop Homing
-5	Double Ended Clockwise Hard Stop Homing with scaler calculation
-6	Double Ended Counterclockwise Hard Stop Homing with scaler calculation

Table 197: Supported CANopen Homing Methods

When using homing methods that need end switch inputs or home switch inputs, please take care of their configuration (object 2005h).

8.1.1 Homing Method 17 and 18: Homing without Index Pulse

For these methods the home position only depends on the relevant home or limit switch transitions.

Homing Methods 17, 18	
Method	Description
17	Search the left end switch.
18	Search the right end switch.

Table 198: Homing Methods 17,18

8.1.2 Homing Method 35: Current Position as Home Position

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

8.1.3 Homing Method -1: Single Ended Clockwise Hard Stop Homing

For this homing method, the motor is driving with a constant positive velocity (6099h:2) clockwise into a hardstop (1) as shown in Figure 7. While driving into the hardstop, the actual motor current is measured and compared with the current threshold (2025h:4). If the current threshold is reached, the motor moves back by -PositionOffset_CW (2025h:2) encoder steps (2). Then the motor is stopped and the actual position (6064h) is set to 0. The max position limit (607Dh:2) is also set to 0. The velocity is limited to (6099h:1). The min position limit (607Dh:1) will not be changed during this homing method and can be set before homing to limit the position range.

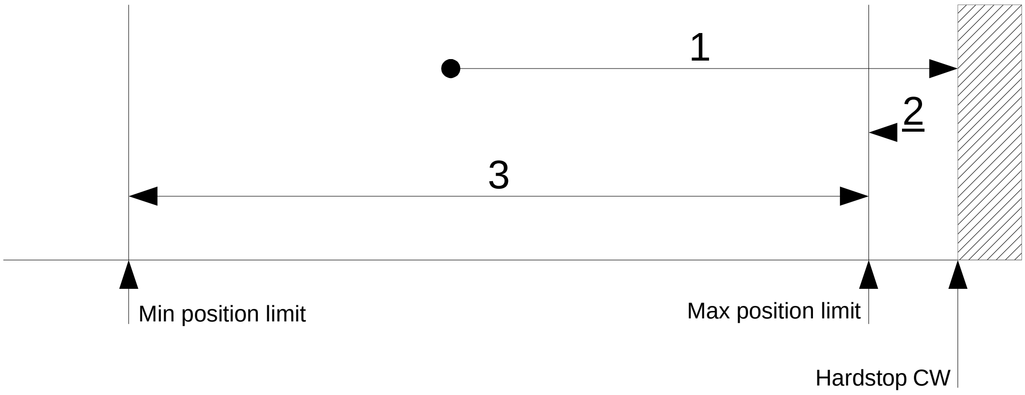


Figure 7: Single ended homing CW

Upon completion, the motor can be moved in torque, velocity, or position mode within the position limits 607Dh :1 and 607Dh:2(3).

8.1.4 Homing Method -2: Single Ended Counterclockwise Hard Stop Homing

For this homing method, the motor is driving with a constant negative velocity (6099h:2) counterclockwise into a hardstop (1) as shown in Figure 8. While driving into the hardstop, the actual motor current is measured and compared with the negative current threshold (2025h:4). If the negative current threshold is reached, the motor moves back by +PositionOffset_CCW (2025h:3) encoder steps (2). Then the motor is stopped and the actual position (6064h) is set to 0. The min position limit (607Dh:1) is also set to 0. The velocity is limited to (6099h:1). The max position limit (607Dh:2) will not be changed during this homing method and can be set before homing to limit the position range.

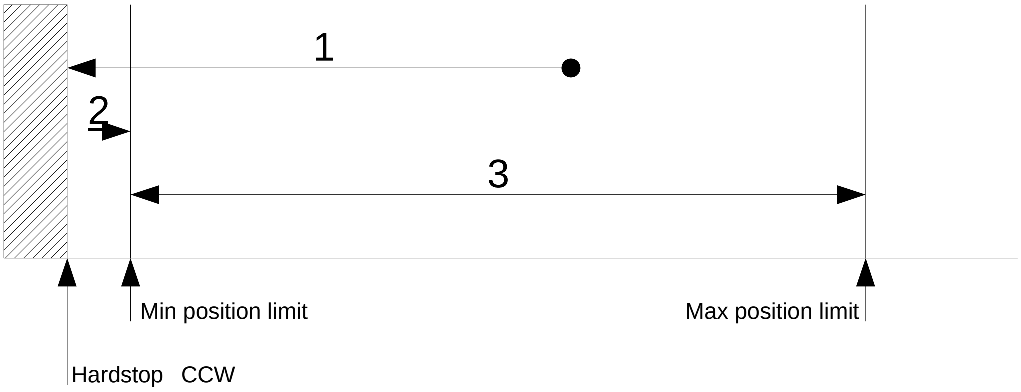


Figure 8: Single ended homing CCW

Upon completion, the actuator can be moved in torque, velocity, or position mode within the position limits 607Dh :1 and 607Dh:2(3).

8.1.5 Homing Method -3: Double Ended Clockwise Hard Stop Homing

For this homing method, the motor is driving with a constant positive velocity (6099h:2) clockwise into a hardstop (1) as shown in Figure 9. While driving into the hardstop the actual motor current is measured and compared with the current threshold (2025h:4). If the current threshold is reached, the motor moves back by -PositionOffset_CW (2025h:2) encoder steps (2). There, the actual position (6064h) and the max position limit (607Dh:2) are marked and the motor drives in the negative (counter clockwise) direction with a constant negative velocity (3). If the negative current threshold is reached, the motor moves back by +PositionOffset_CCW (2025h:3) encoder steps (4). Then the motor is stopped and the min position limit (607Dh:1) and the actual position value (6064h) are set to 0. The velocity is limited to (6099h:1).

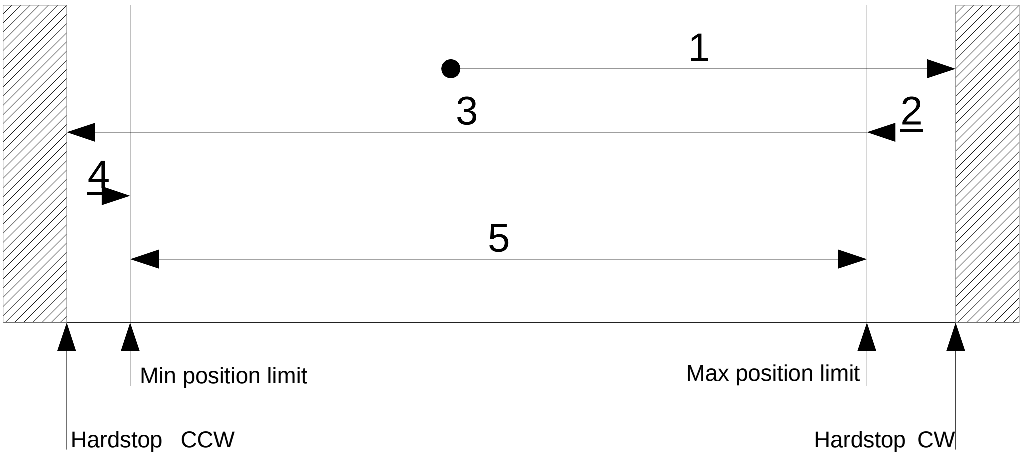


Figure 9: Double ended homing CW

Upon completion, the motor can be moved in torque, velocity, or position mode within the position limits 607Dh :1 and 607Dh:2(5).

8.1.6 Homing Method -4: Double Ended Counterclockwise Hard Stop Homing

For this homing method, the motor is driving with a constant negative velocity (6099h:2) counterclockwise into a hardstop (1) as shown in Figure10. While driving into the hardstop the actual motor current is measured and compared with the negative current threshold (2025h:4). If the negative current threshold is reached, the motor moves back by +PositionOffset_CCW (2025h:3) encoder steps (2). There, the actual position (6064h) and the min position limit (607Dh:1) are set to 0 and the motor drives in the positive (clockwise) direction with a constant positive velocity (3). If the positive current threshold is reached, the motor moves back by -PositionOffset_CW (2025h:2) encoder steps (4). Then the motor is stopped and the max position limit (607Dh:2) is set to the actual position value (6064h). The velocity is limited to (6099h:1).

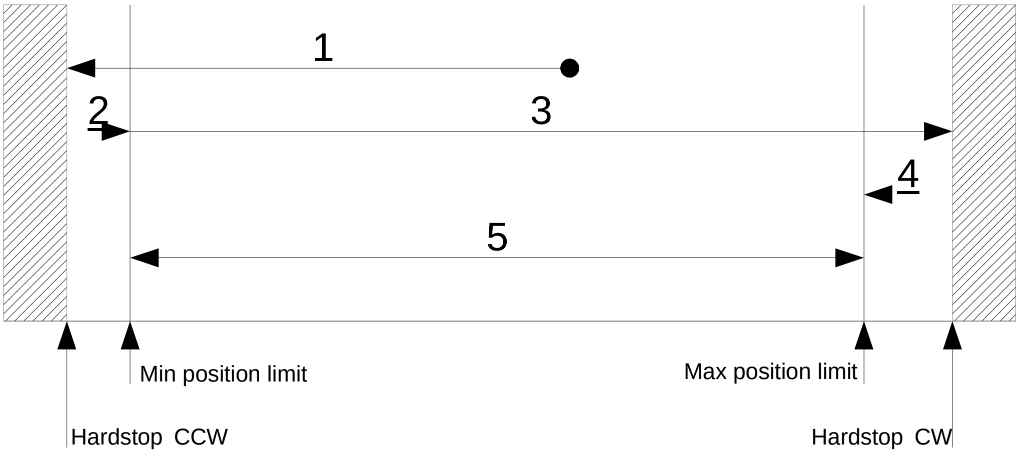


Figure 10: Double ended homing CCW

Upon completion, the motor can be moved in torque, velocity, or position mode within the position limits 607Dₕ :1 and 607Dₕ:2 (5).

8.1.7 Homing Method -5: Double Ended Clockwise Hard Stop Homing (compute scaler)

Homing method -5 uses the same homing process as homing method -3, but in addition the position scaler 2050ₕ:8 is automatically calculated and stored in the module. Thereby, the hardstop offsets (2) and (4) are used as unscaled encoder steps and the position limit range (5) of the application is scaled to 0 and 65535.

8.1.8 Homing Method -6: Double Ended Counterclockwise Hard Stop Homing (compute scaler)

Homing method -6 uses the same homing process as homing method -4, but in addition the position scaler 2050ₕ:8 is automatically calculated and stored in the module. Thereby, the hardstop offsets (2) and (4) are used as unscaled encoder steps and the position limit range (5) of the application is scaled to 0 and 65535.

8.1.9 Teach Mode

For certain applications, the homing range can be defined directly in teach mode without the hardstops

1. Jog the actuator to the min position, write2025ₕ:05=1 to mark the min software limit (607Dₕ:1) as 0.
2. Then jog the actuator to the max position, write 2025ₕ:05=2 to set the max software limit (607Dₕ:2) as the actual position
3. If the scaled homing range is needed, write 2025ₕ:05=3 to update the position scaler, and the max software limit (607Dₕ:2) will be changed to 65535

8.1.10 Parameter Saving After Homing

For RSF, RSA and FHA-Mini series actuators, the new homing parameters will be saved automatically to flash after each homing sequence, including Position Scaler (2050ₕ:8) , Min Position Limit (607Dₕ:1), Max Position Limit (607Dₕ:2), PositionOffset_CW (2025ₕ:2) and PositionOffset_CCW (2025ₕ:3).

For LPA and SHA series actuators, due to the longer flash saving time, all of the new homing parameters will only be updated in RAM and there is no automatic flash saving after homing. The homing parameters can be saved manually through SDO in Pre-operational mode when the PDOs are disabled.

8.2 Detailed Object Specifications

8.2.1 Object 6040$_h$: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to Figure 5 for detailed information.

Structure of the Control Word									
15 11	10	9	8	7	6 4	3	2	1	0
nu	r	oms	h	fr	oms	eo	qs	ev	so
MSB									LSB

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 199: Structure of the Control Word in hm Mode

Operation Mode specific Bits in hm Mode		
Bit	Name	Definition
4	Homing operation start	0: stop homing 1: start homing;
8	Halt	Not supported.

Table 200: Operation Mode specific Bits in hm Mode

Command Coding						
Command	Bits of Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
Disable voltage	0	x	x	0	x	7,9,10,12
Quick stop	0	x	0	1	x	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0-to-1	x	x	x	x	15

Table 201: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040_h	Controlword	Variable	UNSIGNED16

Table 202: Object Description (6040_h in hm Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	Yes	See command coding above.	

Table 203: Entry Description (6040_h in hm Mode)

8.2.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to Figure 5 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

Structure of the Status Word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dir	mot	oms		ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Legend: dir=direction of rotation; mot=motor activity; oms=operation mode specific; ila=internal limit active; tr=target reached; rm=remote; ms=manufacturer spec; w=warning; sod=switch on disabled; qs=quick stop; ve=voltage enabled; f=fault; oe=operation enabled; so=switch on; rtso=ready to switch on.

Table 204: Structure of the Status Word in hm Mode

HDLLC Specific Bits		
Bit	Name	Definition
14	Motor activity	0: Motor stands still. 1: Motor rotates.
15	Direction of rotation	This bit shows the direction of rotation.

Table 205: HDLLC Specific Bits

Operation Mode specific Bits in hm Mode		
Bit	Name	Definition
10	Target reached	Set when the zero position has been found or homing has been stopped by setting controlword bit 4 to zero.
12	Home attained	Set when zero position has been found.
13	Homing error	Not supported.

Table 206: Operation Mode specific Bits in hm Mode

State Coding	
Status word	FSA state
xxxx xxxx x0xx 0000_b	Not ready to switch on
xxxx xxxx x1xx 0000_b	Switch on disabled
xxxx xxxx x01x 0001_b	Ready to switch on
xxxx xxxx x01x 0011_b	Switched on
xxxx xxxx x01x 0111_b	Operation enabled
xxxx xxxx x00x 0111_b	Quick stop active
xxxx xxxx x0xx 1111_b	Fault reaction active
xxxx xxxx x0xx 1000_b	Fault

Table 207: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041_h	StatusWord	Variable	UNSIGNED16

Table 208: Object Description (6041_h in hm Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	Yes	See state coding above.	

Table 209: Entry Description (6041_h in hm Mode)

8.2.3 Object $606C_h$: Velocity Actual Value

This object shows the actual velocity value derived from the encoder.

Object Description			
Index	Name	Object Type	Data Type
$606C_h$	Velocity Actual Value	Variable	SIGNED32

Table 210: Object Description ($606C_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Velocity Actual Value	yes	-2147483648	2147483647	0	[rpm]	ro

Table 211: Entry Description ($606C_h$)

8.2.4 Object 607C$_h$: Home Offset

This object indicates the configured difference between the zero position for the application and the machine home position/home switch (found during homing). While 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. The effect of setting the home position to a non-zero value depends on the selected homing method. Negative values indicate the opposite direction.

Object Description			
Index	Name	Object Type	Data Type
607C$_h$	Home Offset	Variable	SIGNED32

Table 212: Object Description (607C$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Home Offset	no	-2147483648	2147483647	0		rw

Table 213: Entry Description (607C$_h$)

8.2.5 Object 6098$_h$: Homing Method

The actual homing method.

Object Description			
Index	Name	Object Type	Data Type
6098$_h$	Homing Method	Variable	SIGNED8

Table 214: Object Description (6098$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Homing Method	no	-6	35	0		rw

Table 215: Entry Description (6098$_h$)

8.2.6 Object 6099$_h$: Homing Speeds

This object indicates the configured speeds used during fast and slow homing procedure. In most homing modes, the home switch is searched with the fast speed first. When the home switch has been found, the motor will be decelerated to the slow speed (using the homing acceleration, object 609A$_h$) to search for the exact switch point. When the switch point has been found the motor will be stopped at that point.

Object Description			
Index	Name	Object Type	Data Type
6099$_h$	Homing Speeds	Array	UNSIGNED32

Table 216: Object Description (6099$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Fast Homing Speed	no	0	4294967295	1000		rw
2	Slow Homing Speed	no	0	4294967295	500		rw

Table 217: Entry Description (6099$_h$)

8.2.7 Object 609A$_h$: Homing Acceleration

This object indicates the configured acceleration and deceleration to be used during homing operation.

Object Description			
Index	Name	Object Type	Data Type
609A$_h$	Homing Acceleration	Variable	UNSIGNED32

Table 218: Object Description (609A$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Homing Acceleration	no	0	100000	2000	[rpm/s]	rw

Table 219: Entry Description (609A$_h$)

8.2.8 Object 2100$_h$: Home Offset Display

This object shows the home offset. The value is given in encoder increments.

Object Description			
Index	Name	Object Type	Data Type
2100$_h$	Home Offset Display	Variable	SIGNED32

Table 220: Object Description (2100$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Home Offset Display	no	-2147483648	2147483647	0		ro

Table 221: Entry Description (2100$_h$)

8.3 Example of Homing in hm Mode

With the actuator having been reset and then switched to pre-operational or operational by NMT commands, the following is an example of hard-stop homing in hm mode (the values are decimal):

- Select hm mode by writing 6 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Select homing method -1 (or -2) by writing -1 (or -2) to object 6098_h.
- Set the homing speeds by writing e.g. 500 to object 6099_h sub index 1 and e.g. 200 to object 6099_h sub index 2.
- Write 31 to object 6040_h to start the homing process.
- When homing has finished, write 15 to object 6040_h again.

9 Cyclic Synchronous Position Mode

The cyclic synchronous position mode is used to directly control the position of the motor. It contains limit functions, but not a trajectory generator. The trajectory generator is located in the control device (the master), not in the drive device. In cyclic synchronous manner, the control device provides a target position to the drive device, which performs position control, velocity control and torque control.

The main control parameters are the target position (object 607A$_h$, see section 0) and the interpolation time period (object 60C2$_h$, see section 9.1.10). The drive automatically sets the velocity in such a manner that the next target position is reached within the interpolation time period. Acceleration and deceleration ramps are not used in this mode.

The cyclic synchronous position mode covers the following sub-functions:

- Position demand value input directly via an object.
- Monitoring of the position.
- Limiting the position using the software limits or the hardware limit switches.

9.1 Detailed Object Specifications

9.1.1 Object 6040$_h$: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to Figure 5 for detailed information. The cyclic synchronous position mode does not use any mode specific bits of the control word.

Structure of the Control Word							
15 9	8	7	6 4	3	2	1	0
nu	h	fr	nu	eo	qs	ev	so
MSB							LSB

Legend: nu=not used; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 222: Structure of the Control Word in csp Mode

Command Coding						
Command	Bits of Control Word					Transitions
	Bit 7	Bit3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3,4
Disable voltage	0	x	x	0	x	7,9,10,12
Quick stop	0	x	0	1	x	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0-to-1	x	x	x	x	15

Table 223: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040_h	Controlword	Variable	UNSIGNED16

Table 224: Object Description (6040_h in csp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See command coding above	

Table 225: Entry Description (6040_h in csp Mode)

9.1.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to Figure 5 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

Structure of the Status Word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dir	mot	oms		ila	r	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Legend: dir=direction of rotation; mot=motor activity; oms=operation mode specific; ila=internal limit active; r=reserved; rm=remote; ms=manufacturer spec; w=warning; sod=switch on disabled; qs=quick stop; ve=voltage enabled; f=fault; oe=operation enabled; so=switch on; rtso=ready to switch on.

Table 226: Structure of the Status Word in csp Mode

Specific Bits		
Bit	Name	Definition
14	Motor activity	0: Motor stands still. 1: Motor rotates.
15	Direction of rotation	This bit shows the direction of rotation.

Table 227: Specific Bits

Operation Mode specific Bits in csp Mode		
Bit	Name	Definition
10	Reserved	Not used.
12	Target position ignored	0: Target position ignored. 1: Target position used as input to position controller.
13	Following error	0: No following error. 1: Following error.

Table 228: Operation Mode specific Bits in csp Mode

State Coding	
Status word	FSA state
xxxx xxxx x0xx 0000b	Not ready to switch on
xxxx xxxx x1xx 0000b	Switch on disabled
xxxx xxxx x01x 0001b	Ready to switch on
xxxx xxxx x01x 0011b	Switched on
xxxx xxxx x01x 0111b	Operation enabled
xxxx xxxx x00x 0111b	Quick stop active
xxxx xxxx x0xx 1111b	Fault reaction active
xxxx xxxx x0xx 1000b	Fault

Table 229: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041h	StatusWord	Variable	UNSIGNED16

Table 230: Object Description (6041h in csp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See state coding above	

Table 231: Entry Description (6041h in csp Mode)

9.1.3 Object 6062h: Position Demand Value

This object provides the demanded position value. The value is given in microsteps. Object 6062h indicates the actual position that the motor should have. It is not to be confused with objects 6063h and 6064h.

Object Description			
Index	Name	Object Type	Data Type
6062h	Position Demand Value	Variable	SIGNED32

Table 232: Object Description (6062h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 233: Entry Description (6062h)

9.1.4 Object 6063h: Position Actual Internal Value

This object provides the actual value of the encoder or the motor. Please use the sensor selection object 608Fh (see section 5.1.8) for selecting the motor or the encoder first. Object 6063h indicates the actual position of the encoder or the motor, re-scaled to the microstep resolution. The value is given in microsteps.

Object Description			
Index	Name	Object Type	Data Type
6063$_h$	Position Actual Internal Value	Variable	SIGNED32

Table 234: Object Description (6063$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 235: Entry Description (6063$_h$)

9.1.5 Object 6064$_h$: Position Actual Value

This object provides the actual value of the position measurement device. It always contains the same value as object 6063$_h$.

Object Description			
Index	Name	Object Type	Data Type
6064$_h$	Position Actual Value	Variable	SIGNED32

Table 236: Object Description (6064$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 237: Entry Description (6064$_h$)

9.1.6 Object 606C$_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in units of rpm.

Object Description			
Index	Name	Object Type	Data Type
606C$_h$	Velocity Actual Value	Variable	SIGNED32

Table 238: Object Description (606C$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 239: Entry Description (606C$_h$)

9.1.7 Object 607A$_h$: Target Position

The target position is the position that the drive should move to in cyclic synchronous position mode using the current interpolation time period. In csp mode this value is always interpreted as an absolute value.

Object Description			
Index	Name	Object Type	Data Type
607A$_h$	Target Position	Variable	SIGNED32

Table 240: Object Description (607A$_h$ in csp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	SIGNED32	0

Table 241: Entry Description (607A$_h$ in csp Mode)

9.1.8 Object 607D$_h$: Software Position Limit

This object indicates the configured maximal and minimal software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

Corrected_min_position_limit = min_position_limit - home_offset

Corrected_max_position_limit = max_position_limit - home_offset

Object Description			
Index	Name	Object Type	Data Type
607D$_h$	Software Position Limit	Array	SIGNED32

Table 242: Object Description (607D$_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Minimum Position Limit	rw	no	SIGNED32	-2147483648
2	Maximum Position Limit	rw	no	SIGNED32	2147483647

Table 243: Entry Description (607D$_h$)

9.1.9 Object 60B0$_h$: Position Offset

This object provides an offset to the target position (object 607A$_h$, see section 9.1.7). The value is given in microsteps and will be added to the target position.

Object Description			
Index	Name	Object Type	Data Type
60B0$_h$	Position Offset	Variable	SIGNED32

Table 244: Object Description (60B0$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	-2147483648 ... 2147483647	0

Table 245: Entry Description (60B0$_h$)

9.1.10 Object 60C2$_h$: Interpolation Time Period

This object indicates the interpolation cycle time. The interpolation time period (sub-index 01$_h$) is given in $10^{interpolation\_time\_index}$ s. The interpolation time index (sub-index 02$_h$) is dimensionless.

Object Description			
Index	Name	Object Type	Data Type
60C2$_h$	Offset Torque	Record	Interpolation time period record (0080$_h$)

Table 246: Object Description (60C2$_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Highest sub-index supported	ro	no	UNSIGNED8	2
1	Interpolation time period value	rw	no	UNSIGNED8	1
2	Interpolation time index	rw	no	-3... 3	-3

Table 247: Entry Description (60C2$_h$)

9.2 Example of csp Mode Operation

The following is an example of running in pp mode (the values are decimal), assume that the actuator has been reset and then switched to pre-operational or operational by NMT commands

- Write the new interpolation time period to 60C2$_h$:1 and 60C2$_h$:2
- Select CSP mode by writing 8 to object 6060h.
- Write 6 to object 6040$_h$ to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040$_h$ to switch to SWITCHED_ON state.
- Write the desired target position (e.g. 500000) to object 607A$_h$.
- Write 15 to object 6040$_h$ to switch to OPERATION_ENABLED state. The motor now accelerates to the target position.
- Write the new target position to object 607A$_h$.in each Interpolation time interval.

10 Cyclic Synchronous Velocity Mode

The cyclic synchronous velocity mode is used to directly control the velocity of the motor. It contains limit functions, but not a trajectory generator. The trajectory generator is located in the control device (the master), not in the drive device. In cyclic synchronous manner, the control device provides a target velocity to the drive device, which performs position control, velocity control and torque control.

The main control parameters are the target velocity (object 60FF$_h$, see section 10.1.4) and the interpolation time period (object 60C2$_h$, see section 10.1.7). The drive automatically sets the acceleration in such a manner that the next target velocity is reached within the interpolation time period. Acceleration and deceleration ramps are not used in this mode.

The cyclic synchronous velocity mode covers the following sub-functions:

- Velocity demand value input directly via an object.
- Monitoring of the position.
- Limiting the position using the software limits or the hardware limit switches.

10.1 Detailed Object Specifications

10.1.1 Object 6040$_h$: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to Figure 5 for detailed information. The cyclic synchronous velocity mode does not use any mode specific bits of the control word.

Structure of the Control Word							
15 9	8	7	6 4	3	2	1	0
nu	h	fr	nu	eo	qs	ev	so
MSB							LSB

Legend: nu=not used; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 248: Structure of the Control Word in csv Mode

Command Coding						
Command	Bits of Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
Disable voltage	0	x	x	0	x	7,9,10,12
Quick stop	0	x	0	1	x	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0-to-1	x	x	x	x	15

Table 249: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040_h	Controlword	Variable	UNSIGNED16

Table 250: Object Description (6040_h in csv Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	Yes	See command coding above.	

Table 251: Entry Description (6040_h in csv Mode)

10.1.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to Figure 5 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

Structure of the Status Word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dir	mot	oms		ila	r	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Legend: dir=direction of rotation; mot=motor activity; oms=operation mode specific; ila=internal limit active; r=reserved; rm=remote; ms=manufacturer spec; w=warning; sod=switch on disabled; qs=quick stop; ve=voltage enabled; f=fault; oe=operation enabled; so=switch on; rtso=ready to switch on.

Table 252: Structure of the Status Word in csv Mode

Specific Bits		
Bit	Name	Definition
14	Motor activity	0: Motor stands still. 1: Motor rotates.
15	Direction of rotation	This bit shows the direction of rotation.

Table 253: Specific Bits

Operation Mode specific Bits in csv Mode		
Bit	Name	Definition
10	Reserved	Not used.
12	Target position ignored	0: Target velocity ignored. 1: Target velocity used as input to velocity controller.
13	Reserved	Not used.

Table 254: Operation Mode Specific Bits in csv Mode

State Coding	
Status word	FSA state
xxxx xxxx x0xx 0000$_b$	Not ready to switch on
xxxx xxxx x1xx 0000$_b$	Switch on disabled
xxxx xxxx x01x 0001$_b$	Ready to switch on
xxxx xxxx x01x 0011$_b$	Switched on
xxxx xxxx x01x 0111$_b$	Operation enabled
xxxx xxxx x00x 0111$_b$	Quick stop active
xxxx xxxx x0xx 1111$_b$	Fault reaction active
xxxx xxxx x0xx 1000$_b$	Fault

Table 255: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041 $_h$	Status Word	Variable	UNSIGNED16

Table 256: Object Description (6041$_h$ in csv Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See state coding above	

Table 257: Entry Description (6041$_h$ in csv Mode)

10.1.3 Object 606C$_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in units of rpm.

Object Description			
Index	Name	Object Type	Data Type
606C$_h$	Velocity Actual Value	Variable	SIGNED32

Table 258: Object Description (606C$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 259: Entry Description (606C$_h$)

10.1.4 Object 60FF$_h$: Target Velocity

In csv mode the target velocity specifies the velocity that is to be reached within the interpolation time period. The values are given in units of rpm.

Object Description			
Index	Name	Object Type	Data Type
60FF$_h$	Target Velocity	Variable	SIGNED32

Table 260: Object Description (60FF$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	SIGNED32	0

Table 261: Entry Description (60FF$_h$)

10.1.5 Object 607D$_h$: Software Position Limit

This object indicates the configured maximal and minimal software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

$$Corrected\_min\_position\_limit = min\_position\_limit - home\_offset$$
$$Corrected\_max\_position\_limit = max\_position\_limit - home\_offset$$

Object Description			
Index	Name	Object Type	Data Type
607D$_h$	Software Position Limit	Array	SIGNED32

Table 262: Object Description (607D$_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Minimum Position Limit	rw	no	SIGNED32	-2147483648
2	Maximum Position Limit	rw	no	SIGNED32	2147483647

Table 263: Entry Description (607D$_h$)

10.1.6 Object 60B1$_h$: Velocity Offset

This object provides an offset to the target velocity (object 60FF$_h$, see section 10.1.4). The value will be added to the target velocity.

Object Description			
Index	Name	Object Type	Data Type
60B1$_h$	Velocity Offset	Variable	INTEGER32

Table 264: Object Description (60B1$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	-2147483648. . . 2147483647	0

Table 265: Entry Description (60B1$_h$)

10.1.7 Object 60C2$_h$: Interpolation Time Period

This object indicates the interpolation cycle time. The interpolation time period (sub-index 01$_h$) is given in $10^{interpolation\_time\_index}$ s. The interpolation time index (sub-index 02$_h$) is dimensionless.

Object Description			
Index	Name	Object Type	Data Type
60C2$_h$	Offset Torque	Record	Interpolation time period record (0080h)

Table 266: Object Description (60C2$_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Highest sub-index supported	ro	no	UNSIGNED8	2
1	Interpolation time period value	rw	no	UNSIGNED8	1
2	Interpolation time index	rw	no	-3. . . 3	-3

Table 267: Entry Description (60C2$_h$)

10.2 Example of csv Mode Operation

- Write the new interpolation time period to 60C2$_h$:1 and 60C2$_h$:2
- Select CSV mode by writing 9 to object 6060$_h$.
- Write 6 to object 6040$_h$ to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040$_h$ to switch to SWITCHED_ON state.
- Write the desired target velocity (e.g. 2000) to object 60FF$_h$.
- Write 15 to object 6040$_h$ to switch to OPERATION_ENABLED state. The motor now accelerates to the target velocity.
- Write the new target velocity to object 60FF$_h$.in each Interpolation time interval
- Stop the motor by writing 0 to object 60FF$_h$.

11 Cyclic Synchronous Torque Mode

The cyclic synchronous torque mode is used to directly control the torque of the motor, without the need for position or velocity control. It contains limit functions, but not a trajectory generator.

The cyclic synchronous torque mode covers the following sub-functions:

- Demand value input directly via an object.
- Monitoring and limiting the torque.

11.1 Detailed Object Specifications

11.1.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to Figure 5 for detailed information. The cyclic synchronous torque mode does not use any mode specific bits of the control word.

Structure of the Control Word							
15 9	8	7	6 4	3	2	1	0
nu	h	fr	nu	eo	qs	ev	so
MSB							LSB

Legend: nu=not used; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 268: Structure of the Control Word in cst Mode

Command Coding						
Command	Bits of Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
Disable voltage	0	x	x	0	x	7,9,10,12
Quick stop	0	x	0	1	x	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0-to-1	x	x	x	x	15

Table 269: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040_h	Controlword	Variable	UNSIGNED16

Table 270: Object Description (6040_h in cst Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See command coding above.	

Table 271: Entry Description (6040_h in cst Mode)

11.1.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to Figure 5 for detailed information. The object is structured as defined below.

For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

Structure of the Status Word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dir	mot	oms		ila	r	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Legend: dir=direction of rotation; mot=motor activity; oms=operation mode specific; ila=internal limit active; r=reserved; rm=remote; ms=manufacturer spec; w=warning; sod=switch on disabled; qs=quick stop; ve=voltage enabled; f=fault; oe=operation enabled; so=switch on; rtso=ready to switch on.

Table 272: Structure of the Status Word in cst Mode

HDLLC Specific Bits		
Bit	Name	Definition
14	Motor activity	0: Motor stands still. 1: Motor rotates.
15	Direction of rotation	This bit shows the direction of rotation.

Table 273: HDLLC Specific Bits

Operation Mode specific Bits in cst Mode		
Bit	Name	Definition
10	Reserved	Not used.
12	Target torque ignored	0: Target torque ignored. 1: Target torque used as input to control loop.
13	Reserved	Not used.

Table 274: Operation Mode Specific Bits in cst Mode

State Coding	
Status word	FSA state
xxxx xxxx x0xx 0000$_b$	Not ready to switch on
xxxx xxxx x1xx 0000$_b$	Switch on disabled
xxxx xxxx x01x 0001$_b$	Ready to switch on
xxxx xxxx x01x 0011$_b$	Switched on
xxxx xxxx x01x 0111$_b$	Operation enabled
xxxx xxxx x00x 0111$_b$	Quick stop active
xxxx xxxx x0xx 1111$_b$	Fault reaction active
xxxx xxxx x0xx 1000$_b$	Fault

Table 275: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041$_h$	StatusWord	Variable	UNSIGNED16

Table 276: Object Description (6041$_h$ in cst Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See state coding above	

Table 277: Entry Description (6041$_h$ in cst Mode)

11.1.3 Object 6071$_h$: Target Torque

This object gives the target motor current.

Object Description			
Index	Name	Object Type	Data Type
6071$_h$	Target Torque	Variable	SIGNED32

Table 278: Object Description (6071$_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Target Torque	yes	Depends on actuator	Depends on actuator	0	[mA]	rw

Table 279: Entry Description (6071$_h$)

11.1.4 Object 6077h: Torque Actual Value

The actual motor current.

Object Description			
Index	Name	Object Type	Data Type
6077h	Torque Actual Value	Variable	SIGNED32

Table 280: Object Description (6077h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Torque Actual Value	yes	-2147483648	2147483647	0	[mA]	ro

Table 281: Entry Description (6077h)

11.1.5 Object 60B2h: Torque offset

The actual set torque offset.

Object Description			
Index	Name	Object Type	Data Type
60B2h	Torque offset	Variable	SIGNED32

Table 282: Object Description (60B2h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Torque offset	no	-2147483648	2147483647	0	[mA]	rw

Table 283: Entry Description (60B2h)

11.2 Example of cst Mode Operation

With the actuator having been reset and then switched to pre-operational or operational by NMT commands, the following is an example of running in cst mode (the values are decimal):

- If you do not have any limit switches connected, first disable the limit switch inputs by writing 3 to object 2005h.
- Select cst mode by writing 10 to object 6060h.
- Write 6 to object 6040h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040h to switch to SWITCHED_ON state.
- Write 15 to object 6040h to switch to OPERATION_ENABLED state.
- Write the desired torque (e.g. 1000) to object 6071h to start the motor.
- To stop the motor, write 0 to object 6071h.

12 IO Summary

12.1 IO port mapping

IO ports can be used for simple signal conditioning or status indicator near the actuator. There are typically 2 or 4 IO ports in each actuator with 3 different categories—Programmable IO, Programmable Input and Opto-isolated Input as shown in the table 284

Actuator (IO QTY)	[1]Programmable IO	[2]Programmable input	[3]Opto isolated input
RSF-5 (2)	IO1	IN2	-
RSA-8 (2)	IO1	IN2	-
FHA-8/11/14 (4)	IO1, IO2	-	IN3, IN4
LPA-20 (4)	IO1, IO2	-	IN3, IN4
SHA-20/25/32 (4)	IO1, IO2	-	IN3, IN4

[1] Programmable IO: Analog Input, Digital Input or Digital Output
[2] Programmable Input: Analog Input or Digital Input
[3] Opto-Isolated Input: Digital input referred to the isolated ground (or COM port)

Table 284: IO Summary

The general electrical specification of each type of IO port is as below:

IO port type	Descriptions
Programmable IO (Non-isolated)	A: Analog Input --input 0-10.5V→ADC 0-4095 (12-bit resolution) --allowed range -0.5V-24V --typical value 4.1V, with pull-up
	B: Digital Input --Logic ‘0’: <0.8V --Logic ‘1’: >2V --With PULL_IN option to support NPN or PNP sensor --Programmable threshold
	C: Digital Output (open drain) --Logic ‘0’: switch on, output is pulled down, max 1A --Logic ‘1’: switch off, pulled up by supply voltage (24V typical,30V max)
Programmable Input (Non-isolated, without Output)	A: Analog Input --input 0-10.5V→ADC 0-4095 (12-bit resolution) --allowed range -0.5V-24V --typical value 4.1V, with pull-up
	B: Digital Input --Logic ‘0’: <0.8V --Logic ‘1’: >2V --With PULL_IN option to support NPN or PNP sensor --Programmable threshold
Opto-isolated Input	Digital Input --'Logic ‘0’: 0-18 V --Logic ‘1’: 19-24V, typical 24V --Max Frequency: 20kHz

Table 285: The electrical specification of IO ports

There are two different actuator groups that have different IO channel configurations.

The RSF-5 and RSA-8 have one programmable IO and one programmable input.

a. The single programmable IO can be configured to either become the function of an input or output.
b. The single programmable input can support NPN, and PNP sensors by a programmable threshold. The port mapping is as below

IO port Number	Mapping Objects
IO1 (Non-isolated)	A. Analog reading from Object **270E$_h$:05**
	B. Digital input, read from **2702$_h$→bit0** with programmable threshold **2020$_h$:04**
	C. Digital Output, please refer to table 287, 288 for more detail Logic **1**, to enable digital output, and the port will be connected to GND Logic **0**, to disable digital output, and the port will be pulled up by 5V or by external voltage
IN2 (Non-isolated)	A. Analog reading from Object **270E$_h$:06**
	B. Digital input, read from **2702$_h$→bit 1** with programmable threshold **2020$_h$:04** C. No Digital Output

Table 286: Group1, RSF-5 and RSA-8 actuator with two IO ports

The detailed Digital Output mapping for RSF-5 and RSA-8 are as below

	Bit3	Bit2	Bit1	Bit0
Digital Output 2703$_h$:01	PULLUP2	NA	NA	DOUT1
Output Mask 2703$_h$:02	MASK-PULLUP2	NA	NA	MASK-DOUT1

Table 287: Digital Output mapping 2703$_h$ for RSF-5

	Bit3	Bit2	Bit1	Bit0
Digital Output 2703$_h$:01	PULLUP2	PULLUP1	NA	DOUT1
Output Mask 2703$_h$:02	MASK-PULLUP2	MASK-PULLUP1	NA	MASK-DOUT1

Table 288: Digital Output mapping 2703$_h$ for RSA-8

The FHA, LPA and SHA series have a total of 4 IO ports

a. There are two programmable IOs, which can each be programmed as an analog input, a digital input or a digital output;
b. These actuators include two opto-isolated inputs that can support NPN and PNP sensors.

IO port Number	Mapping Objects
IO1 (Non-isolated)	A. analog reading from Object **270E$_h$:05**
	B. Digital input, read from **2702$_h$→Bit0**
	C. Digital Output, please refer to table 290 for more detail Logic **1**, to enable digital output, and the port will be connected to GND Logic **0**, to disable digital output, and the port will be pulled up by 5V or by external voltage
IO2 (Non-isolated)	A. analog reading from Object **270E$_h$:06**
	B. Digital input, read from **2702$_h$→Bit1**
	C. Digital Output, please refer to table 290 for more detail Logic **1**, to enable digital output, and the port will be connected to GND Logic **0**, to disable digital output, and the port will be pulled up by 5V or by external voltage
IN3 (Isolated)	Digital input, read from **2702$_h$→Bit2**
IN4 (Isolated)	Digital input, read from **2702$_h$→Bit3**

Table 289: Group 2, FHA, LPA and SHA actuators with four IO ports

The detailed Digital Output mapping for FHA, LPA and SHA actuators are as below

	Bit3	Bit2	Bit1	Bit0
Digital Output 2703$_h$:01	NA	NA	DOUT2	DOUT1
Output Mask 2703$_h$:02	NA	NA	MASK-DOUT2	MASK-DOUT1

Table 290: Digital Output mapping 2703$_h$ for FHA, LPA and SHA actuators

12.2 Digital Input as Limit switch

When the limit switches are enabled in Object 2005$_h$ the left switch (negative) and right switch (positive) are tied internally to Bit0 and Bit1 respectively of Digital Input 2702$_h$.
Bit0: IN1 (REF_L)
Bit1: IN2 (REF_R)

12.3 LED for NMT State

A green LED is used to show the state of NMT state machine.

Green LED	NMT State
Green LED blinks slow, on for 200ms in every 1s	STOPPED
Green LED blinking fast, 200ms on and 200ms off	PRE-OP
Green LED on (solid)	OP

Table 291: LED for NMT state

12.4 Motor Torque Off

LPA and SHA actuators include the Motor Torque Off protection which is designed to STO (Safe Torque Off) specification but has not undergone certification. MTO1 and MTO2 ports should be connected to +24V and 0VDC separately. The actuator will be disabled when either MTO port is open.

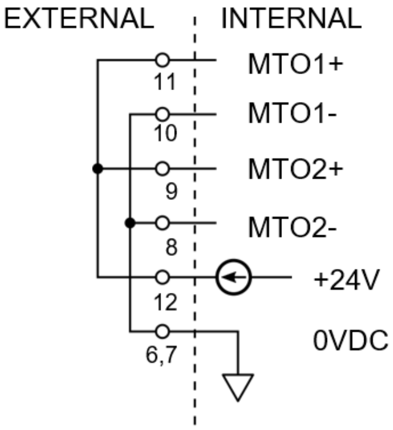


Figure 11: Motor Torque Off connection

13 Emergency Messages (EMCY)

The module sends an emergency message if an error occurs. The message contains information about the error type. The module can map internal errors and object 1001_h (error register) is part of every emergency object.

13.1 Format of EMCY Messages

The COB-ID of an EMCY is 80_h+ID of the node. So for example it is 81_h for node #1. The first two bytes contain the error code with LSB first. The third data byte contains the content of the error register (object 1001_h). The other five bytes can contain additional information, depending on the error code.

EMCY								
COB-ID	1	2	3	4	5	6	7	8
0x80_h+Node ID	Error code		Error register	Additional bytes				
	LSB	MSB	(1001_h)	#1…#5				

Table 292: Format of EMCY Messages

13.2 Error Code

The following tables shows all EMCY error coders used by.

Emergency Messages (EMCY)						
Error code	Additional byte					Description
	1	2	3	4	5	
0000_h	0	0	0	0	0	**Fault reset** The fault reset command has been executed.
4310_h	1	0	0	0	0	**IIT 1 error** The motor driver has been switched off because the IIT 1 limit has been reached.
4310_h	2	0	0	0	0	**IIT 2 error** The motor driver has been switched off because the IIT 2 limit has been exceeded.
5441_h	0	255	0	0	0	**Shutdown switch active** The enable signal is missing (due to the shutdown switch) and the motor driver has been switched off.
6320_h	0	255	0	0	0	**Parameter error** The data in the received PDO is either wrong or cannot be accepted due to the internal state of the drive.
8100_h	0	255	0	0	0	**Communication error** General CAN bus communication error.

Error code	Additional byte					Description
	1	2	3	4	5	
8110h	1	255	0	0	0	**CAN controller overflow** The receive message buffer of the CAN controller hardware is full and some CAN messages are lost.
8110h	2	255	0	0	0	**CAN Tx buffer overflow** The software CAN transmit buffer is full and thus some CAN messages are lost.
8110h	3	255	0	0	0	**CAN Rx buffer overflow** The software CAN receive buffer is full and so some CAN messages are lost.
8120h	0	255	0	0	0	**CAN error passive** The CAN controller has detected communication errors and has entered the CAN Error passive state.
8130h	0	255	0	0	0	**Heartbeat or lifeguard error** The module did not receive a heartbeat or lifeguard message in time.
8140h	0	255	0	0	0	**CAN controller recovered from bus-off state** The CAN controller has detected too many errors and has changed into the bus-off state. The drive has been stopped and disabled. This message is sent after the CAN controller has recovered from bus-off state and is bus-on again.
8210h	0	255	0	0	0	**PDO not processed due to length error** A PDO sent to the module could not be processed because too few bytes were supplied.
8220h	0	255	0	0	0	**PDO length exceeded** A PDO sent to the module could not be processed because too many bytes were supplied.
8611h	1	0	0	0	0	**Position window following error** The deviation between motor position counter and encoder position counter has exceeded the position following error window.
8611h	2	0	0	0	0	**Velocity window following error** The deviation between commanded velocity and measured velocity has exceeded the velocity following error window.
FF00h	0	0	0	0	0	**Undervoltage** The supply voltage is too low to drive a motor.
FF01h	1	0	0	0	0	**Positive software limit** The actual position is outside the range defined by object 607Dh.
FF01h	2	0	0	0	0	**Negative software limit** The actual position is outside the range defined by object 607Dh.

Table 293: Emergency Messages (EMCY)

14 SDO Abort Codes

Trying to access an object via SDO read or SDO write may result in an error. In such a case an SDO abort transfer message containing an abort code will be sent. The following table lists all SDO abort codes defined by the CiA-301 standard. Not all of these are used by the HDLLC actuators.

SDO Abort Codes	
Abort code	Description
05030000_h	Toggle bit not alternated.
05040000_h	SDO protocol timed out.
05040001_h	Client/server command specifier not valid or unknown.
05040002_h	Invalid block size.
05040003_h	Invalid sequence number.
05040004_h	CRC error.
05040005_h	Out of memory.
06010000_h	Unsupported access to an object.
06010001_h	Attempt to read a write only object.
06010002_h	Attempt to write a read only object.
06020000_h	Object does not exist in object dictionary.
06040041_h	Object cannot be mapped to the PDO.
06040042_h	The number and length of the objects to be mapped would exceed the PDO length.
06040043_h	General parameter incompatibility reason.
06040047_h	General internal incompatibility in the device.
06060000_h	Access failed due to a hardware error.
06070010_h	Data type does not match, length of service parameter does not match.
06070012_h	Data type does not match, length of service parameter too high.
06070013_h	Data type does not match, length of service parameter too low.
06090011_h	Sub-index does not exist.
06090030_h	Invalid value for parameter.
06090031_h	Value of parameter too high.
06090032_h	Value of parameter too low.
06090036_h	Maximum value is less than minimum value.
$060A0023_h$	Resource not available.
08000000_h	General error.
08000020_h	Data cannot be transferred or stored to the application.
08000021_h	Data cannot be transferred or stored to the application because of local control.

Abort code	Description
08000022h	Data cannot be transferred or stored to the application because of the present device state.
08000023h	Object dictionary dynamic generation failed or no object dictionary is present.
08000024h	No data available.

Table 294: SDO Abort Codes

15 Figures Index

Figure 1: NMT State Machine 11
Figure 2: Communication Architecture 12
Figure 3: System Model 13
Figure 4 IIt Monitor Windows 38
Figure 5: DS402 Finite State Machine 50
Figure 6: Homing Mode Function 67
Figure 7: Single ended homing CW................................ 69
Figure 8: Single ended homing CCW 69
Figure 9: Double ended homing CW 70
Figure 10: Double ended homing CCW.......................... 71
Figure 11: Motor Torque Off connection.................... 98

16 Tables Index

Table 1: Abbreviations used in this Manual..................... 8
Table 2: Service Primitives 10
Table 3: Service Types 10
Table 4: Object Dictionary.. 13
Table 5: Object Description (1000$_H$) 15
Table 6: Entry Description (1000$_H$) 15
Table 7: Object Description (1001$_H$) 15
Table 8: Entry Description (1001$_H$) 15
Table 9: Error Register Bits... 16
Table 10: Value Definition (1005$_H$) 16
Table 11: Object Description (1005$_H$) 16
Table 12: Entry Description (1005$_H$) 16
Table 13: Object Description (1008$_H$) 17
Table 14: Entry Description (1008$_H$) 17
Table 15: Object Description (1009$_H$) 17
Table 16: Entry Description (1009$_H$) 17
Table 17: Object Description (100A$_H$).............................. 18
Table 18: Entry Description (100A$_H$) 18
Table 19: Save Signature 18
Table 20: Object Description (1010$_H$) 19
Table 21: Entry Description (1010$_H$) 19
Table 22: Load Signature... 19
Table 23: Object Description (1011$_H$) 20
Table 24: Entry Description (1011$_H$) 20
Table 25: Object Description (1014$_H$) 20
Table 26: Entry Description (1014$_H$) 20
Table 27: Object Description (1015$_H$) 20
Table 28: Entry Description (1015$_H$) 20
Table 29: Value Definition (1016$_H$) 21
Table 30: Object Description (1016$_H$) 21
Table 31: Entry Description (1016$_H$) 21
Table 32: Object Description (1017$_H$) 21
Table 33: Entry Description (1017$_H$) 21
Table 34: Object Description (1018$_H$) 22
Table 35: Entry Description (1018$_H$) 22
Table 36: Object Description (1029$_H$) 22
Table 37: Entry Description (1029$_H$) 22
Table 38: Object Description (1400$_H$) 23
Table 39: Entry Description (1400$_H$).............................. 23
Table 40: Object Description (1600$_H$) 24
Table 41: Entry Description (1600$_H$).............................. 24
Table 42: Object Description (1800$_H$) 25
Table 43: Entry Description (1800$_H$).............................. 25
Table 44: Object Description (1A00$_H$) 26
Table 45: Entry Description (1A00$_H$) 26
Table 46: Object Description (2005$_H$) 27
Table 47: Entry Description (2005$_H$).............................. 27
Table 48: Bit Definitions (2005$_H$).................................. 27
Table 49: Status Flags (200D$_H$) 28
Table 50: Object Description (200D$_H$)............................ 28
Table 51: Entry Description (200D$_H$) 29
Table 52: Object Description (200E$_H$) 29
Table 53: Entry Description (200E$_H$).............................. 29
Table 54: Object Description (200F$_H$) 29
Table 55: Entry Description (200F$_H$).............................. 29
Table 56: Object Description (2010$_H$) 30
Table 57: Entry Description (2010$_H$).............................. 30
Table 58: Object Description (2015$_H$) 30
Table 59: Entry Description (2015$_H$).............................. 30
Table 60: Object Description (2020$_H$) 31
Table 61: Entry Description (2020$_H$)............................. 31
Table 62: Maximum Values for Object (2020$_H$).............. 32
Table 63: Object Description (2025$_H$)........................... 33
Table 64: Entry Description (2025$_H$).............................. 33
Table 65: Object Description (2030$_H$) 33
Table 66: Entry Description (2030$_H$).............................. 34
Table 67: Object Description (2040$_H$) 34
Table 68: Entry Description (2040$_H$).............................. 34
Table 69: Object Description (2050$_H$) 35
Table 70: Entry Description (2050$_H$).............................. 35
Table 71: Object Description (2055$_H$) 35
Table 72: Entry Description (2055$_H$).............................. 36
Table 73: Object Description (2056$_H$) 36
Table 74: Entry Description (2056$_H$).............................. 36
Table 75: Object Description (2060$_H$) 36
Table 76: Entry Description (2060$_H$).............................. 36
Table 77: Object Description (2080$_H$) 36
Table 78: Entry Description (2080$_H$).............................. 37

Table 79: Objects Description (2090$_H$) ... 37
Table 80: Entry Description (2090$_H$) ... 37
Table 81: Object Description(2095$_H$) ... 38
Table 82: Entry Description (2095$_H$) ... 38
Table 83: Object Description (2096$_H$) ... 38
Table 84: Entry Description (2096$_H$) ... 39
Table 85: Object Description (2100$_H$) ... 39
Table 86: Entry Description (2100$_H$) ... 39
Table 87: Object Description (2702$_H$) ... 39
Table 88: Entry Description (2702$_H$) ... 39
Table 89: Bit Mapping (2702$_H$) ... 39
Table 90: Object Description (2703$_H$) ... 40
Table 91: Entry Description (2703$_H$) ... 40
Table 92:Bit Mapping(2703$_H$) ... 40
Table 93: Object Description (2704$_H$) ... 40
Table 94: Entry Description (2704$_H$) ... 40
Table 95: Object Description (2705$_H$) ... 41
Table 96: Entry Description (2705$_H$) ... 41
Table 97: Object Description (2706$_H$) ... 41
Table 98: Entry Description (2706$_H$) ... 41
Table 99: Object Description (270E$_H$) ... 41
Table 100: Entry Description (270E$_H$) ... 42
Table 101: Object Description (5FFF$_H$) ... 42
Table 102: Entry Description (5FFF$_H$) ... 42
Table 103: Value Description (605A$_H$) ... 43
Table 104: Object Description (605A$_H$) ... 43
Table 105: Entry Description (605A$_H$) ... 43
Table 106: Value Description (605B$_H$) ... 44
Table 107: Object Description (605B$_H$) ... 44
Table 108: Entry Description (605B$_H$) ... 44
Table 109: Value Description (605C$_H$) ... 44
Table 110: Object Description (605C$_H$) ... 44
Table 111: Entry Description (605C$_H$) ... 44
Table 112: Value Description (605D$_H$) ... 45
Table 113: Object Description (605D$_H$) ... 45
Table 114: Entry Description (605D$_H$) ... 45
Table 115: Value Description (605E$_H$) ... 45
Table 116: Object Description (605E$_H$) ... 45
Table 117: Entry Description (605E$_H$) ... 45
Table 118: Value Description (6060$_H$) ... 46
Table 119: Object Description (6060$_H$) ... 46
Table 120: Entry Description (6060$_H$) ... 46
Table 121: Value Description (6061$_H$) ... 46
Table 122: Object Description (6061$_H$) ... 47
Table 123: Entry Description (6061$_H$) ... 47
Table 124: Object Description (608F$_H$) ... 47
Table 125: Entry Description (608F$_H$) ... 47
Table 126: Position Encoder Resolution ... 47
Table 127: Object Description (60FD$_H$) ... 48
Table 128: Entry Description (60FD$_H$) ... 48
Table 129: Value Definition (6502$_H$) ... 48
Table 130: Object Description (6502$_H$) ... 48
Table 131: Entry Description (6502$_H$) ... 49
Table 132: Structure of the Control Word in pp Mode ... 51
Table 133: Operation Mode specific Bits in pp Mode ... 51
Table 134: Command Coding ... 51
Table 135: Object Description (6040$_H$ in pp Mode) ... 51
Table 136: Entry Description (6040$_H$ in pp Mode) ... 52
Table 137: Structure of the Status Word in pp Mode ... 52
Table 138: HDLLC Specific Bits ... 52
Table 139 Operation Mode specific Bits in pp Mode ... 52
Table 140: State Coding ... 53
Table 141: Object Description (6041$_H$ in pp Mode) ... 53
Table 142: Entry Description (6041$_H$ in pp Mode) ... 53
Table 143: Object Description (6062$_H$) ... 53
Table 144: Entry Description (6062$_H$) ... 53
Table 145: Object Description (6063$_H$) ... 54
Table 146: Entry Description (6063$_H$) ... 54
Table 147: Object Description (6064$_H$) ... 54
Table 148: Entry Description (6064$_H$) ... 54
Table 149: Object Description (6067$_H$) ... 55
Table 150: Entry Description (6067$_H$) ... 55
Table 151: Object Description (606B$_H$) ... 55
Table 152: Entry Description (606B$_H$) ... 55
Table 153: Object Description (606C$_H$) ... 55
Table 154: Entry Description (606C$_H$) ... 56
Table 155: Object Description (607A$_H$) ... 56
Table 156: Entry Description (607A$_H$) ... 56
Table 157: Object Description (607D$_H$) ... 56
Table 158: Entry Description (607D$_H$) ... 56
Table 159: Object Description (6081$_H$) ... 57
Table 160: Entry Description (6081$_H$) ... 57
Table 161: Object Description (6082$_H$) ... 57
Table 162: Entry Description (6082$_H$) ... 57
Table 163: Object Description (6083$_H$) ... 57
Table 164: Entry Description (6083$_H$) ... 58
Table 165: Object Description (6084$_H$) ... 58
Table 166: Entry Description (6084$_H$) ... 58
Table 167: Object Description (6085$_H$) ... 58
Table 168: Entry Description (6085$_H$) ... 58
Table 169: Structure of the Control Word in pv Mode 60
Table 170: Command Coding ... 60
Table 171: Object Description (6040$_H$ in pv Mode) ... 61
Table 172: Entry Description (6040$_H$ in pv Mode) ... 61
Table 173: Structure of the Status Word in pv Mode ... 61
Table 174: HDLLC Specific Bits ... 61
Table 175: Operation Mode Specific Bits in pv Mode ... 61
Table 176: State Coding ... 62
Table 177: Object Description (6041$_H$ in pv Mode) ... 62
Table 178: Entry Description (6041$_H$ in pv Mode) ... 62
Table 179: Object Description (6062$_H$) ... 62
Table 180: Entry Description (6062$_H$) ... 62
Table 181: Object Description (6063$_H$) ... 63
Table 182: Entry Description (6063$_H$) ... 63
Table 183: Object Description (6064$_H$) ... 63
Table 184: Entry Description (6064$_H$) ... 63
Table 185: Object Description (606B$_H$) ... 63
Table 186: Entry Description (606B$_H$) ... 63
Table 187: Object Description (606C$_H$) ... 64
Table 188: Entry Description (606C$_H$) ... 64

Table 189: Object Description (607D$_H$) 64
Table 190: Entry Description (607D$_H$) 64
Table 191: Object Description (6083$_H$) 64
Table 192: Entry Description (6083$_H$) 65
Table 193: Object Description (6085$_H$) 65
Table 194: Entry Description (6085$_H$) 65
Table 195: Object Description (60FF$_H$) 65
Table 196: Entry Description (60FF$_H$) 65
Table 197: Supported CANopen Homing Methods.......... 68
Table 198: Homing Methods 17,18.......... 68
Table 199: Structure of the Control Word in hm Mode .. 72
Table 200: Operation Mode specific Bits in hm Mode.......... 72
Table 201: Command Coding 72
Table 202: Object Description (6040$_H$ in hm Mode) 73
Table 203: Entry Description (6040$_H$ in hm Mode).......... 73
Table 204: Structure of the Status Word in hm Mode.. 73
Table 205: HDLLC Specific Bits 73
Table 206: Operation Mode specific Bits in hm Mode 73
Table 207: State Coding 74
Table 208: Object Description (6041$_H$ in hm Mode) 74
Table 209: Entry Description (6041$_H$ in hm Mode).......... 74
Table 210: Object Description (606C$_H$) 74
Table 211: Entry Description (606C$_H$) 74
Table 212: Object Description (607C$_H$) 75
Table 213: Entry Description (607C$_H$) 75
Table 214: Object Description (6098$_H$) 75
Table 215: Entry Description (6098$_H$) 75
Table 216: Object Description (6099$_H$) 75
Table 217: Entry Description (6099$_H$) 76
Table 218: Object Description (609A$_H$).......... 76
Table 219: Entry Description (609A$_H$) 76
Table 220: Object Description (2100$_H$) 76
Table 221: Entry Description (2100$_H$) 76
Table 222: Structure of the Control Word in csp Mode78
Table 223: Command Coding 78
Table 224: Object Description (6040$_H$ in csp Mode) 79
Table 225: Entry Description (6040$_H$ in csp Mode).......... 79
Table 226: Structure of the Status Word in csp Mode.. 79
Table 227: Specific Bits 79
Table 228: Operation Mode specific Bits in csp Mode 79
Table 229: State Coding 80
Table 230: Object Description (6041$_H$ in csp Mode) 80
Table 231: Entry Description (6041$_H$ in csp Mode).......... 80
Table 232: Object Description (6062$_H$) 80
Table 233: Entry Description (6062$_H$) 80
Table 234: Object Description (6063$_H$) 81
Table 235: Entry Description (6063$_H$) 81
Table 236: Object Description (6064$_H$) 81
Table 237: Entry Description (6064$_H$) 81
Table 238: Object Description (606C$_H$) 81
Table 239: Entry Description (606C$_H$) 81
Table 240: Object Description (607A$_H$ in csp Mode) 82
Table 241: Entry Description (607A$_H$ in csp Mode) 82
Table 242: Object Description (607D$_H$).......... 82
Table 243: Entry Description (607D$_H$) 82
Table 244: Object Description (60B0$_H$) 83
Table 245: Entry Description (60B0$_H$) 83
Table 246: Object Description (60C2$_H$) 83
Table 247: Entry Description (60C2$_H$) 83
Table 248: Structure of the Control Word in csv Mode85
Table 249: Command Coding 85
Table 250: Object Description (6040$_H$ in csv Mode) 86
Table 251: Entry Description (6040$_H$ in csv Mode).......... 86
Table 252: Structure of the Status Word in csv Mode.. 86
Table 253: Specific Bits 86
Table 254: Operation Mode Specific Bits in csv Mode 86
Table 255: State Coding 87
Table 256: Object Description (6041$_H$ in csv Mode) 87
Table 257: Entry Description (6041$_H$ in csv Mode).......... 87
Table 258: Object Description (606C$_H$) 87
Table 259: Entry Description (606C$_H$) 87
Table 260: Object Description (60FF$_H$).......... 88
Table 261: Entry Description (60FF$_H$) 88
Table 262: Object Description (607D$_H$).......... 88
Table 263: Entry Description (607D$_H$) 88
Table 264: Object Description (60B1$_H$) 89
Table 265: Entry Description (60B1$_H$) 89
Table 266: Object Description (60C2$_H$) 89
Table 267: Entry Description (60C2$_H$) 89
Table 268: Structure of the Control Word in cst Mode91
Table 269: Command Coding 91
Table 270: Object Description (6040$_H$ in cst Mode).......... 92
Table 271: Entry Description (6040$_H$ in cst Mode) 92
Table 272: Structure of the Status Word in cst Mode.. 92
Table 273: HDLLC Specific Bits 92
Table 274: Operation Mode Specific Bits in cst Mode 92
Table 275: State Coding 93
Table 276: Object Description (6041$_H$ in cst Mode).......... 93
Table 277: Entry Description (6041$_H$ in cst Mode) 93
Table 278: Object Description (6071$_H$) 93
Table 279: Entry Description (6071$_H$).......... 93
Table 280: Object Description (6077$_H$) 94
Table 281: Entry Description (6077$_H$).......... 94
Table 282: Object Description (60B2$_H$) 94
Table 283: Entry Description (60B2$_H$) 94
Table 284: IO Summary.......... 95
Table 285: The electrical specification of IO ports 95
Table 286: Group1, RSF-5 and RSA-8 actuator with two IO ports.......... 96
Table 287: Digital Output mapping 2703$_H$ for RSF-5..... 96
Table 288: Digital Output mapping 2703$_H$ for RSA-8..... 96
Table 289: Group 2, FHA, LPA and SHA actuators with four IO ports.......... 97
Table 290: Digital Output mapping 2703$_H$ for FHA, LPA and SHA actuators.......... 97
Table 291: LED for NMT state 97
Table 292: Format of EMCY Messages.......... 99
Table 293: Emergency Messages (EMCY).......... 100
Table 294: SDO Abort Codes 102

17 Document Revision

Version	Date	Author	Description
03222020	Mar 20th, 2020	BPC/AIZ	Initial revised and updated.
08302022	Aug 30th, 2022	AZ	100C$_h$, 100D$_h$ are removed, no node guard. 200D$_h$ Status flag updated. 2020$_h$:04 Digital Input threshold. 2015$_h$ brake settings. 2025$_h$:05 Teach mode for homing. 270E$_h$:05 and 270E$_h$:06 for AIN1 AIN2 with PDO mapping. 2090$_h$, IIt1 IIt2 monitor. 2095$_h$ Velocity following window. 2096$_h$ position following window. 6060$_h$ mode of operation, torque mode (=4) removed. 6061$_h$ mode of operation display, torque mode (=4) removed. IO mapping updated. EMCY messages updated. SDO Abort codes.
12072022	Dec 7th,2022	AZ	Format of EMCY message added.
01092023	Jan 9th, 2023	AZ	200D$_h$ Status flag, Velocity/Position window, IIT1/IIT2 bit added. Section 8.1.10 on parameter saving after homing added.

18 Supplemental Directives

18.1 Producer Information

18.2 Copyright

Harmonic Drive LLC owns the content of this user manual in its entirety, including but not limited to pictures, logos, trademarks, and resources. © Copyright 2022 Harmonic Drive LLC. All rights reserved. Electronically published by Harmonic Drive LLC.

Redistributions of source or derived format (for example, Portable Document Format or Hypertext Markup Language) must retain the above copyright notice, and the complete Datasheet User Manual documentation of this product including associated Application Notes; and a reference to other available product-related documentation.

18.3 Trademark Designations and Symbols

Trademark designations and symbols used in this documentation indicate that a product or feature is owned and registered as trademark and/or patent either by Harmonic Drive LLC or by other manufacturers, whose products are used or referred to in combination with Harmonic Drive LLC's products and Harmonic Drive LLC's product documentation.

This CANopen Firmware Manual is a non-commercial publication that seeks to provide concise scientific and technical user information to the target user. Thus, trademark designations and symbols are only entered in the Short Spec of this document that introduces the product at a quick glance. The trademark designation /symbol is also entered when the product or feature name occurs for the first time in the document. All trademarks and brand names used are property of their respective owners.

18.4 Target User

The documentation provided here, is for programmers and engineers only, who are equipped with the necessary skills and have been trained to work with this type of product.

The Target User knows how to responsibly make use of this product without causing harm to himself or others, and without causing damage to systems, in which the user incorporates the product.

18.5 Disclaimer: Life Support Systems

Harmonic Drive LLC does not authorize or warrant any of its products for use in life support systems, without the specific written consent of Harmonic Drive LLC.

Life support systems are equipment intended to support or sustain life, and whose failure to perform, when properly used in accordance with instructions provided, can be reasonably expected to result in personal injury or death.

Information given in this document is believed to be accurate and reliable. However, no responsibility is assumed for the consequences of its use nor for any infringement of patents or other rights of third parties which may result from its use. Specifications are subject to change without notice.

18.6 Disclaimer: Intended Use

The data specified in this user manual is intended solely for the purpose of product description. No representations or warranties, either express or implied, of merchantability, fitness for a particular purpose or of any other nature are made hereunder with respect to information/specification or the products to which information refers and no guarantee with respect to compliance to the intended use is given.

In particular, this also applies to the stated possible applications or areas of applications of the product. Harmonic Drive LLC products are not designed for and must not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (safety-Critical Applications) without Harmonic Drive LLC’s specific written consent.

Harmonic Drive LLC products are not generally designed nor intended for use in military or aerospace applications or environments or in automotive applications unless specifically designated for such use by Harmonic Drive LLC. Harmonic Drive LLC conveys no patent, copyright, or other trademark right to this product. Harmonic Drive LLC assumes no liability for any patent and/or other trademark rights of a third party resulting from processing or handling of the product and/or any other use of the product.

18.7 Supplemental Documents & Tools

This product documentation is related and/or associated with additional tool kits, firmware and other items, as provided on the website at: www.harmonicdrive.net.

HarmonicDrive.net / 978-532-1800

Harmonic Drive LLC
Headquarters/Manufacturing
42 Dunham Ridge, Beverly, MA 01915

New York Office
100 Motor Parkway Suite 116, Hauppauge, NY 11788

Chicago Office
137 N. Oak Park Ave. Suite 410, Oak Park, IL 60301

California Office
333 W. San Carlos St. Suite 1070, San Jose, CA 95110

Group Companies
Harmonic Drive Systems, Inc.
6-25-3 Minami-Ohi, Shinagawa-ku
Tokyo 141-0013, Japan

Harmonic Drive SE
Hoenbergstrasse, 14, D-65555
Limburg/Lahn Germany