Software Manual

LMD CANopen

CiA 301 and CiA 402 Implementation

Publication LMD-CANOPEN-REV-E

02/2022

CANopen

c UL us LISTED CE RoHS REACH

Novanta IMS

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof.

Neither Novanta IMS nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

You agree not to reproduce, other than for your own personal, noncommercial use, all or part of this document on any medium whatsoever without permission of Novanta IMS, given in writing. You also agree not to establish any hypertext links to this document or its content.

Novanta IMS does not grant any right or license for the personal and noncommercial use of the document or its content, except for a non-exclusive license to consult it on an “as is” basis, at your own risk. All other rights are reserved.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed. Failure to use Novanta IMS software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

For information on the availability of products, go to https://novantaims.com/

The information contained in the present document is subject to change without notice. The technical characteristics of the devices described in the present document also appear online. The characteristics that are presented in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If there is a difference between the document and online information, use the online information as reference. All details provided are technical data which do not constitute warranted qualities.

Some of the product designations are registered trademarks of their respective owners, even if this is not explicitly indicated.

Table of Contents

Safety Information

Important Information

NOTICE Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label or message indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages and labels that follow this symbol to avoid possible injury or death.

⚠ DANGER
DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING
WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION
CAUTION indicates a hazardous situation which, if not avoided, **could result in** minor or moderate injury.

NOTICE
NOTICE is used to address practices not related to physical injury.

Qualification of Personnel

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Novanta IMS for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

Intended Use

This product is a motor with an integrated drive and is intended for industrial use according to this manual.

This product may only be used in compliance with all applicable safety regulations and directives, the specified requirements, and the technical data.

Prior to using this product, a risk assessment must be performed in view of the planned application. Based on the results, the appropriate safety measures must be implemented.

Since this product is used as a component in an entire system, the safety of persons must be ensured by means of the design of the entire system (e.g., machine design).

This product may only be operated with the specified cables and accessories. Use only genuine accessories and spare parts. This product must NEVER be operated in explosive atmospheres (e.g., hazardous locations, Ex areas).

Any use other than the use explicitly permitted is prohibited and can result in hazards.

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel.

Product Related Information

When the system is started, the drives are usually out of the operator's view and cannot be visually monitored.

⚠ DANGER

UNINTENDED CONSEQUENCES OF EQUIPMENT OPERATION

- Only start the system if there are no persons in the zone of operation.

Failure to follow these instructions will result in death or serious injury.

The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop, overtravel stop, power outage, and restart.

⚠ WARNING

LOSS OF CONTROL

- Separate or redundant control paths must be provided for critical functions.
- System control paths may include communication links. Consideration must be given to the implication of unanticipated transmission delays or failures of the link.

Observe all accident prevention regulations and local safety guidelines. [1]

- Each implementation of the product must be individually and thoroughly tested for proper operation before being placed into service.

Failure to follow these instructions can result in death or serious injury.

[1] For USA: Additional information, refer to NEMA ICS 1.1 (latest edition), "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control" and to NEMA ICS 7.1 (latest edition), "Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems".

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Novanta IMS for use with this equipment.
- Update the application program every time the physical hardware configuration is changed.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Terminology Derived from Standards

The technical terms, terminology, symbols, and the corresponding descriptions in this manual, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives, and general automation, this may include, but is not limited to, terms such as safety, safety function, safe state, fault, fault reset, malfunction, failure, error, error message, dangerous , etc.

Among others, these standards include:

EN 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2008	Safety of machinery: Safety-related parts of control systems. General principles for design.
EN 61496-1:2013	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction.
IEC/EN60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements.
EN 1088:2008 ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection.
ISO 13850:2006	Safety of machinery - Emergency stop - Principles for design.
IEC/EN 62061:2005	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems.
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2008	Digital data communication for measurement and control: Functional safety field buses.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term “zone of operation” may be used in conjunction with the description of specific hazards, and is defined as it is for a hazard zone or danger zone in the Machinery Directive (2006/42/EC) and ISO 12100:2010.

NOTE*:* The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.

About This Manual

Source Documents and Software

The information provided in this manual supplements, but is not a substitute for, the product hardware manual.

Source Manuals

The latest versions of the manuals can be downloaded from the Internet at: https://novantaims.com/dloads/

Source EDS Files

For easier engineering, Electronic Datasheet Files and product master data are available for download from the Internet at: https://novantaims.com/dloads/

Graphic User Interface Software

For easier configuration, a Graphic User Interface (GUI) is available for use with Liberty MDrive (LMD) and LMD products in conjunction with the optional MD-CC501-000 USB to CANopen interface cable kit. This software is available for download from the Internet at: https://novantaims.com/dloads/

CANopen Configurator

The software associated with LMD CANopen products is contained within the LMD Software Suite (LSS). This software package is available for download at the Novanta IMS web site at https://novantaims.com/dloads/

Applicable module:

CANopen Configuration Interface: Graphic User Interface (GUI) for commissioning the LMD.

See the *LMD Software Suite manual* for installation and use instructions.

Further Reading

Recommended literature for further reading.

CAN Users and Manufacturers Organization

CiA - CAN in Automation
Am Weichselgarten 26
D-91058 Erlangen

http://www.can-cia.org/

CANopen Standards

- CiA Standard 301 (CiA 301): CANopen application layer and communication profile
- CiA Standard 402 (CiA 402): Device profile for drives and motion control
- ISO 11898: Controller Area Network (CAN) for high speed communication
- EN 50325-4: Industrial communications subsystem based on ISO 11898 for controller device interfaces (CANopen)

Chapter 1
CANopen Protocol Basics

What's in this Chapter?

This chapter includes the following topics:

Introduction to CANopen Technology

CANopen Description Language

CANopen is a device- and manufacturer-independent description language for communication via the CAN bus. CANopen provides a common basis for interchanging commands and data between CAN bus devices.

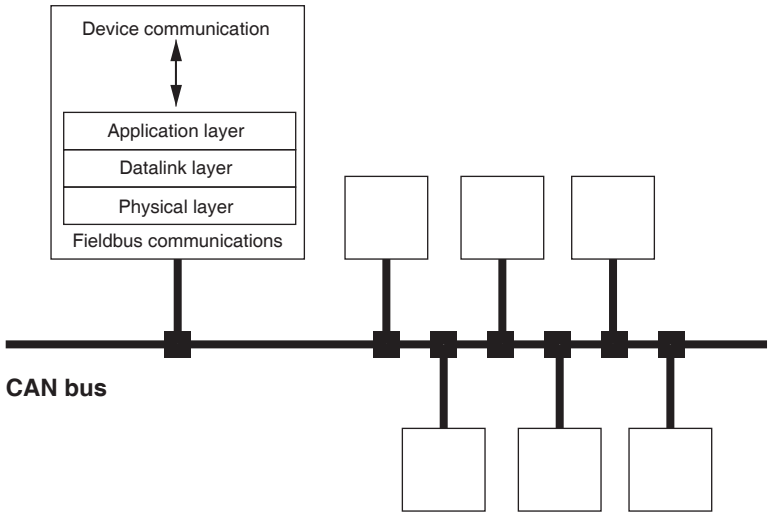
Communication Layers

CANopen uses the CAN bus technology for data communication.

CANopen is based on the basic network services for data communication as per the ISO-OSI model. 3 layers enable data communication via the CAN bus.

- Physical Layer
- Data Link Layer
- Application Layer

Figure 1.1 - CANopen Layer Model



Physical Layer

The physical layer defines the electrical properties of the CAN bus such as connectors, cable length and cable properties as well as bit coding and bit timing.

Data Link Layer

The data link layer connects the network devices. It assigns priorities to individual data packets and monitors and corrects errors.

Application Layer

The application layer uses communication objects (COB) to exchange data between the various devices. Communication objects are elementary components for creating a CANopen application.

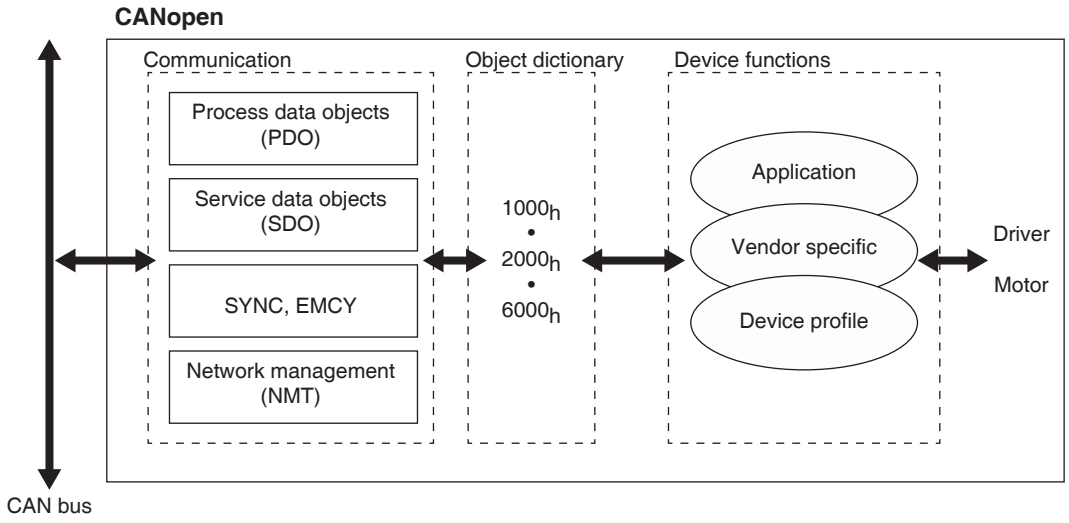
Objects

Processes under CANopen are executed via objects. Objects carry out different tasks; they act as communication objects for data transport to the fieldbus, control the process of establishing a connection, or monitor the network devices. If objects are directly linked to the device (device specific objects), the device functions can be used and changed via these objects.

Object Dictionary

The object dictionary of each network device allows for communication between the devices. Other devices find the objects with which they can communicate in this dictionary.

Figure 1.2 - Device Model with Object Dictionary



The object dictionary contains objects for describing the data types and executing the communication tasks and device functions under CANopen.

Object Index

Each object is addressed by means of a 16 bit index, which is represented as a four-digit hexadecimal number. The objects are arranged in groups in the object dictionary. The following table shows an overview of the object dictionary supported by LMD products as per the CANopen definition.

Table 1.1 - Object Directory Overview

Index range (hex)	Object group
1000_h - $1FFF_h$	Communications profile
2000_h - $5FFF_h$	Vendor specific objects
6000_h - $9FFF_h$	Standardized device profiles

For a list of all CANopen objects, see Section Appendix B, "Object Dictionary".

Communication Profile

CANopen manages communication between the network devices with object dictionaries and objects. A network device can use process data objects (PDO) and service data objects (SDO) to request the object data from the object dictionary of another device and, if permissible, write back modified values.

The following can be done by accessing the objects of the network devices:

- Exchange parameter values
- Start motion functions of individual CAN bus devices
- Request status information

Object Dictionary

Each CANopen device manages an object dictionary which contains the objects for communication.

Index, Subindex

The objects are addressed in the object dictionary via a 16 bit index. One or more 8 bit subindex entries for each object specify individual data fields in the object. Index and subindex are shown in hexadecimal notation with a subscript "$_h$".

Example

The following table shows index and subindex entries using the example of the object Homing Speeds (6098_h) for specifying the fast and slow speeds for homing functions.

Table 1.2 - Example Index and Subindex Entries

Index	Subindex	Name	Meaning
6098_h	00_h	—	Number of data fields
	01_h	Homing speed fast	High speed during homing
	02_h	Homing speed slow	Low speed during homing

Object Descriptions in the Manual

For CAN programming of a device, the objects of the following object groups are described in detail:

$1xxx_h$ objects: Communication objects.

$2xxx_h$ objects: Vendor-specific objects required to control the vendor specific functions of the device.

$6xxx_h$ objects: standardized objects of the device profile.

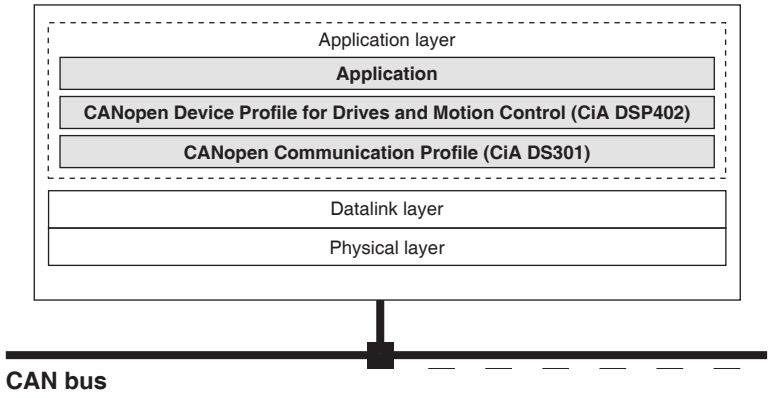
CANopen Profiles

Standardized Profiles

Standardized profiles describe objects that are used with different devices without additional configuration. The users and manufacturers organization CAN in Automation has standardized various profiles. These include:

- CiA 301 communication profile
- CiA 402 device profile

Figure 1.3 - CANopen Reference Model



CiA 301 Communication Profile

The CiA 301 communication profile is the interface between device profiles and CAN bus. It was specified in 1995 under the name CiA 301 and defines uniform standards for common data exchange between different device types under CANopen.

The objects of the communication profile in the device carry out the tasks of data exchange and parameter exchange with other network devices and initialize, control, and monitor the device in the network.

CiA 402 Device Profile

The CiA 402 device profile describes standardized objects for positioning, monitoring, and setting of drives. The tasks of the objects include:

- Device monitoring and status monitoring (Device Control)
- Standardized parameterization
- Changing, monitoring, and execution of operating modes

Vendor-specific Profiles

The basic functions of a device can be used with objects of standardized device profiles. Only vendor-specific device profiles offer the full range of functions. The objects with which the special functions of a device can be used under CANopen are defined in these vendor-specific device profiles.

Standardized Objects

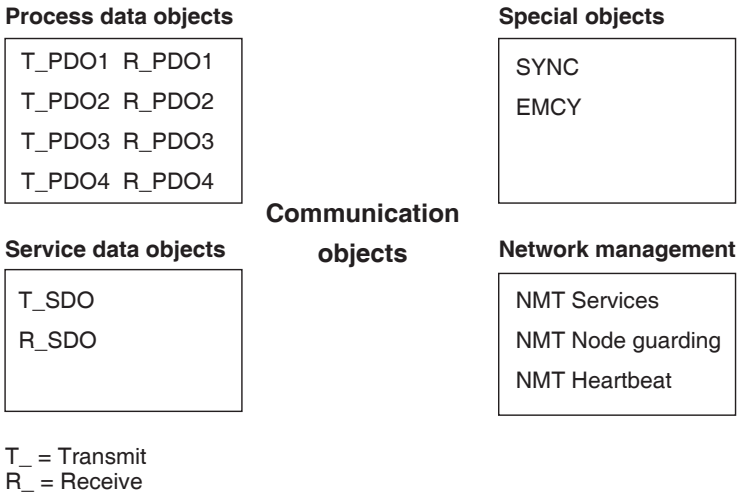
Standardized objects allow the use of the same application program for different network devices of the same device type. This requires these objects to be contained in the object dictionary of the network devices. Standardized objects are defined in the CiA 301 communication profile and the CiA 402 device profile.

Communication Objects

Overview

The communication objects are standardized with the CiA 301 CANopen communication profile. The objects can be classified into 4 groups according to their tasks.

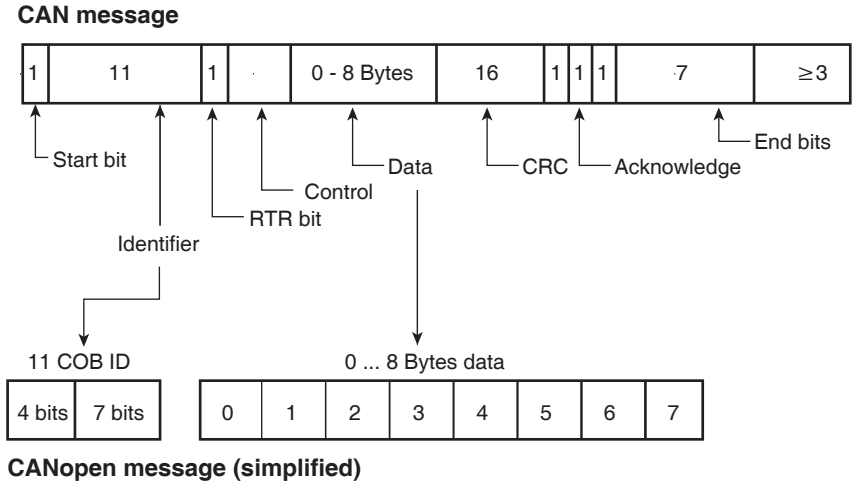
Figure 1.4 - Communication Objects



- PDOs (process data objects) for real-time transmission of process data
- SDOs (service data objects) for read and write access to the object dictionary
- Objects for controlling CAN messages:
    1. SYNC object (synchronization object) for synchronization of network devices
    2. EMCY object (emergency object) for signaling errors of a device or its peripherals.
- Network management services:
    1. NMT services for initialization and network control (NMT: network management)
    2. NMT Node Guarding for monitoring the network devices
    3. NMT Heartbeat for monitoring the network device's CAN Message

Data is exchanged via the CAN bus in the form of CAN messages. A CAN message transmits the communication object as well as numerous administration and control data.

Figure 1.5 - CAN Message and Simplified CANopen Message



CANopen Message

For work with CANopen objects and for data exchange, the CAN message can be represented in simplified form because most of the bits are used for error correction. These bits are automatically removed from the receive message by the data link layer of the OSI model, and added to a message before it is transmitted.

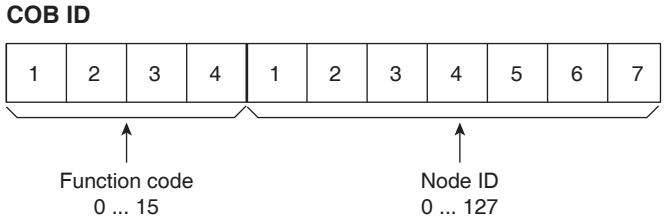
The two bit fields “Identifier” and “Data” form the simplified CANopen message. The “Identifier” corresponds to the “COB ID” and the “Data” field to the data frame (maximum length 8 bytes) of a CANopen message.

COB ID

The COB ID (**C**ommunication **OB**ject Identifier) has 2 tasks as far as controlling communication objects is concerned:

- Bus arbitration: Specification of transmission priorities
- Identification of communication objects: An 11 bit COB identifier as per the CAN 3.0A specification is defined for CAN communication; it is comprised of 2 parts:
  – Function code, 4 bits
  – Node address (node ID), 7 bits.

Figure 1.6 - COB ID with Function Code and Node Address



COB IDs of the Communication Objects

The following table shows the COB IDs of all communication objects with the factory settings. The column «Index of object parameters» shows the index of special objects with which the settings of the communication objects can be read or modified via an SDO.

Table 1.3 - COD IDs of Communication Objects

Communications Object	Function Code	Node Address, Node ID [1...127]	COB ID Decimal (hex)	Index of Object Parameters
NMT Start/Stop Service	0 0 0 0	0 0 0 0 0 0 0	0 (0_h)	—
SYNC object	0 0 0 1	0 0 0 0 0 0 0	128 (80_h)	1005_h....1007_h
EMCY object	0 0 0 1	x x x x x x x	128 (80_h) + node ID	1014_h, 1015_h
T_PDO1	0 0 1 1	x x x x x x x	384 (180_h) + node ID	1800_h
R_PDO1	0 1 0 0	x x x x x x x	512 (200_h) + node ID	1400_h
T_PDO2	0 1 0 1	x x x x x x x	640 (280_h) + node ID	1801_h
R_PDO2	0 1 1 0	x x x x x x x	768 (300_h) + node ID	1401_h
T_PDO3	0 1 1 1	x x x x x x x	896 (380_h) + node ID	1802_h
R_PDO3	1 0 0 0	x x x x x x x	1024 (400_h) + node ID	1402_h
T_PDO4	1 0 0 1	x x x x x x x	1152 (480_h) + node ID	1803_h
R_PDO4	1 0 1 0	x x x x x x x	1280 (500_h) + node ID	1403_h
R_SDO	—	x x x x x x x	1408 (580_h) + node ID	—

Communications Object	Function Code	Node Address, Node ID [1...127]	COB ID Decimal (hex)	Index of Object Parameters
R_SDO	—	x x x x x x x	1536 (600_h) + node ID	—
NMT error control	1 1 1 0	x x x x x x x	1792 (700_h) + node ID	—
LMT Services	1 1 1 1	1 1 0 0 1 0 x	2020 ($7E4_h$), 2021 ($7E5_h$)	—
NMT Identify Service	1 1 1 1	1 1 0 0 1 1 0	2022 ($7E6_h$)	—
NMT Services	1 1 1 1	1 1 0 1 0 0 x	2025 ($7E9_h$), 2026 ($7EA_h$)	—

Function Code

The function code classifies the communication objects. Since the bits of the function code in the COB ID are more significant, the function code also controls the transmission priorities: objects with a lower function code are transmitted with higher priority. For example, an object with function code “1” is transmitted prior to an object with function code “3” in the case of simultaneous bus access.

Node Address

Each network device is configured before it can be operated on the network. The device is assigned a unique 7 bit node address (node ID) between 1 (01_h) and 127 ($7F_h$). The device address “0” is reserved for “broadcast transmissions” which are used to send messages to all reachable devices simultaneously.

Example

Selection of a COB ID

For a device with the node address 5, the COB ID of the communication object T_PDO1 is:

384+node ID = 384 (180_h) + 5 = 389 (185_h).Data Frame

The data frame of the CANopen message can hold up to 8 bytes of data. In addition to the data frame for SDOs and PDOs, special frame types are specified in the CANopen profile:

- Error data frame
- Remote data frame for requesting a message

The data frames contain the respective communication objects.

Communication Relationships

CANopen uses 3 relationships for communication between network devices:

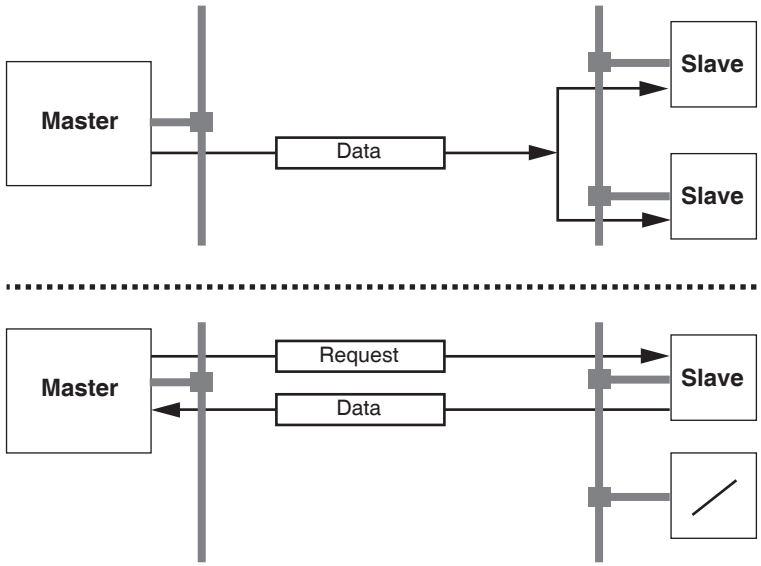
- Master-slave relationship
- Client-server relationship
- Producer-consumer relationship

Master-Slave Relationship

A network master controls the message traffic. A slave only responds when it is addressed by the master.

The master-slave relationship is used with network management objects for a controlled network start and to monitor the connection of devices.

Figure 1.7 - Master - Slave Relationship



Messages can be interchanged with and without confirmation. If the master sends an unconfirmed CAN message, it can be received by a single slave or by all reachable slaves or by no slave.

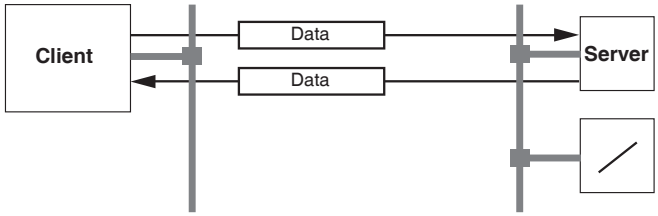
To confirm the message, the master requests a message from a specific slave, which then responds with the desired data.

Client-Server Relationship

A client-server relationship is established between 2 devices. The «server» is the device whose object dictionary is used during data exchange. The «client» addresses and starts the exchange of messages and waits for a confirmation from the server.

A client-server relationship with SDOs is used to send configuration data and long messages.

Figure 1.8 - Client - Server Relationship



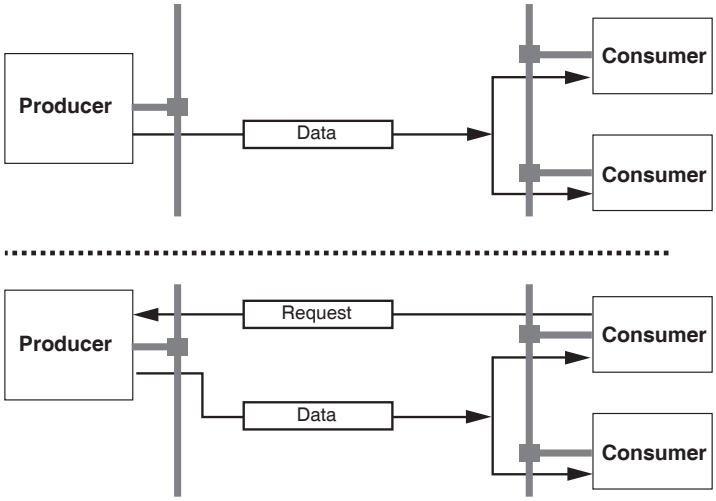
The client addresses and sends a CAN message to a server. The server evaluates the message and sends the response data as an acknowledgement.

Producer-Consumer Relationship

The producer-consumer relationship is used for exchanging messages with process data, because this relationship enables fast data exchange without administration data.

A “Producer” sends data, a “Consumer” receives data.

Figure 1.9 - Producer - Consumer Relationship



The producer sends a message that can be received by one or more network devices. The producer does not receive an acknowledgement to the effect that the message was received. The message transmission can be triggered by:

- An internal event, for example, “target position reached”
- The synchronization object SYNC
- A request of a consumer

See "Process Data Communication" on page 24 for details on the function of the producer-consumer relationship and on requesting messages.

Service Data Communication

Overview

SDOs can be used to access the entries of an object dictionary via index and subindex. The values of the objects can be read and, if permissible, also be changed.

Every network device has at least one server SDO to be able to respond to read and write requests from a different device. A client SDO is only required to request SDO messages from the object dictionary of a different device or to change them in the dictionary.

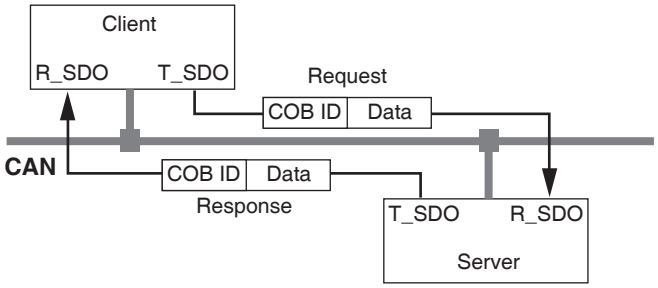
The T_SDO of an SDO client is used to send the request for data exchange; the R_SDO is used to receive. The data frame of an SDO consist of 8 bytes.

SDOs have a higher COB ID than PDOs; therefore, they are transmitted over the CAN bus at a lower priority.

SDO Data Exchange

An SDO transmits parameter data between 2 devices. The data exchange conforms to the client-server relationship. The server is the device to whose object dictionary an SDO message refers.

Figure 1.10 - SDO Message Exchange with Request and Response



Message Types

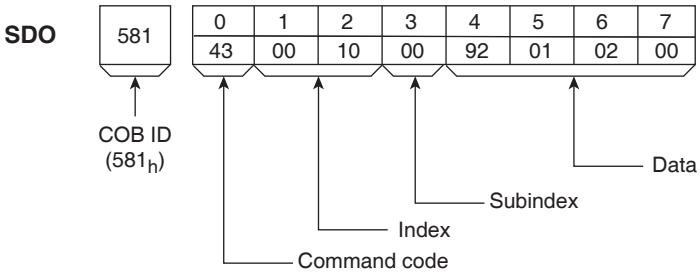
Client-server communication is triggered by the client to send parameter values to the server or to get them from the server. In both cases, the client starts the communication with a request and receives a response from the server.

SDO message

Put simply, an SDO message consists of the COB ID and the SDO data frame, in which up to 4 bytes of data can be sent. Longer data sequences are distributed over multiple SDO messages with a special protocol.

The device transmits SDOs with a data length of up to 4 bytes. Greater amounts of data such as 8 byte values of the data type «Visible String 8» can be distributed over multiple SDOs and are transmitted successively in blocks of 7 bytes.

Figure 1.11 - SDO Message Example



COB ID and data frame R_SDO and T_SDO have different COB IDs.

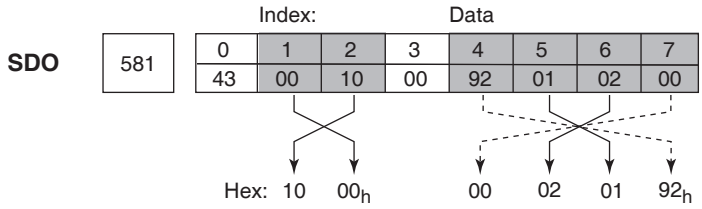
The data frame of an SDO messages consists of:

- Command code (ccd) which contains the SDO message type and the data length of the transmitted value
- Index and subindex which point to the object whose data is transported with the SDO message
- Data of up to 4 bytes

Evaluation of Numeric Values

Index and data are transmitted left-aligned in Intel, or little endian format. If the SDO contains numerical values of more than 1 byte in length, the data must be rearranged byte-by-byte before and after a transmission.

Figure 1.12 - Example of Rearranging Values > 1 Byte

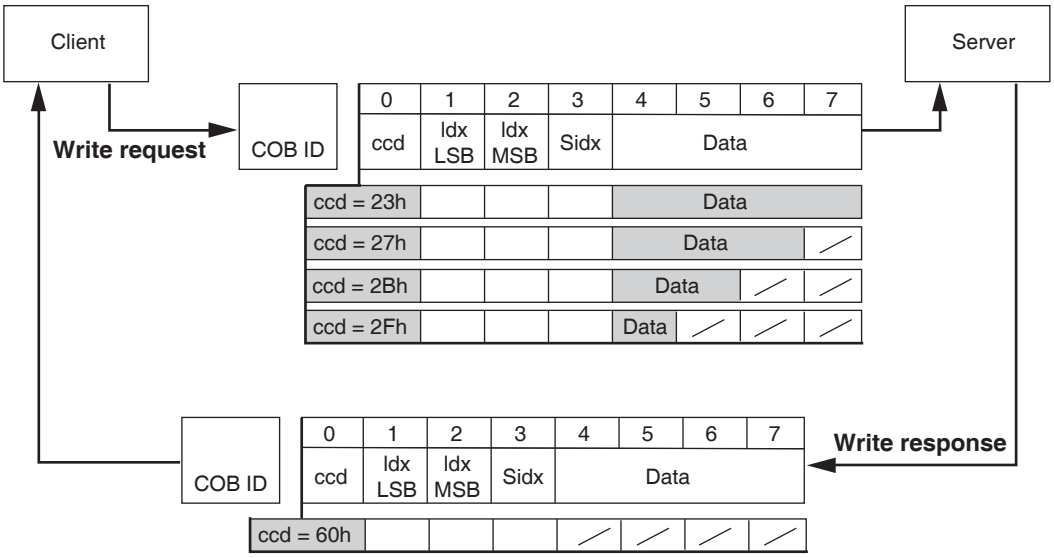


Reading and Writing Data

Writing Data

The client starts a write request by sending index, subindex, data length, and value. The server sends a confirmation indicating whether the data was correctly processed. The confirmation contains the same index and subindex, but no data.

Figure 1.13 - Writing Parameter Values



Unused bytes in the data field are shown with a slash in the graphic. The content of these data fields is not defined.

CCD Coding

The table below shows the command code for writing parameter values. It depends on the message type and the transmitted data length.

Table 1.4 - Command Code for Writing Parameter Values

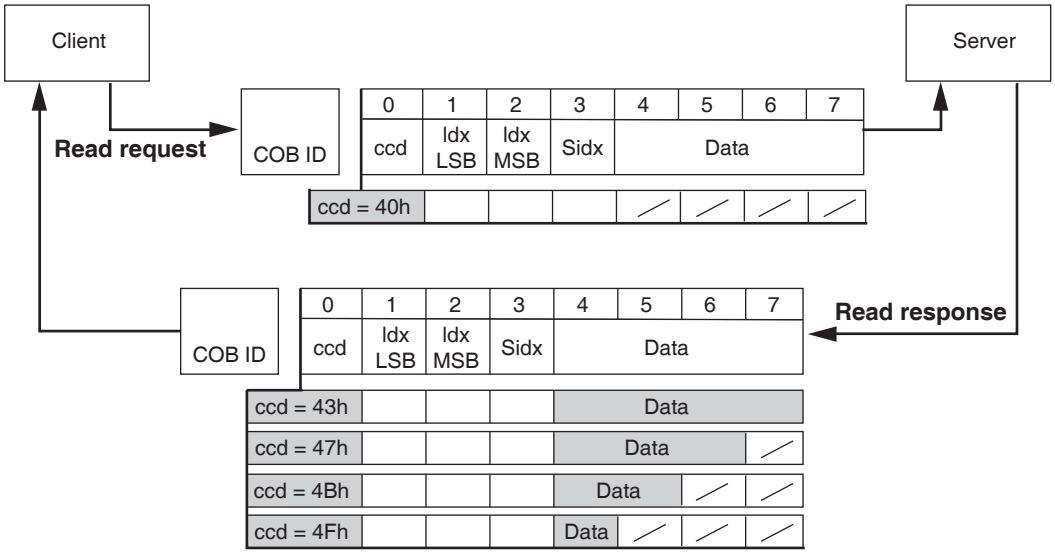
Message type	Data length used				
	4 byte	3 byte	2 byte	1 byte	
Write request	23_h	27_h	$2B_h$	$2F_h$	Transmitting parameters
Write response	60_h	60_h	60_h	60_h	Confirmation
Error response	80_h	80_h	80_h	80_h	Error

Reading Data

The client starts a read request by transmitting the index and subindex that point to the object or part of the object whose value it wants to read.

The server confirms the request by sending the desired data. The SDO response contains the same index and subindex. The length of the response data is specified in the command code «ccd».

Figure 1.14 - Reading a Parameter Value



The table below shows the command code transmitting a read value. It depends on the message type and the transmitted data length.

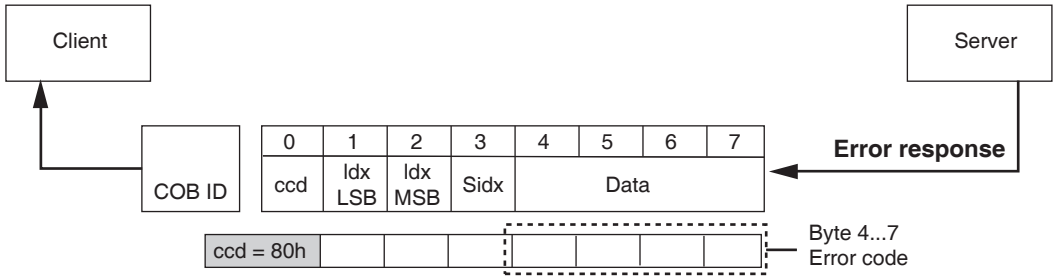
Table 1.5 - Command Code for Transmitting a Read Value

Message type	Data length used				
	4 byte	**3 byte**	**2 byte**	**1 byte**	
Read request	43_h	47_h	$4B_h$	$4F_h$	Request read value
Read response	40_h	40_h	40_h	40_h	Return read value
Error response	80_h	80_h	80_h	80_h	Error

Error Response

If a message could not be evaluated, the server sends an error message. See "Messages on Device Status" on page 92, for details on the evaluation of the error message.

Figure 1.15 - Response with Error Message (Error Response)



Reading data longer than 4 bytes

If values of more than 4 bytes are to be transmitted with an SDO message, the message must be divided into several frames. Each frame consists of 2 parts:

- Request by the SDO client,
- Confirmation by the SDO server.

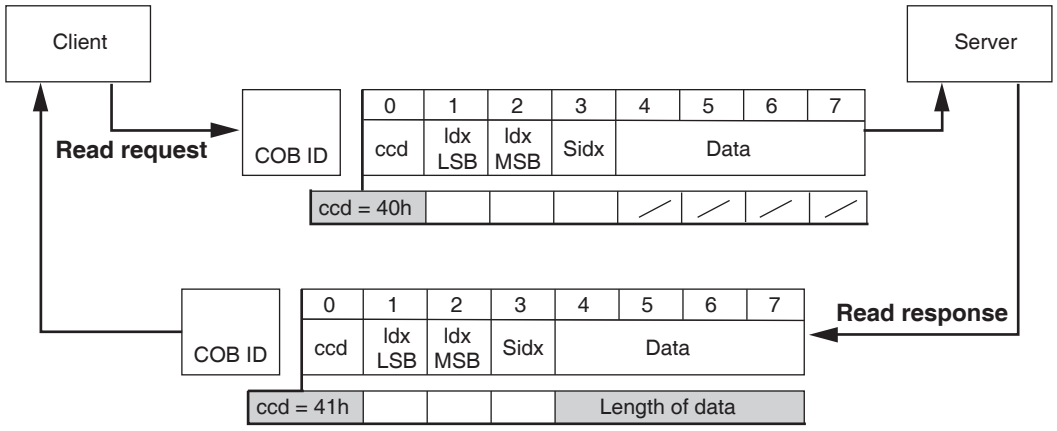
The request by the SDO client contains the command code «ccd» with the toggle bit and a data segment. The confirmation frame also contains a toggle bit in the «ccd» segment. In the first frame, the toggle bit has the value «0», in the subsequent frames it toggles between 1 and 0.

Reading Data

The client starts a read request by transmitting the index and subindex that point to the object or the object value whose value it wants to read.

The server confirms the request by transmitting index, subindex, data length, and the first 4 bytes of the requested data. The command code specifies that data of more than 4 bytes are transmitted. The command code of the read response from the server to the first message is 41_h.

Figure 1.16 - Transmitting the First Message



In the next frames, the remaining data is requested and transmitted in packets of 7 bytes from the server.

CCD Coding

The table below shows the command code for transmitting a read value. It depends on the message type, the value of the toggle bit, the transmitted data length, and the value of the bit that indicates the end of the entire SDO message.

Table 1.6 - Command Code Data Lengths > 4 Bytes

Message type	Data length used							
	7 byte	6 byte	5 byte	4 byte	3 byte	2 byte	1 byte	
Read request Toggle Bit = 0		60_h	60_h		60_h	60_h	60_h	Confirmation with Toggle Bit = 0
Read request Toggle Bit = 1	70_h	70_h	70_h	70_h	70_h	70_h	70_h	Confirmation with Toggle Bit = 1
Read response Toggle Bit = 0	00_h	—	—	—	—	—	—	Send parameter with Toggle Bit = 0
Read response Toggle Bit = 1	10_h	—	—	—	—	—	—	Send parameter with Toggle Bit = 1
Read response last message Toggle Bit = 0	01_h	03_h	05_h	07_h	09_h	$0B_h$	$0D_h$	Transmit parameter with last message and Toggle Bit = 0
Read response last message Toggle Bit = 1	11_h	13_h	15_h	17_h	19_h	$1B_h$	$1D_h$	Transmit parameter with last message and Toggle Bit = 1
Error response	80_h	80_h	80_h	80_h	80_h	80_h	80_h	Error

Refer to CiA 301 of the CiA for additional information on this procedure.

Process Data Communication

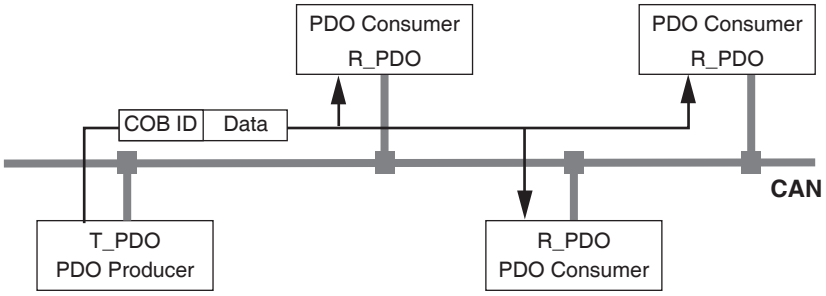
Overview

Process data objects (PDO) are used for real time data exchange of process data such as actual and reference values or the operating state of the device. Transmission is very fast because the data is sent without additional administration data and data transmission acknowledgement from the recipient is not required.

The flexible data length of a PDO message also increases the data throughput. A PDO message can transmit up to 8 bytes of data. If only 2 bytes are assigned, only 2 data bytes are sent. The length of a PDO message and the assignment of the data fields are specified by PDO mapping. See "PDO Mapping" on page 28, for additional information.

PDO messages can be exchanged between devices that generate or process data.

Figure 1.17 - PDO Data Exchange



Data exchange with PDOs follows to the producer-consumer relationship and can be triggered in 3 ways:

- Synchronized
- Event-driven
- Asynchronous

The SYNC object controls synchronized data processing. Synchronous PDO messages are transmitted immediately like the standard PDO messages, but are only evaluated on the next SYNC. For example, several drives can be started simultaneously via synchronized data exchange.

The device immediately evaluates PDO messages that are called on request or in an event-driven way.

The transmission type can be specified separately for each PDO with subindex 02_h (transmission type) of the PDO communication parameter. The objects are listed in Table 2.5.

PDO Message

T_PDO, R_PDO

One PDO each is available for sending and receiving a PDO message:

- T_PDO to transmit the PDO message (T: Transmit),
- R_PDO to receive PDO messages (R: Receive).

The following settings for PDOs correspond to the defaults read and set via objects of the communication profile.

The device uses 6 PDOs, 3 receive PDOs and 3 transmit PDOs. By default, the PDOs are evaluated or transmitted in an event-driven way.

PDO settings

The PDO settings can be read and changed with 8 communication objects.

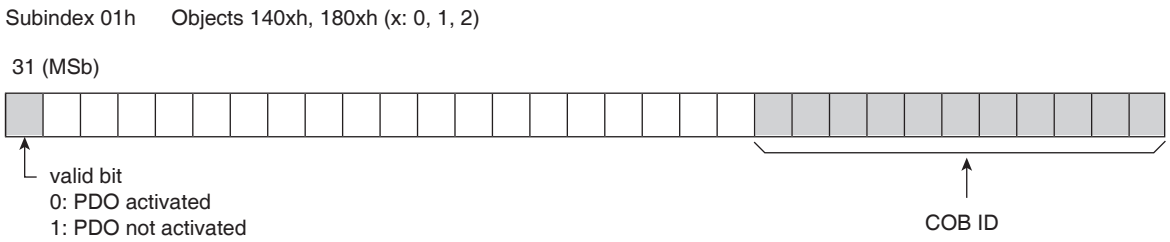
Table 1.7 - PDO Communication Objects

Object	Meaning
1st receive PDO parameter (1400_h)	Settings for R_PDO1
2nd receive PDO parameter (1401_h)	Settings for R_PDO2
3rd receive PDO parameter (1402_h)	Settings for R_PDO3
4th receive PDO parameter (1403_h)	Settings for R_PDO4
1st transmit PDO parameter (1800_h)	Settings for T_PDO1
2nd transmit PDO parameter (1801_h)	Settings for T_PDO2
3rd transmit PDO parameter (1802_h)	Settings for T_PDO3
4th transmit PDO parameter (1803_h)	Settings for T_PDO4

Activating PDOs

With the default PDO settings, R_PDO1 and T_PDO1 are activated. The other PDOs must be activated first. A PDO is activated with bit 31 (valid bit) in subindex 01_h of the respective communication object:

Figure 1.18 - Activating PDOs via Subindex 01_h, Bit 31



Example

Setting for R_PDO3 in object 1402_h

- Subindex 01_h = 8000 $04xx_h$: R_PDO3 not activated
- Subindex 01_h = 0000 $04xx_h$: R_PDO3 activated.

Values for “x” in the example depend on the COB ID setting.

PDO Time Intervals

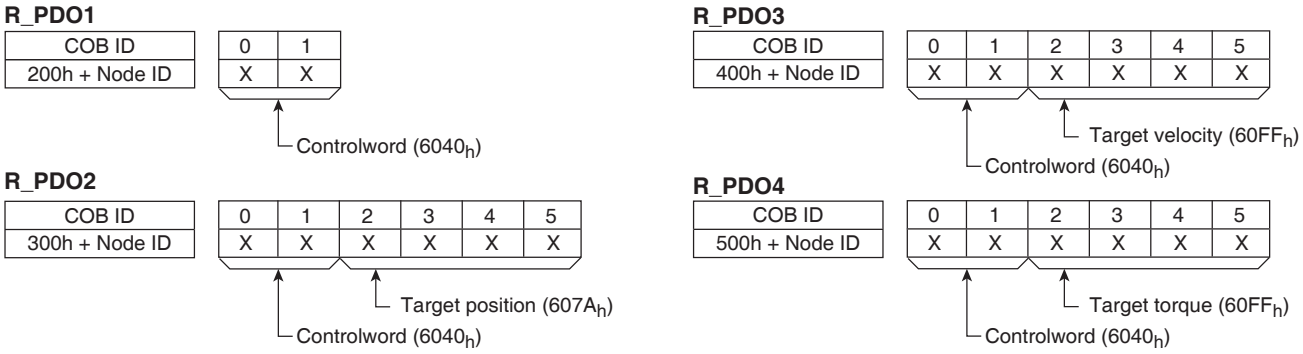
The time intervals «inhibit time» and «event timer» can be set for each transmit PDO.

- The time interval «inhibit time» can be used to reduce the CAN bus load, which can be the result of continuous transmission of T_PDOs. If an inhibit time not equal to zero is entered, a transmitted PDO will only be re-transmitted after the inhibit time has elapsed. The time is set with subindex 03_h.
- The time interval «event timer» cyclically triggers an event message. After the time interval has elapsed, the device transmits the event controlled T_PDO. The time is set with subindex 05_h.

Receive PDOs

The objects for R_PDO1, R_PDO2, R_PDO3 and R_PDO4 are preset. They may be re-mapped to suit the end-user application.

Figure 1.19 - Receive PDOs



R_PDO1

R_PDO1 contains the control word, object **`controlword (6040h)`**, of the state machine which can be used to set the operating state of the device.

R_PDO1 is evaluated asynchronously (i.e., it is event-driven). R_PDO1 is preset.

R_PDO2

With R_PDO2, the control word and the target position of a motion command, object **`target position (607Ah)`**, are received for a movement in the operating mode “Profile Position”.

R_PDO2 is evaluated asynchronously (i.e., it is event-driven). R_PDO2 is preset.

For details on the SYNC object, see "Synchronization" on page 31.

R_PDO3

R_PDO3 contains the control word and the target velocity, object **`Target velocity (60FFh)`**, for the operating mode “Profile Velocity”.

R_PDO3 is evaluated asynchronously (i.e., it is event-driven). R_PDO3 is preset.

R_PDO4

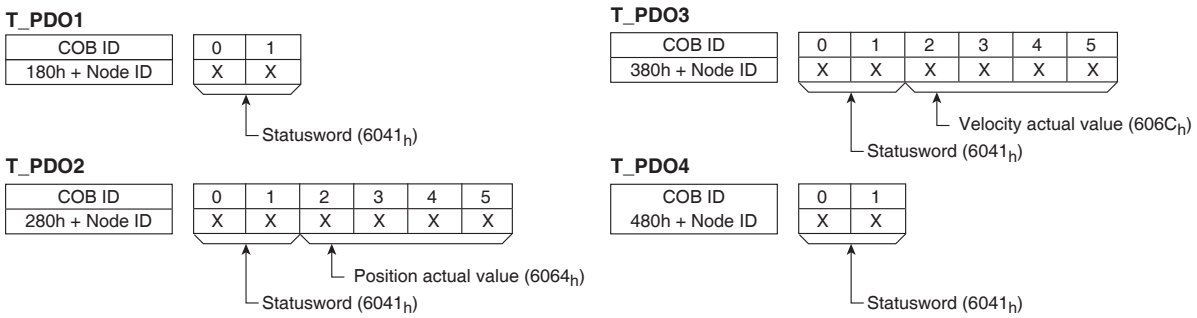
R_PDO4 contains the control word and the target torque, object **`Target torque (6071h)`**, for the operating mode “Profile Torque” (closed loop LMD products only).

R_PDO4 is evaluated asynchronously (i.e., it is event-driven). R_PDO4 is preset. Transmit PDOs

Transmit PDOs

The objects for T_PDO1, T_PDO2 and T_PDO3 can be changed by means of PDO mapping.

Figure 1.20 - Transmit PDOs



T_PDO1

T_PDO1 contains the status word, object **`statusword (6041h)`**, of the state machine.

T_PDO1 is transmitted asynchronously and in an event-driven way whenever the status information changes.

T_PDO2

T_PDO2 contains the status word and the actual position of the motor, object **`Position actual value (6064h)`**, to monitor movements in the operating mode “Profile Position”.

T_PDO2 is transmitted after receipt of a SYNC object and in an event driven way.

T_PDO3

T_PDO3 contains the status word and the actual velocity, object **`Velocity actual value (606Ch)`**, for monitoring the velocity profile in the operating mode “Profile Velocity”.

T_PDO3 is transmitted asynchronously and in an event-driven way whenever the status information changes.

T_PDO4

T_PDO4 contains the status word, object **`statusword (6041h)`**, of the state machine.

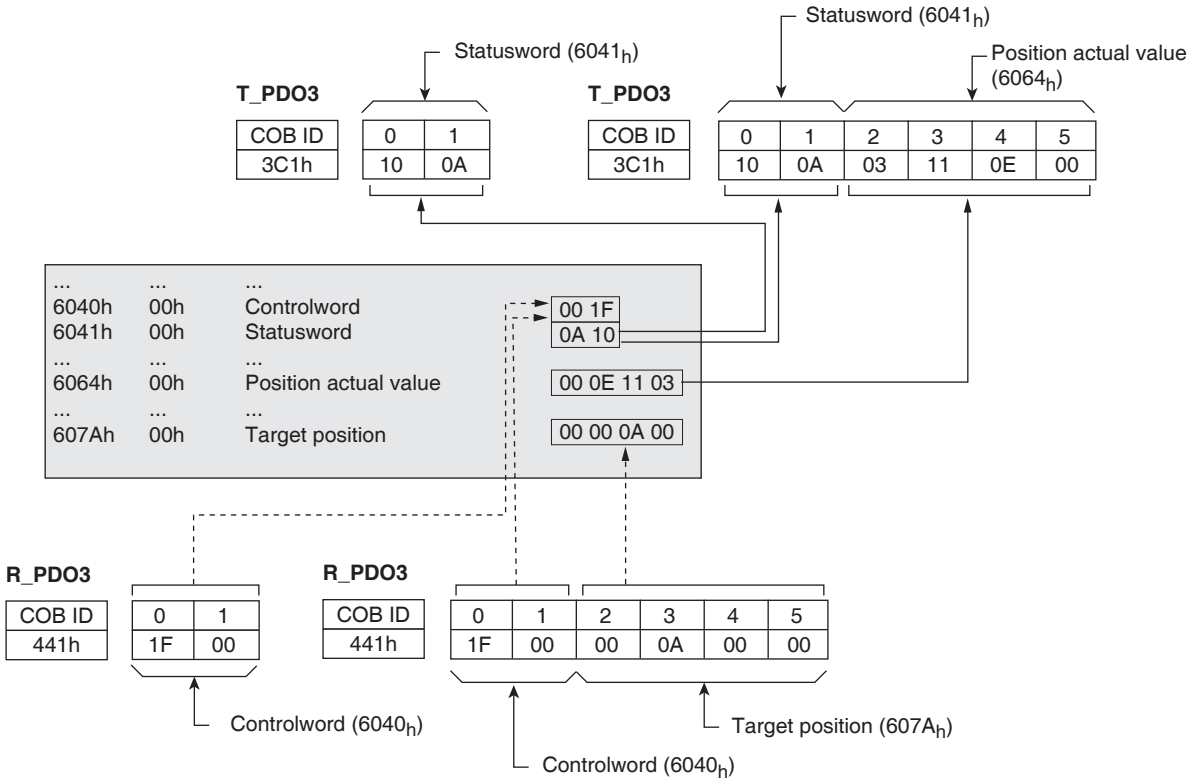
T_PDO4 is transmitted asynchronously and in an event-driven way whenever the status information changes.

PDO Mapping

Up to 8 bytes of data from different areas of the object directory can be transmitted with a PDO message. Mapping of data to a PDO message is referred to as PDO mapping.

The following diagram shows the data exchange between PDOs and the object directory on the basis of two examples of objects in T_PDO3 and R_PDO3 of the PDOs.

Figure 1.21 - PDO Mapping



Dynamic PDO Mapping

The device uses dynamic PDO mapping. Dynamic PDO mapping means that objects can be mapped to the corresponding PDO using adjustable settings.

The settings for PDO mapping are defined in an assigned communication object for each PDO.

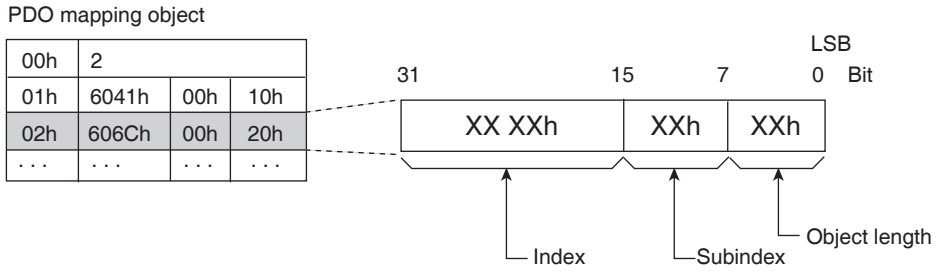
Table 1.8 - Dynamic PDO Mapping Parameters

Object	PDO mapping for:	Type
1st receive PDO mapping (1600_h)	R_PDO1	Dynamic
2nd receive PDO mapping (1601_h)	R_PDO2	Dynamic
3rd receive PDO mapping (1602_h)	R_PDO3	Dynamic
4th receive PDO mapping (1603_h)	R_PDO4	Dynamic
1st transmit PDO mapping ($1A00_h$)	T_PDO1	Dynamic
2nd transmit PDO mapping ($1A01_h$)	T_PDO2	Dynamic
3rd transmit PDO mapping ($1A02_h$)	T_PDO3	Dynamic
4th transmit PDO mapping ($1A03_h$)	T_PDO4	Dynamic

Structure of the Entries

Up to 8 bytes of 8 different objects can be mapped in a PDO. Each communication object for setting the PDO mapping provides 4 subindex entries. A subindex entry contains 3 pieces of information on the object: the index, the subindex, and the number of bits that the object uses in the PDO.

Figure 1.22 - Structure of PDO Mapping Entries



Subindex 00_h of the communication object contains the number of valid subindex entries.

Table 1.9 - Subindex Object Length Entries

Object length	Bit value
08_h	8 bits
10_h	16 bits
20_h	30 bits

PDO Mapping Objects

Table 1.10 - Supported PDO Mapping Entries

Index	Sub-index	Object	PDO	Data type	Parameter name
2009_h	1	Read value of output error	T_PDO	UINT8	**Out_Error**
2010_h	1	Read value of analog input	T_PDO	UINT16	**Analog_In_Reading**
2014_h	1	Read Aux voltage level	T_PDO	UINT16	**AuxPower_Voltage**
2015_h		Read +V level	T_PDO	UINT16	**Vin_Voltage**
2018_h	1	Read internal temperature	T_PDO	INT8	**Temperature_Reading**
2019_h	1	Read output h-bridge temperature	T_PDO	UINT16	**TempC_Bridge_Reading**
2033_h	4	Read captured position	T_PDO	INT32	**CaptureInPositn_user**
2741_h	0	Read hybrid status byte	T_PDO	UINT8	**HybridStatusByte**
$603F_h$	0	Read error code	T_PDO	UINT16	**ErrorCode**
6040_h	0	Controlword	R_PDO	UINT16	**Controlword**
6041_h	0	Statusword	T_PDO	UINT16	**Statusword**
6060_h	0	Modes of operation	R_PDO	INT8	**Modes_of_operation**
6061_h	0	Read mode of operation	T_PDO	INT8	**Modes_of_operation_display**
6062_h	0	Read position value	T_PDO	INT32	**Position_demand_value_user**

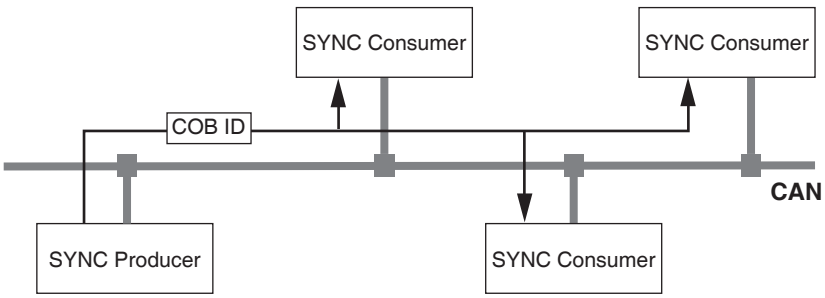
Index	Sub-index	Object	PDO	Data type	Parameter name
6063_h	0	Position actual value	T_PDO	INT32	**Position_actual_value_inc**
6064_h	0	Position actual value	T_PDO	INT32	**Position_actual_value**
$606C_h$	0	Velocity actual value	T_PDO	INT32	**Velocity_actual_value**
6071_h	0	Target torque	R_PDO	INT16	**Target_torque**
$607A_h$	0	Target position	R_PDO	INT32	**Target_position**
$607E_h$	0	Polarity	R_PDO	UINT8	**Polarity**
6081_h	0	Profile velocity	R_PDO	UINT32	**Profile_velocity**
6082_h	0	Initial velocity	R_PDO	UINT32	**Initial_velocity**
6083_h	0	Profile acceleration	R_PDO	UINT32	**Profile_acceleration**
6084_h	0	Profile deceleration	R_PDO	UINT32	**Profile_deceleration**
6086_h	0	Motion profile type	R_PDO	INT16	**Motion_profile_type**
$60FD_h$	0	Digital inputs	T_PDO	UINT32	**Digital_inputs**
$60FE_h$	1	Digital outputs	R_PDO	UINT32	**Digital_outputs**
$60FF_h$	0	Target velocity	R_PDO	INT32	**Target_velocity**

Synchronization

The synchronization object SYNC controls the synchronous exchange of messages between network devices for purposes such as the simultaneous start of multiple drives.

The data exchange conforms to the producer-consumer relationship. The SYNC object is transmitted to all reachable devices by a network device and can be evaluated by the devices that support synchronous PDOs.

Figure 1.23 - SYNC Message

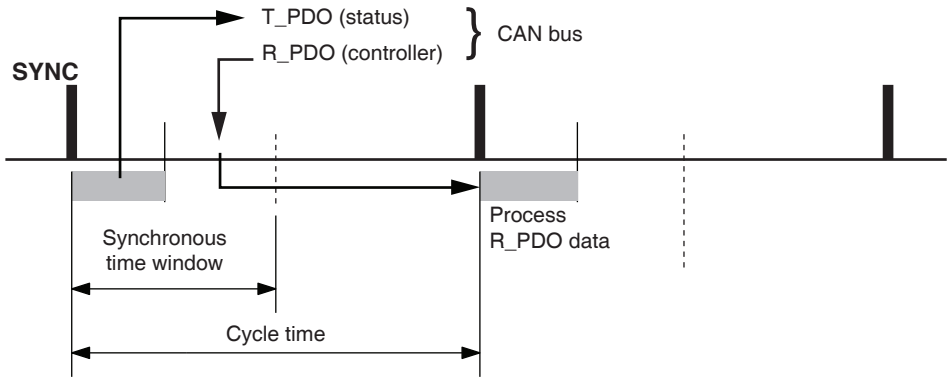


Time Values for Synchronization

Two time values define the behavior of synchronous data transmission:

- The cycle time specifies the time intervals between 2 SYNC messages. It is set with the object **Communication cycle period(1006h).**
- The synchronous time window specifies the time span during which the synchronous PDO messages must be received and transmitted. The time window is defined with the object **Synchronous window length (1007h).**

Figure 1.24 - Synchronization Times



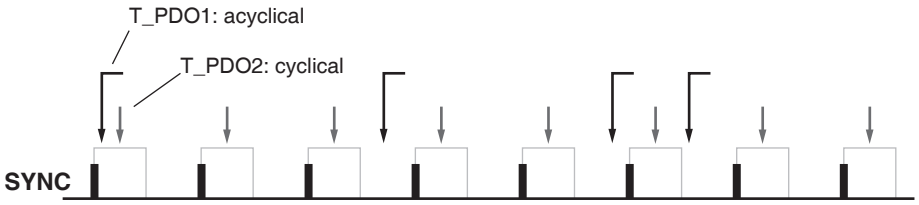
Synchronous Data Transmission

From the perspective of a SYNC recipient, in one time window the status data is transmitted first in a T_PDO, then new control data is received via an R_PDO. The control data is only processed when the next SYNC message is received. The SYNC object itself does not transmit data.

Cyclic and Acyclic Data Transmission

Synchronous exchange of messages can be cyclic or acyclic.

Figure 1.25 - Cyclic and Acyclic Transmission



In the case of cyclic transmission, PDO messages are exchanged continuously in a specified cycle, for example with each SYNC message.

If a synchronous PDO message is transmitted acyclically, it can be transmitted or received at any time; however, it will not be valid until the next SYNC message.

Cyclic or acyclic behavior of a PDO is specified in the subindex transmission type (02_h) of the corresponding PDO parameter, for example, in the object 1st receive PDO parameter (1400_h:02_h) for R_PDO1.

COB ID, SYNC Object

For fast transmission, the SYNC object is transmitted unconfirmed and with high priority.

The COB ID of the SYNC object is set to the value 128 (80_h) by default. The value can be changed after initialization of the network with the object COB-ID SYNC Message (1005_h).

“Start” PDO

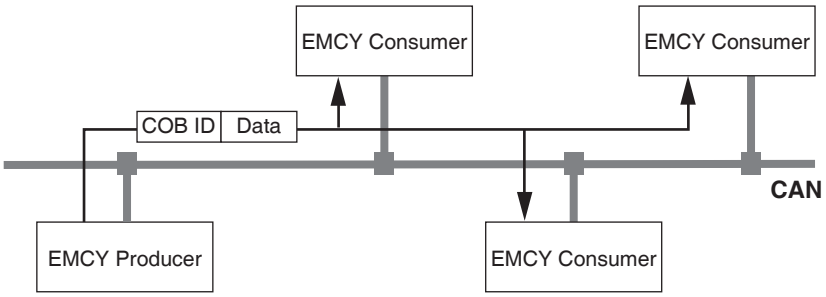
With the default settings of the PDOs, R_PDO1 ... R_PDO4 and T_PDO1 ... T_PDO4 are received and transmitted asynchronously. T_PDO2 ... T_PDO3 are transmitted additionally after the event timer has elapsed. The synchronization allows an operating mode to be

started simultaneously on multiple devices so that, for example, the feed of a portal drive with several motors can be synchronized.

Emergency Service

The Emergency Service signals internal device errors via the CAN bus. The error message is transmitted to the network devices with an Emergency Object (EMCY) according to the consumer-producer relationship.

Figure 1.26 - Error Message via EMCY Objects



Boot-up Message

The communication profile CiA 301, version 3.0, defines an additional task for the EMCY object: sending a boot-up message. A boot-up message informs the network devices that the device that transmitted the message is ready for operation in the CAN network.

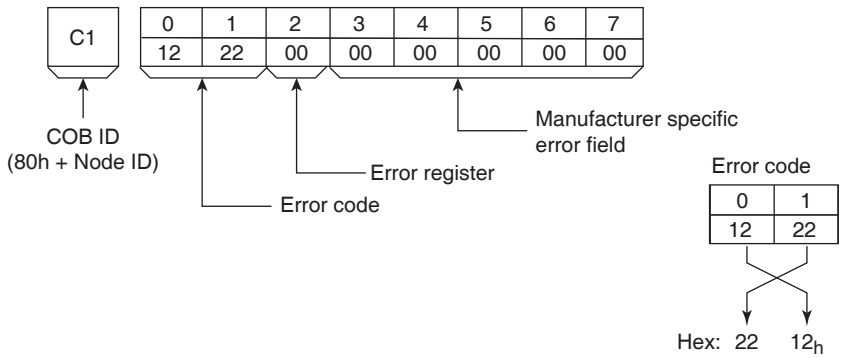
The boot-up message is transmitted with the COB ID 700_h + node ID and one data byte (00_h).

Error Evaluation and Handling

EMCY Message

If an internal device error occurs, the device switches to the operating state 9 Fault as per the CANopen state machine. At the same time, it transmits an EMCY message with error register and error code.

Figure 1.27 - EMCY Message



Bytes 0, 1 - Error code, value is also saved in the object **`Error code (603Fh)`**

Byte 2 - Error register, value is also saved in the object **`Error register (1001h)`**

Bytes 3, 4 - Reserved

Byte 5 - PDO: Number of the PDO

Bytes 6, 7 - Vendor-specific error code

COB ID

The COB ID for each device on the network supporting an EMCY object is determined on the basis of the node address:

COB ID = Function code EMCY object (80_h) + node ID

The function code of the COB ID can be changed with the object **COBID emergency(1014h).**

Error Register and Error Code

The error register contains bit-coded information on the error. Bit 0 remains set as long as an error is active. The remaining bits identify the error type. The exact cause of error can be determined on the basis of the error code. The error code is transmitted in Intel format as a 2 byte value; the bytes must be reversed for evaluation.

See Section Appendix A "Diagnostics and Troubleshooting" on page 91 for a list of the error messages and error responses by the device as well as remedies.

Error Memory

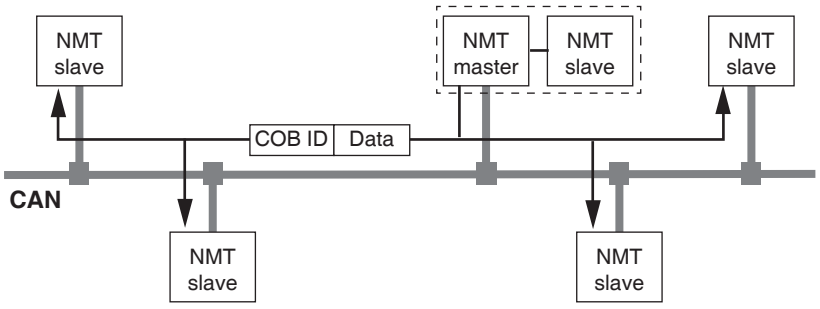
The device saves the error register in the object **Error register (1001h)** and the last error that occurred in the object **Error code (603Fh)**.

Network Management Services

Network management (NMT) is part of the CANopen communication profile; it is used to initialize the network and the network devices and to start, stop, and monitor the network devices during operation on the network.

NMT services are executed in a master-slave relationship. The NMT master addresses individual NMT slaves via their node address. A message with node address «0» is broadcast to all reachable NMT slaves simultaneously.

Figure 1.28 - NMT Services via Master - Slave Relationship



The LMD and LMD devices can only take on the function of an NMT slave.

NMT Services

NMT services can be divided into 2 groups:

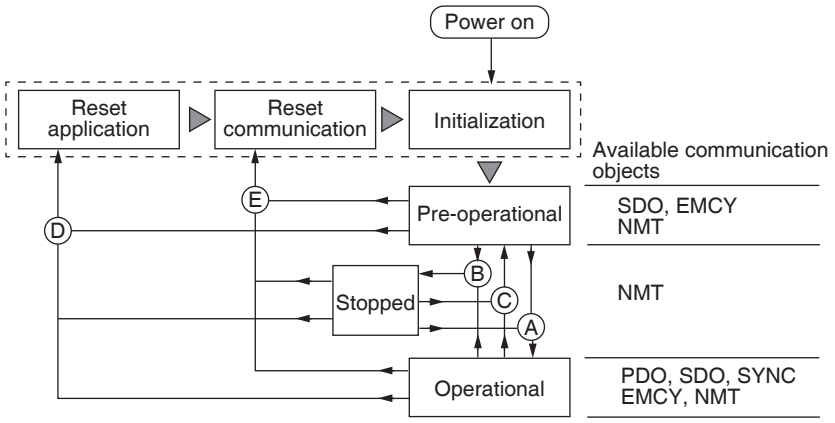
- Services for device control, to initialize devices for CANopen communication and to control the behavior of devices during operation on the network
- Services: for connection monitoring

NMT Services for Device Control

NMT State Machine

The NMT state machine describes the initialization and states of an NMT slave during operation on the network.

Figure 1.29 - NMT State Machine and Available Communication Objects



In the figure above, the communication objects available for use in the specific network state are shown on the right.

Initialization

An NMT slave automatically runs through an initialization phase after the supply voltage is switched on (power on) to prepare it for CAN bus operation. On completion of the initialization, the slave switches to the operating state “Pre Operational” and sends a boot-up message. From now on, an NMT master can control the operational behavior of an NMT slave on the network via 5 NMT services, represented in the figure above by the letters A to E.

Table 1.11 - NMT State Machine Transitions

NMT service	Transition	Meaning
Start remote node (Start network node)	A	Transition to operating state “Operational” Start normal operation on the network
Stop remote node (Stop network node)	B	Transition to operating state “Stopped” Stops communication of the network device on the network. If connection monitoring is active, it remains on. If the power stage is enabled (operating state “Operation Enabled” or “Quick Stop”), an error of error class 2 is triggered. The motor is stopped and the power stage disabled.

NMT service	Transition	Meaning
Enter Pre-Operational (Transition to “Pre-Operational”)	C	Transition to operating state “Pre-Operational” The communication objects except for PDOs can be used. The operating state “Pre-Operational” can be used for configuration via SDOs: - PDO mapping - Start of synchronization - Start of connection monitoring
Reset node (Reset node)	D	Transition to operating state “Reset application” Load stored data of the device profiles and automatically switch via operating state “Reset communication” to “Pre-Operational”.
Reset communication (Reset communication data)	E	Transition to operating state “Reset communication” Load stored data of the communication profile and automatically transition to operating state “Pre-Operational”. If the power stage is enabled (operating state “Operation Enabled” or “Quick Stop”), an error of error class 2 is triggered. The motor is stopped and the power stage disabled.

Persistent Data Memory

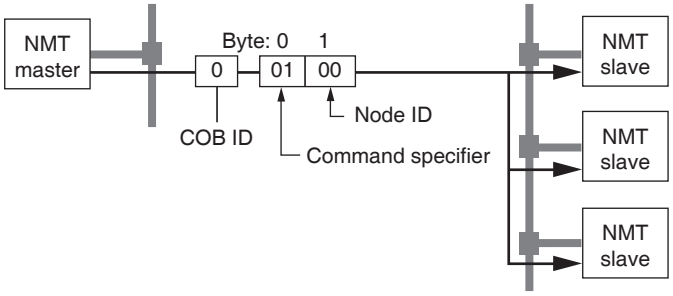
When the supply voltage is switched on (power on), the device loads the saved object data from the non-volatile Electrically Erasable Programmable Read Only Memory (EEPROM) for persistent data to the RAM.

NMT Message

The NMT services for device control are transmitted as unconfirmed messages with the COB ID = 0. By default, they have the highest priority on the CAN bus.

The data frame of the NMT device service consists of 2 bytes.

Figure 1.30 - NMT Message



The first byte, the “Command specifier”, indicates the NMT service used.

Table 1.12 - NMT Command Specifiers.

Command specifier	NMT service	Transition
1 (01_h)	Start remote node	A
2 (02_h)	Stop remote node	B
128 (80_h)	Enter pre-operational	C
129 (81_h)	Reset node	D
130 (82_h)	Reset communication	E

The second byte addresses the recipient of an NMT message with a node address between 1 and 127 ($7F_h$). A message with node address “0” is broadcast to all reachable NMT slaves.

NMT Services for Connection Monitoring

Connection monitoring checks the communication status of network devices.
3 NMT services for connection monitoring are available:

- «Node guarding» for monitoring the connection of an NMT slave
- «Life guarding» for monitoring the connection of an NMT master
- «Heartbeat» for unconfirmed connection messages from network devices.

Node Guarding / Life Guarding

COB ID

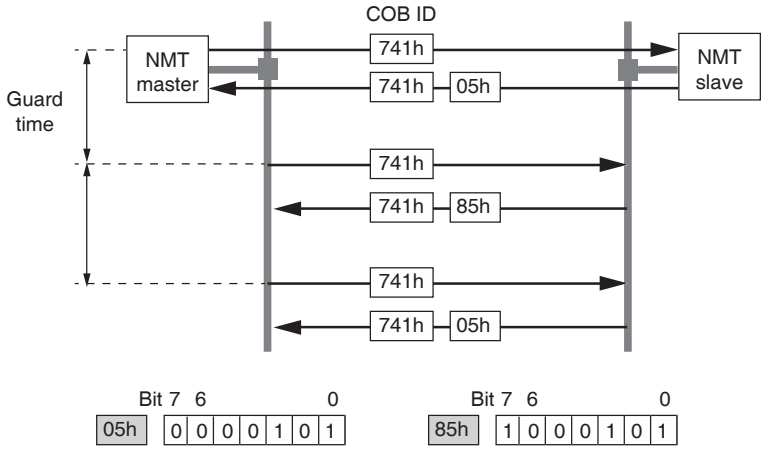
The communication object **`NMT error control (700h+node-Id)`** is used for connection monitoring. The COB ID for each NMT slave is determined on the basis of the node address:

COB ID = function code **`NMT error control (700h) + node-Id`**.

Structure of the NMT Message

After a request from the NMT master, the NMT slave responds with one data byte.

Figure 1.31 - Acknowledgement of the NMT Slave



Bits 0 to 6 identify the NMT state of the slave:
- 4 (04_h): “Stopped”
- 5 (05_h): “Operational”
- 127 ($7F_h$): “Pre-Operational”

After each “guard time” interval, bit 7 switches toggles between “0” and “1”, so the NMT master can detect and ignore a second response within the “guard time” interval. The first request when connection monitoring is started begins with bit 7 = 0.

Connection monitoring must not be active during the initialization phase of a device. The status of bit 7 is reset as soon as the device runs though the NMT state “Reset communication”.

Connection monitoring remains active in the NMT state “Stopped”.

Configuration

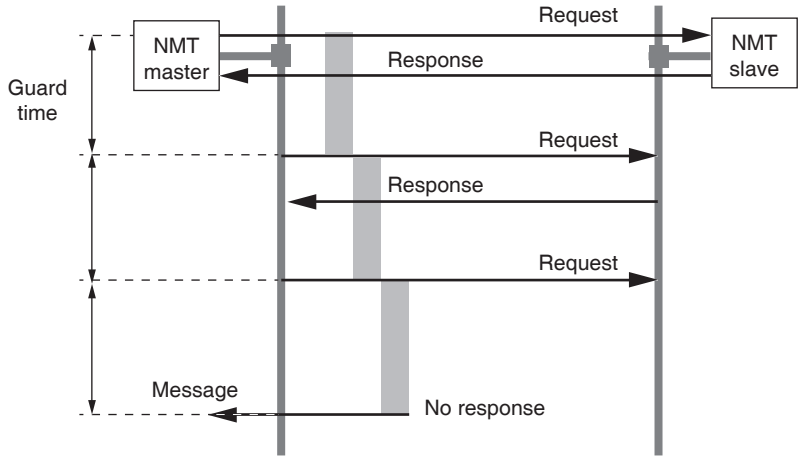
Node Guarding/Life Guarding is configured via:
- Guard time ($100C_h$)
- Life time factor ($100D_h$)

Connection Error

The NMT master signals a connection error to the master program if:
- the slave does not respond within the «guard time» period, or
- the NMT state of the slave has changed without a request by the NMT master.

The following figure shows an error message after the end of the third cycle because of a missing response from an NMT slave:

Figure 1.32 - «Node Guarding» and «Life Guarding» with Time Intervals



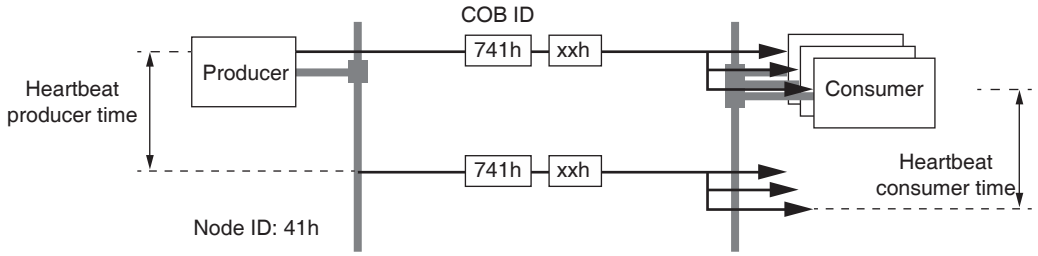
Heartbeat

The optional Heartbeat protocol replaces the node guarding/life guarding protocol. It is recommended for new device versions.

A heartbeat producer transmits a heartbeat message cyclically at the frequency defined in the object **`Producer heartbeat time (1017h)`**. One or several consumers can receive this message. **`Producer heartbeat time (1016h) = 0`** deactivates heartbeat monitoring.

The relationship between producer and consumer can be configured with objects. If a consumer does not receive a signal within the period of time set with **Consumer heartbeat time (1016h)**, it generates an error message (heartbeat event). Consumer heartbeat time (**1016h**) = 0 deactivates monitoring by a consumer.

Figure 1.33 - «Heartbeat» Monitoring



Data byte for NMT state evaluation of the “Heartbeat” producer:

- 0 (00_h): “Boot-Up”
- 4 (04_h): “Stopped”
- 5 (05_h): “Operational”
- 127 ($7F_h$): “Pre-Operational”

Time Intervals

The time intervals are set in increments of 1 ms steps; the values for the consumer must not be less than the values for the producer. Whenever the “Heartbeat” message is received, the time interval of the producer is restarted.

Start of Monitoring

“Heartbeat” monitoring starts as soon as the time interval of the producer is greater than zero. If “Heartbeat” monitoring is already active during the NMT state transition to “Pre-Operational”, “Heartbeat” monitoring starts with sending of the boot-up message. The boot-up message is a “Heartbeat” message with one data byte 00_h.

Devices can monitor each other via “Heartbeat” messages. They assume the function of consumer and producer at the same time.

Chapter 2
Commissioning

What's in this Chapter?

This chapter includes the following topics:

The product is unable to detect an interruption of the network link if connection monitoring is not active.

⚠WARNING

LOSS OF CONTROL

- Verify that connection monitoring is on.
- Set the shortest, practical monitoring time cycles to detect communication interruptions as quickly as possible.

Failure to follow these instructions can result in death, serious injury or equipment damage.

⚠WARNING

UNINTENDED OPERATION

- Do not write values to reserved parameters.
- Do not write values to parameters unless you fully understand the function.
- Run initial tests without coupled loads.
- Verify that the system is free and ready for the movement before changing parameters.
- Verify the use of the word sequence for fieldbus communication.
- Do not establish a fieldbus connection unless you have fully understood the communication principles.
- Only start the system if there are no persons or obstructions in the zone of operation.

Failure to follow these instructions can result in death, serious injury or equipment damage.

Commissioning the Device

For installation in the network, the device must first be properly installed (mechanically and electrically) and commissioned. Install the device as per the related hardware manual.

Commissioning the device can be accomplished via one of two methods:

1. Using the CANopen Configuration Utility, part of the LSS, which may be downloaded from our website at: https://novantaims.com/.
2. By using Layer Setting Services, as defined in CiA 305: Layer Setting Services

Two conditions must be fulfilled in order for a CANopen device to operate on a network:

- It must have a unique Node ID
- It must have the same communication baud rate as all other devices on the network

Address and Baud Rate

Up to 64 devices can be addressed in one CAN bus network branch and up to 127 devices in the extended network. Each device is identified by a unique address or Node ID. The default Node ID for an LMD is 41_h.

The default baud rate is 1 MBaud (1000 kbps).

Each device must be assigned a unique node address (i.e., any given node address may be assigned only once in the network).

The baud rate must match the baud rate setting of the network into which the device is being installed.

Commissioning via CANopen Configuration Utility

The NODE ID and BAUD rate may be configured using the CANopen Configuration Utility, which is part of the LSS.

Instructions for installing and using this tool to commission the LMD CANopen device are available in the LSS Manual.

Both the software and user manual are available for download at:
https://novantaims.com/dloads/

Commissioning via Layer Setting Services

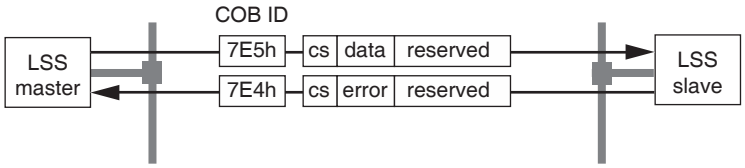
The device may be commissioned using CAN Layer Setting Services. Reference CiA 305.

The scope of Layer Setting Services is to allow the Node ID and communication baud rate to be read or written through the network.

The Node ID and baud rate must be set using Layer Setting Service commands. It is recommended that the parameters be set prior to the installation of the device into a network.

Layer Setting Service messages are 8 bytes in length.

Figure 2.1 - Layer Setting Service Message Structure



The Layer Setting Service message consists of a COB IDs specific to the Layer Setting Service master and Layer Setting Service slave.

The command specifier identifies the action to be taken.

COB ID

Layer Setting Service commands use two specific COB IDs to request and respond to Layer Setting Service commands:

- $07E5_h$ – COB ID of the Layer Setting Service slave device
- $07E4_h$ – COB ID of the Layer Setting Service master device to which message responses are sent

Switch Modes

There are two methods of initiating communications with the device to be commissioned:

- Switch mode global: can set all connected devices into configuration mode. Can be used to set the baud rate of multiple connected devices, can only be used to set the Node ID if one device is connected.
- Switch mode selective: can be used to set the parameters of a single device in the network using vendor specific objects such as the serial number to communicate directly to the device.

The switch mode commands are used to set the device into either operational or configuration mode. In order to make changes to the parameters it must be in configuration mode.

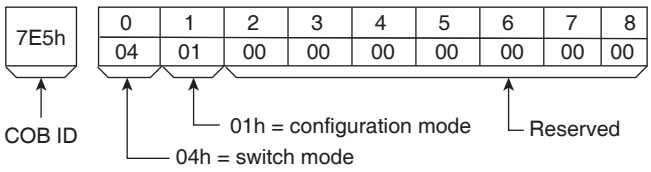
Commissioning via Switch Mode Global

Procedure

The following procedure will step through configuring the Node ID and baud rate parameters from the perspective of a single device connected to the Layer Setting Service master.

1. Transmit the command to the device setting it into switch mode global - configuration mode:

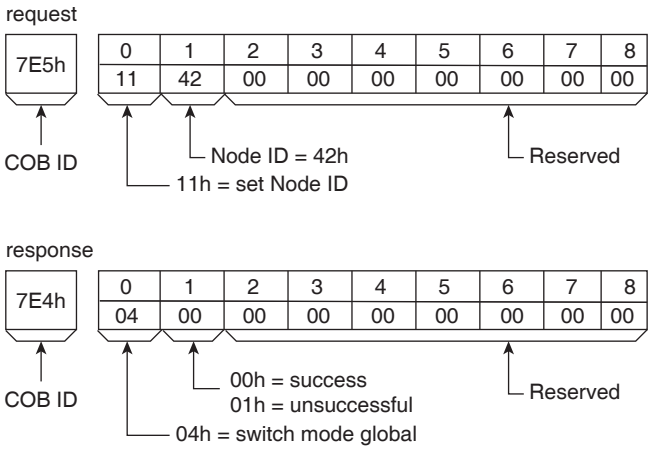
Figure 2.2 - Setting Global - Configuration Mode



This will place the device in configuration mode. This is an unacknowledged Layer Setting Service; there will be no response.

2. Set the new Node ID (42_h used in the following example).

Figure 2.3 - Configuration Mode Example

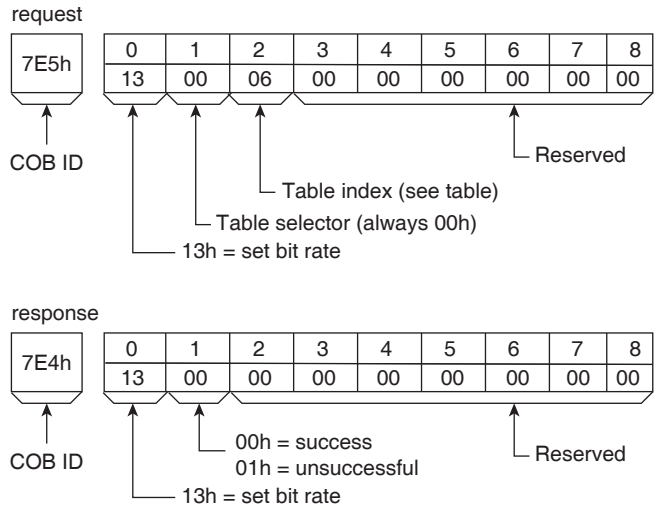


3. Set the desired baud rate using the table below as a reference (for example purposes 50 kbps will be used):

Table 2.1 - Baud Rate Settings

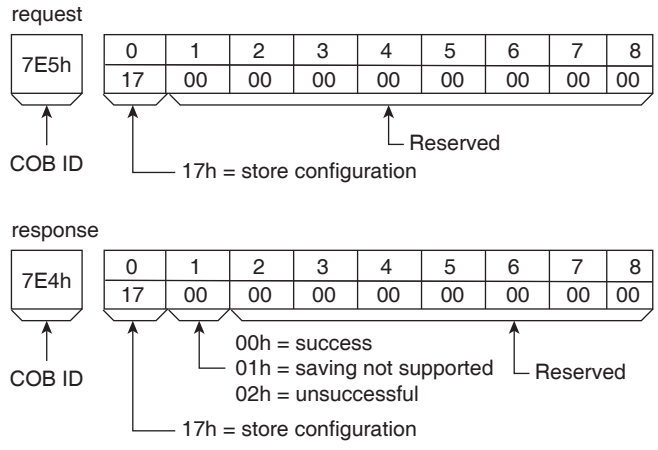
Table index	Baud rate (kbps)
00_h	1000 (default)
01_h	800*
02_h	500
03_h	250
04_h	125
05_h	100
06_h	50
07_h	20
08_h	10
*Not available if using MD-CC500-000 USB to CANopen converter cable.	

Figure 2.4 - Baud Rate Setting Example



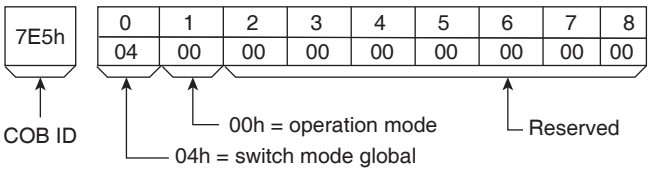
4. Save the changed parameters using the “Store configuration” service (17_h).

Figure 2.5 - Save Configuration Example



5. Change the mode from configuration to operational.

Figure 2.6 - Change Mode Example



6. The new Node ID is now active. To make the new bit rate active, cycle power to the device.
7. The device is now commissioned and ready to be placed in a network having a Node ID of 42_h and a bitrate of 50 kbps.

Commissioning via Switch Mode Selective

Using switch mode selective it is possible to isolate a single unit installed on a CANopen network and commission it alone by sending out a four-part signature.

The signature consists of:

- VendorID = 0800005A$_h$
- Product code = 00
- Revision number = 507$_h$
- Serial number

The first three parameters are identical throughout the product line. The serial number may be retrieved from the label on the device or from the Indexes 1018$_h$4 or 5002$_h$1.

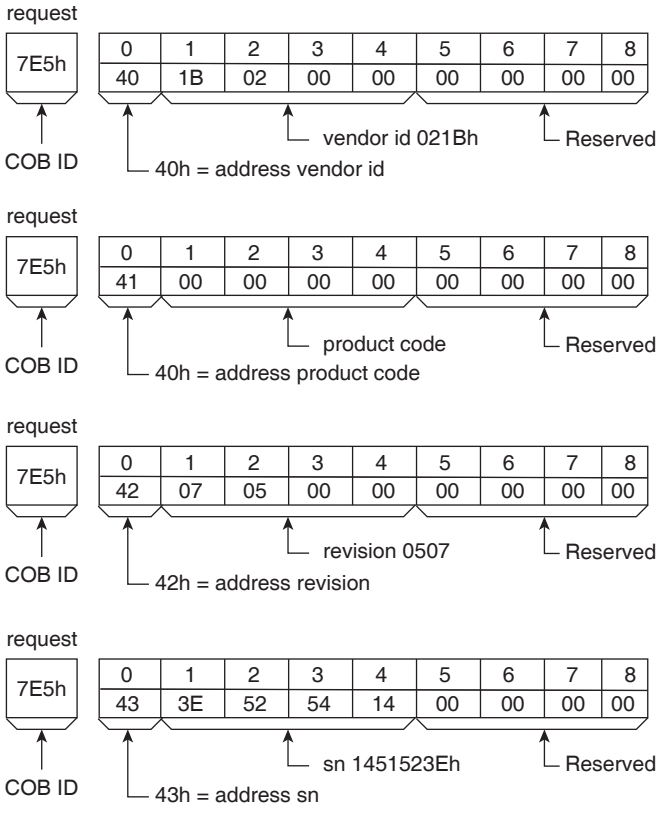
Using only the numeric portion of the serial, convert it to a 4 byte hex number, for example:

341070398 = 1454523E

Procedure

1. Send the four parameters to place the desired device in configuration mode.

Figure 2.7 - Device Parameters in Configuration Mode



2. Perform steps 2 – 7 of the Switch Mode Global procedure to commission the device.

Chapter 3
Operation

What's in this Chapter?

The section "Operation" describes the basic operating states, operating modes, and functions of the device.

This chapter includes the following topics:

⚠WARNING

UNINTENDED OPERATION

- Do not write values to reserved parameters.
- Do not write values to parameters unless you fully understand the function.
- Run initial tests without coupled loads.
- Verify that the system is free and ready for the movement before changing parameters.
- Verify the use of the word sequence for fieldbus communication.
- Do not establish a fieldbus connection unless you have fully understood the communication principles.
- Only start the system if there are no persons or obstructions in the zone of operation.

Failure to follow these instructions can result in death, serious injury or equipment damage.

Operating States

State Diagram

After switching on and when an operating mode is started, the product goes through a number of operating states.

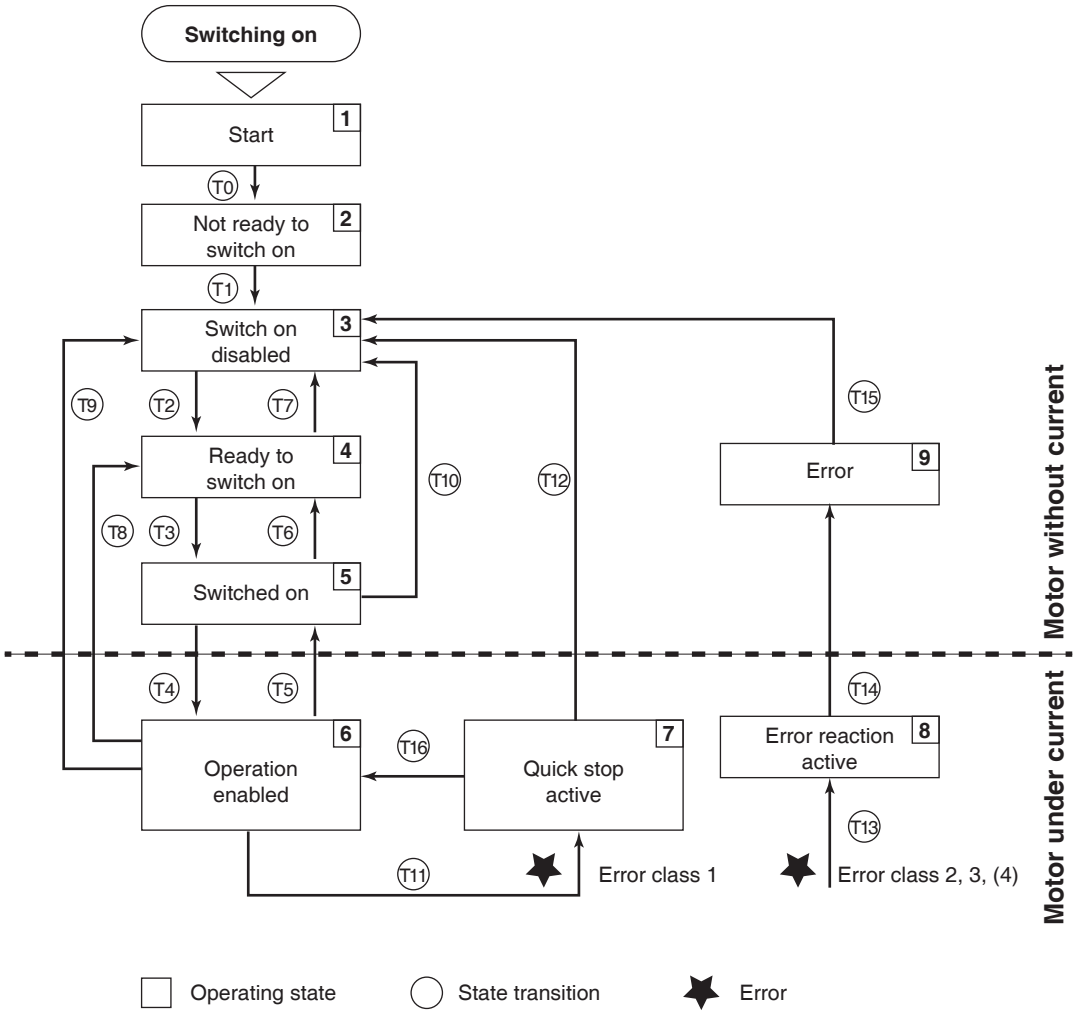
The state diagram (state machine) shows the relationships between the operating states and the state transitions.

The operating states are monitored and influenced by internal monitoring functions and system functions such as temperature monitoring or current monitoring.

Graphical Representation

The state diagram is represented as a flowchart:

Figure 3.1 - State Diagram



Operating States

Refer to Table 3.1 for an explanation of each operating states shown in Figure 3.1.

Table 3.1 - Operating States

Operating state		Description
1	Start	Controller supply voltage switched on Electronics are initialized
2	Not Ready To Switch On	The power stage is not ready to switch on
3	Switch On Disable	Impossible to enable the power stage
4	Ready To Switch On	The power stage is ready to switch on.
5	Switched On	Power stage is switched on
6	Operation Enabled	Power stage is enabled Selected operating mode is active
7	Quick Stop Active	“Quick Stop” is being executed
8	Error Reaction Active	Error response is active
9	Error	Error response terminated, Power stage is disabled

Error Class

The product triggers an error response if an error is detected. Depending upon the severity of the error, the device responds in accordance with one of the following error classes:

Table 3.2 - Error Class

Class	Response	Description
0	Advisory	A monitoring function has detected a problem. No interruption of the movement.
1	“Quick Stop”	Motor stops with “Quick Stop”, the power stage remains enabled.
2	“Quick Stop” with switch-off	Motor stops with “Quick Stop”, the power stage is disabled after standstill has been achieved.
3	Recoverable error	The power stage is immediately disabled without stopping the motor first.
4	Unrecoverable error	The power stage is immediately disabled without stopping the motor first. The error can only be reset by switching off the product.

Error Response

The state transition T13 (error class 2, 3, or 4) initiates an error response as soon as an internal occurrence signals an error to which the device must react.

Table 3.3 - Error Response

Class	State from -> to	Description
2	x -> 8	Stop movement with “Quick Stop” Power stage is disabled
3, 4, or Safety function STO	x -> 8 -> 9	Power stage is disabled immediately, even if “Quick Stop” is still active.

When an error is triggered, such as by a temperature sensor, the product cancels the running movement and performs an error response, such as stopping with «Quick Stop» or disabling the power stage. Subsequently, the operating state changes to 9 (Error). To exit the Error operating state, the cause of the error must be remedied and an Error Reset must be executed. Once the Error Reset is executed, the error message will clear.

Control and Status

Controlling the State Machine (Controlword 6040_h)

`Controlword 6040h` is a mandatory index which sets the operating states and modes of the state machine.

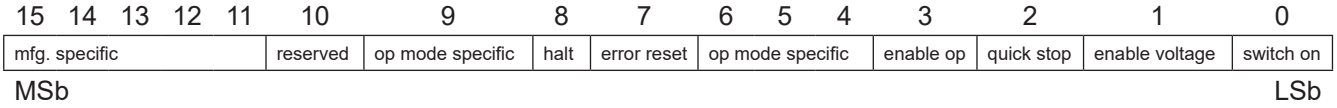


Table 3.4 - Controlword Value Range

Command	Bits of the 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
Error reset	0 –> 1	X	X	X	X	15

Op mode	Bits of the control word			
	Bit 8	Bit 6	Bit 5	Bit 4
Profile position	Halt	Abs/rel	Change set	New setpoint
Profile velocity	Halt	Reserved	Reserved	Reserved
Homing	Halt	Reserved	Reserved	Homing start

Table 3.5 - 6040_h Object Description

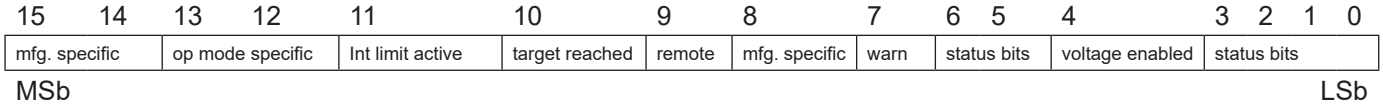
Index	6040_h
Name	Control word
Object code	VAR
Data type	Unsigned16
Category	Mandatory

Table 3.6 - 6040_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	See the Command and Op mode tables above
Default value	Device and operation mode specific

Indication of the Operating State (Statusword $6041:0_h$)

The **`status word 6041:0h`** provides information on the operating state of the device and the processing status of the operating mode



Status Bits

Bit 6: Switch on disabled
Bit 5: Quick stop
Bit 3: Error
Bit 2: Operation enabled
Bit 1: Switch on
Bit 0: Ready to switch on

Table 3.7 - Statusword Status Bit States

Status	Bits of the status word					
	Bit 6	Bit 5	Bit 3	Bit 2	Bit 1	Bit 0
Not Ready To Switch On	0	X	0	0	0	0
Switch On Disabled	1	X	0	0	0	0
Ready To Switch On	0	1	0	0	0	1
Switched On	0	1	0	0	1	1
Operation Enabled	0	1	0	1	1	1
Quick Stop Active	0	0	0	1	1	1
Error Reaction Active	0	X	1	1	1	1
Error	0	X	1	0	0	0

Bit 4: Voltage Enabled

Bit 4=1 indicates whether the DC bus voltage is correct. If the voltage is missing or too low, the device does not transition from operating state 3 to operating state 4.

Bit 5: Quick Stop Active

When reset, this bit indicates that the drive is reacting on a quick stop request. Bits 0, 1, and 2 of the **statusword** must be set to 1 to indicate that the drive is capable to regenerate. The setting of the other bits indicates the status of the drive (e.g. the drive is performing a quick stop as result of a reaction to a recoverable error. The error bit is set as well as bits 0, 1, and 2).

Bit 7: Advisory

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

Bit 8: Manufacturer Specific

This bit may be used by a drive manufacturer to implement any manufacturer specific functionality. This bit is not used by LMD products.

Bit 9: Remote

If bit 9 is set, then parameters may be modified via the CAN-network, and the drive executes the content of a command message. If the bit remote is reset, then the drive is in local mode and will not execute the command message. The drive may transmit messages containing valid actual values like a **position_actual_value**, depending on the actual drive configuration. The drive will accept accesses via service data objects (SDOs) in local mode.

Bit 10: Target Reached

If bit 10 is set by the drive, then a setpoint has been reached (e.g., torque, speed, or position, depending on the **modes_of_operation**). The change of a target value by software alters this bit. If **quickstop_option_code** is 5, 6, 7, or 8, this bit must be set, when the quick stop operation is finished and the drive is halted. If Halt occurred and the drive has halted then this bit is set too.

Bit 11: Internal Limit Active

This bit is set by the drive and indicates that an internal limitation is active (e.g., **position_range_limit**).

Bits 12-13: Operation Mode Specific

Op mode	Bits of the control word	
	Bit 13	Bit 12
Profile position	Following error	Set point acknowledge
Profile velocity	Max slippage error	Speed
Homing	Homing error	Homing attained

Bit 14-15: Unused

Option Code Objects

Abort Connection (6007_h)

This object indicates what action is performed when one of the following events occurs: bus-off, heartbeat, life guarding, NMT stopped state entered, reset application, and reset communication

Table 3.8 - 6007_h Object Description

Index	6007_h
Name	Abort connection
Object code	VAR
Data type	Integer16
Category	Optional

Table 3.9 - 6007_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	0 to 3
Default value	0

Table 3.10 - Abort Connection Option Code

Value	Meaning
0	No action
1	Error signal
2	Disable voltage command
3	Quick stop command

Error Code ($603F_h$)

This object provides the error code of the last error that occurred in the drive device. This object provides the same information as the lower 16-bit of sub-index 01_h of the pre-defined error field (**1003h**).

Table 3.11 - $603F_h$ Object Description

Index	$603F_h$
Name	Error code
Object code	VAR
Data type	Unsigned16
Category	Optional

Table 3.12 - $603F_h$ Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	Unsigned16
Default value	0

Quick Stop ($605A_h$)

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 operations.

Table 3.13 - $605A_h$ Object Description

Index	$605A_h$
Name	Quick stop
Object code	VAR
Data type	Integer16
Category	Optional

Table 3.14 - $605A_h$ Entry Description

Sub-index	00_h
Access	rw
PDO mapping	No
Value range	0 to 6
Default value	2

Table 3.15 - Quick Stop Option Codes

Value	Meaning
0	Disable drive function
1	Slow down on slow down ramp and transit into Switch On Disabled
2	Slow down on quick stop ramp and transit into Switch On Disabled
3	Slow down on current limit and transit into Switch On Disabled
4	Slow down on voltage limit and transit into Switch On Disable
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

Shutdown ($605B_h$)

This object indicates what action is performed if there is a transition from ‘Operation Enabled’ state to ‘Ready To Switch On’ state. The slow down ramp is the deceleration value of the used mode of operations.

Table 3.16 - $605B_h$ Object Description

Index	$605B_h$
Name	Shutdown
Object code	VAR
Data type	Integer16
Category	Optional

Table 3.17 - $605B_h$ Entry Description

Sub-index	00_h
Access	rw
PDO mapping	No
Value range	0 to 1
Default value	0

Table 3.18 - Shutdown Option Codes

Value	Meaning
0	Disable drive function (Switch off the drive power stage)
1	Slow down with slow down ramp; disable the drive function

Disable Operation ($605C_h$)

This object indicates what action is performed if there is a transition from ‘Operation Enabled’ state to ‘Switched On’ state. The slow down ramp is the deceleration value of the used mode of operation.

Table 3.19 - $605C_h$ Object Description

Index	$605C_h$
Name	Disable operation
Object code	VAR
Data type	Integer16
Category	Optional

Table 3.20 - $605C_h$ Entry Description

Sub-index	00_h
Access	rw
PDO mapping	No
Value range	0 to 1
Default value	0

Table 3.21 - Disable Operation Option Codes

Value	Meaning
0	Disable drive function (Switch off the drive power stage)
1	Slow down with slow down ramp; disable of the drive function

Halt ($605D_h$)

This object indicates what action is performed if there is a transition from ‘Operation Enabled’ state to ‘Switched On’ state. The slow down ramp is the deceleration value of the used mode of operations.

Table 3.22 - $605D_h$ Object Description

Index	$605D_h$
Name	Halt
Object code	VAR
Data type	Integer16
Category	Optional

Table 3.23 - $605D_h$ Entry Description

Sub-index	00_h
Access	rw
PDO mapping	No
Value range	1 or 2
Default value	2

Table 3.24 - Halt Option Codes

Value	Meaning
1	Slow down on slow down ramp and stay in Operation Enabled
2	Slow down on quick stop ramp and stay in Operation Enabled

Error Reaction ($605E_h$)

This object indicates what action is performed when an error is detected in the PDS. The slow down ramp is the deceleration value of the used mode of operations.

Table 3.25 - $605E_h$ Object Description

Index	$605E_h$
Name	Halt
Object code	VAR
Data type	Integer16
Category	Optional

Table 3.26 - $605E_h$ Entry Description

Sub-index	00_h
Access	rw
PDO mapping	No
Value range	0 to 2
Default value	2

Table 3.27 - Error Reaction Option Codes

Value	Meaning
0	Disable drive function, motor is free to rotate
1	Slow down on slow down ramp
2	Slow down on quick stop ramp

Supported Modes of Operation

The function of the product depends on the selected modes of operation. It is not possible to operate the modes in parallel. The mode of operation must be selected. An example of exclusive functions are Profile Velocity and Profile Position modes. Supported modes are:

1. Profile position
2. Homing mode
3. Profile velocity

This product allows dynamic switching between various operation modes.

The mode of operation is set or read using Mode of Operation (6060_h) and Mode of Operation display (6061_h).

Mode of Operation (6060_h)

This object indicates the requested operation mode.

Table 3.28 - 6060_h Object Description

Index	6060_h
Name	Mode of operation
Object code	VAR
Data type	Integer8
Category	Optional

Table 3.29 - 6060_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	No
Value range	1, 3 or 6
Default value	1

Table 3.30 - Mode of Operation

Value	Meaning
1	Profile Position
3	Profile Velocity
6	Homing

Mode of Operation Display (6061_h)

The Modes of Operation Display represents the current mode of operation. The meaning of the returned value corresponds to that of the Modes of Operation option code (index 6060_h).

Table 3.31 - 6061_h Object Description

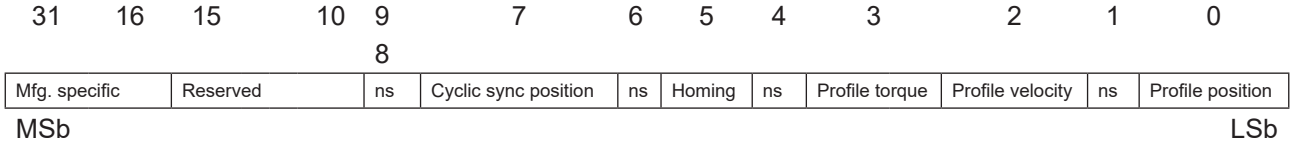
Index	6061_h
Name	Mode of operation display
Object code	VAR
Data type	Integer8
Category	Optional

Table 3.32 - 6061_h Entry Description

Sub-index	00_h
Access	ro
PDO mapping	Yes
Value range	1, 3, or 6 (Refer to Mode of Operation table in the “Mode of Operation (6060h)” section above)
Default value	1

Supported Drive Modes (6502_h)

This object provides information on the supported drive modes.



ns = not supported

Table 3.33 - 6502_h Object Description

Index	6502_h
Name	Supported drive modes
Object code	VAR
Data type	Unsigned32
Category	Optional

Table 3.34 - 6502_h Entry Description

Sub-index	00_h
Access	ro
PDO mapping	No
Value range	Unsigned32
Default value	00000025_h

Profile Position Mode

Overview

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 section. These two function blocks are optionally controlled by individual parameter sets.

At the input to the trajectory generator, parameters may have optional limits applied before being normalized to internal units. Normalized parameters are denoted with an asterisk. The simplest form of a trajectory generator is just to pass through a **target_position** and to transform it to a **position_demand_value** with internal units (increments) only.

Figure 3.2 - Trajectory Generator for Profile Position Block Diagram

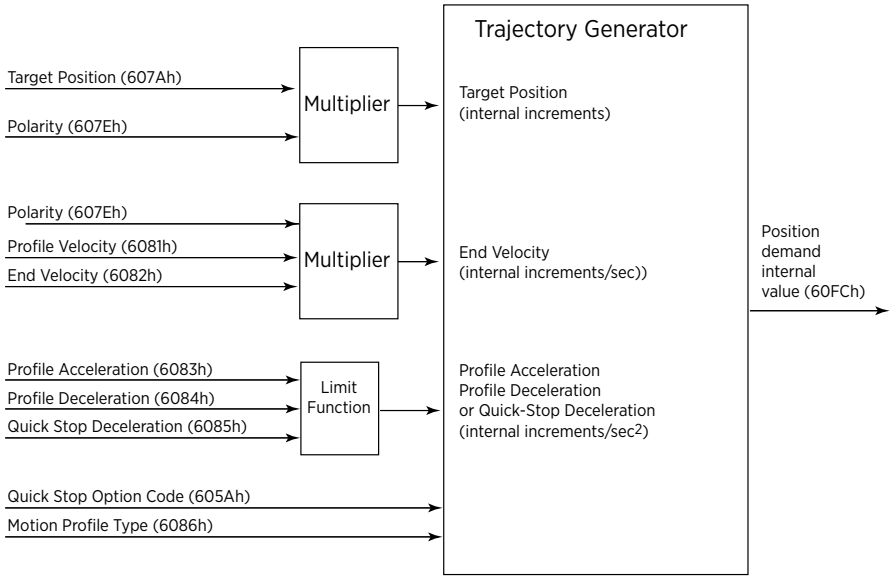


Table 3.35 - Input Data Objects for Profile Position:

Object	Name	Description
$607A_h$	**target_position**	Defines the targeted absolute or relative position for a move
$607E_h$	**polarity**	Sets the polarity for position or speed commands
6081_h	**profile_velocity**	Sets the velocity for profile position motion
6082_h	**initial_velocity**	Sets the velocity upon reaching target
6083_h	**profile_acceleration**	Sets the acceleration for profile position and profile velocity
6084_h	**profile_deceleration**	Sets the deceleration for profile position and profile velocity
6085_h	**quick_stop_decel**	Sets the deceleration for quick stop active

Table 3.36 - Output Data Objects for Profile Position:

Object	Name	Description
$607A_h$	**position_demand_value**	Displays the motor position

Functional Description

There are two different ways to apply **target_positions** to a drive, which are supported by this device profile.

1. Set of set-points:
   After reaching the **target_position** the drive unit immediately processes the next **target_position** which results in a move where the velocity of the drive normally is not reduced to zero after achieving a set-point.
2. Single set-point:
   After reaching the **target_position** the drive unit signals this status to a host computer and then receives a new set-point. After reaching a **target_position** the velocity normally is reduced to zero before starting a move to the next set-point.

The two modes are controlled by the timing of the bits **new_set-point** and **change_set_immediately** in the controlword and **set-point_acknowledge** in the **statusword**.

These bits allow the set up a request-response mechanism in order to prepare a group of set-points while another set is still processed in the drive unit. This minimizes reaction times within a control program on a host computer. The following graphic and table shows the set-point transmission from the host and host bit rates:

Figure 3.3 - Set-Point Transmission from Host

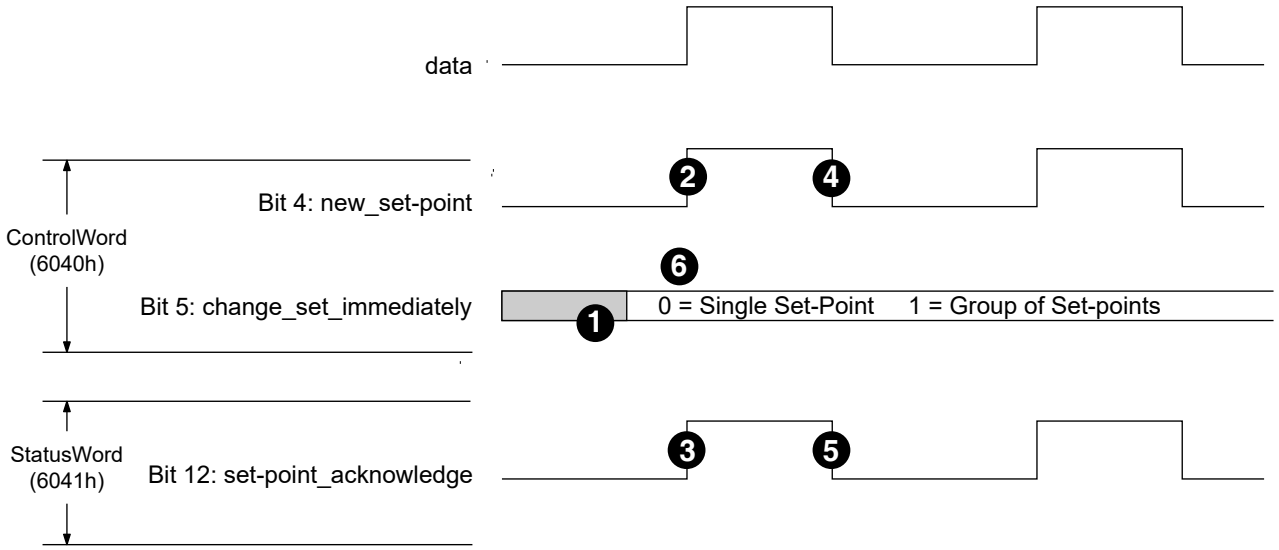


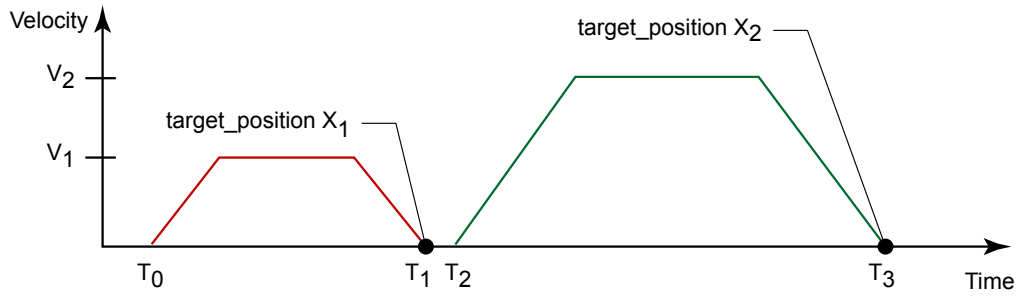
Table 3.37 - Set-point Transmission from Host Bit States

Bit state #	Meaning
(1)	Single set-point is expected by device
(2)	Host signals “data is valid” new set-point = 1
(3)	Device response: sets bit 12, **set-point acknowledge** = 1
(4)	Data is validated, host may release new set-point
(5)	Device response: sets bit 12, **set-point acknowledge** = 0 Device ready to receive new data
(6)	Indicates state of **change_set_immediately** = 1

Figure 3.3, Figure 3.4, and Figure 3.5 illustrate the difference between the “group of set-points” mode and the “single set-point” mode. The initial status of the bit **change_set_immediately** in the controlword determines which mode is used. Trapezoidal moves are used as this is the only **motion_profile_type** the LMD CANopen supports.

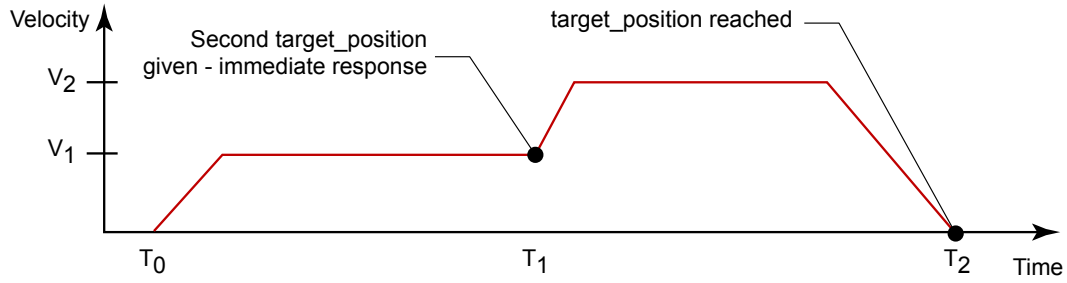
If the bit **change_set_immediately** is “0” (shaded area in Figure 3.3) a single set-point is expected by the drive ***(1)***. After data is applied to the drive, a host signals that the data is valid by changing the bit **new_setpoint** to “1” in the controlword ***(2)***. The drive responds with **set-point_acknowledge** set to “1” in the statusword ***(3)*** after it is recognized and has buffered the new valid data. Now the host may release **new_setpoint** (4) and afterwards the drive signals with **set-point_acknowledge** equal “0” its ability to accept new data again ***(5)***. In , this mechanism results in a velocity of zero after ramping down in order to reach a **target_position** X_1 at T_1. After signaling to the host that the set-point has been reached as described above, the next **target_position** X_2 is processed at T_2 and reached at T_3. The following graphic shows the Single Set-point Mode (Move After a Move) 6040_h Bit 5=0.

Figure 3.4 - Single Set-point Mode (Move After Move) 6040_h Bit 5=0



With **change_set_immediately** set to “1” ***(6)***, symbolized by the clear area in Figure 3.3, the host advises the drive to apply a new set-point immediately after reaching the last one. The relative timing of the other signals is unchanged. This behavior causes the drive to already process the next set-point X_2 and to keep its velocity when it reaches the **target_position** X_1 at T_1. The drive then moves immediately to the already calculated next **target_position** X_2.

Figure 3.5 - Group of Set-points (Move on a Move) 6040_h Bit 5=1



Controlword Definition for Profile Position

15 ... 9	8	7	6	5	4	3 ... 0
See Table 3.4 on page 51	halt	See Table 3.4 on page 51	abs/rel	change set immediately	new set point	See Table 3.4 on page 51

Table 3.38 - Profile Position Mode Controlword (6040_h) Bit State Meanings

Bit	Name	Value	Meaning
4	New set-point	0	Does not assume target position
		1	Assume target position
5	Change set immediately	0	Finish the actual positioning and then start the next positioning
		1	Interrupt the actual positioning and start the next positioning
6	abs/rel	0	Target position is an absolute value
		1	Target position is a relative value
8	Halt	0	Execute positioning
		1	Stop motion with profile deceleration

Statusword Definition for Profile Position

15 ... 14	13	12	11	10	9 ... 0
See Table 3.7 on page 52	following error	set-point acknowledge	See Table 3.7 on page 52	target reached	See Table 3.7 on page 52

Table 3.39 - Profile Position Mode Statusword (6041_h) Bit State Meanings

Bit	Name	Value	Meaning
10	Target reached	0	Halt=0: Target position not reached Halt=1: Axis decelerating
		1	Halt=0: Target position reached Halt=1: Axis velocity is 0
12	Set-point acknowledge	0	Trajectory generator has not assumed the positioning values yet
		1	Trajectory generator has assumed the positioning values
13	Following error	0	No following error
		1	Following error

Position, Velocity, and Acceleration Objects

$607A_h$ Target Position

The target position is the position that the drive should move to in position profile mode using parameters such as velocity, acceleration, deceleration, motion profile type, etc. The target position is given in terms of 51,200 units per motor shaft revolution. The target position will be interpreted as absolute or relative depending on the absolute-relative flag (bit 6) in the controlword.

Table 3.40 - $607A_h$ Object Description

Index	$607A_h$
Name	Target position
Object code	VAR
Data type	Integer32
Category	Optional

Table 3.41 - $607A_h$ Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	8000 0000_h to 7FFF $FFFF_h$
Default value	0000 0000_h

$607E_h$ Polarity

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

7	6	5 ... 0
position polarity	velocity polarity	reserved

Table 3.42 - $607E_h$ Object Description

Index	$607E_h$
Name	Polarity
Object code	VAR
Data type	Unsigned8
Category	Optional

Table 3.43 - $607E_h$ Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	Unsigned8
Default value	00_h

6081_h Profile Velocity

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

Table 3.44 - 6081_h Object Description

Index	6081_h
Name	Profile velocity
Object code	VAR
Data type	Unsigned32
Category	Optional

Table 3.45 - 6081_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	Unsigned32
Default value	512000_d

6082_h Initial Velocity

The initial velocity defines the velocity which the drive must have upon reaching the target position. Normally, the drive stops at the target position (i.e., the **`initial_velocity`** = 0). The initial velocity is given in the same units as profile velocity.

Table 3.46 - 6082_h Object Description

Index	6082_h
Name	Initial Velocity
Object code	VAR
Data type	Unsigned32
Category	Optional

Table 3.47 - 6082_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	Unsigned32
Default value	51200_d

6083_h Profile Acceleration

Profile acceleration is given in steps/sec^2

Table 3.48 - 6083_h Object Description

Index	6083_h
Name	Profile acceleration
Object code	VAR
Data type	Unsigned32
Category	Optional

Table 3.49 - 6083_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	Unsigned32
Default value	100000_d

6084_h Profile Deceleration

Profile deceleration is given in steps/sec^2

Table 3.50 - 6084_h Object Description

Index	6084_h
Name	Profile deceleration
Object code	VAR
Data type	Unsigned32
Category	Optional

Table 3.51 - 6084_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	Unsigned32
Default value	100000_d

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 (**Idx 605Ah**) is set to 2 or 4. The quick stop deceleration is also used if the error reaction code object (**Idx 605Eh**) is 2 and the halt option code object (**Idx 605Dh**) is 2. The value is given in the same physical unit as profile acceleration object (**Idx 6083h**).

Table 3.52 - 6085_h Object Description

Index	6085_h
Name	Quick stop deceleration
Object code	VAR
Data type	Unsigned32
Category	Optional

Table 3.53 - 6085_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	Unsigned32
Default value	200000_d

6086_h Motion Profile Type

The motion profile type is used to select the type of motion profile used to perform a move. The represented devices are fixed at value 0: linear ramp (trapezoidal profile)

Table 3.54 - 6085_h Object Description

Index	6085_h
Name	Motion profile type
Object code	VAR
Data type	Integer16
Category	Optional

Table 3.55 - 6085_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	0_d
Default value	0_d

Profile Position Application Example

The represented device(s) support relative and absolute moves to position. Using either relative or absolute moves, it is possible to select (by the controlword data) if the target position should be reached before another target position is allowed (finish first) or if the product should execute a newly received target position even if still in motion (immediate).

The following example sets typical motion profile commands a system would configure[1], enabling the motor power[2] and the four different move types[3] supported in profile position mode using SDOs with Node $ID41_h$.

All values shown are hexadecimal.

1 Typical motion profile commands could be set each time on power up from host or set using a configuration file and stored to non-volatile memory (NVM) once.
2 Enabling the motor power only has to be done once on power up.
3 The Control Word data selects the move type.

Table 3.56 - Profile Position Mode Application Example

ID	RTR	Data String	Action
Typical motion parameters			
0641	00	2F 04 22 00 50 00 00 00	Set run current to 80%
0641	00	23 84 60 00 40 42 0F 00	Set deceleration to 1M steps/sec^2
0641	00	23 83 60 00 40 42 0F 00	Set acceleration to 1M steps/sec^2
0641	00	23 81 60 00 00 D0 07 00	Set max velocity to 512000 steps/sec
Enable motor power - CiA 402 state machine			
0641	00	2B 40 60 00 06 00 00 00	Ready to switch on
0641	00	2B 40 60 00 07 00 00 00	Switched on
0641	00	2B 40 60 00 0F 00 00 00	Operation enable
Set to profile position mode			
0641	00	2F 60 60 00 01 00 00 00	Set to profile position mode
Perform absolute move, finish before performing additional moves			
0641	00	23 7A 60 00 30 75 00 00	Set target position to 30000 steps
0641	00	2B 40 60 00 1F 00 00 00	Set control word bit 4 to 1
0641	00	2B 40 60 00 0F 00 00 00	Set control word bit 4 to 0
Perform absolute move, move immediate			
0641	00	23 7A 60 00 B8 0B 00 00	Set target position to 3000 steps
0641	00	2B 40 60 00 3F 00 00 00	Set control word bit 4 to 1
0641	00	2B 40 60 00 2F 00 00 00	Set control word bit 4 to 0
Perform relative move, finish before performing additional moves			
0641	00	23 7A 60 00 A0 86 01 0	Set target position to 100000
0641	00	2B 40 60 00 5F 00 00 00	Set control word bit 4 to 1
0641	00	2B 40 60 00 4F 00 00 00	Set control word bit 4 to 0
Perform relative move, move immediate			
0641	00	23 7A 60 00 B8 0B 00 00	Set target position to 3000 steps
0641	00	2B 40 60 00 7F 00 00 00	Set control word bit 4 to 1
0641	00	2B 40 60 00 6F 00 00 00	Set control word bit 4 to 0

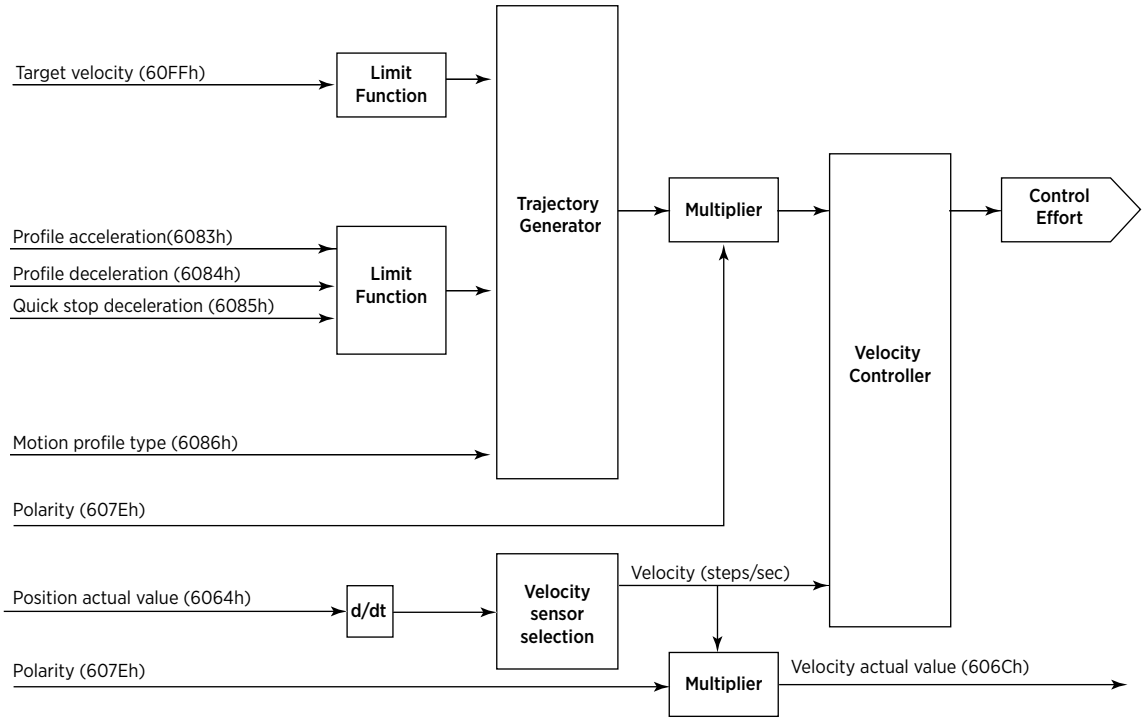
Profile Velocity Mode

Overview

The profile velocity mode covers the following sub-functions:

- Demand value input via trajectory generator
- Velocity capture using position sensor or velocity sensor
- Velocity control function with appropriate input and output signals
- Monitoring of the profile velocity using a window-function
- Monitoring of velocity actual value using a threshold

Figure 3.6 - Profile Velocity Control Function



Controlword Definition for Profile Velocity

15 ... 9	8	7	6 ... 4	3 ... 0
See Table 3.4 on page 51	halt	See Table 3.4 on page 51	reserved	See Table 3.4 on page 51

Table 3.57 - Profile Velocity Mode Controlword (6040_h) Bit State Meanings

Bit	Name	Value	Meaning
8	Halt	0	Execute the motion
		1	Stop axis

Statusword Definition for Profile Velocity

15 ... 14	13	12	11	10	9 ... 0
See Table 3.7 on page 52	max slippage error	speed	See Table 3.7 on page 52	target reached	See Table 3.7 on page 52

Table 3.58 - Profile Velocity Mode Statusword (6041_h) Bit State Meanings

Bit	Name	Value	Meaning
10	Target reached	0	Halt=0: Target position not reached Halt=1: Axis decelerating
		1	Halt=0: Target position reached Halt=1: Axis velocity is 0
12	Speed	0	Speed is not equal to 0
		1	Speed is equal to 0
13	Max slippage error	0	Maximum slippage not reached
		1	Maximum slippage reached

Profile Velocity Mode Objects

$606C_h$ Velocity Actual Value

This object provides the actual velocity value derived either from the velocity sensor or the position sensor. The value is given in microsteps per second.

Table 3.59 - $606C_h$ Object Description

Index	$606C_h$
Name	Velocity actual value
Object code	VAR
Data type	Integer32
Category	Optional

Table 3.60 - $606C_h$ Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	Integer32
Default value	0000 0000_h

$60F8_h$ Maximum Slippage

This object indicates the configured maximum slippage of an asynchronous motor. When the maximum slippage has been reached, the corresponding bit 13 max slippage error in the statusword is set to 1. The reaction of the drive device, when the max slippage error occurs, is manufacturer-specific. This value is given in microsteps.

Table 3.61 - $60FF_h$ Object Description

Index	$60FF_h$
Name	Maximum slippage
Object code	VAR
Data type	Integer32
Category	Optional

Table 3.62 - $60FF_h$ Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	Integer32
Default value	512_d

60FF$_h$ Target Velocity

The Target Velocity is the input to the trajectory generator and the value is given in micro-steps/second.

Table 3.63 - 60FF$_h$ Object Description

Index	60FF$_h$
Name	Target velocity
Object code	VAR
Data type	Integer32
Category	Optional

Table 3.64 - 60FF$_h$ Entry Description

Sub-index	00$_h$
Access	rw
PDO mapping	Yes
Value range	8000 0000$_h$ to 7FFF FFFF$_h$
Default value	0000 0000$_h$

Profile Velocity Application Example

The represented device(s) supports the ability to move in velocity mode. Once in Profile Velocity Mode, any new target velocity will be executed immediately.

The below example sets typical motion profile commands a system would configure[1], enabling the motor power[2] and sending a new target velocity using SDOs with Node ID41$_h$.

All values shown are hexadecimal.

Table 3.65 - Profile Velocity Mode Application Example

ID	RTR	Data String	Action
Typical motion parameters			
0641	00	2F 04 22 00 50 00 00 00	Set run current to 80%
0641	00	23 84 60 00 40 42 0F 00	Set deceleration to 1M steps/sec^2
0641	00	23 83 60 00 40 42 0F 00	Set acceleration to 1M steps/sec^2
0641	00	23 81 60 00 00 D0 07 00	Set max velocity to 512000 steps/sec
Enable motor power - CiA 402 state machine			
0641	00	2B 40 60 00 06 00 00 00	Ready to switch on
0641	00	2B 40 60 00 07 00 00 00	Switched on
0641	00	2B 40 60 00 0F 00 00 00	Operation enable
Set to profile velocity mode			
0641	00	2F 60 60 00 03 00 00 00	Set to profile velocity mode
Send new target velocity			
0641	00	23 FF 60 00 50 C3 00 00	Set target velocity 50000 steps/sec

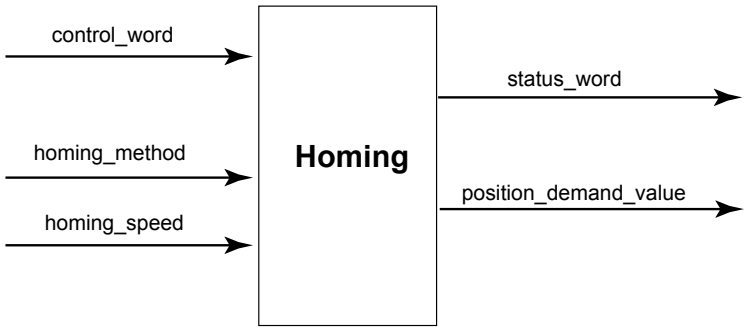
1 Typical motion profile commands could be set each time on power up from host or set using a configuration file and stored to NVM once.
2 Enabling the motor power only has to be done once on power up.

Homing Mode

Overview

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

Figure 3.7 - The Homing Function



Input Data Description

The speeds and method of homing can be specified. There are two homing speeds. In a typical cycle, the faster speed is used to find the home switch and the slower speed is used to find the index pulse. Discretion is allowed in the use of these speeds as the response to the signals is dependent on the hardware used.

Output Data Description

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.

Controlword Definition for Homing Mode

15 ... 9	8	7	6 ... 5	4	3 ... 0
Table 3.4 on page 51	halt	Table 3.4 on page 51	reserved	homing operation start	Table 3.4 on page 51

Table 3.66 - Homing Mode Controlword (6040_h) Bit State Meanings

Bit	Name	Value	Meaning
4	Homing operation start	0	Execute the motion
		0 ⇨ 1	Start homing mode
		1	Homing mode active
		1 ⇨ 0	Interrupt homing mode
8	Halt	0	Execute the instruction of bit 4
		1	Stop axis

Statusword Definition for Homing Mode

15 ... 14	13	12	11	10	9 ... 0
Table 3.7 on page 52	homing error	homing attained	Table 3.7 on page 52	target reached	Table 3.7 on page 52

Table 3.67 - Homing Mode Statusword (6041_h) Bit State Meanings

Bit	Name	Value	Meaning
10	Target reached	0	Halt=0: Target position not reached Halt=1: Axis decelerating
		1	Halt=0: Target position reached Halt=1: Axis velocity is 0
12	Homing Attained	0	Homing mode not yet complete
		1	Homing mode carried out successfully
13	Homing error	0	No homing error
		1	Homing error

Homing Mode Objects

$607C_h$ Homing Offset

This object indicates the configured difference between the zero position for the application and the machine home position (found during homing). During homing, the machine home position is found. Once the homing is completed, the zero position is offset from the home position by adding the home offset to the home position. All subsequent absolute moves are taken relative to this new zero position. If this object is not implemented, then the home offset is regarded as zero. The value of this object is given in microsteps. Negative values indicate the opposite direction.

Figure 3.8 - The Homing Offset

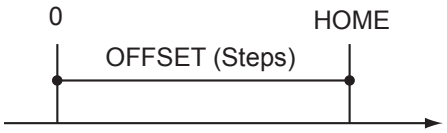


Table 3.68 - $607C_h$ Object Description

Index	$607C_h$
Name	Homing offset
Object code	VAR
Data type	Unsigned32
Category	Optional

Table 3.69 - $607C_h$ Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	Unsigned32
Default value	0_d

6098_h Homing Method

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

Table 3.70 - 6098_h Object Description

Index	6098_h
Name	Homing method
Object code	VAR
Data type	Integer8
Category	Optional

Table 3.71 - 6098_h Entry Description

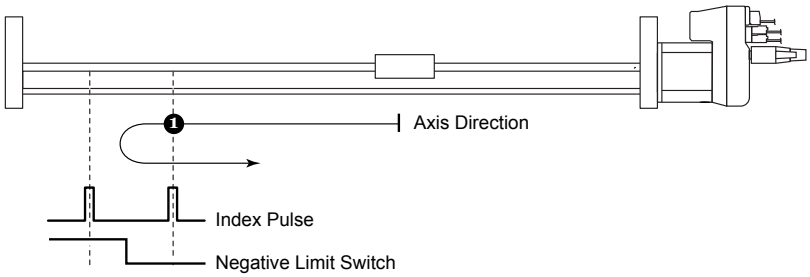
Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	0_d (no homing) 1 – 35_d (method)
Default value	0_d

Functional Description of Homing Methods

Method 1: Homing on the negative limit switch and index pulse

Using this method the initial direction of movement is leftward if the negative limit switch is inactive (here shown as low). The home position is at the first index pulse to the right of the position where the negative limit switch becomes inactive.

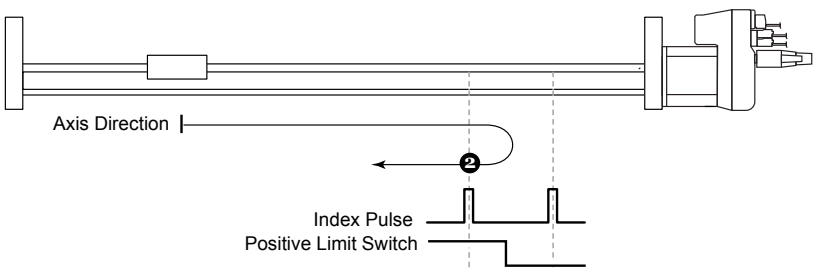
Figure 3.9 - Homing on the Negative Limit Switch and Index Pulse



Method 2: Homing on the positive limit switch and index pulse

Using this method the initial direction of movement is rightward if the positive limit switch is inactive (here shown as low). The position of home is at the first index pulse to the left of the position where the positive limit switch becomes inactive.

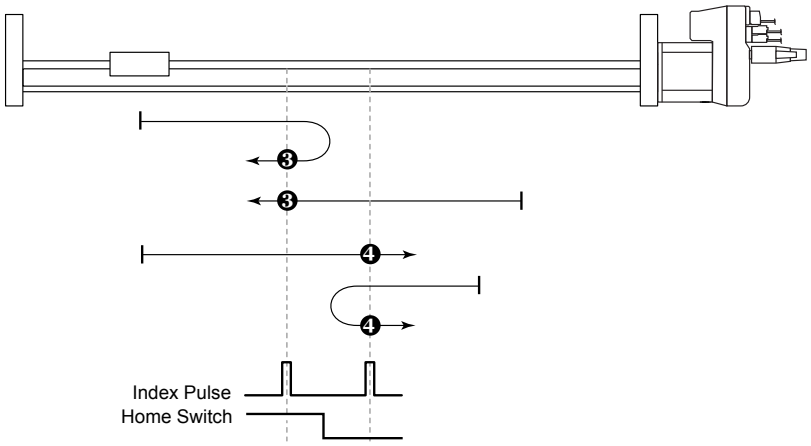
Figure 3.10 - Homing on the Positive Limit Switch and Index Pulse



Methods 3 and 4: Homing on the positive home switch and index pulse

Using methods 3 or 4, the initial direction of movement is dependent on the state of the home switch. The home position is at the index pulse to either the left or the right of the point where the home switch changes state. If the initial position is set so that the direction of movement must reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.

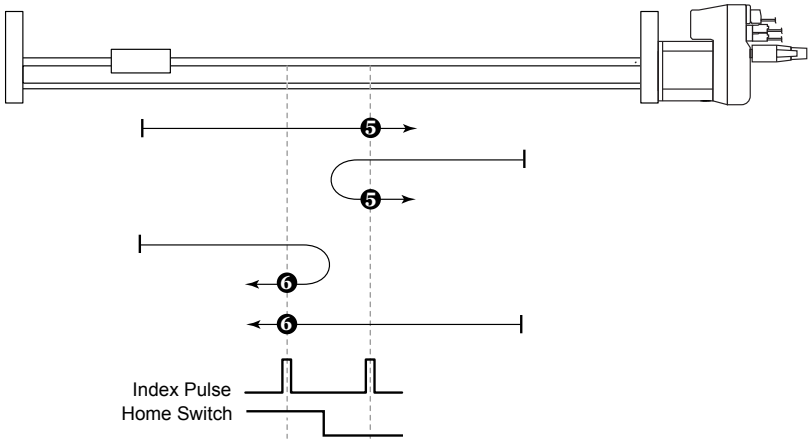
Figure 3.11 - Homing on the Positive Home Switch and Index Pulse



Methods 5 and 6: Homing on the negative home switch and index pulse

Using methods 5 or 6, the initial direction of movement is dependent on the state of the home switch. The home position is at the index pulse to either the left or the right of the point where the home switch changes state. If the initial position is set so that the direction of movement must reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.

Figure 3.12 - Homing on the Negative Home Switch and Index Pulse



Methods 7 to 14: Homing on the Home Switch and Index Pulse

These methods use a home switch which is active over only portion of the travel, in effect the switch has a ‘momentary’ action as the axle`s position sweeps past the switch.

Using methods 7 to 10, the initial direction of movement is to the right, and using methods 11 to 14, the initial direction of movement is to the left except if the home switch is active at the start of the motion. In this case the initial direction of motion is dependent on the edge being sought. The home position is at the index pulse on either side of the rising or falling edges of the home switch, as shown in the following two diagrams.

Figure 3.13 - Homing on the Home Switch and Index Pulse - Positive Initial Move

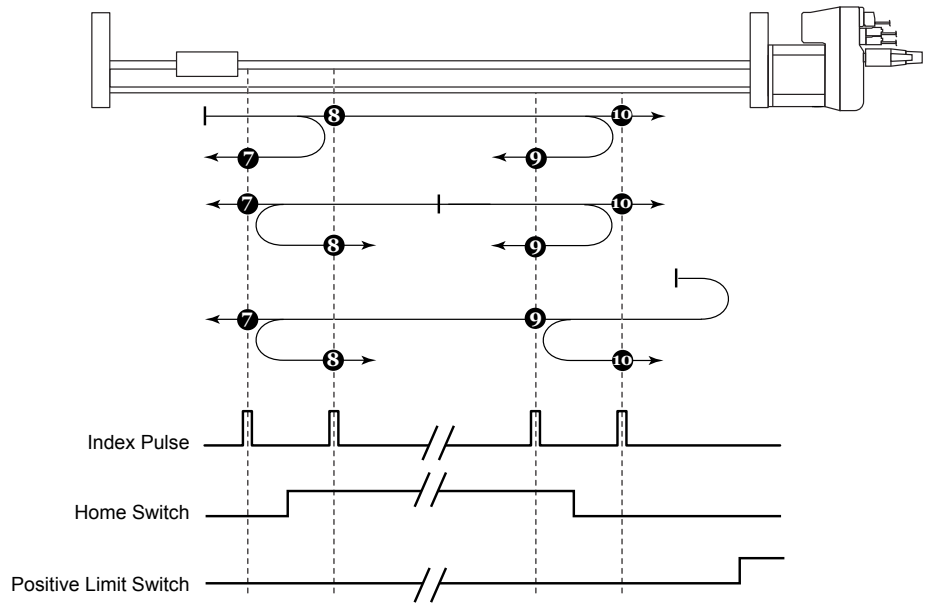
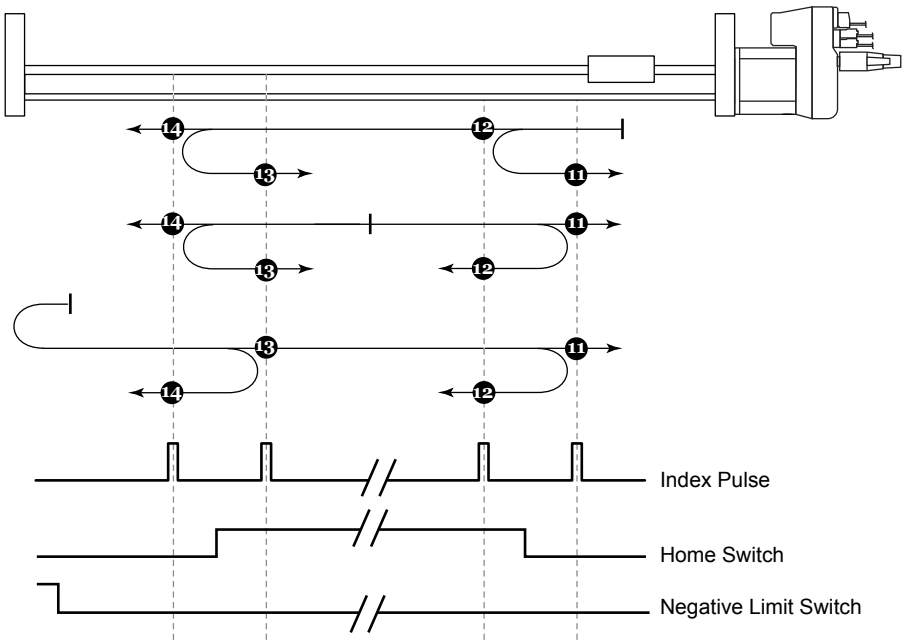


Figure 3.14 - Homing on the Home Switch and Index Pulse - Negative Initial Move



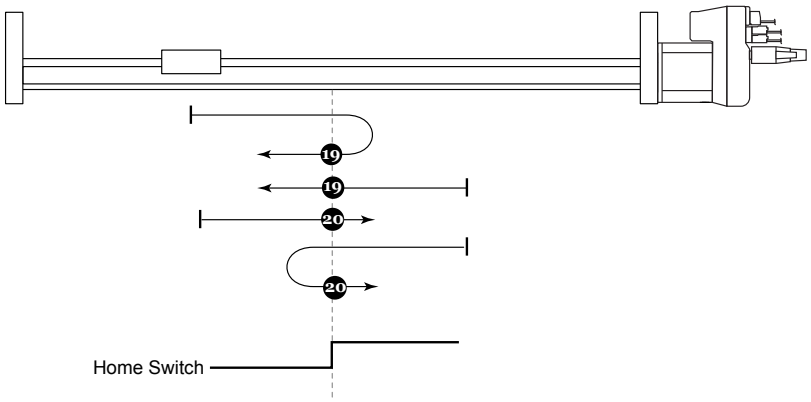
Methods 15 and 16: Reserved

These methods are reserved for future expansion of the homing mode.

Methods 17 to 30: Homing without an index pulse

These methods are similar to methods 1 to 14 except that the home position is not dependent on the index pulse but only Dependent on the relevant home or limit switch transitions. For example methods 19 and 20 are similar to methods 3 and 4 as shown in the following diagram.

Figure 3.15 - Homing Without an Index Pulse



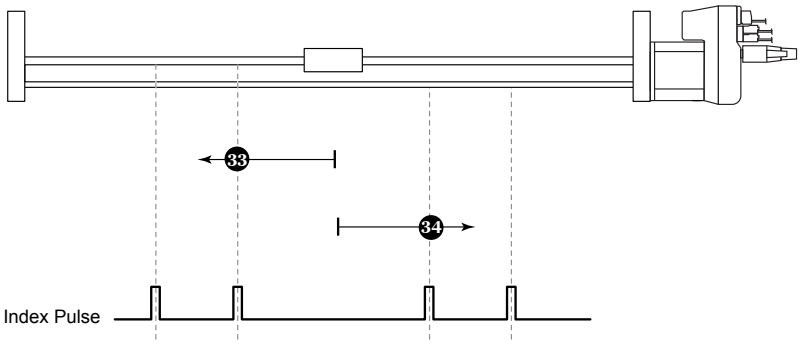
Methods 31 and 32: Reserved

These methods are reserved for future expansion of the homing mode.

Methods 33 and 34: Homing on an index pulse

Using methods 33 or 34, the direction of homing is negative or positive respectively. The home position is at the index pulse found in the selected direction.

Figure 3.16 - Homing on an Index Pulse



Method 35: Homing on the current position

In method 35 the current position is taken to be the home position.

6099_h Homing Speeds

The homing speeds object determines the fast and slow speeds that will be used during homing.

Table 3.72 - 6099_h Object Description

Index	6098_h
Name	Homing speeds
Object code	ARRAY
Data type	Unsigned32
Category	Optional

Table 3.73 - 6099_h Entry Description

Sub-index	00_h
Name	Number of entries
Access	ro
PDO mapping	—
Value range	—
Default value	02_h

Sub-index	01_h
Name	Homing speed fast
Access	rw
PDO mapping	—
Value range	0000 0000_h to 7FFF $FFFF_h$
Default value	102400_d

Sub-index	02_h
Name	Homing speed slow
Access	rw
PDO mapping	—
Value range	0000 0000_h to 7FFF $FFFF_h$
Default value	6400_d

Homing Mode Application Example

Homing Mode – demonstrates home method 18 decimal using Service Data Objects (SDOs).

Devices represented by this manual support the ability to move in homing mode.

The below example sets typical motion profile commands which a system would configure[1], enabling the motor power[2] and executing a homing function using SDOs with Node ID 41_h.

All values shown are hexadecimal.

1 Typical motion profile commands could be set each time on power up from host or set using a configuration file and stored to NVM once.
2 Enabling the motor power only has to be done once on power up.

Table 3.74 - Homing Mode Application Example

ID	RTR	Data String	Action
Typical motion parameters			
0641	00	2F 04 22 00 50 00 00 00	Set run current to 80%
0641	00	23 84 60 00 40 42 0F 00	Set deceleration to 1M steps/sec^2
0641	00	23 83 60 00 40 42 0F 00	Set acceleration to 1M steps/sec^2
0641	00	23 81 60 00 00 D0 07 00	Set max velocity to 512000 steps/sec
Enable motor power - CiA 402 state machine			
0641	00	2B 40 60 00 06 00 00 00	Ready to switch on
0641	00	2B 40 60 00 07 00 00 00	Switched on
0641	00	2B 40 60 00 0F 00 00 00	Operation enable
Set to homing mode			
0641	00	2F 60 60 00 06 00 00 00	Set to homing mode
Configure I/O and homing method			
0641	00	22 00 20 01 00 00 00 00	Set I/O as inputs
0641	00	22 00 20 02 00 00 00 00	Set I/O as sinking
0641	00	22 00 20 04 01 00 00 00	Set I1 as polarity
0641	00	22 02 20 01 01 00 00 00	Set I1 as home switch
0641	00	22 06 20 01 0A 00 00 00	Set I1 filter to 10ms
Set homing method, offset and speeds			
0641	00	22 98 60 00 13 00 00 00	Homing Method 18 decimal
0641	00	2F 98 20 00 01 00 00 00	Apply home offset to pos counter
0641	00	22 7C 60 00 00 00 00 00	Home offset = 0
0641	00	22 99 60 01 00 C8 00 00	Home speed fast
0641	00	22 99 60 02 00 14 00 00	Home speed slow
Start homing			
0641	00	2B 40 60 00 1F 00 00 00	Start homing
After home switch toggles			
0641	00	2B 40 60 00 00 00 00 00	Stop homing

Torque Mode (Closed Loop Models Only)

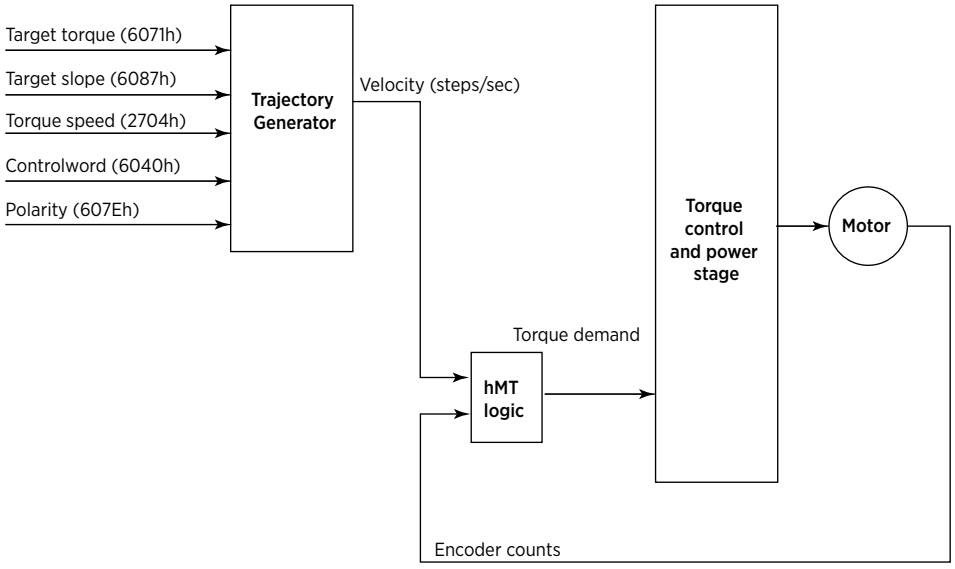
Overview

The profile torque mode allows the LMD to transmit the target torque value, which is processed via the trajectory generator and the hMTechnology logic.

The axis will ramp to the **TARGET_TORQUE (6071h)** as specified by the **TORQUE_RAMP (6087h).**

The manufacturer-specific **TORQUE_SPEED (2704h)** specifies the axis velocity at which the **TARGET_TORQUE** will be maintained.

Figure 3.17 - Profile Torque Mode



Controlword Definition for Profile Torque Mode

15 ... 9	8	7	6 ... 4	3 ... 0
See Table 3.4 on page 51	halt	See Table 3.4 on page 51	reserved	See Table 3.4 on page 51

Table 3.75 - Profile Torque Mode Control Word (6040h) Bit State Meanings

Bit	Value	Meaning
8	0	The motion is executed 8 or continued
	1	1 Axis is stopped according to the halt option code (605Dh)

Statusword Definition for Profile Torque Mode

15 ... 14	13	12 ... 11	10	9 ... 0
See Table 3.7 on page 52	reserved	See Table 3.7 on page 52	Target reached	See Table 3.7 on page 52

Table 3.76 - Profile Torque Mode Status Word (6041h) Bit State Meanings

Bit	Value	Meaning
10	0	Halt (Bit 8 in statusword) = 0: Target torque not reached Halt (Bit 8 in statusword) = 1: Axis decelerates
	1	Halt (Bit 8 in statusword) = 0: Target torque reached Halt (Bit 8 in statusword) = 1: Velocity of axis is 0

Torque Mode Objects

Object 6071_h: Target Torque

This object indicates the configured input value for the torque controller in profile torque mode. The value is given per thousand of rated torque.

Table 3.77 - 6071_h Object Description

Index	6071_h
Name	Target torque
Object code	VAR
Data type	Integer16
Category	Mandatory

Table 3.78 - 6071_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	-1000_d to 1000_d
Default value	0_d

Object 6077_h: Torque Actual Value

Provides the actual value of the torque. It corresponds to the instantaneous torque in the motor. The value is given per thousand of rated torque.

Table 3.79 - 6077_h Object Description

Index	6077_h
Name	Torque actual value
Object code	VAR
Data type	Integer16
Category	Optional

Table 3.80 - 6077_h Entry Description

Sub-index	00_h
Access	ro
PDO mapping	Yes
Value range	-1000_d to 1000_d
Default value	0_d

Object 6087_h: Torque Slope

This object indicates the configured rate of change of torque. The value is given in units of per thousand of rated torque per second.

Table 3.81 - 6087_h Object Description

Index	6087_h
Name	Torque slope
Object code	VAR
Data type	Unsigned32
Category	Mandatory

Table 3.82 - 6087_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	Yes
Value range	Unsigned32
Default value	1000_d

Profile Torque Example

This example will enter profile torque and set the target torque, speed, and torque ramp.

Table 3.83 - Profile Torque Application Example

ID	RTR	Data String	Action
0641	00	2F 60 60 00 04 00 00 00	Enable profile torque mode
0641	00	23 87 60 00 E8 03 00 00	Set torque ramp to 100%/sec
0641	00	2F 04 27 00 62 00 00 00	Set max torque speed to 120000 µsteps/sec
0641	00	2B 71 60 00 F4 01 00 00	Set torque to 50%

Cyclic Synchronous Position

Overview

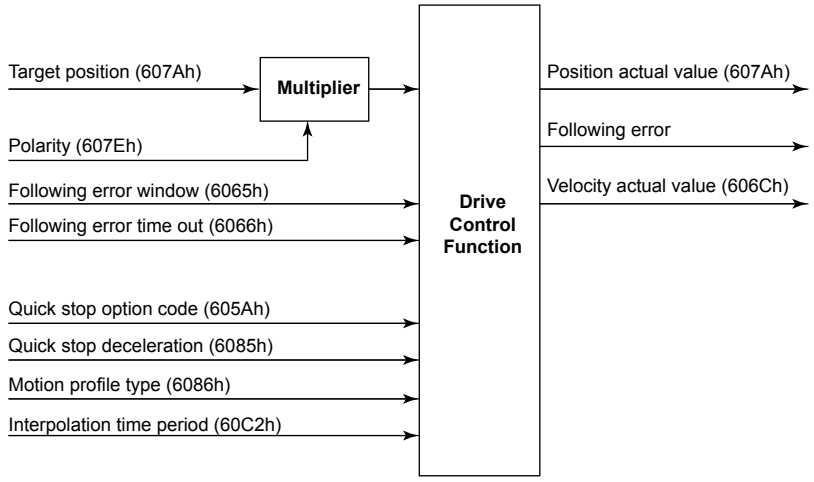
With this mode, the trajectory generator is located in the control device, not in the drive device. In a cyclic synchronous manner, it provides a target position to the drive device, which performs position and velocity control. Optionally, additive velocity values can be provided by the control system in order to allow for velocity feed-forward.

The behavior of the control function is influenced by control parameters, which are externally applied. The interpolation time period defines the time period between two updates of the target position and/or additive position and is used for inter-cycle interpolation.

The target position is interpreted as absolute value. The position actual value is used as mandatory output to the control device. Additionally, the velocity actual value is output.

The following error is used to set an error bit in the statusword (6041_h).

Figure 3.18 - Cyclic Synchronous Position Control Function



Controlword and Statusword for Cyclic Sync Position

The cyclic synchronous position mode uses no mode-specific bits of the controlword and three bits of the statusword for mode-specific purposes.

15 ... 14	13	12	11	10	9 0
Table 3.7 on page 52	following error	Table 3.7 on page 52	Target position ignored	reserved	Table 3.7 on page 52

Table 3.84 - Statusword Bits for Cyclic Synchronous Position Mode

Bit	Value	Meaning
10	0	reserved
	1	reserved
11	0	Target position ignored
	1	Target position is used as input to position control loop
13	0	No following error
	1	Following error

Position Control Function

Overview

In this section, all parameters are described which are necessary for a closed loop position control. The control loop is fed with the **`position_demand_value`** as one of the outputs of the trajectory generator and with the output of the position detection unit (**`position_actual_value`**), like a resolver or encoder, as input parameters.

6062_h Position Demand Actual Value

This object provides the demanded position value. The value is given in motor steps.

Table 3.85 - 6062_h Object Description

Index	6062_h
Name	Position demand actual value
Object code	VAR
Data type	Integer32
Category	Optional

Table 3.86 - 6062_h Entry Description

Sub-index	00_h
Access	ro
PDO mapping	Yes
Value range	Integer32
Default value	0000 0000_h

6063_h Position Actual Value Internal

This object provides the actual value of the position measurement device, which is one of the two input values of the closed-loop position control.

Table 3.87 - 6063_h Object Description

Index	6063_h
Name	Position actual value internal
Object code	VAR
Data type	Integer32
Category	Optional

Table 3.88 - 6063_h Entry Description

Sub-index	00_h
Access	ro
PDO mapping	Yes
Value range	Integer32
Default value	0000 0000_h

6064_h Position Actual Value

This object represents the actual value of the position measurement device in microsteps.

Table 3.89 - 6064_h Object Description

Index	6064_h
Name	Position actual value
Object code	VAR
Data type	Integer32
Category	Optional

Table 3.90 - 6064_h Entry Description

Sub-index	00_h
Access	ro
PDO mapping	Yes
Value range	Integer32
Default value	0000 0000_h

6065_h Following Error Window

This object indicates the proportional range of acceptable values to the position demand. A Following Error will occur when the difference between the position demand and position actual exceeds the Following Error Window value. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients. The value is given in user defined position units. If the value of the following error window is FFFF $FFFF_h$, the following control is switched off.

Table 3.91 - 6065_h Object Description

Index	6065_h
Name	Following error window
Object code	VAR
Data type	Unsigned32
Category	Optional

Table 3.92 - 6065_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	No
Value range	Unsigned32
Default value	0000 0512_h

6066_h Following Error Timeout

This object is the configured time for a Following Error condition, after that the bit 13 of the statusword is set to 1. The reaction of the drive when a following error occurs is manufacturer-specific. The value is given in milliseconds.

Table 3.93 - 6066_h Object Description

Index	6066_h
Name	Following error timeout
Object code	VAR
Data type	Unsigned16
Category	Optional

Table 3.94 - 6066_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	No
Value range	Unsigned16
Default value	0000^h

6067_h Position Window

This object indicates the configured proportional range of accepted positions relative to the target position. The target position is reached when the actual value of the position encoder is within the position window. The target position is handled in the same manner as in the trajectory generator. This limits functions and transforms into internal machine units before it may be used The value is given in user-defined position units. If the value of the position window is FFFF $FFFF_h$, the position window control is switched off.

Table 3.95 - 6067_h Object Description

Index	6067_h
Name	Position window
Object code	VAR
Data type	Unsigned32
Category	Optional

Table 3.96 - 6067_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	No
Value range	Unsigned32
Default value	0FFF $FFFF_h$

6068_h Position Window Time

This object indicates the configured time, during which the actual position within the position window is measured. The value is given in milliseconds.

Table 3.97 - 6066_h Object Description

Index	6066_h
Name	Position window time
Object code	VAR
Data type	Unsigned16
Category	Optional

Table 3.98 - 6066_h Entry Description

Sub-index	00_h
Access	rw
PDO mapping	No
Value range	Unsigned16
Default value	0000_h

Factors

$608F_h$ Position Encoder Resolution

This object defines the ratio of encoder increments per motor revolution:

$$\textit{position encoder resolution} = \frac{\textit{encoder increments (sub } 01_h\textit{)}}{\textit{motor revolutions (sub } 02_h\textit{)}}$$

The default values assume a 1000 line (4000 edges) encoder.

Table 3.99 - $608F_h$ Object Description

Index	$608F_h$
Name	Position encoder resolution
Object code	ARRAY
Data type	—
Category	Optional

Table 3.100 - $608F_h$ Entry Description

Sub-index	00_h
Name	Number of entries
Access	ro
PDO mapping	—
Value range	—
Default value	02_h

Sub-index	01_h
Name	Resolution numerator
Access	rw
PDO mapping	—
Value range	Unsigned32 (1 to 65535_d)
Default value	4000_d

Sub-index	02_h
Name	Resolution denominator
Access	rw
PDO mapping	—
Value range	Unsigned32 (1 to 65535_d)
Default value	1_d

6092_h Feed and Drive Shaft Resolution

This object defines the ratio of user units per shaft revolution:

$$\textit{feed and drive shaft resolution} = \frac{\textit{feed user units (sub } 01_h\textit{)}}{\textit{shaft revolutions (sub } 02_h\textit{)}}$$

The default values assume a 200 step/rev motor at a step resolution of 256, or 51200 steps/rev as the factor.

Table 3.101 - 6092_h Object Description

Index	6092_h
Name	Feed and drive shaft resolution
Object code	ARRAY
Data type	—
Category	Optional

Table 3.102 - 6092_h Entry Description

Sub-index	00_h
Name	Number of entries
Access	ro
PDO mapping	—
Value range	—
Default value	02_h

Sub-index	01_h
Name	Feed user units numerator
Access	ro
PDO mapping	—
Value range	Unsigned32
Default value	51200_d

Sub-index	02_h
Name	Shaft revolutions denominator
Access	ro
PDO mapping	—
Value range	Unsigned32
Default value	1_d

Optional Application FE (General I/O)

$60FD_h$ Digital Inputs

This object reads the digital inputs.

31 ... 24	23	22	21	20	19	18	17	16	15 ... 4	3	2	1	0
X	X	X	X	X	4	3	2	1	reserved	X	home switch	+ limit	– limit
MSb	Input points												LSb

Table 3.103 - $60FD_h$ Object Description

Index	$60FD_h$
Name	Digital inputs
Object code	VAR
Data type	Unsigned32
Category	Optional

Table 3.104 - $60FD_h$ Entry Description

Sub-index	00_h
Access	ro
PDO mapping	Yes
Value range	Unsigned32
Default value	$0000\ 0000_h$

$60FE_h$ Digital Outputs

This object reads the digital inputs.

31 ... 24	23	22	21	20	19	18	17	16	15 ... 1	0
X	X	X	X	X	X	3	2	1	reserved	brake
MSb	Output points									LSb

Table 3.105 - $60FE_h$ Object Description

Index	$60FE_h$
Name	Digital outputs
Object code	ARRAY
Data type	Unsigned32
Category	Optional

Table 3.106 - $60FE_h$ Entry Description

Sub-index	00_h
Name	Number of entries
Access	ro
PDO mapping	No
Value range	01_h
Default value	01_h

Sub-index	01_h
Name	Digital outputs
Access	rw
PDO mapping	Yes
Value range	Unsigned32
Default value	$0000\ 0000_h$

Appendix A
Diagnostics and Troubleshooting

Fieldbus Communication Error Diagnostics

A properly operating fieldbus is essential for evaluating operation and error messages.

Connections for Fieldbus Mode

If the product cannot be addressed via the fieldbus, first check the connections. The product manual contains the technical data of the device and information on network and device installation. Verify the following:

- Power connections to the device
- Fieldbus cable and fieldbus wiring

Baud Rate and Address

If it is possible to connect to a device, verify the baud rate and node address.

- The baud rate must be the same for all devices in the network.
- The node address of each device must be between 1 and 127 and unique for each device.

To set the baud rate and node address see Chapter 2, “Commissioning”.

Fieldbus Function Test

After correct configuration of the transmission data, test fieldbus mode. This requires installation of a CAN configuration tool that displays CAN messages. Feedback from the product is indicated in the form of a bootup message:

- Switch the power supply off and on again.
- Observe the network messages after switching on. After initialization of the bus, the device sends a boot-up message (COB ID 700_h + node ID, and 1 data byte with the content 00_h).
- With the factory setting 65 (41_h) for the node address, the boot-up message is sent via the bus. The device can then be put into operation via NMT services.

Error Diagnostics via Fieldbus

Message Objects

A number of objects provide information on the operating state and on errors:

- Object **`Statusword (6041h)`**, see “Operating States” on page 48
- Object **`EMCY (80h+ Node-ID)`** Error message from a device with error and error code.
- Object **`Error register (1001h)`** Error
- Object **`Error code (603Fh)`** Error code of the most recent error. See “Error Code ($603F_h$)” on page 54

Messages on Device Status

Synchronous and asynchronous errors are distinguished in terms of evaluation and handling of errors.

Synchronous Errors

The device signals a synchronous error directly as a response to a message that cannot be evaluated. Possible causes comprise transmission errors or invalid data. See “Error Register (1001_h)” below for a list of synchronous errors.

Asynchronous Errors

Asynchronous errors are signaled by the monitoring units in the device as soon as a device error occurs. An asynchronous error is signaled via bit 3, Error, of the object **`statusword (6041h)`**. In the case of errors that cause an interruption of the movement, the device transmits an EMCY message.

CANopen Error Messages

CANopen error messages are signaled in the form of EMCY messages. They are evaluated via the objects **`Error register (1001h)`** and **`Error code (603Fh)`**.

CANopen signals errors that occur during data exchange via SDO with the special SDO error message ABORT.

Error Register (1001_h)

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

Table A.1 - 1001_h Object Description

Index	1001_h
Name	Error register
Object code	VAR
Data type	Unsigned8
Category	Mandatory

Table A.2 - 1001_h Entry Description

Sub-index	00_h
Access	ro
PDO mapping	Optional
Value range	Unsigned8
Default value	—

Table A.3 - Abort Connection Option Code

Bit	M/O	Meaning
0	M	Generic Error
1	O	Current
2	O	Voltage
3	O	Temperature
4	O	Communication error (Overrun, Error State)
5	O	Device profile specific
6	O	Reserved (always 0)
7	O	Manufacturer specific

Pre-defined Error (1003_h)

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

1. The entry at sub-index 0 contains the number of actual errors that are recorded in the array starting at sub-index 1.
2. Every new error is stored at sub-index 1, the older ones move down the list.
3. Writing a “0“ to sub-index 0 deletes the entire error history (empties the array). Values higher than 0 lead to an abort message (error code: $0609\ 0030_h$) and should not to be written.
4. The error numbers are of type UNSIGNED32 and are composed of a 16 bit error code and a 16 bit additional error information field which is manufacturer specific. The error code is contained in the lower 2 bytes (LSb) and the additional information is included in the upper 2 bytes (MSb). If this object is supported, it must consist of the length entry on subindex 00_h and at least one error entry at sub-index 01_h.

Table A.4 - 1003_h Object Description

Index	1003_h
Name	Pre-defined error field
Object code	ARRAY
Data type	Unsigned32
Category	Optional

Table A.5 - 1003_h Entry Description

Sub-index	00_h
Description	Number of errors
Access	rw
Entry category	Mandatory
PDO mapping	No
Value range	0 – 254
Default value	0

Sub-index	01_h
Description	Standard error field
Access	ro
Entry category	Optional
PDO mapping	No
Value range	Unsigned32
Default value	—

Sub-index	02_h – FE_h
Description	Standard error field
Access	ro
Entry category	Optional
PDO mapping	No
Value range	Unsigned32
Default value	—

Table A.6 - Error Code Descriptions

Error description	Add'tl info byte	Error code byte
No error	0_h	0000_h
CAN overrun	0_h	8110_h
CAN in error passive mode	0_h	8120_h
Lifeguard or heartbeat error	0_h	8130_h
Recovered from "bus off" state	0_h	8140_h
Bus off state occurred	0_h	8141_h
PDO not processed - length error	0_h	8210_h
Over temperature error	8_h	4210_h
Pending over temperature warning	16_h	4210_h
Motor idle during commanded move	1_h	$FF01_h$
Motor should be idle	$2 – 8_h$	$FF01_h$
Undershot warning	9_h	$FF01_h$

Status LED

The LMD has two light-emitting diodes (LEDs) for status indication.

- LED 1: Status of the power supply
- LED 2: The CANopen status LED shows the states as specified in CiA DR-303-3, Indicator Specification.

Table A.7 - LED 1 Power Indication

Color	Status
Off	No Power
Green	+VDC supply in range
Flashing green	+VDC off, drive on AUX power
Red	+VDC supply out of range
Flashing red	+VDC off, AUX power out of range

Table A.8 - LED 2 CANopen Status

Color	Status
Red - single Flash	At least one of the error counters of the CAN controller has reached or exceeded the advisory level (too many error frames).
Red - double flash	A guard event (NMT-Slave or NMT-master) or a heartbeat event (Heartbeat consumer) has occurred.
Red - triple flash	The SYNC message has not been received within the configured communication cycle period time out (see Object Dictionary Entry 0x1006).
Red - on	The CAN controller bus is off
Green - single flash	The Device is in STOPPED state
Green - blinking	The Device is in the PREOPERATIONAL state
Green - on	The Device is in the OPERATIONAL state

Appendix B
Object Dictionary

Specification for the Objects

Index

The index specifies the position of the object in the object dictionary. The index value is specified as a hexadecimal value.

Object Code & Data Types

The object code specifies the data structure of the object.

Table B.1 - CANopen Object Codes

Object code	Meaning	Coding
VAR	A simple value, for example of the type Integer8 or Unsigned32	7
ARR (ARRAY)	A data field in which the entries have the same data type.	8
REC (RECORD)	A data field that contains entries that are a combination of simple data types.	9

Table B.2 - CANopen Data Types

Data type	Value range	Data length	CiA 301 coding
Boolean	0 = false, 1 = true	1 byte	0001
Integer8	–128 ... +127	1 byte	0002
Integer16	–32768 ... +32767	2 byte	0003
Integer32	–2147483648 ... +2147483647	4 byte	0004
Unsigned8	0 ... 255	1 byte	0005
Unsigned16	0 ... 65535	2 byte	0006
Unsigned32	0 ... 4294967295	4 byte	0007
Visible String8	ASCII chars	8 byte	0009
Visible String168	ASCII chars	16 byte	0010

RO/RW

Indicates read and/or write values
RO: values can only be read
RW: values can be read and written

PDO

R_PDO: Mapping for R_PDO possible
T_PDO: Mapping for T_PDO possible
No specification: PDO mapping not possible with the object

Min/max Values

Specifies the permissible range in which the object value is defined and valid.

Factory Default

Factory default settings when the product is shipped.

Overview of Object Group 1000_h

No objects from object group 1000_h are PDO mappable.

Table B.3 - Object Group 1000_h Overview

Index	Sub-index	Name	Obj. code	Data type	Access	Description
1000_h		Device type	VAR	Unsigned32	ro	Device type and profile
1001_h		Error register	VAR	Unsigned8	ro	Error register
1003_h		Predefined error field	ARR		rw	Error history, memory for error messages
	00_h	Number of errors	VAR	Unsigned8	rw	Number of error entries
	01_h - 04_h	Error field	VAR	Unsigned32	ro	Error number
1005_h		COB-ID SYNC	VAR	Unsigned32	rw	Identifier of the synchronization object
1007_h		Sync window length	VAR	Unsigned32	rw	Time window for synchronous PDOs in µS
1008_h		Mfg. device name	VAR	Vis String8	ro	Manufacturer's designation
1009_h		Mfg. hardware version	VAR	Vis String8	ro	Hardware version
$100A_h$		Mfg. software version	VAR	Vis String8	ro	Software version
$100C_h$		Guard time	VAR	Unsigned16	rw	Time span for Node Guarding [ms]
$100D_h$		Life time factor	VAR	Unsigned8	rw	Repeat factor for Node Guarding
1010_h		Store parameters	ARR	Unsigned32		Store parameters
	00_h	Largest sub-index	VAR	Unsigned8	ro	Largest sub-index supported» 04_h
	01_h	Save all parameters	VAR	Unsigned32	rw	Save all parameters
	02_h	Save communication	VAR	Unsigned32	rw	Save Communication Parameters
	03_h	Save application	VAR	Unsigned32	rw	Save Application Parameters
	04_h	Save manufacturer	VAR	Unsigned32	rw	Save Manufacturer Parameters
1011_h		Restore defaults	ARR	Unsigned32		Restore defaults as group
	00_h	Largest sub-index	VAR	Unsigned8	ro	Largest sub-index supported» 04_h
	01_h	Restore all defaults	VAR	Unsigned32	rw	Restore all defaults
	02_h	Restore communication	VAR	Unsigned32	rw	Restore Communication defaults
	03_h	Restore application	VAR	Unsigned32	rw	Restore Application defaults
	04_h	Restore manufacturer	VAR	Unsigned32	rw	Restore Manufacturer defaults
1012_h		COB-ID time stamp	VAR	Unsigned32	rw	COB-ID time stamp message
1014_h		COB-ID EMCY	VAR	Unsigned32	rw	80_h + Node ID
1015_h		Inhibit time EMCY	VAR	Unsigned16	rw	Wait time for the repeated transmission of EMCY x 100 µS
1017_h		Producer Heartbeat Time	VAR	Unsigned16	rw	Time interval for producer «Heartbeat»
1018_h		Identity	REC		ro	Identification object
	00_h	Largest sub-index	VAR	Unsigned8	ro	Largest sub-index supported» 04_h
	01_h	Vendor ID	VAR	Unsigned32	ro	Vendor ID
	02_h	Product code	VAR	Unsigned32	ro	Product code
	03_h	Revision number	VAR	Unsigned32	ro	Revision number
	04_h	Serial number	VAR	Unsigned32	ro	Serial number

Index	Sub-index	Name	Obj. code	Data type	Access	Description
1400_h		1st R_PDO parameter	REC			1st receive PDO parameter
	00_h	Largest sub-index	VAR	Unsigned8	ro	Largest sub-index supported» 05_h
	01_h	COB-ID used	VAR	Unsigned32	rw	COB-ID used: 200_h + Node ID
	02_h	Transmission type	VAR	Unsigned8	rw	Default type = 255 (asynchronous)
	03_h	Inhibit time	VAR	Unsigned16	rw	Default = 0
	05_h	Event timer	VAR	Unsigned16	rw	Default = 0
1401_h		2nd R_PDO parameter	REC			2nd receive PDO parameter
	00_h	Largest sub-index	VAR	Unsigned8	ro	Largest sub-index supported» 05_h
	01_h	COB-ID used	VAR	Unsigned32	rw	COB-ID used: 300_h + Node ID
	02_h	Transmission type	VAR	Unsigned8	rw	Default type = 255 (asynchronous)
	03_h	Inhibit time	VAR	Unsigned16	rw	Default = 0
	05_h	Event timer	VAR	Unsigned16	rw	Default = 0
1402_h		3rd R_PDO parameter	REC			3rd receive PDO parameter
	00_h	Largest sub-index	VAR	Unsigned8	ro	Largest sub-index supported» 05_h
	01_h	COB-ID used	VAR	Unsigned32	rw	COB-ID used: 400_h + Node ID
	02_h	Transmission type	VAR	Unsigned8	rw	Default type = 255 (asynchronous)
	03_h	Inhibit time	VAR	Unsigned16	rw	Default = 0
	05_h	Event timer	VAR	Unsigned16	rw	Default = 0
1403_h		4th R_PDO parameter	REC			4th receive PDO parameter
	00_h	Largest sub-index	VAR	Unsigned8	ro	Largest sub-index supported» 05_h
	01_h	COB-ID used	VAR	Unsigned32	rw	COB-ID used: 500_h + Node ID
	02_h	Transmission type	VAR	Unsigned8	rw	Default type = 255 (asynchronous)
	03_h	Inhibit time	VAR	Unsigned16	rw	Default = 0
	05_h	Event timer	VAR	Unsigned16	rw	Default = 0
1600_h		1st R_PDO mapping	REC		ro	PDO mapping for R_PDO1, settings
	00_h	# of mapped objects	VAR	Unsigned8	rw	Number of mapped objects, range 1 – 64
	01_h - 08_h	Application Objects	VAR	Unsigned32	rw	R_PDO1 mapping application objects
1601_h		2nd R_PDO mapping	REC		ro	PDO mapping for R_PDO2, settings
	00_h	# of mapped objects	VAR	Unsigned8	rw	Number of mapped objects, range 1 – 64
	01_h - 08_h	Application Objects	VAR	Unsigned32	rw	R_PDO2 mapping application objects
1602_h		3rd R_PDO mapping	REC		ro	PDO mapping for R_PDO3, settings
	00_h	# of mapped objects	VAR	Unsigned8	rw	Number of mapped objects, range 1 – 64
	01_h - 08_h	Application Objects	VAR	Unsigned32	rw	R_PDO3 mapping application objects
1603_h		4th R_PDO mapping	REC		ro	PDO mapping for R_PDO4, settings
	00_h	# of mapped objects	VAR	Unsigned8	rw	Number of mapped objects, range 1 – 64
	01_h - 08_h	Application Objects	VAR	Unsigned32	rw	R_PDO4 mapping application objects
1800_h		1st T_PDO parameter	REC			1st transmit PDO parameter
	00_h	Largest sub-index	VAR	Unsigned8	ro	Largest sub-index supported» 05_h
	01_h	COB-ID used	VAR	Unsigned32	rw	COB-ID used: 180_h + Node ID
	02_h	Transmission type	VAR	Unsigned8	rw	Default type = 255 (asynchronous)
	03_h	Inhibit time	VAR	Unsigned16	rw	Default = 0
	05_h	Event timer	VAR	Unsigned16	rw	Default = 0
1801_h		2nd T_PDO parameter	REC			2nd transmit PDO parameter
	00_h	Largest sub-index	VAR	Unsigned8	ro	Largest sub-index supported» 05_h
	01_h	COB-ID used	VAR	Unsigned32	rw	COB-ID used: 280_h + Node ID
	02_h	Transmission type	VAR	Unsigned8	rw	Default type = 255 (asynchronous)
	03_h	Inhibit time	VAR	Unsigned16	rw	Default = 0
	05_h	Event timer	VAR	Unsigned16	rw	Default = 0

Index	Sub-index	Name	Obj. code	Data type	Access	Description
1802$_h$		3rd T_PDO parameter	REC			3rd transmit PDO parameter
	00_h	Largest sub-index	VAR	Unsigned8	ro	Largest sub-index supported» 05_h
	01_h	COB-ID used	VAR	Unsigned32	rw	COB-ID used: 380_h + Node ID
	02_h	Transmission type	VAR	Unsigned8	rw	Default type = 255 (asynchronous)
	03_h	Inhibit time	VAR	Unsigned16	rw	Default = 0
	05_h	Event timer	VAR	Unsigned16	rw	Default = 0
1803$_h$		4th T_PDO parameter	REC			4th transmit PDO parameter
	00_h	Largest sub-index	VAR	Unsigned8	ro	Largest sub-index supported» 05_h
	01_h	COB-ID used	VAR	Unsigned32	rw	COB-ID used: 480_h + Node ID
	02_h	Transmission type	VAR	Unsigned8	rw	Default type = 255 (asynchronous)
	03_h	Inhibit time	VAR	Unsigned16	rw	Default = 0
	05_h	Event timer	VAR	Unsigned16	rw	Default = 0
1A00$_h$		1st T_PDO mapping	REC		ro	PDO mapping for T_PDO1, settings
	00_h	# of mapped objects	VAR	Unsigned8	rw	Number of mapped objects, range 1 – 64
	01_h - 08_h	Application Objects	VAR	Unsigned32	rw	T_PDO1 mapping application objects
1A01$_h$		2nd T_PDO mapping	REC		ro	PDO mapping for T_PDO2, settings
	00_h	# of mapped objects	VAR	Unsigned8	rw	Number of mapped objects, range 1 – 64
	01_h - 08_h	Application Objects	VAR	Unsigned32	rw	T_PDO2 mapping application objects
1A02$_h$		3rd T_PDO mapping	REC		ro	PDO mapping for T_PDO3, settings
	00_h	# of mapped objects	VAR	Unsigned8	rw	Number of mapped objects, range 1 – 64
	01_h - 08_h	Application Objects	VAR	Unsigned32	rw	T_PDO3 mapping application objects
1A03$_h$		4th T_PDO mapping	REC		ro	PDO mapping for T_PDO4, settings
	00_h	# of mapped objects	VAR	Unsigned8	rw	Number of mapped objects, range 1 – 64
	01_h - 08_h	Application Objects	VAR	Unsigned32	rw	T_PDO4 mapping application objects

Overview of Manufacturer Specific Objects Group 2000_h

Table B.4 - Object Group 2000_h Overview

Index	Sub	Name	Obj. code	Data type	Access	PDO	Description
2000$_h$		Configure GPIO	ARR				Configure the general purpose I/O points
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 05_h
	01_h	Reserved	VAR	Unsigned8	rw		Configure the I/O point as an output or input
	02_h	Reserved	VAR	Unsigned8	rw		Configure the I/O point as sourcing or sinking
	03_h	Reserved	VAR	Unsigned8	rw		Configure as both sink/source or source only
	04_h	Configure polarity in	VAR	Unsigned8	rw		Configure input logic polarity
	05_h	Reserved	VAR	Unsigned8	rw		Configure output logic polarity
2002$_h$		Configure digital inputs	ARR				Configure the functions of inputs
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 04_h
	01_h	Define as Home switch	VAR	Unsigned8	rw		Configure input as homing switch
	02_h	Define as positive limit	VAR	Unsigned8	rw		Configure input as positive limit switch
	03_h	Define as negative limit	VAR	Unsigned8	rw		Configure input as negative limit switch
	04_h	Define as inhibit switch	VAR	Unsigned8	rw		Configure input as inhibit switch (see 2007_h)
2004$_h$		Input mask	ARR				Configure input filter mask
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 08_h
	01_h - 08_h	Input filter mask	VAR	Unsigned8	rw		Defines inputs to apply filtering (see 2006_h)
2006$_h$		Input filter time	ARR				Configure input filter time
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 08_h
	01_h - 08_h	Input filter time	VAR	Unsigned8	rw		Defines inputs filter time in ms

Index	Sub	Name	Obj. code	Data type	Access	PDO	Description
2007_h		Inhibit switch	ARR				Configure inhibit switch
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 01_h
	01_h	Input switch action	VAR	Unsigned8	rw		Defines the action of the inhibit switch
2008_h		Configure digital outputs	ARR				Configure output functions
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 02_h
	01_h	Brake output defined	VAR	Unsigned8	rw		Defines the output(s) used for braking
	02_h	Target reached output	VAR	Unsigned8	rw		Defines the output used to indicate target reached
2010_h		Analog input configuration	ARR				Configure the analog input
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 03_h
	01_h	Analog input read value	VAR	Unsigned16	ro	T_PDO	Analog input value
	02_h	Analog input scale	VAR	Unsigned8	rw		Sets the mode as 0 - 5V, 0 - 10V or 0 - 20mA
	03_h	Analog input filter	VAR	Unsigned8	rw		Defines the filtering for the analog input
2014_h		Aux-power monitoring	ARR				Monitor the level of the Aux-power input
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 03_h
	01_h	Aux-power voltage	VAR	Unsigned8	ro	T_PDO	Read Aux-power voltage value
	02_h	Aux-power low advisory	VAR	Unsigned8	ro		Read Aux-power low level advisory
	03_h	Aux-power high advisory	VAR	Unsigned8	ro		Read Aux-power high level advisory
2015_h		Input voltage monitoring	ARR				Monitors the level of the +VDC input voltage
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 03_h
	01_h	Input +V	VAR	Unsigned8	ro	T_PDO	Read Input +V voltage value
	02_h	Input +V low advisory	VAR	Unsigned8	ro		Read Input +V low level advisory
	03_h	Input +V high advisory	VAR	Unsigned8	ro		Read Input +V high level advisory
2016_h		Abs. Encoder Back-up Voltage	ARR				
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 03_h
	01_h	Voltage level	VAR	Unsigned16	ro	T_PDO	Read the voltage of the abs. encoder battery backup input
	02_h	Low-level advisory	VAR	Unsigned16	rw		Set low-level advisory threshold (abs. encoder battery back-up)
	03_h	High-level advisory	VAR	Unsigned16	rw		Set high-level advisory threshold (abs. encoder battery back-up)
2017_h		Abs. encoder internal voltage	VAR	Unsigned16	rw		Read the level of the internal back-up voltage (abs. encoder models only)
2018_h		Board Temperature	ARR				Set the board temperature parameters
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 03_h
	01_h	Board temperature	VAR	Integer8	ro	T_PDO	Board temperature reading
	02_h	Set temperature advisory	VAR	Integer8	rw		Set board temperature advisory threshold
	03_h	Set temperature error	VAR	Integer8	rw		Set board temperature error threshold
2019_h		H-Bridge Temperature	ARR				Set the bridge temperature parameters
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 03_h
	01_h	H-Bridge temperature	VAR	Integer8	ro	T_PDO	Bridge temperature reading
	02_h	Set temperature advisory	VAR	Integer8	rw		Set bridge temperature advisory threshold
	03_h	Set temperature error	VAR	Integer8	rw		Set board temperature error threshold
2020_h		Set temperature error	VAR	Integer8	rw		Set bridge temperature error threshold
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 02_h
	01_h	Limit reached flag	VAR	Unsigned8	rw		Indicates type of limit reached
	02_h	Limit reached mask	VAR	Unsigned8	rw		Defines limits to act upon
2022_h		Software limit	ARR				Actual position software limit
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 02_h
	01_h	Actual negative limit	VAR	Interger32	rw		Actual negative limit
	02_h	Actual positive limit	VAR	Interger32	rw		Actual positive limit
2030_h		Output h-bridge polarity	VAR	Integer8	rw		Defines the polarity of the output bridge
2031_h		Unit options	VAR	Unsigned8	rw		Enable encoder, capture/trip functions

Index	Sub	Name	Obj. code	Data type	Access	PDO	Description
2033_h		Capture input parameters	REC				Capture input parameters
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 04_h
	01_h	Capture input control	VAR	Unsigned8	rw		Enables the capture input
	02_h	Capture input flag	VAR	Unsigned8	rw		Displays the status of a position capture
	03_h	Capture input filter	VAR	Unsigned8	rw		Sets the filtering for the capture input
	04_h	Captured position	VAR	Integer32	ro	T_PDO	Stores the captured position
2034_h		Bridge on settle time	ARR				Settling time after bridge power on
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 01_h
	01_h	Bridge settle time	VAR	Unsigned8	rw		Bridge settling time in ms
2035_h		Brake settle allow time					Settling time after brake on/off
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 02_h
	01_h	Brake on settle time	VAR	Unsigned8	rw		Brake on settling time in ms
	02_h	Brake off settle time	VAR	Unsigned8	rw		Brake off settling time in ms
2036_h		Hold current delay time	VAR	Unsigned16	rw		Defines time in ms to transition to hold current following cessation of motion
2037_h		Bridge on to encoder settle time	VAR	Unsigned16	rw		Time between switching into operation enable to resynching the encoder position
2038_h		Trip output configuration	REC				Trip output parameters
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 03_h
	01_h	Trip output control	VAR	Unsigned16	rw		Controls logic and trip points
	02_h	1st position of a series	VAR	Integer32	rw		First trip position
	03_h	Multiple trip point spacing	VAR	Integer32	rw		Defines the spacing between subsequent trip points
2098_h		Homing configuration	VAR	Unsigned8	rw		Defines the counter status following a home
2099_h		Index offset	VAR	Unsigned32	rw		Defines the offset of the index in microsteps
2204_h		Run current	VAR	Unsigned8	rw		Sets the motor run current percent
2205_h		Hold current	VAR	Unsigned8	rw		Sets the motor hold current percent
2211_h		Position present point target	VAR	Integer32	ro		Position present point target
2212_h		Position final point target	VAR	Integer32	ro		Position final point target
2221_h		Following Error	ARR				Following Error reaction code
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 01_h
	01_h	Reaction code	VAR	Unsigned8	rw		Following error reaction code
2231_h		Encoder following mode	ARR				Allows LMD to follow an external encoder input
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 06_h
	01_h	Direction of rotation	VAR	Unsigned8	rw		Direction of rotation for encoder following
	02_h	Sample rate	VAR	Unsigned16	rw		Sets the sample rate of the input for encoder following
	03_h	Target reached delay time	VAR	Unsigned16	rw		Sets delay before triggering target reached.
	04_h	Minimum Threshold	VAR	Signed16	rw		Sets counts required to activate x 1 gain
	05_h	Minimum Move	VAR	Signed16	rw		Sets the minimum move distance
	06_h	Maximum Move	VAR	Signed16	rw		Sets the maximum move distance
	07_h	Gain	VAR	Signed16	rw		Gain occurring between min. and max. distance
	08_h	Threshold spacing	VAR	Signed16	rw		Gain increase or decrease
2401_h		Gen Purpose user variable	VAR	Unsigned8	rw		May be used to store 8 bits of data
2402_h		Gen Purpose user variable	ARR	Unsigned32	rw		Set of user 32-bit user variables
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 04_h
	01_h	Gen Purpose user variable	VAR	Unsigned32	rw		General purpose 32-bit variable
	02_h	Gen Purpose user variable	VAR	Unsigned32	rw		General purpose 32-bit variable
	03_h	Gen Purpose user variable	VAR	Unsigned32	rw		General purpose 32-bit variable
2701_h		hMTechnology enable	VAR	Unsigned8	rw		Enable/disable hMT control (closed loop only)
2702_h		Make up mode	VAR	Unsigned8	rw		Defines hMT make up mode (closed loop only)
2703_h		Make up velocity	VAR	Unsigned32	rw		Defines the velocity for make up (closed loop only)
2704_h		Torque Speed	VAR	Unsigned8	rw		Defines the velocity for torque mode (closed loop only)

Index	Sub	Name	Obj. code	Data type	Access	PDO	Description
2708_h		Velocity Actual Filter	VAR	Unsigned8	rw		
2710_h		Locked Rotor Timeout	VAR	Unsigned8	rw		Defines the time period for a locked rotor condition to assert. (closed loop only)
2711_h		Locked Rotor Opcode	VAR	Integrer16			Defines the response to a locked rotor condition (closed loop only)
2712_h		Following Error Opcode	VAR	Integrer16			Defines the response to a following error (closed loop only)
2740_h		hMTechnology control	VAR	Unsigned8	rw	R_PDO	Defines the control parameters for hMTechnology. (closed loop only)
2741_h		hMTechnology status (filtered)	VAR	Unsigned8	ro	T_PDO	Reads the hMT status as filtered by 2743_h (closed loop only)
2742_h		hMTechnology status (unfiltered)	VAR	Unsigned8	ro		Reads the hMT status unfiltered (closed loop only)
2743_h		hMTechnology status filter	VAR	Unsigned8	rw		Determines which bits of the status byte are filtered in 2741_h
2840_h		Multi-turn Control Byte	VAR	Integrer16	rw		
2841_h		Multi-turn Status Byte	VAR	Integrer16	ro		

Overview of Assignment Objects Group 6000_h

Table B.5 - Object Group 6000_h Overview

Index	Sub	Name	Obj. code	Data type	Access	PDO	Description
6007_h		Abort connection opcode	VAR	Integer16	rw		Controls the process for abort connection
$603F_h$		Error code	VAR	Unsigned16	ro	T_PDO	Stores the last error
6040_h		Control word	VAR	Unsigned16	rw	R_PDO	Control word
6041_h		Status word	VAR	Unsigned16	ro	T_PDO	Status word
$605A_h$		Quick stop option code	VAR	Integer16	rw		Defines the method for quick stop
$605B_h$		Shutdown option code	VAR	Integer16	rw		Defines the method for shutdown
$605C_h$		Disable operation opcode	VAR	Integer16	rw		Defines the method for disable operation
$605D_h$		Halt operation opcode	VAR	Integer16	rw		Defines the method for halt operation
$605E_h$		Error reaction opcode	VAR	Integer16	rw		Defines the reaction to a error
6060_h		Modes of operation	VAR	Integer8	rw	R_PDO	Set the mode of operation
6061_h		Modes of operation display	VAR	Integer8	ro	T_PDO	Read the mode of operation
6062_h		Position demand value	VAR	Integer32	ro	T_PDO	Read the motor position in user units
6063_h		Position actual value	VAR	Integer32	ro	T_PDO	Read the motor position
6064_h		Position actual value	VAR	Integer32	ro	T_PDO	Read the motor position
6065_h		Following error window	VAR	Unsigned32	rw		Defines range of tolerated positions symmetrical to 6062_h
6066_h		Following error window time	VAR	Unsigned16	rw		Defines the timeout for the next error window
6067_h		Position window	VAR	Unsigned32	rw		Defines accepted positions relative to target
6068_h		Position window timeout	VAR	Unsigned16	rw		Defines time to indicate target reached
$606C_h$		Velocity actual value	VAR	Integer32	ro	T_PDO	Actual velocity of the motor
6071_h		Target torque	VAR	Integer16	rw	R_PDO	Defines the target torque for Profile Torque mode
6077_h		Torque actual value	VAR	Integer16	ro	T_PDO	Actual torque value for Profile Torque
$607A_h$		Profiled target position	VAR	Integer32	rw	R_PDO	Defines target position for absolute or relative move
$607C_h$		Homing offset	VAR	Integer32	rw		Defines offset from homing zero position
$607E_h$		Polarity	VAR	Unsigned8	rw	R_PDO	Sets polarity for position/speed commands
6081_h		Profile velocity	VAR	Unsigned32	rw	R_PDO	Sets the velocity for the profile position motion
6082_h		Initial velocity	VAR	Unsigned32	rw	R_PDO	Sets the terminal (max) velocity
6083_h		Profile acceleration	VAR	Unsigned32	rw	R_PDO	Sets the acceleration for profile position and profile velocity motion
6084_h		Profile deceleration	VAR	Unsigned32	rw	R_PDO	Sets the deceleration for profile position and profile velocity motion
6085_h		Quick stop deceleration	VAR	Unsigned32	rw		Sets the deceleration for a quick stop state

Index	Sub	Name	Obj. code	Data type	Access	PDO	Description
6086_h		Motion profile type	VAR	Integer16	rw	R_PDO	Defines method by which profile motion is evaluated
6087_h		Torque slope	VAR	Unsigned32	rw		Defines the torque ramp for Profile Torque mode
$608F_h$		Position encoder resolution	ARR				Defines relation between motor revolution and position increments
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 02_h
	01_h	Resolution numerator	VAR	Unsigned32	rw		# of encoder increments
	02_h	Resolution denominator	VAR	Unsigned32	rw		# of motor revolutions
6092_h		Factor group feed and driveshaft	ARR				Defines relation between feed user units and drive shaft revolutions
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 02_h
	01_h	Feed numerator	VAR	Unsigned32	rw		# of feed increments
	02_h	Driveshaft denominator	VAR	Unsigned32	rw		# of driveshaft revolutions
6098_h		Homing method	VAR	Integer8	rw		Defines the method for homing operation
6099_h		Homing speed	ARR				Defines the high and low speeds for homing
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 02_h
	01_h	Homing speed fast	VAR	Unsigned32	rw		Defines the high speed for homing
	02_h	Homing speed slow	VAR	Unsigned32	rw		Defines the low speed for homing
$60C2_h$		Interpolated position time period	REC				Defines time for interpolation position trajectory.
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 02_h
	01_h	Interpolation period	VAR	Unsigned8	rw		Interpolation time period
	02_h	Interpolation factor	VAR	Integer8	rw		Interpolation factor
$60F8_h$		Maximum slippage	VAR	Integer32	rw		Maximum slippage
$60FD_h$		Digital inputs	VAR	Unsigned32	ro	T_PDO	Reads the state of digital inputs
$60FE_h$		Digital outputs	ARR				Sets the state of digital outputs
	00_h	Number of entries	VAR	Unsigned8	ro		Number of entries = 01_h
	01_h	Digital outputs	VAR	Unsigned32	rw	R_PDO	Reads the state of digital inputs
$60FF_h$		Target velocity	VAR	Integer32	rw	R_PDO	Defines the target velocity
6402_h		Motor types	VAR	Unsigned16	ro		Motor type = 9: Stepper motor
6502_h		Supported drive modes	VAR	Unsigned32	ro		Profile position, profile velocity, homing

Details of Object Group 1000_h

1000_h Device Type

The object specifies the device profile used as well as the device type.

Table B.6 - 1000_h Object Description

Index	1000_h
Name	Device type
Object code	VAR
Data type	Unsigned32

Table B.7 - 1000_h Value Description

Sub-index	00_h, device type
Meaning	Device type and profile
Access	Read only
PDO mapping	—
Value range	—
Default value	0044 0192_h
Category	—

Table B.8 - Bit Coding Sub-index 00_h

Bit	Access	Value	Meaning
31-16	ro	0044_h	Stepper motor
15-0	ro	0192_h	Device profile CiA 402

1001_h Error Register

The object specifies the error of the device. The detailed cause of error can be determined with the object **`predefined error field (1003h)`** and, for reasons of compatibility with devices with other fieldbus profiles, with the object **`error code (603Fh)`**.

Errors are signaled by an EMCY message as soon as they occur.

Table B.9 - 1001_h Object Description

Index	1001_h
Name	Error register
Object code	VAR
Data type	Unsigned8

Table B.10 - 1001_h Value Description

Sub-index	00_h, error register
Meaning	Error register
Access	Read only
PDO mapping	—
Value range	—
Default value	—
Category	—

Table B.11 - Bit Coding Sub-index 00_h

Bit	Access	Value	Meaning
0	ro	—	Error (generic error)
1	ro	—	Reserved
2	ro	—	Reserved
3	ro	—	Temperature
4	ro	—	Communication profile (communication error)
5	ro	—	Reserved
6	ro	—	Reserved
7	ro	—	Manufacturer specific

1003_h Pre-defined Error Field

The object contains the latest error messages that were shown as EMCY messages.

- The sub-index 00_h entry contains the number of saved error messages.
- The current error message is stored at sub-index 01_h, older messages are moved to higher sub-index entries.
- Writing 0 to sub-index 00_h resets the error list.

Table B.12 - 1003_h Object Description

Index	1003_h
Name	Pre-defined error field
Object code	ARRAY
Data type	Unsigned32

Table B.13 - 1003_h Value Description

Sub-index	00_h, number of errors
Meaning	Number of error entries
Access	Read-write
PDO mapping	—
Value range	0 ... 4
Default value	0
Category	—

Sub-index	01_h – 04_h, error field
Meaning	Error number
Access	Read only
PDO mapping	—
Value range	—
Default value	0
Category	—

Bit Coding Sub-index 00_h ... 04_h

Bytes 0 through 15 are dedicated for error codes.
Bytes 16 through 31 are dedicated for additional error information, not assigned in the device.

1005_h COB ID SYNC Message

The object specifies the COB ID of the SYNC object.

The device can only receive SYNC messages.

The COB ID can be changed in the NMT state “Pre-Operational”.

Table B.14 - 1005_h Object Description

Index	1005_h
Name	COB ID SYNC
Object code	VAR
Data type	Unsigned32

Table B.15 - 1005_h Value Description

Sub-index	00_h, COB ID SYNC
Meaning	Identifier of the synchronization object
Access	Read-write
PDO mapping	—
Value range	0...4294967295
Default value	0000 0080_h
Category	Yes

1007_h Sync Window Length

Contains the length of the time window for synchronous PDOs in microseconds.

Table B.16 - 1007_h Object Description

Index	1007_h
Name	Sync window length
Object code	VAR
Data type	Unsigned32

Table B.17 - 1007_h Value Description

Sub-index	00_h, Sync window length
Meaning	Timing for sync PDOs
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	0000 0000_h
Category	Yes

1008_h Mfg. Device Name

Provides the name of the device as given by the manufacturer.

Table B.18 - 1008_h Object Description

Index	1008_h
Name	Manufacturer device name
Object code	VAR
Data type	Visible String

Table B.19 - 1008_h Value Description

Sub-index	00_h, Manufacturer device name
Meaning	Manufacturer device name
Access	Read only
PDO mapping	—
Value range	Visible String
Default value	CANopen LMD Motion Control Node
Category	—

1009_h Mfg. Hardware Version

Provides the hardware version of the device as given by the manufacturer.

Table B.20 - 1009_h Object Description

Index	1009_h
Name	Manufacturer hardware version
Object code	VAR
Data type	Visible String

Table B.21 - 1009_h Value Description

Sub-index	00_h, Manufacturer hardware version
Meaning	Manufacturer hardware version
Access	Read only
PDO mapping	—
Value range	Visible String
Default value	V1.00
Category	—

$100A_h$ Mfg. Software Version

Provides the software version of the device as given by the manufacturer.

Table B.22 - $100A_h$ Object Description

Index	$100A_h$
Name	Manufacturer software version
Object code	VAR
Data type	Visible String

Table B.23 - $100A_h$ Value Description

Sub-index	00_h, Manufacturer software version
Meaning	Manufacturer software version
Access	Read only
PDO mapping	—
Value range	Visible String
Default value	V5.48
Category	—

$100C_h$ Guard Time

The object specifies the time span for connection monitoring (Node Guarding) of an NMT slave.

The time span for connection monitoring of an NMT master results from the time span “guard time” multiplied by the factor “life time”, object **`Life time factor(100Dh)`** .

The time span can be changed in the NMT state “Pre-Operational”.

Table B.24 - $100C_h$ Object Description

Index	$100C_h$
Name	Guard time
Object code	VAR
Data type	Unsigned16

Table B.25 - $100C_h$ Value Description

Sub-index	00_h, Guard time
Meaning	Guard time
Access	Read-write
PDO mapping	—
Value range	0...65535
Default value	0000_h
Category	Yes

$100D_h$ Life Time Factor

The object specifies the factor that, together with the time span “guard time”, results in the time interval for connection monitoring of an NMT master. Within this period, the NMT slave device expects a monitoring request via Node Guarding from the NMT master.

life time = guard time * life time factor

The value “0” deactivates monitoring of the NMT master.

If there is no connection monitoring through the NMT master during the time interval “life time”, the device signals an error and switches to the operating state Error.

The time factor can be changed in the NMT state “Pre-Operational”. The time span “guard time” is set with the object **Guard time (100Ch).**

Table B.26 - $100D_h$ Object Description

Index	$100D_h$
Name	Life time factor
Object code	VAR
Data type	Unsigned8

Table B.27 - $100D_h$ Value Description

Sub-index	00_h, Life time factor
Meaning	Life time factor
Access	Read-write
PDO mapping	—
Value range	0...255
Default value	00_h
Category	Yes

1010_h Store Parameters

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

- Sub-Index 0 contains the largest sub-index that is supported.
- Sub-Index 1 refers to all parameters that can be stored on the device.
- Sub-Index 2 refers to communication related parameters (**Index 1000_h - $1FFF_h$** manufacturer specific communication parameters).
- Sub-Index 3 refers to application related parameters (**Index 6000_h - $9FFF_h$** manufacturer specific application parameters).
- Sub-index 4 refers to manufacturer specific parameters.

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

Figure B.1 - Storage Write Access Signature

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

On reception of the correct signature in the appropriate sub-index, the device stores the parameter and then confirms the SDO transmission (initiate download response). If the storing failed, the device responds with an Abort SDO Transfer (**abort code: $0606\ 0000_h$**).

If an incorrect signature is written, the device refuses to store and responds with Abort SDO Transfer (**abort code: 0800 002x**$_h$).

On read access to the appropriate Sub-Index the device provides information about its storage functionality with the following format:

Figure B.2 - Storage Read Access Structure

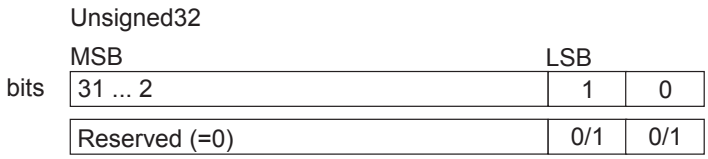


Table B.28 - Structure of Read Access

Bit	Value	Meaning
31 ... 2	0	Reserved (=0)
1	0	Device does not save the parameters autonomously
	1	Device does save the parameters autonomously
2	0	Device does not save the parameters on command
	1	Device does save the parameters on command

Autonomous saving means that a device stores the storable parameters in a non-volatile manner without user request.

Table B.29 - 1010$_h$ Object Description

Index	1010$_h$
Name	Store parameters
Object code	Array
Data type	Unsigned32

Table B.30 - 1010$_h$ Value Description

Sub-index	00$_h$, Largest supported sub-index
Meaning	Largest supported sub-index
Access	Read only
PDO mapping	—
Value range	1$_h$ - 4$_h$
Default value	4$_h$
Category	—

Sub-index	01$_h$, Save all parameters
Meaning	Save all parameters
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	—
Category	—

Sub-index	02_h, Save communication parameters
Meaning	Save communication parameters
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	—
Category	—

Sub-index	03_h, Save application parameters
Meaning	Save application parameters
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	—
Category	—

Sub-index	04_h, Save manufacturer parameters
Meaning	Save manufacturer parameters
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	—
Category	—

1011_h Restore default parameters

With this object, the default values of parameters according to the communication or device profile are restored. By read access the device provides information about its capabilities to restore these values. Several parameter groups are distinguished:

1. Sub-Index 0 contains the largest sub-index that is supported.
2. Sub-Index 1 refers to all parameters that can be restored.
3. Sub-Index 2 refers to communication related parameters (**`Index 1000h - 1FFFh`** manufacturer specific communication parameters).
4. Sub-Index 3 refers to application related parameters (**`Index 6000h - 9FFFh`** manufacturer specific application parameters).
5. At Sub-Index 4 - 127 manufacturers may restore their individual choice of parameters.
6. Sub-Index 128 - 254 are reserved for future use.

In order to avoid the restoring of default parameters by mistake, restoring is only executed when a specific signature is written to the appropriate sub-index. The signature is “load”.

Figure B.3 - Restore Default Parameters Write Access Signature.

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

On reception of the correct signature in the appropriate sub-index, the device restores the default parameters and then confirms the SDO transmission (initiate download response). If the restoring failed, the device responds with an Abort SDO Transfer (**abort code: 0606 0000h**). If an incorrect signature is written, the device refuses to restore the defaults and responds with an Abort SDO Transfer (**abort code: 0800 002xh**).

The default values are set valid after the device is reset (reset node for sub-index 1_h – 4_h, reset communication for sub-index 2_h) or power cycled.

On read access to the appropriate sub-index the device provides parameter restoring capability with the following format:

Figure B.4 - Restore Default Parameters Write Access Structure.

	Unsigned32	
	MSB	LSB
bits	31 ... 1	0
	Reserved (=0)	0/1

Table B.31 - Structure of Write Access

Bit	Value	Meaning
31 ... 1	0	Reserved (=0)
1	0	Device does not restore the default parameters
	1	Device does restore the default parameters

Table B.32 - 1011_h Object Description

Index	1011_h
Name	Restore default parameters
Object code	Array
Data type	Unsigned32

Table B.33 - 1011_h Value Description

Sub-index	00_h, Largest supported sub-index
Meaning	Largest supported sub-index
Access	Read only
PDO mapping	—
Value range	1_h - 4_h
Default value	4h
Category	—

Sub-index	01_h, Restore all parameters
Meaning	Restore all parameters
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	—
Category	—

Sub-index	02_h, Restore communication parameters
Meaning	Restore communication parameters
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	—
Category	—

Sub-index	03_h, Restore application parameters
Meaning	Restore application parameters
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	—
Category	—

Sub-index	04_h, Restore manufacturer parameters
Meaning	Restore manufacturer parameters
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	—
Category	—

1012_h COB-ID Time Stamp Object

Index 1012_h defines the COB-ID of the time-stamp object (TIME). Further, it defines whether the device consumes the TIME or whether the device generates the TIME.

Figure B.5 - Structure of the COB-ID TIME Entry

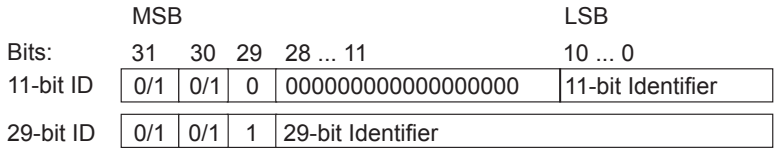


Table B.34 - Description of the TIME COB-ID Entry

Bit	Value	Meaning
31 (MSb)	0	Device does not consume the TIME message
	1	Device consumes the TIME message
30	0	Device does not produce the TIME message
	1	Device produces the TIME message
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
28 ... 11	0	If bit 29=0
	X	If bit 29=1: bits 28 ... 11 of 29 bit TIME-COB-ID
10 ... 0 (LSb)	X	Bits 10 ... 0 of TIME-COB-ID

Bits 29, 30 may be static (not changeable). If a device is not able to generate TIME messages, an attempt to set bit 30 results in an abort message (**`abort code: 0609 0030`**$_h$).

Table B.35 - Object Description

Index	1012_h
Name	COB-ID time stamp message
Object code	VAR
Data type	Unsigned32

Table B.36 - Value Description

Sub-index	00_h, COB-ID time stamp message
Meaning	COB-ID time stamp message
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	0000 0100_h
Category	—

1014_h COB-ID Emergency Error Object

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

Figure B.6 - Structure of the EMCY Identifier Entry

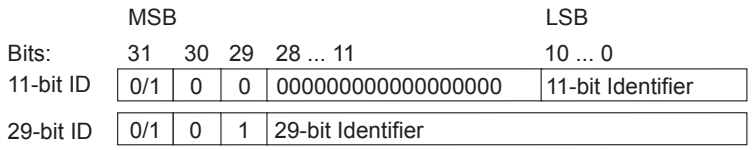


Table B.37 - Description of the COB-ID Entry

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

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

Table B.38 - 1014_h Object Description

Index	1014_h
Name	COB-ID emergency message
Object code	VAR
Data type	Unsigned32

Table B.39 - 1014_h Value Description

Sub-index	00_h, COB-ID emergency message
Meaning	COB-ID emergency message
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	0080_h + node ID
Category	—

1015_h Inhibit Time EMCY Object

The inhibit time for the EMCY message can be adjusted via this entry. If this entry exists it must be writable in the object dictionary. The time has to be a multiple of 100µs.

Table B.40 - 1015_h Object Description

Index	1015_h
Name	Inhibit time EMCY
Object code	VAR
Data type	Unsigned16

Table B.41 - 1015_h Value Description

Sub-index	00_h, Inhibit time EMCY
Meaning	Inhibit time EMCY
Access	Read-write
PDO mapping	—
Value range	Unsigned16
Default value	0000_h
Category	—

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.

Table B.42 - 1017_h Object Description

Index	1017_h
Name	Producer heartbeat time
Object code	VAR
Data type	Unsigned16

Table B.43 - 1017_h Value Description

Sub-index	00_h, Producer heartbeat time
Meaning	Producer heartbeat time
Access	Read-write
PDO mapping	—
Value range	Unsigned16
Default value	0000_h
Category	—

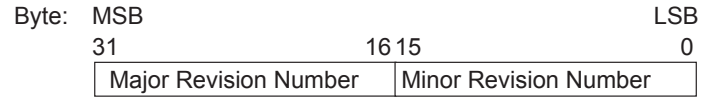
1018_h Identity Object

The object at index 1018_h contains general information about the device. The Vendor ID (sub-index 1_h) contains a unique value allocated to each manufacturer.

The manufacturer-specific product code (sub-index 2_h) identifies a specific device version. The manufacturer-specific revision number (sub-index 3_h) consists of a major revision number and a minor revision number. The major revision number identifies a specific CANopen

behavior. If the CANopen functionality is expanded, the major revision has to be incriminated. The minor revision number identifies different versions with the same CANopen behavior.

Figure B.7 - Structure of the Revision Number



The manufacturer-specific serial number (sub-index 4_h) identifies a specific device.

Table B.44 - 1018_h Object Description

Index	1018_h
Name	Identity object
Object code	Array
Data type	Unsigned32

Table B.45 - 1018_h Value Description

Sub-index	00_h, Largest supported sub-index
Meaning	Largest supported sub-index
Access	Read only
PDO mapping	—
Value range	1_h - 4_h
Default value	4_h
Category	—

Sub-index	01_h, Vendor ID
Meaning	Vendor ID
Access	Read-only
PDO mapping	—
Value range	Unsigned32
Default value	0000 $021B_h$
Category	—

Sub-index	02_h, Product code
Meaning	Product code
Access	Read only
PDO mapping	—
Value range	Unsigned32
Default value	0000 0000_h
Category	—

Sub-index	03_h, Revision number
Meaning	Revision number
Access	Read only
PDO mapping	—
Value range	Unsigned32
Default value	$0000\ 0507_h$
Category	—

Sub-index	04_h, Serial number
Meaning	Serial number
Access	Read only
PDO mapping	—
Value range	Unsigned32
Default value	—
Category	—

$1400 – 1403_h$ Receive PDO Communications Parameter

These objects contain the communication parameters for the PDOs the device is able to receive. The type of the PDO communication parameter (20_h) is described in CiA 301: CANopen Application Layer and Communications Profile. The sub-index 0_h contains the number of valid entries within the communication record. Its value is at least 2. If inhibit time supported the value is 3. The COB-ID of the PDO resides at sub-index 1_h. This entry has been defined as UNSIGNED32 in order to cater for 11-bit CAN Identifiers (CAN 2.0A) as well as for 29-bit CAN identifiers (CAN 2.0B).

Figure B.8 - Structure of the PDO COB-ID Entry

	MSB				LSB
Bits:	31	30	29	28 ... 11	10 ... 0
11-bit ID	0/1	0/1	0	00000000000000000	11-bit Identifier
29-bit ID	0/1	0/1	1	29-bit Identifier	

Table B.46 - Description of the PDO COB-ID Entry

Bit	Value	Meaning
31 (MSb)	0	PDO Exists/Is Valid
	1	PDO Does Not Exist/Is Not Valid
30	0	RTR is Allowed on this PDO
	1	RTR is Not Allowed on this PDO
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
28 ... 11	0	If bit 29=0
	X	If bit 29=1: bits 28 ... 11 of 29 bit COB-ID
10 ... 0 (LSb)	X	Bits 10 ... 0 of COB-ID

The PDO valid/not valid allows the selection of which PDOs are used in the operational state. There may be PDOs fully configured (by default) but not used, and therefore set to “not valid” (deleted). This feature is necessary for devices supporting more than 4 receive process data objects (RPDOs) or 4 transfer process data objects (TPDOs), because each device has only default identifiers for the first four RPDOs/TPDOs. Devices supporting the standard CAN frame type only or which do not support Remote Frames, an attempt to set bit 29 to 1 or bit 30 to 0 results in an abort message (**`abort code: 0609 0030h`**). It is not allowed to change bit 0-29 while the PDO exists (Bit 31=0).

The transmission type (sub-index 2) defines the transmission/reception character of the PDO. On an attempt to change the value of the transmission type to a value that is not supported by the device, an abort message (**`abort code: 0609 0030`**$_h$) is generated.

Table B.47 - Description of the PDO COB-ID Entry

Transmission type	PDO transmission				
	cyclic	acyclic	sync	async	RTR only
0		X	X		
1 – 240	X		X		
241 – 251	Reserved				
252			X		X
253				X	X
254				X	
255				X	

Synchronous (transmission types 0-240 and 252) means that the transmission of the PDO is related to the SYNC object. Preferably the devices use the SYNC as a trigger to output or actuate based on the previous synchronous Receive PDO respectively to update the data transmitted at the following synchronous Transmit PDO. Details of this mechanism depend on the device type and are defined in the device profile if applicable.

Asynchronous means that the transmission of the PDO is not related to the SYNC object. A transmission type of zero means that the message is transmitted synchronously with the SYNC object, but not periodically. A value between 1 and 240 means that the PDO is transferred synchronously and cyclically. The transmission type indicating the number of SYNC which are necessary to trigger PDO transmissions.

Receive PDOs are always triggered by the following SYNC upon reception of data and independent of the transmission types 0 - 240. The transmission types 252 and 253 mean that the PDO is only transmitted on remote transmission request. At transmission type 252, the data is updated (but not sent) immediately after reception of the SYNC object.

At transmission type 253, the data is updated at the reception of the remote transmission request (hardware and software restrictions may apply). These value are only possible for T_PDOs.

For T_PDOs transmission, type 254 means the application event is manufacturer specific (manufacturer specific part of the Object Dictionary). Transmission type 255 means the application event is defined in the device profile. R_PDOs with that type trigger the update of the mapped data with the reception.

Sub-index 3_h contains the inhibit time. This time is a minimum interval for PDO transmission. The value is defined as a multiple of 100µs. It is not allowed to change the value while the PDO exists (Bit 31 of sub-index 1 is 0).

Sub-index 4h is reserved. It does not have to be implemented, in this case read or write access leads to Abort SDO Transfer (**`abort code: 0609 0011h`**).

In mode 254/255 additionally an event time can be used for T_PDO. If an event timer exists for a T_PDO (value not equal to 0), the elapsed timer is considered to be an event. The event timer elapses as a multiple of 1 ms of the entry in sub-index 5_h of the T_PDO. This event will cause the transmission of this T_PDO in addition to otherwise defined events. The occurrence of the events set the timer. Independent of the transmission type the R_PDO event timer is used to recognize the expiration of the R_PDO.

Table B.48 - 1400 – 1403_h Object Description

Index	$1400 – 1403_h$
Name	1st, 2nd, 3rd and 4th receive PDO parameters
Object code	Record
Data type	—

Table B.49 - 1400 – 1403_h Value Description

Sub-index	00_h, Largest supported sub-index
Meaning	Largest supported sub-index
Access	Read only
PDO mapping	—
Value range	Unsigned8
Default value	05_h
Category	—

Sub-index	01_h, COB-ID used by PDO
Meaning	COB-ID used by PDO
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	$1400_h = 0200_h$ + node ID $1401_h = 0300_h$ + node ID $1402_h = 0400_h$ + node ID $1403_h = 0500_h$ + node ID
Category	Yes

Sub-index	02_h, Transmission type
Meaning	Transmission type
Access	Read-write
PDO mapping	—
Value range	Unsigned8
Default value	255 (asynchronous)
Category	Yes

Sub-index	03_h, Inhibit time
Meaning	Inhibit time
Access	Read-write
PDO mapping	—
Value range	Unsigned16
Default value	0000_h
Category	Yes

Sub-index	05_h, Event timer
Meaning	Event timer
Access	Read-write
PDO mapping	—
Value range	Unsigned16
Default value	0000_h
Category	Yes

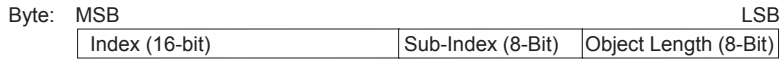
1600 – 1603_h Receive PDO Mapping Parameter

Contains the mapping for the PDOs the device is able to receive. The type of the PDO mapping parameter (21_h) is described in CiA 301: CANopen Application Layer and Communications Profile. The sub-index 0_h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which is transmitted/received with the corresponding PDO. The sub-indices from 1h to number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index, and length. All three values are hexadecimal coded. The length entry contains the length of the object in bit ($1..40_h$).

This parameter can be used to verify the overall mapping length. It is mandatory.

The structure of the entries from sub-index 1_h – 40_h is as follows:

Figure B.9 - Structure of the PDO Mapping Entry



If the change of the PDO mapping cannot be executed (i.e., the PDO length is exceeded or the SDO client attempts to map an object that cannot be mapped) the device responds with an Abort SDO Transfer Service.

Sub-index 0 determines the valid number of objects that have been mapped. To change the PDO mapping, first the PDO has to be deleted and the sub-index 0 must be set to 0 (mapping is deactivated). Then the objects can be remapped. When a new object is mapped by writing a sub-index between 1 and 64, the device may check whether the object specified by index / sub-index exists. If the object does not exist or the object cannot be mapped, the SDO transfer must be aborted with the Abort SDO Transfer Service using one of the abort codes 0602 0000_h or 0604 0041_h.

After all objects are mapped, sub-index 0 is set to the valid number of mapped objects. Finally the PDO will be created by writing to its communication parameter COB-ID. When sub-

index 0 is set to a value >0 the device may validate the new PDO mapping before transmitting the response of the SDO service. If an error is detected the device has to transmit the Abort SDO Transfer Service with one of the abort codes 0602 0000_h, 0604 0041_h, or 0604 0042_h.

When sub-index 0 is read, the actual number of valid mapped objects is returned. If data types (Index 1_h-7_h) are mapped they serve as “dummy entries”. The corresponding data in the PDO is not evaluated by the device. This optional feature is useful to transmit data to several devices using one PDO, each device only utilizing a part of the PDO. It is not possible to create a dummy mapping for a T_PDO.

A device that supports dynamic mapping of PDOs must support this during the PREOPERATIONAL state. If dynamic mapping during the state OPERATIONAL is supported, the SDO client is responsible for data consistency.

Figure B.10 - Principle of PDO Mapping

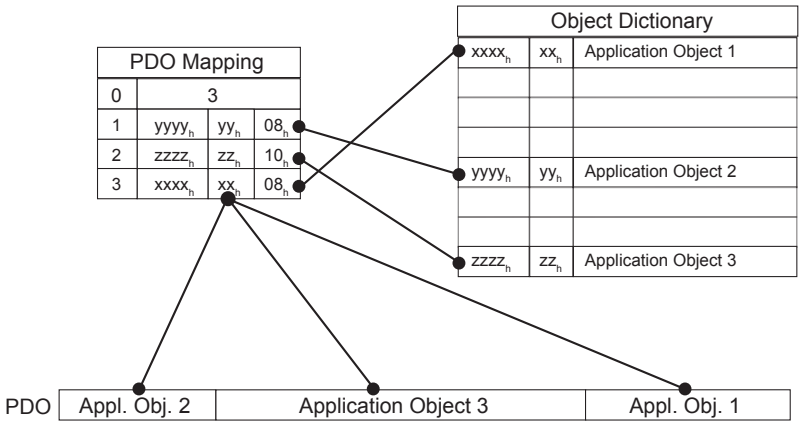


Table B.50 - 1600 – 1603_h Object Description

Index	1600 – 1603_h
Name	1st, 2nd, 3rd, and 4th receive PDO mapping
Object code	Record
Data type	—

Table B.51 - 1600 – 1603_h Value Description

Sub-index	00_h, Number of mapped application objects
Meaning	Number of mapped application objects
Access	Read-write
PDO mapping	—
Value range	1 – 64
Default value	1600_h= 1 1601_h= 2 1602_h= 2 1603_h= 2
Category	Yes

Sub-index	01_h, PDO mapping 1st application object
Meaning	PDO mapping 1st application object
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	1600_h= 6040 0010_h 1601_h= 6040 0010_h 1602_h= 6040 0010_h 1603_h= 6040 0010_h
Category	Yes

Sub-index	02_h, PDO mapping 2nd application object
Meaning	PDO mapping 2nd application object
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	1600_h= 0000 0000_h 1601_h= 607A 0020_h (Profile position – target position) 1602_h= 60FF 0020_h (Profile velocity – target velocity) 1603_h= 6071 0020_h (Profile torque – target torque)
Category	Yes

Sub-index	03 – 08_h, PDO mapping *n*th application object
Meaning	PDO mapping *n*th application object
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	1600_h= 0000 0000_h 1601_h= 0000 0000_h 1602_h= 0000 0000_h 1603_h= 0000 0000_h
Category	Yes

1800 – 1803_h Receive PDO Mapping Parameter

Contains the communication parameters for the PDOs the device is able to transmit. The type of the PDO communication parameter (20_h) is described in CiA 301: CANopen Application Layer and Communications Profile. A detailed description of the entries is done in the section for the Receive PDO Communication Parameter (1400_h – 1403_h).

Table B.52 - 1800 – 1803_h Object Description

Index	1800 – 1803_h
Name	1st, 2nd, 3rd, and 4th transmit PDO parameters
Object code	Record
Data type	—

Table B.53 - 1800 – 1803_h Value Description

Sub-index	00_h, Largest supported sub-index
Meaning	Largest supported sub-index
Access	Read only
PDO mapping	—
Value range	Unsigned8
Default value	05_h
Category	—

Sub-index	01_h, COB-ID used by PDO
Meaning	COB-ID used by PDO
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	1800_h= 0180_h+ node ID 1801_h= 0280_h+ node ID 1802_h= 0380_h+ node ID 1803_h= 0480_h+ node ID
Category	Yes

Sub-index	02_h, Transmission type
Meaning	Transmission type
Access	Read-write
PDO mapping	—
Value range	Unsigned8
Default value	255 (asynchronous)
Category	Yes

Sub-index	03_h, Inhibit time
Meaning	Inhibit time
Access	Read-write
PDO mapping	—
Value range	Unsigned16
Default value	0000_h
Category	Yes

Sub-index	05_h, Event timer
Meaning	Event timer
Access	Read-write
PDO mapping	—
Value range	Unsigned16
Default value	1800_h= 0 1801_h= 100 1802_h= 100
Category	Yes

1A00 – $1A03_h$ Transmit PDO Mapping Parameter

These objects contain the mapping for the PDOs the device is able to transmit. The type of the PDO mapping parameter (21_h) is described in CiA 301: CANopen Application Layer and Communications Profile. A detailed description of the entries can be found in "1600 – 1603h Receive PDO Mapping Parameter" on page 121.

Table B.54 - 1A00 – $1A03_h$ Object Description

Index	1A00 – $1A03_h$
Name	1st, 2nd, and 3rd transmit PDO mapping
Object code	Record
Data type	—

Table B.55 - 1A00 – $1A02_h$ Value Description

Sub-index	00_h, Number of mapped application objects
Meaning	Number of mapped application objects
Access	Read-write
PDO mapping	—
Value range	1 – 64
Default value	$1A00_h$= 1 $1A01_h$= 2 $1A02_h$= 2 $1A03_h$= 1
Category	Yes

Sub-index	01_h, PDO mapping 1st application object
Meaning	PDO mapping 1st application object
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	$1A00_h$= 6041 0010_h (Status Word) $1A01_h$= 6041 0010_h (Status Word) $1A02_h$= 6041 0010_h (Status Word) $1A03_h$= 6041 0010_h (Status Word)
Category	Yes

Sub-index	02_h, PDO mapping 2nd application object
Meaning	PDO mapping 2nd application object
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	1800_h= 0000 0000_h 1801_h= 6064 0020_h (Profile position – position actual value) 1802_h= 606C 0020_h (Profile velocity – velocity actual value) 1803_h= 0000 0000_h
Category	Yes

Sub-index	$03 – 08_h$, PDO mapping nth application object
Meaning	PDO mapping nth application object
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	1600_h= 0000 0000_h 1601_h= 0000 0000_h 1602_h= 0000 0000_h 1603_h= 0000 0000_h
Category	Yes

Details of Object Group 2000_h (Mfg Specific)

The objects detailed in this section are Novanta IMS manufacturer specific configuration objects to configure the manufacturer object specific to the LMD CANopen node.

2000_h I/O Configuration

This object facilitates the configuration of the input and output points available on LMD CANopen devices. The sub-indexes 4 and 5 may be configured as sinking or sourcing inputs or outputs. Each bit of the sub-indices are mapped to a particular I/O point with I/O 12 being the Most Significant bit (MSb) and I/O 1 being the Least Significant bit (LSb). The configuration options are:

1. Sub-Index 01_h: Reserved
2. Sub-Index 02_h: Reserved
3. Sub-Index 03_h: Reserved
4. Sub-Index 04_h: This allows the configuration of inputs to read inverted polarity. By default this is deactivated and setting the bit to a 1 will change the invert polarity read by the input. See Optional Application FE, Object 60DF.
5. Sub-Index 05_h: This allows the configuration of outputs to invert the output polarity. See Optional Application FE, Object 60FE.

The I/O configuration is saved using the Store Parameters Object (1010_h).

Table B.56 - 2000_h Object Description

Index	2000_h
Name	I/O configuration
Object code	ARRAY
Data type	Unsigned8

Table B.57 - 2000_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	05_h
Category	—

Sub-index	01_h - 03_h
Meaning	Reserved
Access	—
PDO mapping	—
Value range	—
Default value	—
Category	Yes

Sub-index	04_h, Configure as polarity in
Meaning	Configure as polarity in (See object $60FD_h$ sub-index 01_h)
Access	Read-write
PDO mapping	—
Value range	$00 – FF_h$
Default value	00_h
Category	Yes

Sub-index	05_h, Configure as polarity out
Meaning	Configure as polarity out
Access	Read-write
PDO mapping	—
Value range	$00 – FF_h$
Default value	00_h (See object $60FE_h$ sub-index 01_h)
Category	Yes

2002_h Configure Input Switches

Object 2002_h facilitates the configuration of input switches. Input switches may be configured as the following types:

1. Home
2. Positive Limit
3. Negative Limit
4. Inhibit (Inhibit Switch function is configured by Object 2007_h)

Table B.58 - 2002_h Object Description

Index	2002_h
Name	Configure input switches
Object code	ARRAY
Data type	Unsigned8

Table B.59 - 2002_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	04_h
Category	—

Sub-index	01_h, Configure input as home
Meaning	Configure input as home switch (I/O point bit(s) = 1_b to select as home)
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	00_h
Category	Yes

Sub-index	02_h, Configure input as positive limit
Meaning	Configure input as + Limit (I/O point bit(s) = 1b to select as + limit)
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	00_h
Category	Yes

Sub-index	03_h, Configure input as negative limit
Meaning	Configure input as – Limit (I/O point bit(s) = 1_b to select as – limit)
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	00_h
Category	Yes

Sub-index	04_h, Configure inhibit switch
Meaning	Configure inhibit switch (I/O point bit(s) = 1_b to select as inhibit). Use object 2007_h to define inhibit function.
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	00_h
Category	Yes

2004_h Configure Input Filter Mask

The Input filter mask object configures the device to filter the selected inputs. Sub-indices 01_h through 04_h define the inputs to which filtering will be applied. Object 2006_h defines the filter time applied to each input.

Figure B.11 - Input Filter Mask

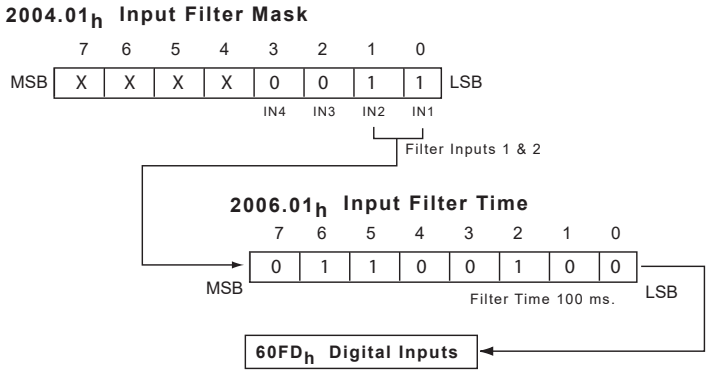


Table B.60 - 2004_h Object Description

Index	2004_h
Name	Configure input mask
Object code	ARRAY
Data type	—

Table B.61 - 2004_h Value Description

Sub-index	00h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	08h
Category	—

Sub-index	01h, Configure mask for Input 1
Meaning	Configure mask for Input 1
Access	Read-write
PDO mapping	—
Value range	00 – FFh
Default value	01h
Category	Yes

Sub-index	02_h, Configure mask for Input 2
Meaning	Configure mask for Input 2
Access	Read-write

PDO mapping	—
Value range	00 – FF_h
Default value	02_h
Category	Yes

Sub-index	03_h, Configure mask for Input 3
Meaning	Configure mask for Input 3
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	04_h
Category	Yes

Sub-index	04_h, Configure mask for Input 4
Meaning	Configure mask for Input 4
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	08_h
Category	Yes

Sub-index	05_h - 08_h, reserved
Meaning	Reserved
Access	Reserved
PDO mapping	—
Value range	—
Default value	—
Category	Yes

2006_h Configure Input Filter Time

This object sets the input filter time in milliseconds. Each sub-index applies to a specific input where sub-index 01_h applies to input 1, sub-index 02_h applies to input 2, etc.

Table B.62 - 2006_h Object Description

Index	2006_h
Name	Configure input filter
Object code	ARRAY
Data type	—

Table B.63 - 2006_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	08_h
Category	—

Sub-index	01_h, Configure filter for Input 1
Meaning	Configure filter for Input 1
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	00_h
Category	Yes

Sub-index	02_h, Configure filter for Input 2
Meaning	Configure filter for Input 2
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	00_h
Category	Yes

Sub-index	03_h, Configure filter for Input 3
Meaning	Configure filter for Input 3
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	00_h
Category	Yes

Sub-index	05_h, Configure filter for Input 4
Meaning	Configure filter for Input 4
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	00_h
Category	Yes

Sub-index	05_h, Configure filter for Input 9
Meaning	Configure filter for Input 9
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	00_h
Category	Yes

Sub-index	06_h, Configure filter for Input 10
Meaning	Configure filter for Input 10
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	00_h
Category	Yes

Sub-index	07_h, Configure filter for Input 11
Meaning	Configure filter for Input 11
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	00_h
Category	Yes

Sub-index	08_h, Configure filter for Input 12
Meaning	Configure filter for Input 12
Access	Read-write
PDO mapping	—
Value range	00 – FF_h
Default value	00_h
Category	Yes

2007_h Inhibit Switch Reaction

This object allows the user to configure different actions for an inhibit switch (see **`Object 2002h, Sub-index 4h`**).

This object will function through controlword overwrites and overrides. The inhibit switch reaction is set using sub-index 01_h.

Table B.64 - 2007_h Object Description

Index	2007_h
Name	Inhibit switch reaction
Object code	ARRAY
Data type	—

Table B.65 - 2007_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	01_h
Category	—

Sub-index	01_h, Inhibit switch reaction
Meaning	Inhibit switch reaction
Access	Read-write
PDO mapping	—
Value range	1 – 12
Default value	12
Category	Yes

Table B.66 - Inhibit Switch Reactions

Value	Meaning
0	No action
1	Error signal - Control word overwrite
2	Error signal - Control word override
3	Disable voltage command - Control word overwrite
4	Disable voltage command - Control word override
5	Quick stop command - Control word overwrite
6	Quick stop command - Control word override
7	Shutdown command - Control word overwrite
8	Shutdown command - Control word override
9	Disable operation command - Control word overwrite
10	Disable operation command - Control word override
11	Halt command - Control word overwrite
12	Halt command - Control word override

2008_h Output Definition

This object allows the user to configure one or more outputs as brake outputs. For an explanation of brake functions, see Object 2035_h: Brake Timers

Table B.67 - 2008_h Object Description

Index	2008_h
Name	Output definition
Object code	ARRAY
Data type	—

Table B.68 - 2008_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	02_h
Category	—

Sub-index	01_h, Brake output defined
Meaning	Brake output defined
Access	Read-write
PDO mapping	—
Value range	Unsigned8
Default value	00_h
Category	Yes

Sub-index	02_h, Target reached output defined
Meaning	Target reached output defined
Access	Read-write
PDO mapping	—
Value range	Unsigned8
Default value	00_h
Category	Yes

Table B.69 - Brake and Target Reached Output Definition

Output selected	Setting
1	Sub-index = 01_h
2	Sub-index = 02_h
3	Sub-index = 04_h

2010_h Analog Input Configuration

This object allows the user to configure the 12-bit Analog Input. There are 3 sub-indices that set the configuration properties for the input:

1. Sub-Index 01_h: Analog Input reading provides the value of the Analog Input.
2. Sub-Index 02_h: This sets the type of device the Analog Input will read. It can be set for two modes, Voltage with ranges of 0 to 5V or 0 to 10V, or Current with an input range of 0 to 20 mA.
3. Sub-Index 03_h: This sets the filtering for the Analog Input. In the 0 (default) setting the filtering is off.

Table B.70 - 2010_h Object Description

Index	2010_h
Name	Analog input configuration
Object code	ARRAY
Data type	—

Table B.71 - 2010_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
Default value	03_h

Sub-index	01_h, Analog input reading
Meaning	Analog input reading
Access	Read-only
PDO mapping	Yes – T_PDO
Value range	$0 – 4095_d$ (0000 – FFFFh – Unsigned16)
Default value	—
Category	—

Sub-index	02_h, Analog input configuration
Meaning	Analog input configuration
Access	Read-write
PDO mapping	—
Value range	00_h, 02_h, or 08_h (Unsigned8)
Default value	00_h
Category	Yes

Sub-index	02_h Analog input mode
00_h	0 to 5 V scale
02_h	0 to 20 mA scale
08_h	0 to 10 V scale

Sub-index	03_h, Analog filter level
Meaning	Analog filter level
Access	Read-write
PDO mapping	—
Value range	$00 – 31_h$ (Unsigned8)
Default value	00_h
Category	Yes

2014_h Auxiliary Power Input Monitoring

This object allows for the user to monitor the voltage level of the aux-power input, as well as establish low and high level advisory limits. This input provides power to logic and feedback circuitry in the event of a main power loss.

The reading is given as voltage X 10 (i.e., 24.4 VDC = 244).

Table B.72 - 2014_h Object Description

Index	2014_h
Name	Auxiliary power input monitoring
Object code	ARRAY
Data type	—

Table B.73 - 2014_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	03_h
Category	—

Sub-index	01_h, Aux-power input reading
Meaning	Aux-power input reading
Access	Read-only
PDO mapping	Yes – T_PDO
Value range	$0 – 280_d$ ($0000 – 0118_h$ – Unsigned16)
Default value	—
Category	—

Sub-index	02_h, Aux-power input low-level advisory
Meaning	Aux-power input low-level advisory
Access	Read-write
PDO mapping	—
Value range	$0 – 280_d$ ($0000 – 0118_h$ – Unsigned16)
Default value	00_h
Category	Yes

Sub-index	03_h, Aux-power input high-level advisory
Meaning	Aux-power input high-level advisory
Access	Read-write
PDO mapping	—
Value range	$0 – 280_d$ ($0000 – 0118_h$ – Unsigned16)
Default value	280_d (0118_h)
Category	Yes

2015_h +VDC Input Monitoring

This object allows for the user to monitor the voltage level of the motor power input (+VDC), as well as establish low and high level advisory limits.

The reading is given as voltage X 10 (i.e., 44.8 VDC = 448).

Table B.74 - 2015_h Object Description

Index	2015_h
Name	+VDC input monitoring
Object code	ARRAY
Data type	—

Table B.75 - 2015_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	03_h
Category	—

Sub-index	01_h, +VDC input monitoring
Meaning	+VDC input monitoring
Access	Read-only
PDO mapping	Yes – T_PDO
Value range	$0 – 800_d$ ($0000 – 0320_h$ – Unsigned16)
Default value	—
Category	—

Sub-index	02_h, +VDC input low-level advisory
Meaning	+VDC input low-level advisory
Access	Read-write
PDO mapping	—
Value range	$0 – 800_d$ ($0000 – 0320_h$ – Unsigned16)
Default value	100_d (64_h)
Category	Yes

Sub-index	03_h, +VDC input high-level advisory
Meaning	+VDC input high-level advisory
Access	Read-write
PDO mapping	—
Value range	$0 – 800_d$ ($0000 – 0320_h$ – Unsigned16)
Default value	580_d (244_h)
Category	Yes

2016_h Backup Voltage Input Monitoring

This object is supported by LMD products with a multi-turn absolute encoder and is used in conjunction with the Absolute Encoder Back-up Battery pack, ICP0531, accessory to provide up to five years position update and retention.

Table B.76 - 2016_h Object Description

Index	2016_h
Name	Encoder battery pack voltage
Object code	ARRAY
Data type	Integer8
Category	Optional

Table B.77 - 2016_h Entry Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	03_h
Category	—

Sub-index	01_h
Meaning	External battery voltage
Access	ro
PDO mapping	Yes
Value range	Unsigned16
Default value	—
Category	—

Sub-index	02h
Meaning	External battery low-level advisory. Setting to 0 disables the advisory.
Access	rw
Value range	Unsigned16
Default value	0
Low limit	0

High limit	6500

Sub-index	03_h
Meaning	External battery high-level advisory. Setting to 0 disables the advisory.
Access	rw
PDO mapping	—
Value range	Unsigned16
Default value	5500 (5.500V)
Category	—

2017_h Backup Voltage Input Monitoring

This object represents the voltage level of the internal back-up voltage on LMD Absolute Encoder products.

Table B.78 - 2017_h Object Description

Index	2017_h
Name	Read internal holding voltage level
Object code	VAR
Data type	Integer8
Category	Optional

Table B.79 - 2017_h Entry Description

Sub-index	00_h
Meaning	Internal holding voltage level in millivolts
Access	Read only
PDO mapping	—
Value range	Unsigned16
Default value	—
Category	—

2018_h PCB Temperature Options

This object represents the temperature of the printed circuit board (PCB) as measured at the microprocessor. There are 3 sub-indices:

1. Sub-Index 01_h: Read-only sub-index that represents the internal temperature of the device. May be mapped to a PDO.
2. Sub-Index 02_h: Temperature Advisory parameter allows for the setting of a parameter that will generate an error (**`Index 1003 - 0016 4210h`**) if the advisory threshold is reached. The default is 80°C.
3. Sub-Index 03_h: This sub-index sets the threshold for a temperature error. Note that the outputs of the device will disable at 85°C regardless of the setting for this parameter. If reached, the error message will be located at **`Index 1003h`**. The error code is one byte in length and reads **`0008 4210h`**.

The units for this object are degrees Celsius (°C).

Table B.80 - 2018_h Object Description

Index	2018_h
Name	PCB temperature options
Object code	ARRAY
Data type	Signed8

Table B.81 - 2018_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read-only
PDO mapping	—
Value range	—
Default value	03_h
Category	—

Sub-index	01_h, PCB temperature reading
Meaning	PCB temperature reading
Access	Read-only
PDO mapping	Yes – T_PDO
Value range	—
Default value	—
Category	—

Sub-index	02_h, PCB temperature advisory threshold
Meaning	PCB temperature advisory threshold
Access	Read-write
PDO mapping	—
Value range	-50 to $+120_d$ (Signed8)
Default value	80_d
Category	Yes

Sub-index	03_h, PCB temperature error
Meaning	PCB temperature error
Access	Read-write
PDO mapping	—
Value range	-50 to $+120_d$ (Signed8)
Default value	85_d
Category	Yes

2019_h Output h-bridge Temperature Options

This object represents the temperature of the output h-bridge as measured at the output h-bridge. There are 3 sub-indices:

1. Sub-Index 01_h: Read-only sub-index that reads the output h-bridge temperature of the device. May be mapped to a PDO.
2. Sub-Index 02_h: Temperature Advisory parameter allows for the setting of a parameter that will generate an error (**`Index 1003 - 0016 4210h`**) if the advisory threshold is reached. The default is 80°C.
3. Sub-Index 03_h: This sub-index sets the threshold for a temperature error. Note that the outputs of the device will disable at 85°C regardless of the setting for this parameter. If reached, the error message will be located at **`Index 1003h`**. The error code is one byte and reads **`0008 4210h`**.

The units for this object are degrees Celsius (°C).

Table B.82 - 2018_h Object Description

Index	2018_h
Name	Output h-bridge temperature options
Object code	ARRAY
Data type	Signed8

Table B.83 - 2018_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	03_h
Category	—

Sub-index	01_h, Output h-bridge temperature reading
Meaning	Output h-bridge temperature reading
Access	Read-only
PDO mapping	Yes – T_PDO
Value range	—
Default value	—
Category	—

Sub-index	02_h, Output h-bridge temperature advisory threshold
Meaning	Output h-bridge temperature advisory threshold
Access	Read-write
PDO mapping	—
Value range	-50 to $+120_d$ (Signed8)
Default value	80_d
Category	Yes

Sub-index	03_h, Bridge temperature error
Meaning	Output h-bridge temperature error
Access	Read-write
PDO mapping	—
Value range	-50 to $+120_d$ (Signed8)
Default value	850_d
Category	Yes

2020_h Hardware/Software Limit Reached

This object defines the actions taken when the Position Software Limit for Object 2022_h is reached. It consists of 2 sub-indices.

1. Sub-index 01_h: Limit reached flag. This flag will be set based upon the status of a limit, it will register whether the limit reached is a hardware limit or a software limit. The statusword (6041_h), bit 11, internal limit active will set whenever the flag is not 0. The action taken for a limit reached condition will be determined by the limit mask sub-index.

Table B.84 - Description of Limit Reached Flag 2020.01_h

Status	Bit 3	Bit 4	Bit 1	Bit 0
Negative hardware limit reached	0	0	0	1
Positive hardware limit reached	0	0	1	0
Negative software limit reached	0	1	0	0
Positive software limit reached	1	0	0	0

2. Sub-index 02_h: Limit reached mask

Figure B.12 - Software Limits as Hardware Functions

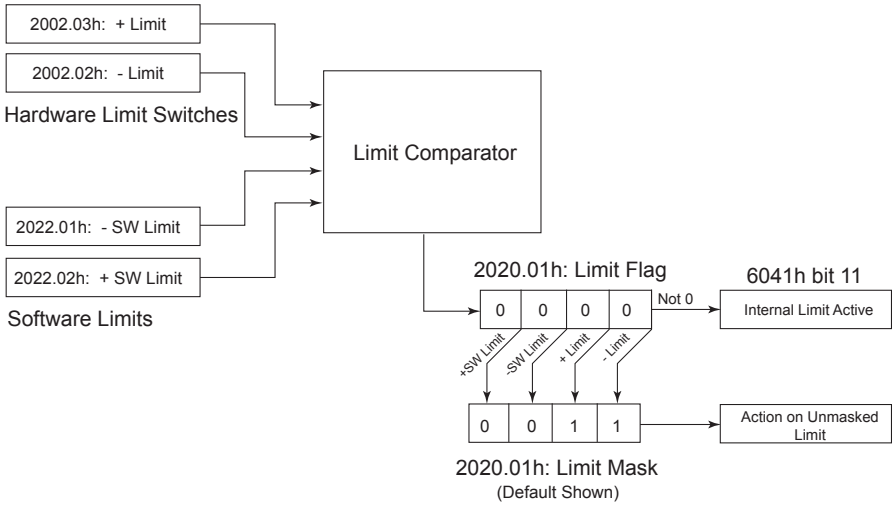


Table B.85 - 2020_h Object Description

Index	2020_h
Name	Software limits as hardware
Object code	ARRAY
Data type	Unsigned8

Table B.86 - 2020_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	02_h
Category	—

Sub-index	01_h, Limit reached flag
Meaning	Limit reached flag
Access	Read-write
PDO mapping	—
Value range	$00 – 0F_h$
Default value	00_h
Category	—

Sub-index	02_h, Limit reached mask
Meaning	Limit reached mask
Access	Read-write
PDO mapping	—
Value range	$00 – 0F_h$
Default value	03_h
Category	Yes

2022_h Software Position Limits

This object defines the software limit based on set negative and positive limits set in position counts. See object 2020_h for a description of software limit functionality and configuration.

1. Sub-index 01h: Software negative limit
2. Sub-index 02h: Software positive limit

Table B.87 - 2022_h Object Description

Index	2022_h
Name	Software position limit
Object code	ARRAY
Data type	Signed32

Table B.88 - 2022_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	02_h
Category	—

Sub-index	01_h, Software negative limit
Meaning	Software negative limit
Access	Read-write
PDO mapping	—
Value range	Signed32
Default value	$8000\ 0000_h$
Category	—

Sub-index	02_h, Software positive limit
Meaning	Software positive limit
Access	Read-write
PDO mapping	—
Value range	Signed32
Default value	$7FFF\ FFFF_h$
Category	Yes

2030_h Output h-bridge Polarity

This object defines the polarity of the h-bridge. By default, positive represents clockwise (cw) and negative represents counter-clockwise (ccw). By changing the polarity of the output h-bridge to a negative integer between -128 to -1, the cw/ccw motor direction can be swapped.

There is no significance assigned to the number; any negative integer from -128 to -1 will reverse the output h-bridge polarity, any positive integer from 0 to 127 will reset the polarity to the default cw/ccw motor direction configuration.

Table B.89 - 2030_h Object Description

Index	2030_h
Name	Output h-bridge polarity
Object code	VAR
Data type	Signed8

Table B.90 - 2030_h Value Description

Sub-index	00_h, Output h-bridge polarity
Meaning	Output h-bridge polarity
Access	Read-write
PDO mapping	—
Value range	-128 to 127_d
Default value	0_d
Category	Yes

2031_h Unit Options (Encoder Enable, Trip/Capture Enable)

This object defines the configuration of the following unit options:

1. Encoder sync actions (bits 5, 4). Encoder sync will determine whether the encoder counter, position counter or -home offset will be the master counter for synchronizing the counters. If -home offset is used, it will function as homing 35.
2. Encoder enable (bit 3)
3. Trip output/capture input (bit 2)

Note: Encoder functions only apply to the LMD products. The MForce products do not have closed-loop capability.

Table B.91 - 2031_h Object Description

Index	2031_h
Name	Unit options (encoder enable, trip/capture enable)
Object code	VAR
Data type	Unsigned8

Table B.92 - 2031_h Value Description

Sub-index	00_h, Unit options (encoder enable, trip/capture enable)
Meaning	Unit options (encoder enable, trip/capture enable)
Access	Read-write
PDO mapping	—
Value range	Unsigned8
Default value	00_h
Category	Yes

Table B.93 - Description of Unit Options Object 2031.00_h

Bits	7	6	5	4	3	2	1	0
Function	X	X	sync_action		e/e	c/t	X	X
Encoder auto-sync action (sync_action)								
Bit 5	Bit 4	Bits 5 and 4 control encoder sync action						
0	0	No action						
0	1	Position synced to encoder master						
1	0	Encoder synced to position master						
1	1	Position and encoder synced to –home offset						
Encoder enable (e/e)								
Bit 3	0	Encoder not enabled						
	1	Encoder operation enabled						
Capture/trip select (c/t)								
Bit 2	0	Will operate as a capture input (configure using 2033h)						
	1	Will operate as a trip output (configure using 2038h)						

2033_h Capture Input Parameters

This object configures the functionality of the capture input.

1. Sub-index 01_h: Capture input control. Sets a bit that will enable the capture input. Note that the capture input must also be selected using **`object 2031h`**.
2. Sub-index 02_h: Position captured flag. Displays the status of a position capture by setting the least significant bit (LSb) of an 8-bit unsigned integer. Write all ones to clear the flag.
3. Sub-index 03_h: Capture input filter time. This sub-index configures the filtering for the capture input.
4. Sub-index 04_h: Captured position. This sub-index holds the captured position.

Table B.94 - 2033_h Object Description

Index	2033_h
Name	Capture input parameters
Object code	REC
Data type	—

Table B.95 - 2033_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	04_h
Category	—

Sub-index	01_h, Capture input enable
Meaning	Capture input enable
Access	Read-write
PDO mapping	—
Value range	0/1 (Unsigned8)
Default value	0 (disabled)
Category	Yes

Sub-index	02_h, Position captured flag
Meaning	Position captured flag
Access	Read-write
PDO mapping	—
Value range	0/1 (Unsigned8)
Default value	0 (no position captured)
Category	Yes

Sub-index	03_h, Capture input filter time
Meaning	Capture input filter time
Access	Read-write
PDO mapping	—
Value range	0 – 9 (Unsigned8)
Default value	0 (50ns)
Category	Yes

Value	**Filter**		**Value**	**Filter**		**Value**	**Filter**
0	50ns		4	500ns		8	6.5µs
1	150ns		5	900ns		9	12.9µs
2	200ns		6	1.7µs			
3	300ns		7	3.3µs			

Sub-index	04_h, Captured position
Meaning	Captured position
Access	Read-only
PDO mapping	—
Value range	Integer32
Default value	0 (no position)
Category	—

2034_h Output h-bridge on Settle Time

This object establishes the time in milliseconds that current in the output h-bridge is allowed to stabilize after power on. This index will delay the device entering operation enabled mode by the time set (0 to 1000ms). It is also a factor in the brake logic block. See **object 2035h** for functional block diagram.

Table B.96 - 2034_h Object Description

Index	2034_h
Name	Output h-bridge on settle time
Object code	Array
Data type	Unsigned16

Table B.97 - 2034_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	01_h
Category	—

Sub-index	01_h, Output h-bridge on settle time
Meaning	Output h-bridge on settle time
Access	Read-write
PDO mapping	—
Value range	$0 – 1000_d$
Default value	0
Category	Yes

2035_h Brake Settle Allow Time

This object establishes the time in milliseconds that power to the driver output h-bridge is allowed to settle prior to releasing the brake and after engaging the brake. This object works in cooperation with **object 2034.01h**, Output h-bridge on settle allow time. The sequence of events follows for a braking operation:

1. Output h-bridge power turns on, **object 2034.01h** begins timing the amount of milliseconds specified in sub-index 01_h. This will be the time between output h-bridge power enabled and brake off. This time will also allow time for settling before initial synchronizing with encoder counts.
2. Break output turns off, **object 2035.02h** specifies the time delay from set brake off to allow for motor movement.
3. Device enters the operation enabled state of the state machine. Motion occurs.
4. The brake will engage after cessation of motion. **object 2035.01h** specifies the time delay from set brake on to removal of output h-bridge power.

Figure B.13 - Output H-bridge to Brake Timing

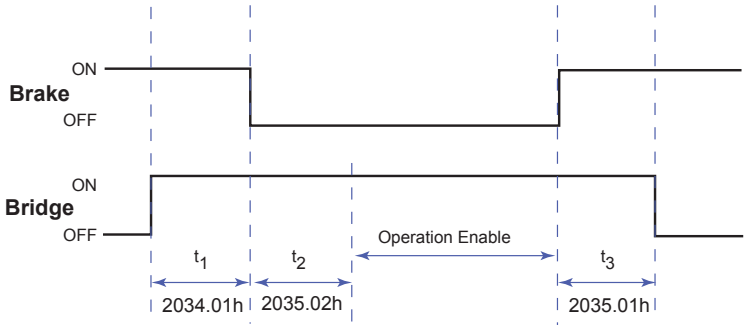


Figure B.14 - Brake Functions Block Diagram

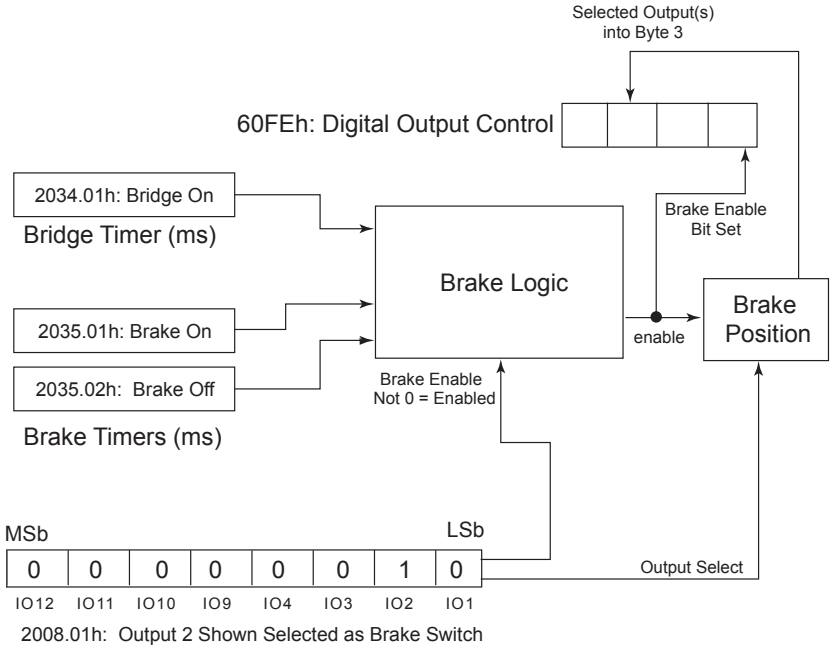


Table B.98 - 2035_h Object Description

Index	2035_h
Name	Brake settle allow time
Object code	ARRAY
Data type	—

Table B.99 - 2035_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	02_h

Category	—

Sub-index	01_h, Brake on settle time
Meaning	Brake on settle time
Access	Read-write
PDO mapping	—
Value range	$0 – 2000_d$ (ms)
Default value	0
Category	Yes

Sub-index	02_h, Brake off settle time
Meaning	Brake off settle time
Access	Read-write
PDO mapping	—
Value range	$0 – 1000_d$ (ms)
Default value	0
Category	Yes

2036_h Hold Current Delay Time

This object defines the delay in milliseconds between the device switching from the run current (**`object 2204h`**) to the holding current (**`object 2205h`**)

Table B.100 - 2036_h Object Description

Index	2036_h
Name	Hold current delay time
Object code	VAR
Data type	Unsigned16

Table B.101 - 2036_h Value Description

Sub-index	00_h, Hold current delay time
Meaning	Hold current delay time
Access	Read-write
PDO mapping	—
Value range	0_d (off) or 2 to 65535_d (ms)
Default value	500d (ms)
Category	Yes

2037_h Output h-bridge on to Encoder Settle Time

This object defines the delay, in milliseconds, between the device switching into operation enable and re-syncing the encoder position. Only applicable on LMD models equipped with an encoder.

Table B.102 - 2037_h Object Description

Index	2037_h
Name	Output h-bridge on to encoder settle time
Object code	VAR
Data type	Unsigned16

Table B.103 - 2037_h Value Description

Sub-index	00_h, Output h-bridge on to encoder settle time
Meaning	Output h-bridge on to encoder settle time
Access	Read-write
PDO mapping	—
Value range	0_d to 3000_d (ms)
Default value	300_d (ms)
Category	Yes

2038_h Trip Output Configuration

This object configures the functionality of the trip output which will pulse the output upon reaching particular **`position_demand_effort_position(s)`**.

Values are in Internal Units, based on 51200/rev and not necessarily scaled to user units. Also note trip is activated on position demand effort, and not actual position or encoder position.

This definition of point(s) begins with sub-index 2036.02_h (1st position of a series) then add sub-index 2036.03_h to form the specified number of trip points.

1. Sub-index 01_h: Trip output control. Controls the logic and trip points and is used to set up one or multiple trip positions. Note that the trip output must also be selected using **`object 2031`**$_h$.
2. Sub-index 02_h: 1st trip point of a series. Defines the first trip point of a series of points.
3. Sub-index 03_h: Capture input filter time. This sub-index configures the filtering for the capture input.
4. Sub-index 04_h: Multiple trip point spacing. This sub-index defines the modulus, or spacing between the trip points, scaled at 51200 steps/rev.

Table B.104 - 2038_h Object Description

Index	2038_h
Name	Trip output configuration
Object code	REC
Data type	—

Table B.105 - 2038_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	03_h
Category	—

Sub-index	01_h, Trip output control
Meaning	Trip output control
Access	Read-write
PDO mapping	—
Value range	Unsigned16
Default value	0000_h
Category	Yes

Sub-index 01_h value range		
Bit: 15 (trip enable)	14 ... 12	Bits: 11 ... 0 (number of trip points)
0= disabled, 1=enabled	0 0 0	0=infinite, 1 – 4095=# trip points from start

Sub-index	02_h, First trip point
Meaning	First trip point
Access	Read-write
PDO mapping	—
Value range	Integer32
Default value	0000 0000_h
Category	Yes

Sub-index	03_h, Multiple trip point spacing
Meaning	Multiple trip point spacing
Access	Read-write
PDO mapping	—
Value range	Integer32
Default value	5120_d
Category	Yes

2098_h Homing Configuration

This object determines the position or encoder counter status following a home. If 0, the position or encoder counter will clear following a home. The position and encoder counter will NOT be assigned, following a homing attained. The exception to this rule is when performing homing method 35.

If 1 (default), the device will subtract the homing offset (**`object 607Ch`**) from the counter and set the counter to the difference.

Table B.106 - 2098_h Object Description

Index	2098_h
Name	Homing configuration
Object code	VAR
Data type	Unsigned8

Table B.107 - 2098_h Value Description

Sub-index	00_h, Homing configuration
Meaning	Homing configuration
Access	Read-write
PDO mapping	—
Value range	0/1
Default value	1
Category	Yes

2099_h Index Offset

This object allows the setting of a position offset from the encoder index mark within a range of motor microsteps.

Table B.108 - 2099_h Object Description

Index	2099_h
Name	Index offset position
Object code	VAR
Data type	Unsigned8

Table B.109 - 2099_h Value Description

Sub-index	00_h, Index offset position
Meaning	Set position offset encoder index by +/- motor steps
Access	Read-write
PDO mapping	—
Value range	-25600 to +25600
Default value	1
Category	Yes

2204_h Run Current

This object sets the percentage of full current at which the device will operate.

Table B.110 - 2204_h Object Description

Index	2204_h
Name	Run current
Object code	VAR
Data type	Unsigned8

Table B.111 - 2204_h Value Description

Sub-index	00_h, Run current
Meaning	Run current
Access	Read-write
PDO mapping	—
Value range	$1 – 100_d$ (%)
Default value	25 (%)

2205_h Hold Current

This object sets the percentage of full current at which the device will transition to when motion ceases.

Table B.112 - 2205_h Object Description

Index	2205_h
Name	Hold current
Object code	VAR
Data type	Unsigned8

Table B.113 - 2205_h Value Description

Sub-index	00_h, Hold current
Meaning	Hold current
Access	Read-write
PDO mapping	—
Value range	$0 – 100_d$ (%)
Default value	5 (%)
Category	Yes

2211_h Position Present Point Target

This object contains the position present point target.

Table B.114 - 2211_h Object Description

Index	2211_h
Name	Position present point target
Object code	VAR
Data type	Integer32

Table B.115 - 2211_h Value Description

Sub-index	00_h, Position present point target
Meaning	Position present point target
Access	Read-only
PDO mapping	—
Value range	±231
Default value	0
Category	—

2212_h Position Final Point Target

This object contains the position final point target.

Table B.116 - 2212_h Object Description

Index	2212_h
Name	Position final point target
Object code	VAR
Data type	Integer32

Table B.117 - 2212_h Value Description

Sub-index	00_h, Position final point target
Meaning	Position final point target
Access	Read-only
PDO mapping	—
Value range	±231
Default value	0
Category	—

2221_h Following Error

This object defines an action to take in the event of a following error (**`Idx 6041 bit 13`**).

Table B.118 - 2221_h Object Description

Index	2221_h
Name	Following error
Object code	VAR
Data type	Integer32

Table B.119 - 2221_h Value Description

Sub-index	00_h, Number of entries
Meaning	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	01_h
Category	—

Sub-index	01_h, Following error reaction code
Meaning	Following error reaction code
Access	Read-only
PDO mapping	—
Value range	—
Default value	0 - 2
Category	—

Sub-index 01_h value range	
Value	**Action**
0	No action taken
1	Post-move to correct for error after initial move
2	Post-move to correct for error after initial move, then adjust for encoder position

2231_h Following Mode Control

Index 2231_h has 9 subindexes which may be used to tune the motor's response to the encoder input. Subindexes 01 through 03 set the direction of rotation, sample rate, and delay before sending a target reached message.

The graph below shows how sub-indexes 04 through 08 impact the move profile. Every sample period, a relative move is sent to the controller per Figure B.15. Default values shown on graph.

Figure B.15 - Index 2231_h Parameters

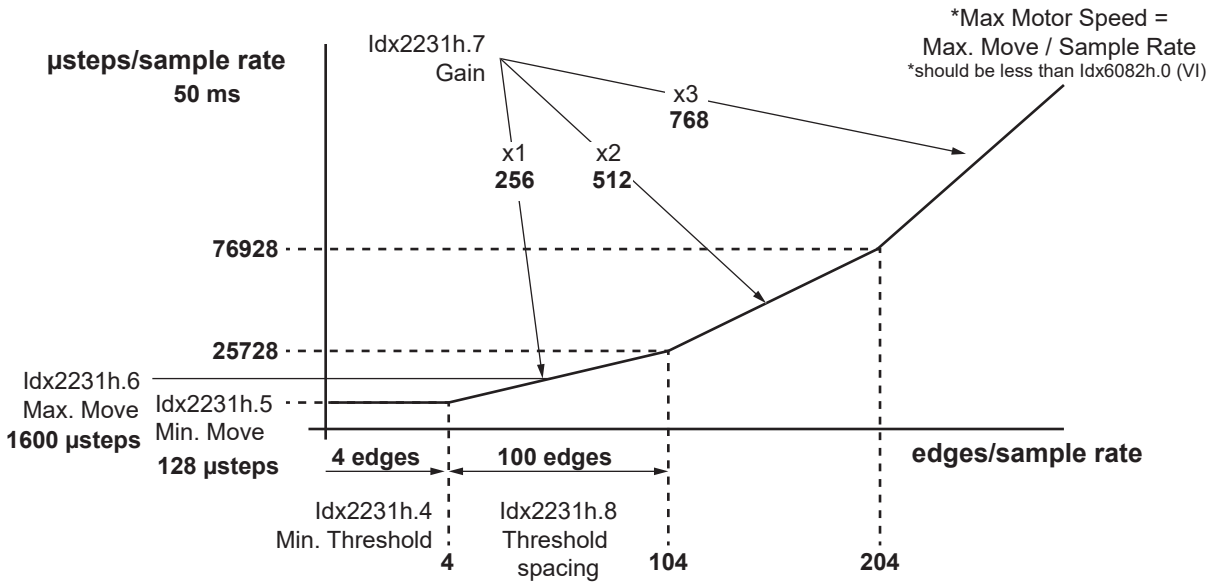


Table B.120 - 2231_h Object Description

Index	2231_h
Name	Encoder following mode control
Object code	ARRAY
Data type	—
Category	Optional

Table B.121 - 2231_h Entry Description

Sub-index	00_h
Name	Number of entries
Access	Read only
PDO mapping	—
Value range	—
Default value	08_h

Sub-index	01_h
Name	Direction of rotation
Description	Sets the default rotational direction of the motor.
Access	Read-write
PDO mapping	—
Value range	00_h - CW 80_h - CCW
Default value	00_h

Sub-index	02_h
Name	Sample rate
Description	Sets the time in milliseconds which the input is sampled.
Access	Read-write
PDO mapping	—
Value range	Unsigned16
Default value	50_d

Sub-index	03_h
Name	Target reached delay time
Description	Sets the time in milliseconds which the LMD will delay before triggering the target reached status bit (6041_h bit 10).
Access	Read-write
PDO mapping	—
Value range	Unsigned16
Default value	1000_d

Sub-index	04_h
Name	Minimum Threshold
Description	Sets the number of encoder counts required to activate x1 gain. At or below this value will result in a minimal move of 4 µsteps/encoder edge.
Access	Read-write
PDO mapping	—
Value range	Signed16
Default value	4_d

Sub-index	05_h
Name	Minimum Move
Description	Defines the minimum distance in µsteps the motor will move provided the encoder count is greater than zero.
Access	Read-write
PDO mapping	—
Value range	Signed16
Default value	128_d

Sub-index	06_h
Name	Maximum Move
Description	Defines the maximum distance in µsteps the motor will move within the specified sample time.
Access	Read-write
PDO mapping	—
Value range	Signed16
Default value	1600_d

Sub-index	07_h
Name	Gain
Description	The gain (µSteps/Encoder counts) that will occur between the min and max move values. There are 3 thresholds of gain (x1, x2, and x3). Based on the Encoder In threshold levels, the gain will be incriminated for the various thresholds. This gives variable gain potential.
Access	Read-write
PDO mapping	—
Value range	Signed16
Default value	256_d

Sub-index	08_h
Name	Threshold spacing
Description	Sets the various thresholds, in encoder edges, for encoder inputs. The lower the value, the faster the gain will increase. The faster encoder pulses are received, the faster the motor will move.
Access	Read-write
PDO mapping	—
Value range	Signed16
Default value	100_d

2401_h General Purpose User Variable

This object is a general purpose user variable which can be used to store 8-bits of data.

Table B.122 - 2401_h Object Description

Index	2401_h
Name	General purpose user variable
Object code	VAR
Data type	Integer32

Table B.123 - 2401_h Value Description

Sub-index	00_h, General purpose user variable
Meaning	General purpose user variable
Access	Read-write
Value range	$00 – FF_h$
Default value	0

2402_h General Purpose User Variable

Table B.124 - 2402_h Object Description

Index	2402_h
Name	32-bit general purpose variables
Object code	ARRAY
Data type	—
Category	Optional

Table B.125 - 2402_h Entry Description

Sub-index	00_h
Name	Number of entries
Access	Read only
Value range	04_h
Default value	08_h

Sub-index	01_h, General purpose user variable

Meaning	General purpose user variable
Access	Read-write
Value range	Unsigned32
Default value	0

Sub-index	02_h, General purpose user variable
Meaning	General purpose user variable
Access	Read-write
Value range	Unsigned32
Default value	0

Sub-index	03_h, General purpose user variable
Meaning	General purpose user variable
Access	Read-write
Value range	Unsigned32
Default value	0

2510_h Set NodeID or BAUD

Table B.126 - 2510_h Object Description

Index	2510_h
Name	Set NodeID or BAUD
Object code	ARRAY
Data type	—
Category	Optional

Table B.127 - 2510_h Entry Description

Sub-index	00_h
Name	Number of entries
Access	ro
Value range	04_h
Default value	02_h

Sub-index	01_h, Set NodeID
Meaning	Set NodeID
Access	Read-write
Value range	Unsigned32
LowLimit	1
HighLimit	127

Sub-index	02_h, Set BAUD
Meaning	Set BAUD
Access	Read-write
Value range	Unsigned32
LowLimit	0
HighLimit	8

2701_h hMT Enable

This object controls the enable/disable state of the Hybrid Motion Technology (hMT)control circuitry.

- Disabled = 00_h
- Enabled = 80_h

Object 2701h is only available on LMD closed loop models.

Table B.128 - 2701_h Object Description

Index	2701_h
Name	Hybrid enable
Object code	VAR
Data type	Unsigned8

Table B.129 - 2701_h Value Description

Sub-index	00_h, Hybrid enable
Meaning	Hybrid enable
Access	Read-write
Value range	00_h or 80_h
Default value	00_h
Category	Yes

2702_h hMT Configuration

This object sets the parameters of the hMT operation. The following parameters are set:

1. Fixed/variable current (bit 7): hMT can operate using two current modes, fixed and variable. In fixed current mode, the device will operate using the settings of the run (**object 2204h**) and hold (**object 2205h**) current set parameters. In variable current mode, the device will adjust the current between the run and hold current settings to what is required to move the load at the desired velocity.
2. Control boundaries (bits 6 ... 5): hMT functions by closely monitoring the relationship between the rotor and stator of the motor to within set boundaries from 1.1 to 1.7 motor full steps. A lower setting gives enhanced torque performance. A higher setting delivers better speed performance.
3. Make-up mode (bits 1 ... 0): If enabled, hMT control will make-up for lost steps in the move profile. This can occur at a fixed velocity of 2.5 MHz or at a speed preset using **object 2703h**.

Object 2701h is only available on LMD closed loop models.

Table B.130 - 2702_h Object Description

Index	2702_h
Name	hMT configuration
Object code	VAR
Data type	Unsigned8

Table B.131 - 2702_h Value Description

Sub-index	00_h, hMTechnology configuration
Meaning	hMT configuration
Access	Read-write
Value range	See Table 6.13 (Unsigned8)
Default value	$A2_h$
Category	Yes

Table B.132 - Description of Clock Options Object 2032.01_h

Bits	7	6	5	4	3	2	1	0
Function	cur_fv	c_bnds		X	X	X	mu	

Current mode			
Bit 7	0	Fixed current (default)	
	1	Variable current	
Control bounds			
Bit 6	Bit 5		
0	0	1.1 full steps (best torque performance)	
0	1	1.3 full steps (default)	(Best overall performance)
1	0	1.5 full steps	
1	1	1.7 full steps (best speed performance)	
Make-up mode			
Bit 1	Bit 0		
0	0	Make-up disabled	
0	1	reserved	
1	0	Make-up velocity = 2.5 MHz (default)	
1	1	Make-up velocity = 2703h setting	

2703_h Make-up Velocity

This object defines the velocity for hybrid make-up (**`Object 2702h, bits 1 ... 0`**) if selected. The make-up period is determined using the equation:

Frequency = (x+2) * 50ns where x is the setting of 2703_h

e.g., (1998+2) * 50 = 100000 steps/sec

`Object 2703h` is only available on LMD closed loop models.

Table B.133 - 2703_h Object Description

Index	2703_h
Name	Make-up velocity
Object code	VAR
Data type	Unsigned32

Table B.134 - 2703_h Value Description

Sub-index	00_h, Make-up velocity
Meaning	Make-up velocity
Access	Read-write
PDO mapping	—
Value range	Unsigned32
Default value	1998_d (100000 steps/sec)
Category	Yes

2704_h Torque Velocity

This object defines the maximum velocity for Profile Torque mode.

Velocity = 12 Mhz (x+2) where x is the setting of **`2703`**$_h$

e.g., 12 x 10^6 / (98+2) = 120000 steps/sec @ torque (**`Idx 6071`**)

`Idx 2704h` is only available on LMD closed loop models.

Table B.135 - 2704_h Object Description

Index	2704_h
Name	Torque velocity
Object code	VAR
Data type	Unsigned8

Table B.136 - 2704_h Value Description

Sub-index	00_h, Torque velocity
Meaning	Torque velocity
Access	Read-write
PDO mapping	—
Value range	Unsigned8
Default value	98_d (120000 steps/sec)
Category	Yes

2708_h Velocity Actual Filter

Object 2708_h takes a value of 0 to 10. It can be defined as 0 = no filtering and 10 = most filtering.

Because the Torque Velocity is computed and the encoder is sampled every millisecond, there can be fluctuation in the result. The filtering compensates for this fluctuation.

Table B.137 - 2708_h Object Description

Index	2708_h
Name	Velocity Actual Filter
Object code	VAR
Data type	Unsigned8

Table B.138 - 2708_h Value Description

Sub-index	00_h, Velocity Actual Filter
Meaning	Velocity Actual Filter
Access	Read-write
Value range	See table
Default value	0

Value	Notes/reading source	Filtering (ms)
0	hMT modes other than torque	No filtering
	Torque mode	256
1	Encoder	64
2	Encoder	128
3	Encoder	196
4	Encoder	256
8	Encoder	16
9	Encoder	24
10	Encoder	32

2710_h Locked Rotor Timeout Indicator

If the rotor is 'locked' for this period of time, the hMT status byte Idx2742_bit3 will be set to 1.

Table B.139 - 2710_h Object Description

Index	2710_h
Name	Locked rotor timeout
Object code	VAR
Data type	Unsigned16

Table B.140 - 2710_h Value Description

Sub-index	00_h, Locked rotor timeout
Meaning	Locked rotor timeout
Access	Read-write
Value range	0 (disabled) or 2 to 65535
Default value	5000
Category	Yes

2711_h Locked Rotor Opcode

The Locked Rotor Opcode object defines the response of the product to a locked rotor condition.

Table B.141 - 2711_h Object Description

Index	2711_h
Name	Locked Rotor Opcode
Object code	VAR
Data type	Unsigned8

Table B.142 - 2711_h Value Description

Sub-index	00_h, Locked Rotor Opcode
Meaning	Locked Rotor Opcode
Access	Read-write
Value range	0 to 3
Default value	2

Value	Locked rotor error response
0	Reserved
1	hMT statusword set, output h-bridge will not disable
2	hMT statusword set, output h-bridge will disable
3	hMT statusword set, output h-bridge will disable

2712_h Following Error Opcode

The Following Error Opcode object defines the response of the product to a following error condition

Table B.143 - 2712_h Object Description

Index	2712_h
Name	Following Error Opcode
Object code	VAR
Data type	Unsigned8

Table B.144 - 2712_h Value Description

Sub-index	00_h,Following Error Opcode
Meaning	Following Error Opcode
Access	Read-write
Value range	0 to 5
Default value	0

Value	Following error response
0	No action
1	Reserved
2	Output h-bridge will disable
3	Error set, output h-bridge will disable
4	Send emergency message, output h-bridge will disable
5	Error set, send emergency message, output h-bridge will disable

2740_h Clear Locked Rotor

This object is used to clear the locked rotor status flag. Sending 04_h will clear a locked rotor indication.

Table B.145 - 2740_h Object Description

Index	2740_h
Name	Clear locked rotor
Object code	VAR
Data type	Unsigned8

Table B.146 - 2740_h Value Description

Sub-index	00_h, Clear locked rotor
Meaning	Clear locked rotor
Access	Read-write
PDO mapping	Yes, R_PDO
Value range	Unsigned8
Default value	—
Category	Yes

2741_h hMT Status (Filtered)

This object indicates the status of the hMT logic. This object may be filtered using **`Idx 2743`**. The purpose behind the filter is that the object is PDO mappable. Bit 4, hMT intervening, may cycle rapidly between states. If mapped to a PDO this can saturate the fieldbus.

7	6	5	4	3	2	1	0
Factory	Reserved	Reserved	hMT intervening	Locked rotor	Reserved	Reserved	Error

Bit	Value	Meaning
0	0	No error
	1	hMT error state exists
3	0	Rotor normal
	1	Rotor is locked
4	0	hMT not intervening
	1	hMT intervening
7	0	N/A
	1	Factory alignment - This bit will always be 1

Object 2741h is only available on LMD closed loop models.

Table B.147 - 2741_h Object Description

Index	2741_h
Name	hMT status
Object code	VAR
Data type	Unsigned8

Table B.148 - 2741_h Value Description

Sub-index	00_h, hMT status
Meaning	hMT status
Access	Read-only
PDO mapping	Yes, T_PDO
Value range	Unsigned8
Default value	80_h
Category	—

2742_h hMT Status (Unfiltered)

This object indicates the status of the hMT logic.

7	6	5	4	3	2	1	0
Factory	Reserved	Reserved	hMT intervening	Locked rotor	Reserved	Reserved	Error

Bit	Value	Meaning
0	0	No error
	1	hMT error state exists
3	0	Rotor normal
	1	Rotor is locked
4	0	hMT not intervening
	1	hMT intervening
7	0	N/A
	1	Factory alignment - This bit will always be 1

Object 2742h is only available on LMD closed loop models.

Table B.149 - 2742_h Object Description

Index	2742_h
Name	hMT status
Object code	VAR
Data type	Unsigned8

Table B.150 - 2742_h Value Description

Sub-index	00_h, hMT status
Meaning	hMT status
Access	Read-only
PDO mapping	—
Value range	Unsigned8
Default value	80_h
Category	—

2743_h hMT Status Byte Filter

This object filters the status byte of the hMT logic.

7	6	5	4	3	2	1	0
Factory	Reserved	Reserved	hMT intervening	Locked rotor	Reserved	Reserved	Error
1	1	1	0	1	1	1	1

Object 2743h is only available on LMD closed loop models.

Table B.151 - 2743_h Object Description

Index	2743_h
Name	hMT status filter
Object code	VAR
Data type	Unsigned8

Table B.152 - 2743_h Value Description

Sub-index	00_h, hMT status filter
Meaning	hMT status filter
Access	Read-write
PDO mapping	—
Value range	Unsigned8
Default value	EF_h
Category	—

2840_h Multi-turn Control Byte

The multi-turn control byte is used to read the status of the encoder and related power system.

Table B.153 - 2840_h Object Description

Index	2840_h
Name	Multi-turn control byte
Object code	VAR
Data type	Integer8
Category	Optional

Table B.154 - 2840_h Entry Description

Sub-index	00_h
Access	Read only
PDO mapping	Yes
Value range	Unsigned 8
Default value	0_h

Bits	7	6	5	4	3	2	1	0
Function	clrstat	clrstat	X	X	rstrt	X	X	X

bit 3	rstrt	Restart state machine
bit 6	clrstat	Clear 6041h bits 4 and 6
bit 7	clrstat	Clear 6041h bits 0, 1, 2, 3, and 6

2841_h Multi-turn Status Byte

The multi-turn status byte is used to read the status of the encoder and related power system.

Table B.155 - 2841_h Object Description

Index	2841_h
Name	Multiturn status byte
Object code	VAR
Data type	Integer8
Category	Optional

Table B.156 - 2841_h Entry Description

Sub-index	00_h
Access	Read only
PDO mapping	Yes
Value range	Unsigned 8
Default value	0_h

Bits		7	6	5	4	3	2	1	0
Function		s-err	b-err	b-warn	oor	X	er	X	warn
bit 0	warn	Encoder advisory							
bit 2	er	Encoder read/write error							
bit 4	oor	External battery voltage out of range advisory							
bit 5	b-warn	Internal voltage advisory							
bit 6	b-err	Voltage error							
bit 7	s-err	Startup error							

In the case of status, bits 0, 2, 6, or 7 will indicate that multi-turn encoder position is lost and a rehome of the system is recommended.

Details of Object Group 5000_h (Mfg Factory Specific)

Object group 5000_h contains objects reserved for factory use.

Details of Assignment Objects Group 6000_h

The objects in group 6000_h are operation specific. See Chapter 3 on page 47 for detailed information on these objects.

Warranty

For the latest warranty and product information, visit: https://novantaims.com/warranty-and-disclaimer/

Document Revision History

LMD CANopen: LMD-CANOPEN-REV-E		
Date	**Revision**	**Changes**
07/03/2013	V1.00, 07.2013	Initial release
08/18/2014	V1.00, 08.2014	Updated to match firmware version 7.15. Notable changes include: hMT defaulted to an “off state, added a fourth RxPDO and TxPDO. Minor updates to indexes impacted by Firmware update.
09/22/2014	LMD-CANOPEN-REV-B	Updated to match firmware version 7.16. Added Index 6077h: Demand Torque Actual Value.
01/20/2019	LMD-CANOPEN-REV-C	Updated to support firmware release 7.23.12 This includes support for absolute encoder and encoder follower features.
01/03/2020	LMD-CANOPEN-REV-D	Updates and corrections throughout.
02/18/2022	LMD-CANOPEN-REV-E	Update to change Brand

Novanta IMS is a part of the Precision Motion Group within Novanta, a leading technology company that delivers innovation to medical and advanced industrial OEMS.

As standards, specifications, and designs may change, confirmation of the information given in this publication can be found in the product disclaimer and most recent product information, available online.
https://novantaims.com/all-products/

Photos: Novanta IMS
Print: Novanta IMS
To provide feedback on Novanta IMS documentation, send an e-mail to:
documentation@imshome.com

Novanta IMS
370 North Main Street
Marlborough, CT 06447
Phone: (860) 295-6102
www.novantaims.com